



---

# **SISTEMA DE INTEROPERABILIDAD PARA TECNOLOGÍAS CELULARES E INALÁMBRICAS ESTABLECIDAS EN LOS ESTÁNDARES IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11**

---

**Autor**

**NELSON GIOVANNI AGUDELO CRISTANCHO**

**Tutor**

**Octavio Salcedo Parra**

**PhD En Ingeniería Informática**

**PhD En Estudios Políticos**

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS**

**Énfasis en Teleinformática**

**Bogotá, Colombia**

**noviembre de 2021**

## CONTENIDO

RESUMEN .....	4
ABSTRACT .....	4
INTRODUCCION .....	5
PALABRAS CLAVE .....	6
1. PROBLEMA DE INVESTIGACION .....	7
1.1 PLANTEAMIENTO DEL PROBLEMA .....	7
1.2 PREGUNTA DE INVESTIGACIÓN .....	8
1.3 OBJETIVOS .....	8
1.4 RESULTADOS ESPERADOS .....	9
1.5 HIPOTESIS .....	9
2. COMUNICACIONES .....	10
2.1 COMUNICACIONES CELULARES .....	10
2.2 COMUNICACIONES INALÁMBRICAS .....	11
2.2.1 WiFi 802.11b/g/n .....	11
2.2.2 BLUETOOTH BLE IEEE 802.15.1 .....	11
2.2.3 LoRa IEEE 802.15.4 .....	12
2.3 INTERCONECTIVIDAD ENTRE REDES .....	12
3. TRABAJOS RELACIONADOS .....	14
4. METODOLOGIA Y DISEÑO DE SISTEMA INTEROPERABLE. ....	22
4.1 METODOLOGIA .....	22
4.2 DISEÑO FUNCIONAL .....	25
5. IMPLEMENTACIÓN DE SISTEMA INTEROPERABLE. ....	27
5.1 INTERCONEXIÓN .....	27
5.2 REDES CELULARES. ....	29
5.3. REDES INALÁMBRICAS .....	32
5.3.1 ESTANDAR IEEE 802.15.1. ....	32
5.2.1. ESTANDAR IEEE 802.15.4. ....	33

5.2.2	ESTANDAR IEEE 802.11.b/g/n .....	34
5.3	SISTEMA DE INFORMACIÓN. ....	36
6.	RESULTADOS. ....	41
7.	PRUEBAS Y DISCUSION DE RESULTADOS. ....	44
7.1	PRUEBAS DE FUNCIONAMIENTO. ....	44
7.2	PRUEBAS DE DESEMPEÑO OUTDOOR. ....	46
7.3	PRUEBAS DE DESEMPEÑO INDOOR. ....	61
7.4	PRUEBAS DE LATENCIA. ....	68
8.	CONCLUSIONES Y FUTUROS TRABAJOS. ....	75
8.1	CONCLUSIONES. ....	75
8.2	FUTUROS TRABAJOS. ....	76
9.	ANEXOS Y COMPLEMENTOS .....	79
10.	REFERENCIAS .....	96

## RESUMEN

---

Este documento presenta los diseños desarrollados y las implementaciones que fueron necesarias para el proceso de generación de un prototipo funcional, que integra tecnologías celulares e inalámbricas en un solo sistema, buscando que estas convivan en una sola interfaz, este prototipo permite al usuario evaluar la interoperabilidad, la interconectividad y analizar el desempeño que ofrece cada una de las tecnologías de acuerdo a las características de implementación frente a las necesidades de usuario final, enfocado a internet de las cosas (IoT) utilizando plataformas de visualización y almacenamiento web. Este prototipo reúne cada una de las tecnologías más utilizadas en cuanto a comunicación celular e inalámbrica soportadas sobre un sistema embebido basado en un micro-procesador y gestionado por una aplicación web.

El producto que ofrece este prototipo es la validación de cada una de estas tecnologías en ambientes de usuario final con funcionamiento en tiempo real, para mejorar la incorporación de aplicaciones actuales y futuras por medio de la interacción e integración de forma adecuada de múltiples tipos de tecnologías encaminadas a mejorar la capacidad y cobertura inalámbrica en diferentes tipos de escenarios, que en la actualidad están orientados al Internet de las Cosas (IoT).

## ABSTRACT

---

This document presents the designs developed and the implementations that were necessary for the process of generating a functional prototype, which integrates cellular and wireless technologies in a single system, seeking that they coexist in a single interface. This prototype allows the user to evaluate the interconnectivity and analyze the performance offered by each of the technologies according to the characteristics of implementation against the needs of the end user, Under the interoperability concept and oriented towards type solutions (IoT) Internet of Things. This prototype brings together each of the most used technologies in terms of cellular and wireless communication controlled by an embedded system based on a micro-processor and managed by a web application.

The product offered by this prototype is the validation of each of these technologies in end-user environments with real-time operation, to improve the incorporation of current and future applications through the interaction and integration of multiple types of technologies. aimed at improving wireless capacity and coverage in different types of scenarios, which are currently oriented to the Internet of Things (IoT).

## INTRODUCCION

---

La particularidad que presenta la investigación aplicada, los desarrollos tecnológicos y los procesos de innovación en el área electrónica y de las telecomunicaciones son generalmente las aplicaciones que involucran comunicaciones inalámbricas. En la actualidad las Redes de Sensores Inalámbricos WSN (*Wireless Sensor Networks*) y el Internet de las Cosas IoT (*Internet of Things*) están creciendo a un ritmo acelerado, tanto en el ámbito de investigación como en la industria. La convergencia entre ambos presenta muchos retos y grandes ventajas (Ruiz et al., 2016).

Las implementaciones orientadas al concepto de IoT inician sobre el año 2013, durante el año 2015 ACM, IEEE y Springer, publicaron 16 artículos en revistas científicas especializadas, los artículos estaban relacionados con diseño, simulación e implementación de aplicaciones de tipo IoT. Para el año 2016 esta cantidad se duplicó y paso a 32 artículos científicos, para un total de 48 en dos años. Se diseñaron e implementaron un total de 15 tipos de aplicaciones, siendo las más frecuentes las orientadas a *Smart home*, *Smart healthcare* y *Smart city* con 11, 9 y 8 respectivamente, se caracterizaron 142 sensores de 24 diferentes tipos (Morais et al., 2019).

Históricamente la arquitectura de red concebida para las diferentes tecnologías inalámbricas que han surgido a través del tiempo ha sido homogénea, diseñada e implementada para brindar servicio con cobertura y capacidad de manera uniforme utilizando elementos de red con parámetros muy similares. Pero la imperiosa necesidad de cumplir con ciertos requerimientos de Calidad de Servicio (QoS, *Quality of Service*) orientados a la cobertura y la capacidad se reflejó en una evolución de la arquitectura de red convencional, impulsando el cambio de una arquitectura homogénea a una nueva idea de arquitectura de red denominada Red Heterogénea (*HetNet*, *Heterogeneous Networks*). Las redes heterogéneas buscan ser una solución coherente con el mercado y sus necesidades actuales y futuras por medio de la interacción e integración inteligente de múltiples tipos de nodos, capaces de brindar y mejorar la capacidad y cobertura inalámbrica en diferentes tipos de escenarios (Probability & Networks, 2013).

Las actuales tecnologías de comunicación que están orientadas a soluciones convencionales o soluciones de tipo IOT combinan varios estándares de comunicación. El proceso de interoperabilidad entre plataformas de comunicación y plataformas en la nube es cada vez más necesario, por lo cual, se recomienda el análisis, la simulación y la evaluación previa que fundamente la selección de una determinada tecnología.

Los denominados Gateway, puertas de enlace o pasarelas vienen implementando soluciones que desarrollan una comunicación inalámbrica punto multipunto, en donde intervienen varios estándares de comunicación.

En este trabajo se describe el proceso de diseño, implementación y evaluación de un prototipo funcional que integra tecnologías de comunicación inalámbrica que apuntan a soluciones de tipo IoT. Se ilustra el diseño de arquitecturas, componentes del sistema, los códigos utilizados, los resultados que se obtienen de la puesta en marcha del prototipo y las conclusiones.

**PALABRAS CLAVE**

---

Gateway, interoperabilidad, compatibilidad, estándares, integración, BLE, LoRa, IoT.

## **1. PROBLEMA DE INVESTIGACION**

---

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

En la actualidad existen varios medios de comunicación, refiriéndonos a las telecomunicaciones inalámbricas y celulares se tiene que estas funcionan de manera independiente de acuerdo con un estándar establecido, la evolución de las tecnologías plantea nuevas soluciones, pero la carencia de sistemas que apunten a la interoperabilidad entre tecnologías emergentes limita el desarrollo de aplicaciones. Estas aplicaciones deben tener una selección adecuada del estándar que mejor solucione la necesidad requerida. Dentro del proceso de implementación de proyectos de I+D+i referentes al Internet de las cosas IoT, existe la necesidad y la tendencia a desarrollar prototipos que integren tecnologías disímiles, emergentes y convergentes.

La selección inadecuada de estándares de comunicación hace que el desarrollo de distintas aplicaciones falle en sus etapas finales debido a la deficiencia en las comunicaciones a causa de la mala elección de la tecnología, lo cual solo se identifica en el proceso de validación posterior a la implementación del prototipo funcional. La escogencia de un estándar de telecomunicación muchas veces se hace de manera intuitiva, a razón de no tener un sistema de validación de tecnologías emergentes. En la actualidad las comunicaciones deben realizarse de acuerdo a las necesidades, pero no existe un sistema que permita realizar un análisis de compatibilidad, latencias, pruebas de rendimiento, pérdida de paquetes, o que realice una evaluación de la interoperabilidad entre tecnologías. La aplicación de tecnologías erradas persiste al igual que el rediseño constante de aplicaciones debido al cambio de tecnologías de comunicación o la evolución de las mismas.

La carencia de desarrollos orientados a la evaluación y análisis de compatibilidad entre tecnologías es necesario en el momento de intentar solucionar una insuficiencia que involucre el uso de telecomunicaciones. Es necesario diseñar, implementar y validar prototipos dotados tanto de tecnologías celulares como inalámbricas buscando que estas convivan en una sola interfaz, esto implica el diseño y desarrollo de hardware, el diseño y desarrollo de software y los componentes de acople necesarios para la interacción y gestión.

Para solucionar una problemática en particular que depende del objetivo del proyecto y que involucra transporte de información, es necesario analizar y evaluar las características intrínsecas de las tecnologías implicadas en la factible convergencia a nivel de redes y servicios. Un problema adicional es la dificultad para la gestión, integración y prueba de las tecnologías inalámbricas. Adicionalmente, cuando se requiere implementar una aplicación que incluye transportar información, es necesario conocer los circuitos integrados o módulos de comunicación específicamente su configuración y caracterización para el desarrollo de los protocolos de comunicaciones.

Una de las tendencias en cuanto a comunicaciones celulares, es la implementación de comunicaciones de cuarta generación, pues, aunque la tecnología LTE se propone como la más

actual, en Colombia existe una alta penetración de redes de segunda y tercera generación. La cobertura 4g en el país aún no está masificada por diversas problemáticas como son, carencia de cobertura a nivel rural, compatibilidad 100% entre redes de acceso, dispositivos móviles y velocidades ofrecidas frente a las entregadas al usuario final.

Los sistemas de comunicación celular se clasifican mediante generaciones que van desde la primera orientada a la comunicación por voz, hasta LTE 4G que permite aplicaciones multimediales, de acuerdo con (Rappaport et al., 2013), la evolución de las tecnologías debe conservar la compatibilidad con generaciones anteriores.

La evolución de IoT permite que múltiples fabricantes de tecnologías inalámbricas participen en el desarrollo de dispositivos y describan las características en las hojas de datos de sus módulos o circuitos integrados, pero hasta el momento en el que las comunicaciones se implementen en ambientes de usuario final se tendrán verificadas completamente las características de una tecnología inalámbrica es posible evaluar parámetros como estabilidad, alcance, cobertura, pérdida de paquetes y latencia. Para poder analizar un sistema IoT y sus tecnologías es necesario diseñar, desarrollar y evaluar un prototipo funcional IoT que interopere con múltiples tecnologías inalámbricas. La conexión a internet es una característica esencial por lo que es posible crear un entorno de visualización local o en la nube y de almacenamiento local o en un servidor web de acuerdo con el desempeño de cada uno durante la implementación, es necesario además llevar el hardware a su máximo rendimiento para implementar y evaluar el concepto de tiempo real, calculando la latencia total entre actuadores y sensores. Los sistemas de IoT se pueden emplear en ambientes cerrados y abiertos, por esta razón se requiere evaluar el comportamiento de cada tecnología en línea de vista y con obstáculos para calcular el alcance y cobertura de cada una.

## **1.2 PREGUNTA DE INVESTIGACIÓN**

¿Cómo diseñar e implementar la interoperabilidad de tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4?

## **1.3 OBJETIVOS**

Implementar un sistema que permita la interoperabilidad de tecnologías celulares 2G, 3G y LTE, e inalámbricas establecidas en los estándares IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4. Los objetivos específicos son:

- Plantear los requerimientos que necesita el sistema de interoperabilidad de tecnologías celulares e inalámbricas en un prototipo de características de firmware.
- Implementar el hardware necesario para interconectar las tecnologías 2G, 3G y LTE, e inalámbricas establecidas en los estándares IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4.
- Desarrollar un software que permita la interacción del usuario con el sistema de interoperabilidad de tecnologías celulares e inalámbricas propuestas.



- Validar y evaluar la eficiencia del sistema y de las tecnologías propuestas en ambientes de usuario mediante resultados de implementación.

#### **1.4 RESULTADOS ESPERADOS**

Este prototipo funcional consiste en un desarrollo tecnológico que pretende generar nuevas ideas de investigación, permitir que tecnologías emergentes puedan interoperar, remediar y ayudar a orientar soluciones referentes a tecnologías IoT en hogares y empresas, los productos que se entregaran son:

- Diagramas de bloque planteados del sistema desarrollado.
- Relación de los códigos de software desarrollados para la gestión del sistema electrónico y la plataforma Web.
- Una solución para compatibilidad en dispositivos móviles.
- Implementación completa de un prototipo funcional.
- Resultados de puesta en marcha del sistema.
- Un artículo publicable de resultados obtenidos en la investigación.

#### **1.5 HIPOTESIS**

Mediante un diseño de interoperabilidad es posible la interconexión y puesta en funcionamiento de los estándares de comunicación inalámbrica IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4.

## 2. COMUNICACIONES

---

### 2.1 COMUNICACIONES CELULARES

El despliegue de (sistemas de comunicaciones móviles de 2ª y 3ª generación) 2G y 3G en el país, sin embargo, el interés de muchos organismos de investigación se enfoca hacia sistemas de última generación y de implementaciones futuras como LTE y 5G.

Con la tecnología 4G, una gama de nuevos servicios y modelos estarán disponibles. Estas interfaces de servicios y modelos necesitan ser analizadas y evaluadas con los sistemas de comunicación celular existentes los cuales se clasifican de acuerdo a la generación. (Rappaport et al., 2013).

Tecnología	Orientación	Transferencia Aprox
1G	Sistemas analógicos	1,9 Kbps
2G	Modulación digital	14,4 Kbps
3G	Acceso a internet	2 Mbps
LTE	Alta eficiencia espectral	200 Mbps
5G	IoT masivo	10 Gbps

Tabla 1 Tecnologías celulares.

Las redes celulares de primera generación fueron sistemas analógicos básicos diseñados para comunicaciones de voz. Posteriormente se realizó un cambio a los primeros servicios de datos y se mejoró la eficiencia espectral en los sistemas 2G. En 2G, haciendo uso de modulaciones digitales y acceso múltiple por división de tiempo o división de código se mejoró la velocidad como se relaciona en la tabla 1. Las redes de tercera generación 3G introdujeron el acceso a Internet de alta velocidad, capacidades de transmisión de video y audio altamente mejoradas mediante el uso de tecnologías como el acceso múltiple por división de código de banda ancha (W-CDMA) y el acceso a paquetes de alta velocidad (HSPA). HSPA es una amalgama de dos protocolos de telefonía móvil, acceso de paquetes de enlace descendente de alta velocidad (HSDPA) y acceso de paquetes de enlace ascendente de alta velocidad (HSUPA) que pretende ampliar y mejorar el rendimiento de las redes de telecomunicaciones móviles convencionales 3G que utilizan protocolos WCDMA. Un estándar mejorado de 3GPP (3G Partnership Project), HSPA +, se lanzó a fines de 2008 con la posterior utilización mundial a partir de 2010. HSPA ha sido implementado en más de 150 países por más de 350 proveedores de servicios de comunicaciones, en múltiples bandas de frecuencia.

Dependiendo de la época, los nuevos sistemas se implementan para mejorar las características de los métodos actuales y así reemplazar las técnicas existentes como son las que se originan luego de la tercera generación denominadas B3G o 4G (sistema de comunicación móvil cuarta generación). (Khan et al., 2009), en la actualidad se proponen diseños para implementación de la tecnología 5G.

Técnicamente, 4G significa un único entorno integrado basado en IP para todas las necesidades de telecomunicaciones incluyendo voz, vídeo, medios de radiodifusión, Internet y por lo tanto las aplicaciones orientadas a IoT que utilizan redes alámbricas como redes inalámbricas. Los usuarios son el foco central de 4G, los cuales por medio de terminales inteligentes pueden obtener desde un acceso sencillo de banda ancha, hasta una serie de servicios que tengan en cuenta sus preferencias personales (Jayanthiladevi et al., 2013).

## **2.2 COMUNICACIONES INALÁMBRICAS**

### **2.2.1 WIFI 802.11B/G/N**

Con el continuo desarrollo de aplicaciones sobre internet con el protocolo IP el uso de redes WIFI se está volviendo cada vez más común. Los denominados puntos de acceso WIFI (AP) se ha convertido en nodos clave para fortalecer la seguridad de los datos y el control de acceso del usuario. El punto de acceso WIFI sirve como un nodo de reenvío intermedio. Inicialmente puede adquirir todos los datos para la interacción del usuario. Y de otro lado, los puntos de acceso WIFI se conectan al ISP, permitiendo que los usuarios accedan a la red. De esta manera los puntos de acceso son la primera barrera para controlar el acceso de usuarios malintencionados a la red y son una base muy importante para el control de acceso (T. Liu & Wen, 2019). Adicionalmente el concepto de IoT requiere que los dispositivos estén conectados a internet y posiblemente requieran de un AP.

La nueva era de Internet de las cosas (IoT) que se refería a objetos identificables de forma única y se representaba en una estructura "similar a Internet", ha jugado un papel importante en nuestra vida diaria en términos de inteligencia y automatización como formas convenientes. Al crear conectividad IoT como sistema inteligente y conectar cosas a la red de internet se tiene como resultado la generación de un gran volumen de datos que necesitan gestión y control. Además, a medida que avanza la tecnología el sistema de gestión de automatización de IoT es más utilizado, se aplica en muchas infraestructuras básicas como los sistemas de medición de sensores, gestión de electricidad, gas y agua, según la conveniencia de individuos y organizaciones. (Hlaing et al., 2017).

### **2.2.2 BLUETOOTH BLE IEEE 802.15.1**

Los dispositivos (Bluetooth low Energy) BLE se incluyen en el estándar Bluetooth desde la especificación Bluetooth 4.0 y define dos topologías de red diferentes para transmitir datos: conexión y difusión. Debido a sus características, cada topología es apropiada para determinar casos de uso con diferentes fortalezas y debilidades. Las versiones posteriores de BLE (4.1, 4.2 y 5.0) mantienen estas topologías y las mejoran, permitiendo la combinación de diferentes roles. Sin embargo, estas últimas versiones mejoradas no se implementan en la mayoría de los dispositivos IoT, razón por la cual si se tiene la intención de trabajar en aplicaciones de tipo IoT es recomendable el uso de dispositivos de tipo BLE 4.0. Las topologías disponibles en el estándar BLE son "*Connection topology*" y "*Broadcast topology*". En la primera topología de conexión dos dispositivos BLE pueden establecer una conexión para intercambiar datos de forma permanente y periódica. Se utilizan dos

roles en esta topología: maestro (también llamado central) y esclavo (o periférico). Es posible conectar un dispositivo maestro con hasta 8 dispositivos esclavos en una topología estrella. Esta topología permite el flujo de datos en ambas direcciones. Además, los dispositivos esclavos proporcionan características de notificación e indicación para enviar datos al maestro en el momento en que cambian. La segunda topología es denominada de difusión, un dispositivo BLE puede usar paquetes de publicidad para transmitir datos a cualquier dispositivo BLE en modo de exploración ubicado dentro de su rango de cobertura, en esta topología se definen dos roles: emisor (que transmite datos) y receptor (que recibe datos), debido a la naturaleza de esta topología, el intercambio de datos es unidireccional, desde un emisor hasta uno o más observadores. Recientemente, la topología de “red mesh” se ha incluido en la Especificación de Bluetooth, aunque el número de dispositivos reales que lo implementan es muy limitado. (Garrido-Hidalgo et al., 2018).

### **2.2.3 LoRa IEEE 802.15.4**

Con el creciente interés en IoT se están desarrollando diversas tecnologías orientadas a abordar los requisitos para la integración de dispositivos inteligentes con el propósito de un bajo consumo de energía y una cobertura de señal de área amplia. Algunas de las tecnologías LPWAN todavía están en desarrollo, tecnologías como LoRa y SigFox ya están ampliamente disponibles en el mercado. Con respecto a la operación de bajo costo, las redes basadas en LoRa presentan una ventaja sobre las redes basadas en SigFox, considerando la necesidad de una suscripción SigFox para cada dispositivo, lo que se refleja en gastos de funcionamiento para cada dispositivo conectado. Sin embargo, las redes LoRa necesitan configurar una red propietaria, como lo implementa (Jörke et al., 2018) en una solución para la evaluación de una red de Smart City basada en LoRa.

La propagación de LoRa a 433MHz está ubicada en la banda de Ultra Alta Frecuencia (UHF). Se ha realizado una gran cantidad de trabajo desde la década de 1960 para comprender la propagación de ondas de radio en el entorno forestal y urbano. Es posible presenciar la ruptura total del enlace de comunicación debido a factores externos, se necesita mucha más investigación, especialmente para el trabajo empírico específico del sitio y aquellos que limitan la aplicación práctica del trabajo de investigación existente, como se describe en (Ahmad et al., 2018) en cual, se describe la transmisión en áreas de entornos forestales.

## **2.3 INTERCONECTIVIDAD ENTRE REDES.**

Con el uso generalizado de los teléfonos inteligentes el tráfico de datos móviles está creciendo exponencialmente. Este hecho plantea un reto importante en términos de capacidad para los operadores móviles cuyas infraestructuras no son capaces de soportar todo el tráfico adicional generado por los usuarios de este tipo de dispositivos IP, de esta situación surgen una serie de enfoques para dar solución a este problema de capacidad, un enfoque posible es mediante el incremento de la granularidad de las antenas desplegadas, es decir, mediante el despliegue de un mayor número de antenas con menor cobertura conocidas como *Small Cells*. Sin embargo, la necesidad de infraestructura adicional tiene costos significativos tanto en el despliegue como en las

fases de planificación y gestión, requiriendo su escalado y una planeación minuciosa, por lo tanto, esta solución de forma aislada (desplegar más infraestructura) no es adecuada para satisfacer la creciente demanda de capacidad de intercambio de información a la que se enfrentan las redes actuales y por consiguiente a la que se enfrentarán las futuras en mayor medida. Otras alternativas emergentes que actualmente se consideran como parte de la evolución 5G (quinta generación) de la red incluyen la migración del tráfico de datos móviles de la infraestructura del operador a los dispositivos de los usuarios (*offloading*) aprovechando las capacidades de conexión de los actuales *Smartphones* para transmitir los datos mediante comunicaciones dispositivo a dispositivo (*Device-To-Device* - D2D) y la integración de las comunicaciones celulares, WiFi y ad-hoc o comunicaciones D2D, constituyendo las denominadas redes celulares multisalto (*Multi-Hop Cellular Networks* - MCN). (Río Jalón et al., 2016). Sin embargo, estas soluciones dependen de los operadores de comunicación celular.

En el futuro, los puntos de acceso deberán ofrecer un camino inalámbrico para cualquier persona o cosa. El acceso inalámbrico llegará más allá para incluir servicios a cualquier objeto que se pueda respaldar de una conexión, este concepto a menudo se conoce como “la Internet de las cosas (IoT)”, “la sociedad interconectada”, “comunicaciones máquina a máquina (M2M)” o “comunicaciones centradas en las máquinas”, cada vez se desarrollan dispositivos tales como medidores inteligentes de servicios, redes de sensores, señalización digital, sistemas de info-entretenimiento en vehículos y conceptos como WSN (*Wireless Sensor Network*) que interactúa con redes alámbricas o inalámbricas, redes móviles (LTE) o migraciones 5G, sensores y protocolos, deben analizar la compatibilidad de tecnologías orientados al IoT. (Río Jalón et al., 2016).

### 3. TRABAJOS RELACIONADOS

---

A continuación, se detallan algunos trabajos relacionados, el documento de (Sehrawat & Gill, 2019) presenta la estructura básica de un sistema IoT describe varios tipos de sensores y varias posibles aplicaciones. Además, después de analizar diferentes aplicaciones de sensores, este artículo aclara qué aplicación de IoT requiere qué tipo de sensor. El artículo de (Ruiz et al., 2016) presenta la simulación necesaria para la implementación de una red de sensores inalámbricos orientados a IoT utilizando la tecnología 6LoWPAN, proponen una Red de Sensores Inalámbricos WSN compuesta de un gran número de nodos sensores desplegados, Se cita este trabajo por describir las dos maneras de integración de sensores para aplicaciones de IoT. El nodo sensor de la WSN está compuesto por un módulo de detección, algoritmos de procesamiento, y elementos de comunicación. Las WSN están tendiendo a integrarse a IoT, con el objetivo de interoperar con tecnologías de comunicación heterogéneas. Actualmente el acceso a Internet por parte de WSN, tiene diferentes maneras de integrarlo, la primera forma consiste en una conexión de una WSN independiente y un acceso a Internet a través de una única pasarela, comúnmente denominada Gateway o puerta de enlace que utiliza una comunicación de largo alcance generalmente una tecnología celular (Figura 1).

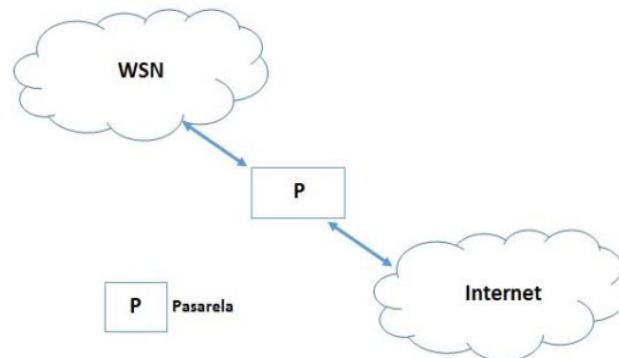


Figura 1 Red independiente (Ruiz et al., 2016).

La segunda forma, se basa en una red híbrida, formada por redes independientes, donde pocos nodos sensores pueden acceder a Internet (Figura 5).

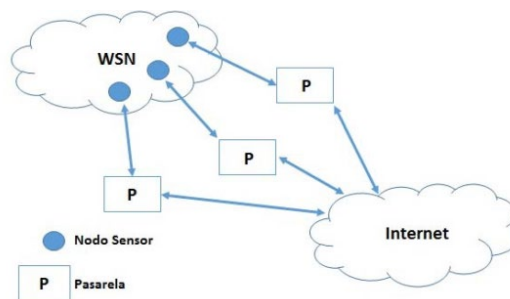


Figura 2 Red híbrida (Ruiz et al., 2016).

En (Khalil & Hamaydeh, 2013) mencionan la investigación que propuso un mecanismo de traspaso entre las redes WiFi y WiMAX utilizando el estándar MIH, se estudió el impacto del factor LGD y la confianza de probabilidad sobre la latencia de traspaso y la pérdida de paquetes. Sin embargo, no se consideró el impacto de algunos indicadores clave en la transferencia vertical como la velocidad del nodo móvil, las tasas de bits de las aplicaciones como parámetros de entrada y sus efectos en el rendimiento medio, el retardo E2E, la pérdida de paquetes y la latencia de traspaso, este trabajo se toma como referencia por la importancia de un protocolo de pruebas de rendimiento como latencia y desempeño de tecnologías en diferentes ambientes.

En el trabajo (Mehaseb et al., 2013) se pretende arrojar alguna luz sobre los posibles desafíos de integración de WSN y LTE, que se manifiestan en la diferencia en características de tráfico a través de IoT. Se realizan también algunos experimentos para explorar el comportamiento del tráfico específico de la aplicación WSN. Posteriormente se analiza la posible adaptación que se requiere para la integración exitosa de las dos tecnologías. El propósito es simular las características de tráfico de una red de sensores inalámbricos de detección de intrusos. De esta implementación se toma el concepto de integración de tecnologías implementadas en un nodo sensor.

En (Simo-Reigadas et al., 2015) se menciona la necesidad de cobertura mediante redes inalámbricas a bajo costo para las zonas rurales. Las áreas urbanas actualmente representan más del 60% de la población mundial por lo tanto las soluciones tecnológicas se centran habitualmente en estas áreas. Las zonas rurales son mucho más grandes en extensión, pero sólo son el 40% de la población en todo el mundo, proporcionar a esta población servicios de telecomunicaciones es mucho más difícil y costoso debido a la falta de accesibilidad y la baja densidad de población. En particular, los operadores de telecomunicaciones en las regiones en desarrollo ofrecen todos los servicios y aplicaciones de telecomunicaciones en las ciudades donde los usuarios se concentran y los ingresos compensan las inversiones requeridas para desplegar las infraestructuras, por esta razón en el artículo citado se presenta un estudio de los métodos más recientes que permiten alcanzar la interoperabilidad entre las tecnologías 3G y WLAN. Para nuestro caso se desarrolló un sistema de interoperabilidad entre redes celulares y tres tecnologías inalámbricas.

El documento de (Kang & Kim, 2016) presenta una nueva arquitectura para los servicios de salud basados en el estándar ISO / IEEE 11073 sobre una plataforma IoT. El dispositivo Bluetooth existente no es compatible para proporcionar servicios de salud basados en IoT, ya que la tecnología Bluetooth no es compatible con el protocolo IP. La arquitectura propuesta satisface los estándares M2M, ISO e IEEE 11073, se proponen protocolos para los dispositivos de salud restringidos a través de la red BLE, se diseña 6LoWPAN como un protocolo de capa IP para la comunicación a través de la red BLE y CoAP como un protocolo de capa de aplicación para comparar la eficiencia del protocolo en los dispositivos restringidos, se analizó el análisis de rendimiento utilizando HTTP y CoAP. A partir de los resultados, se concluye que el CoAP puede ofrecer un mejor rendimiento que HTTP en términos de rendimiento de transmisión de datos.

El concepto de sensores inteligentes es necesario para detección de variables preferiblemente en tiempo real para aplicaciones de IoT, por ejemplo, el índice de contaminantes como en el trabajo de (Duangsuwan et al., 2018) que se propone en la ciudad de Bangkok. En este diseño se plantea la

adopción de sensores LPWAN, para monitorear los datos de la contaminación del aire a través de la red NB-IoT.

La interoperabilidad en una red LoRa WAN y BLE se propone en el trabajo de (Ayele et al., n.d.). Este trabajo propone una nueva arquitectura de red dual de radio para IoT utilizada para el monitoreo de la vida silvestre, esta arquitectura logra un control más amplio sobre el equilibrio entre el consumo de energía y el alcance, cambiando adaptativamente la radio operativa de WMS con base en medidas de proximidad y aplicando un esquema de concatenación de datos en el cabezal del clúster. Los resultados de la evaluación indican que la red propuesta supera a los sistemas tradicionales que utilizan un solo tipo de radio transceptor (es decir, LoRa o BLE). En promedio, se redujo el consumo de energía de LPWAN (LoRa WAN) hasta en un 97%. La arquitectura mejoró el tiempo de vida de la red hasta en un 99% para varias tasas de tráfico de paquetes en la red.

La interoperabilidad entre las tecnologías inalámbricas es necesaria para aplicaciones de tipo IoT, en el trabajo de (Rahman & Chakraborty, 2018) se describe la implementación de las tecnologías inalámbricas (BLE) Bluetooth Low Energy y ZigBee pertenecientes a los estándares IEEE 802.15.1, IEEE 802.15.4 respectivamente, en el trabajo se propone un Gateway o puerta de enlace desarrollado para la interconexión de estas dos tecnologías en una aplicación de tipo IoT que permita reducir costos, energía y latencia. La interoperabilidad es importante para la adopción a gran escala de IoT. La tarea principal de IoT en el desarrollo es recopilar datos de detección mediante los dispositivos conectados y realizar el envío al servidor web. Los protocolos ZigBee de interoperabilidad se utilizan para crear un entorno de red con nodos sensores que tienen una velocidad de datos baja y los protocolos BLE son utilizados para recopilar esta información y transmitirla a velocidades de datos más altas. En el desarrollo del trabajo propuesto en este texto el concepto de interoperabilidad se implementa a través de una puerta de enlace, Gateway o pasarela que se desarrolla para tecnologías inalámbricas orientada a la gestión de actuadores y supervisión de sensores a través de una aplicación web.

Los resultados de implementación de la tecnología inalámbrica LoRa se presentan en el artículo de (Zourmand, 2019), se analiza el rendimiento de la tecnología teniendo en cuenta la cobertura en ambientes cerrados y abiertos modificando los parámetros de configuración de acuerdo a cada condición, los resultados concluyen que en los ambientes cerrados el rendimiento depende tanto de la distancia del Gateway como de la estructura de los elementos de encerramiento, en espacios abiertos la distancia es de 12 a 330 metros teniendo la puerta de enlace desarrollada en un espacio cerrado.

Además de la implementación de redes LoRa WAN se han analizado y realizado pruebas de rendimiento en redes privadas basadas en LoRa para aplicaciones IoT, en las cuales se encontraron algunos problemas al usar el estándar de LoRa en redes privadas con escenarios de trabajo especiales, por lo tanto, se propone un estándar LoRa WAN modificado basado en redes mesh y TEDS, la red de mesh mejora la cobertura y facilita la implementación, con los TEDS, se reduce la tasa de colisión en la red especialmente para los sistemas que tienen muchos nodos finales y necesitan recopilar datos controlados por eventos, se desarrolla una pila TEDS para reducir la tasa de colisión de datos y se vincula el dato de mejor calidad esto evita el aumento simultáneo de la



carga debido a señales de procesamiento de eventos y datos errados. En particular, es adecuado para sistemas de medición remota de agua o gas que funcionan con una batería, sistemas simulados que requieren capacidades casi en tiempo real y sistemas de control de iluminación vial que requieren una respuesta rápida de muchos nodos (Kim, 2019).

Una solución para monitoreo ambiental de forma portátil se describe en (Wu & Christoph, n.d.), el documento presenta un nodo de sensor IoT portátil para aplicaciones de seguridad llamado WE-Safe que se centra principalmente en los desarrollos de hardware, un nodo WE-Safe portátil recolecta algunos datos ambientales de muestra y los envía a la nube a través de la red LoRa. El objetivo del proyecto WE-Safe es proporcionar alertas tempranas para las personas que trabajan en entornos extremos y hostiles.

Orientados a la interoperabilidad entre aplicaciones IoT se relaciona el trabajo de (Georgi et al., 2018) donde resaltan la modalidad de las aplicaciones ofrecidas como servicios y la necesidad de tener versatilidad para no limitar aplicaciones a una tecnología o un sensor específico, esto sugiere un sistema de interoperabilidad que se convierte en un concepto crucial para intercambiar datos de múltiples tipos de sensores y estándares que proporcionen un monitoreo más robusto. Adicionalmente para fomentar la interoperabilidad en los sistemas de salud, se propone un middleware en forma de biblioteca de software basado en el funcionamiento de una API capaz de comunicarse con un conjunto de sensores de tipo WIFI y Bluetooth, utilizando protocolos estándar patentados con el objetivo de acceder a cualquier información. Al usar dicha biblioteca los desarrolladores no necesitan implementar cada protocolo de forma independiente y pueden centrarse en el análisis y procesamiento de datos.

En el documento de (Electrical Engineering/Electronics et al., 2018) se presenta un desarrollo basado en sensores de detección de contaminación del aire, los cuales son gestionados por un microcontrolador y enviados a un sitio web mediante la comunicación inalámbrica NB-IoT, el sistema se diseñó para ciudades inteligentes en Tailandia, el sitio web visualiza la gráfica del índice de calidad del aire y presenta resultados de las variables mediante los sensores configurados. Se cita este trabajo pues se propone una plataforma web para gestión de hardware, visualización de variables y almacenamiento de datos.

El trabajo de (Cheng et al., 2019) describe la implementación de un sistema que utiliza un microprocesador específicamente el STM32F103RCT6, el microprocesador interactúa con el puerto serie en el cual está conectado el sensor de gas electroquímico realizando un envío de datos. El sistema puede aportar datos como temperatura, humedad y material particulado PM2.5, la comunicación de red está basada en el protocolo TCP / IP y la conectividad se implementa mediante el módulo NB-IoT o el módulo GPRS, los datos con marca de tiempo se pueden cargar en tiempo real. Se relaciona este trabajo pues se adopta la lectura de puertos USB para conectar dispositivos y se involucró el concepto de marca de tiempo.

En el documento de (Dambal et al., 2019) se realiza la medición de RSS en un LoRa GW para áreas interiores, suburbanas y urbanas, cuando el transmisor LoRa está en otra ubicación distinta como en un interior o montado en un vehículo aéreo no tripulado (UAV). Para un entorno interior,

concluimos que un LoRa GW puede recibir paquetes si se coloca en la línea de visión directa de una abertura en el edificio como una puerta, una ventana o una repisa. Para el entorno suburbano, la altura del dron y la orientación de la antena juegan un papel crucial en el RSS, ya que no hay reflectores de señal fuertes que provoquen la recepción de múltiples rutas en el LoRa GW. Este trabajo define la RSS como una medida fundamental para evaluar el rendimiento de una tecnología inalámbrica.

El trabajo de (Rivero & Bolaño, 2018) en el capítulo 6 presenta una solución inalámbrica *zigbee* aplicada a un sistema de energía solar de tipo SCADA para monitoreo de variables de interés, se compone de un sistema de almacenamiento, un sistema de visualización y una red de sensores y actuadores. El proceso de interconexión entre la red inalámbrica y el servidor hace posible la adquisición, el procesamiento y la caracterización de señales entregadas al servidor, se cita este trabajo pues se adopta el concepto de conexión de interacción con redes inalámbricas por interfaz UART para procesamiento y almacenamiento.

El tema propuesto en (Keramidas et al., 2016) es la introducción de un concepto que comienza desarrollando una puerta de enlace o Gateway que se puede transferir a un simulador para Internet de las cosas (IoT), para lograr esto, se presenta el simulador de red *OMNeTCC* que puede utilizarse para la investigación de la infraestructura de las comunicaciones e introduce una técnica para simular diferentes estándares de comunicación en IoT, el simulador de red *OMNeTCC* se utiliza para modelar la infraestructura entre nodos virtuales y físicos, la funcionalidad de la puerta de enlace es realizar la traducción entre diferentes estándares de comunicación no se limita a un dispositivo específico, pero puede manejar la simulación de sensores y actuadores, Este trabajo se cita pues sustenta por medio de una simulación la necesidad de un sistema interoperable para IoT, abierto a diferentes estándares de comunicación.

En el trabajo de (Issa et al., 2019) en el capítulo 4 se plantea un sistema de IoT para monitoreo de variables de interés fundamentado en un microprocesador GHI, referencia G120 de 32 bits, tipo ARM Cortex-M3, la solución contiene un sistema de almacenamiento y de visualización web, los sensores son conectados directamente a la tarjeta y tecnologías de tipo satelital, celular de tercera generación y WiFi son utilizadas para el envío de datos a un servicio web de reportes fundamentado en un *Backend*, permitiendo la creación de múltiples *frontend*.

Un punto importante en los sistemas IoT es el análisis de la calidad del servicio QoS factores como la interferencia, el ruido y la distancia pueden afectar la información, en el trabajo de (Mishra et al., 2018) se presenta una implementación algorítmica eficiente para la precodificación óptima global de TX y los parámetros de relación IoT, el sistema IoT es *multicasting MISO SWIPT* compuesto por una multi antena de transmisión para los sistemas IoT de multidifusión que tienen restricciones de QoS individuales. Los resultados mostraron que el diseño propuesto supera a los puntos de referencia existentes y produce una ganancia de rendimiento significativa de más del 20%. Las extensiones futuras del marco presentado incluyen la consideración de múltiples antenas en los dispositivos IoT y conjuntos de antenas masivas en TX, así como de aplicaciones de ondas milimétricas con arquitecturas de formación de haces híbridas. Los resultados toman como base mediciones de potencia de TX individual en un ambiente de multidifusión.

El trabajo de (L. Liu et al., 2018) describe un sistema que propone la trazabilidad de la información desde la recolección automática basado en IoT. La adquisición se compone principalmente de terminales de detección por RFID, una estructura de tipo bus y un sistema de cómputo para la administración. El terminal puede percibir automáticamente la información de cada enlace y transmitir la información de percepción al sistema de administración de la computadora superior a través del bus de datos. El sistema de administración de la computadora superior realiza las funciones de almacenar la información de detección, el análisis inteligente y la activación de alarmas. Los resultados indican que los sistemas de tipo IoT mejoran significativamente las capacidades de gestión de calidad y seguridad de los productos, pero el autor reconoce que es necesario mejorar aún más la capacidad de percepción del terminal, la capacidad de procesamiento de información y la interacción hombre – máquina.

IEEE define la interoperabilidad como “*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*” la capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información que se ha intercambiado (Geraci A, Katki F, McMonegal L, Meyer B, Lane J, Wilson P, Radatz J, Yee M, Porteous H, 1991) En consecuencia, los sistemas interoperables deben orientarse a proporcionar servicios de soporte entre ellos, por esta razón la conectividad y la compatibilidad son conceptos más básicos que el de interoperabilidad esta deducción se plantea como base en el trabajo de (Kirpes et al., 2019), el cual realiza un análisis comparativo del estado de la técnica en los modelos encontrados, en los marcos de arquitectura existentes y describe los requisitos de la interoperabilidad para la movilidad eléctrica, de esta forma proponen el modelo de arquitectura de sistemas de movilidad eléctrica (EMSA), basado en un modelo de arquitectura de sistemas tridimensionales, involucrando las fases de generación, transmisión, distribución y cliente, su estructura se origina del modelo de arquitectura *Smart Grid*.

El crecimiento generalizado de Big Data y la evolución de las tecnologías de Internet de las cosas (IoT) permiten a las ciudades obtener información valiosa de una gran cantidad de datos producidos en tiempo real. En una ciudad inteligente *Smart City*, varios dispositivos de IoT generan flujos de datos continuamente que deben analizarse en un corto período de tiempo usando alguna técnica de Big Data. Los *frameworks* de procesamiento de flujo distribuido (DSPF) tienen la capacidad de manejar el procesamiento de datos en tiempo real para *Smart Cities*, como se describe en el trabajo de (Nasiri et al., 2019) en el cual se examina la aplicabilidad de emplear *frameworks* de procesamiento de flujo distribuido en la capa de procesamiento de datos para una *Smart City* y evaluar el estado actual de su adopción y madurez entre las aplicaciones de IoT. El texto describe la evaluación de tres DSPF: Apache Storm, Apache Spark Streaming y Apache Flink, según los resultados obtenidos en términos de latencia y rendimiento, elegir un *framework* adecuado en la capa de análisis de datos de una Smart City requiere suficiente conocimiento sobre las características de las aplicaciones de destino, cada uno de los *frameworks* tiene sus ventajas y desventajas, *storm* y *flink* tienen un rendimiento muy similar y *Spark Streaming* tiene una latencia mucho mayor mientras que proporciona un mayor rendimiento. También se evalúa la escalabilidad (en términos de velocidad de llegada de datos y el número de nodos del clúster) y la utilización de recursos de estos marcos utilizando dos aplicaciones de referencia del mundo real. Se cita este

trabajo pues las pruebas de rendimiento de tecnologías sobre plataformas web se realizaron de forma similar.

La creciente población de personas afectadas por enfermedades crónicas como diabetes, enfermedades cardiovasculares y obesidad, el apoyo a la salud, tanto mental como física, es cada vez más importante. La detección, el monitoreo remoto de la salud y el reconocimiento de las actividades de la vida diaria se hacen vitales y desde una perspectiva técnica el internet de las cosas (IoT) está ganando una atención cada vez mayor en estas disciplinas especialmente en la atención médica personalizada, mientras tanto, la red de sensores de área corporal (BASN) bajo el marco de IoT se ha aplicado ampliamente para el monitoreo ubicuo de la salud, en el trabajo de (Wan et al., 2018) se presenta el sistema WISE (sistema de monitoreo de salud basado en la nube de IoT) desarrollado para el monitoreo de la salud personal en tiempo real. WISE adopta el marco BASN (red de sensores de área corporal) en apoyo de la supervisión de la salud. La mayoría de los sistemas de monitoreo de salud portátiles requieren un teléfono inteligente como puerta de enlace de procesamiento, visualización y transmisión de datos lo que impacta el uso diario normal del teléfono inteligente. Mientras que en WISE, los datos recopilados de BASN se transmiten directamente a la nube y se puede incrustar una pantalla LCD portátil liviana como una solución alternativa para una vista rápida de los datos en tiempo real.

Las tecnologías y capacidades de los sensores tienen un efecto en la calidad de los datos de observación, por lo general, la gestión de la información comienza cuando un administrador obtiene los datos y necesita referirlos a los consumidores de datos, en aplicaciones de despliegue masivo es necesario asignar roles de acuerdo a la especialidad de cada integrante, a menudo, los métodos de detección no se describen adecuadamente, el administrador de datos desconoce el funcionamiento de los sensores, su configuración y el despliegue, en consecuencia, el conocimiento a menudo permanece oculto en los manuales del sensor y en los registros del operador en campo, la mayoría de los requisitos de metadatos se han simplificado para acomodar esta brecha en el conocimiento, cuando la información se captura, se pueden proporcionar descripciones accionables por máquina para describir adecuadamente los procesos tomados en la generación de observaciones, la información puede asociarse con los datos, ser accesible, detectable, utilizada en el control de calidad de la información por parte de los proveedores de datos y en la evaluación de la calidad de los mismos por parte de los consumidores. En el trabajo de (Fredericks & Botts, 2018) se definen actores y acciones para promover la creación basada en roles de documentos completamente descritos y basados en estándares.

Las barreras sociales de lograr que los fabricantes de sensores y los operadores de campo se involucren en la generación de información pueden superarse al proporcionar acceso a herramientas simples que les permitan contribuir sin ser conscientes de las tecnologías involucradas en habilitar soluciones interoperables para capturar metadatos, como la información se comparte en una comunidad más amplia y los datos están menos asociados con aquellos que los crearon, es imperativo poder comprenderlos y evaluar su reutilización al esperar un conjunto más amplio de descripciones. Los autores están trabajando con la plataforma de interoperabilidad de datos oceánicos (<http://www.odip.org>) y hacia el desarrollo y la adopción de perfiles SWE estandarizados.

El trabajo de (Savaux, 2020) pretende realizar la estimación del identificador de celda (ID) para sistemas (NB-IoT), el principio se extiende al caso práctico donde el canal es desconocido y debe estimarse, en este escenario, la función de probabilidad concentrada necesita ser maximizada en la cual la estimación del canal ML es una función del parámetro de ID de celda desconocido. Los resultados de la simulación han demostrado que el método propuesto alcanza el rendimiento de la búsqueda exhaustiva de ID de celda ML, además, otras series de simulaciones han revelado que la técnica basada en DFT es robusta para el desplazamiento de frecuencia residual ya que experimenta una pérdida de  $\leq 0.5$  dB hasta un desplazamiento de 200 Hz. En futuras investigaciones se propone trabajar en adaptar el estimador a los canales selectivos de frecuencia, el resultado sería muy conveniente teniendo en cuenta que solo la tecnología NB-IoT se encuentra en desarrollo y en Colombia es aprovisionada únicamente por el operador claro.

## 4. METODOLOGIA Y DISEÑO DE SISTEMA INTEROPERABLE.

La metodología propuesta se compone de tres etapas fundamentales, el diseño, la implementación y los resultados, cada uno tiene un capítulo en el desarrollo del documento, durante la transición del diseño a la implementación se debieron ajustar componentes de hardware como el cambio de tecnología en el diseño inicial de la tecnología *zigbee* por la tecnología LoRa, el circuito integrado WiFi esp8266 al componente de *Particle Photon* WiFi y el cambio del Hc05 al *Bluebean*, el hardware definitivo se muestra en el capítulo de implementación en la figura 8, en la construcción del sistema de almacenamiento se encontró inestabilidad en la aplicación, razón por la cual se conservó el modelo cliente servidor que se ejecutaba en la *Raspberry*, pero se migró el *Backend* y el *Frontend* a la plataforma *Azure* para visualización y procesamiento en la nube, el almacenamiento local se migró a un servidor web alojado en la plataforma Heroku, la arquitectura definitiva se muestra en figura 53.

### 4.1 METODOLOGIA.

Para la implementación de la interoperabilidad en tecnologías, especialmente desde el punto de vista de la conectividad el protocolo IP se considera la tecnología más prometedora ya que se diseñó con un objetivo muy similar en entornos de redes informáticas heterogéneas. También se observa que IP ha tenido éxito durante varias décadas y ahora está funcionando en la mayoría de las redes de capa de enlace existentes como Ethernet, WiFi y LTE bajo el concepto de *All-IP*. El intercambio de datos dentro de los entornos de desarrollo del Internet de las Cosas (*IoT Internet of Things*), la conexión de personas, cosas y procesos crece exponencialmente, la omnipresencia de estos dispositivos, el gran número de soluciones tecnológicas sin una interfaz de programación de aplicaciones (*API Application Programming Interface*) estandarizada o protocolos de comunicación oficialmente adoptados hacen de la interoperabilidad en todo el sistema una característica fuerte de la rama de estudio. (Allafi, 2017).



Figura 3. Modelo por capas propuesto para los sistemas IoT

En el artículo de (Mahmoud & Mohamad, 2016) se describe IoT como un sistema basado en miles de millones de sensores y actuadores inteligentes, para construir dicho sistema se deben introducir nuevas ideas sobre sensores inteligentes, cálculos y procesamiento de datos. El primer requerimiento para IoT es la interconexión de dispositivos, sensores y actuadores, el cableado de Ethernet no se encuentra viable, de hecho, solo la tecnología inalámbrica ofrece la flexibilidad, escalabilidad y rentabilidad necesaria para garantizar la adopción viable de Internet de las cosas, en este trabajo también se aclara que la comunicación inalámbrica por radiofrecuencia con su multitud y la infraestructura existente ofrece la mejor solución para el tráfico de datos en sistemas IoT, basados en este concepto se propone el modelo mostrado en la figura 3 y una arquitectura fundamentada en redes inalámbricas.

Este modelo está compuesto por seis capas, las tres capas inferiores incluyen los sistemas de sensores, comunicaciones de corto alcance y el acceso por Gateway, pasarelas o puertas de enlace.

Un tipo de conexiones muy usadas en IoT para la interconexión de dispositivos inteligentes con un acceso inalámbrico corto como *Bluetooth* o *Zigbee*; los nodos de la red son de bajo consumo y se comunican con un *Gateway* el cual habilita la conexión con redes de mayor ancho de banda, en la cuarta capa se integra la tecnología IP. La quinta capa de la arquitectura corresponde a todos los servicios encargados de controlar el sistema *IoT*, se encuentran las API de aplicaciones y se da respuesta a los requerimientos no funcionales como seguridad y disponibilidad.

Finalmente, la capa superior o capa de aplicación incluye un software de tipo web dedicado acorde al sistema propuesto para este desarrollo, se diseñó e implementó un sistema de gestión web para la visualización de señales obtenidas por sensores y el almacenamiento de información en la nube.

El prototipo propuesto interopera con tecnologías celulares inalámbricas WiFi, LoRa y Bluetooth LE en la segunda capa, realizando inicialmente un proceso de interconexión y finalmente una interoperabilidad tanto de tecnologías como de aplicaciones.

El control de hardware es realizado con un sistema embebido *Raspberry Pi* que cuenta con funcionalidades de capa 3 y 4, es gestionado por un sistema operativo y configurado con aplicaciones web en las capas superiores. Las APP modernas tienen una tendencia a modificar sus componentes del sistema orientados a encontrar el mejor desempeño de sus plataformas, para el caso de las herramientas que se implementan en este trabajo el mapa funcional propuesto se puede observar en la figura 4, basados en la tendencia que presentan las webs app modernas.

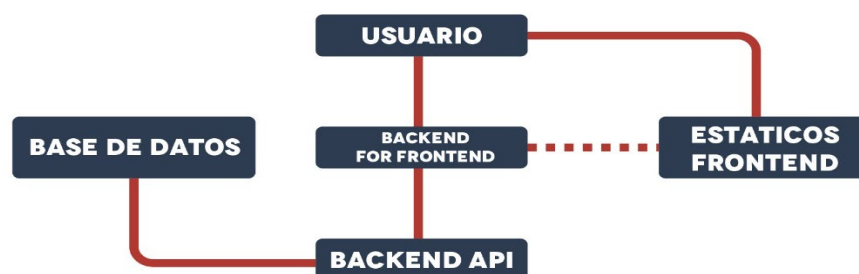


Figura 4 Web App moderna.

El diseño propuesto para la implementación del sistema, está compuesto por una aplicación de *Backend* como unidad de procesamiento, una unidad de almacenamiento de datos recopilados “base de datos”, una unidad de API que permite la interfaz con otras aplicaciones y una aplicación de *Frontend* vista al navegador.

El reto más difícil de los sistemas de IoT consiste en orientarse al concepto de gestión en “tiempo real” teniendo en cuenta la implementación requerida para la interoperabilidad de tecnologías y que el tráfico de datos provenientes del hardware debe implementarse evitando al máximo retardos de procesamiento, en la figura 5 se muestra el diseño propuesto para el flujo de datos hacia el servidor implementado, el servidor se desarrolló en node.js incorporando el concepto de *websocket* mediante la librería de *javascript* denominada *socket.io*, esta librería se orienta a aplicaciones web en tiempo real y permite la comunicación bidireccional entre el cliente que se ejecuta en el navegador y el servidor que inicialmente se ejecuta en la *Raspberry* y posteriormente se ejecuta en la nube.



Figura 5 Trafico de datos a la web.

Teniendo en cuenta que este sistema pretende la integración de tecnologías emergentes hacia IoT se propone desarrollar el procesamiento principal desde un sistema embebido basado en una arquitectura de microprocesador para la gestión de hardware mediante un sistema operativo.

Inicialmente se utilizaron microcontroladores de 8 bits programados bajo lenguaje C, se experimentó inestabilidad en el consumo de servicios web, el diseño final está basado en un sistema operativo raspbian de Linux instalado en un dispositivo embebido *Raspberry pi 3* sobre el cual se implementaron las aplicaciones de *Backend* y de *frontend*, esta arquitectura se adopta para facilitar la visualización y la comunicación con los sistemas inalámbricos como lo muestra la figura 6, sin embargo aunque el desempeño es mejor en la visualización en el proceso de almacenamiento la aplicación presentó algunos problemas.



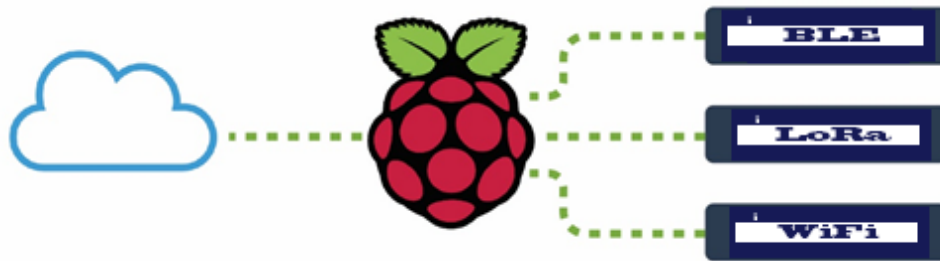


Figura 6 Conectividad propuesta.

## 4.2 DISEÑO FUNCIONAL.

La arquitectura IoT basada en IPv6 es la tecnología más avanzada para IoT. Muchos grupos de trabajo IETF (WG) están desarrollando protocolos de adaptación de IPv6 para el IoT, tecnologías como 6LoWPAN, 6lo, 6tisch, Roll, core, lwig son conocidas para el desarrollo de aplicaciones. En esta tecnología se especifica cómo proporcionar compresión de encabezado IPv6, fragmentación, reensamblado de paquetes IPv6 y reenvío de capa 2 para adaptar IPv6 en el enlace IEEE 802.15.4.

Además del IEEE 802.15.4, 6loWG está extendiendo 6LoWPAN para admitir otras redes restringidas como Bluetooth LE, DECT Ultra LE, red MS/TP y NFC. Roll WG ha desarrollado un nuevo protocolo de enrutamiento, denominado RPL, para baja potencia. (Jung, 2015). La desventaja de la implementación de esta tecnología es su elevado costo y el desarrollo del software que tiene la tecnología no es de carácter *open source*, esto representa una dificultad para este desarrollo orientado a la interoperabilidad de tecnologías, se propone para esta solución un servicio RESTful basado en UDP y se adopta Bluetooth LE 4.0 siendo el más indicado para aplicaciones de IoT

Se propone trabajar con tecnologías inalámbricas de punta como WiFi, LoRa y bluetooth LE 4.0, en busca de una solución completa de IoT, controlada bajo un sistema embebido, gestionada por un sistema operativo y funcionando como aplicación web tal como se ilustra en la figura 7.

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

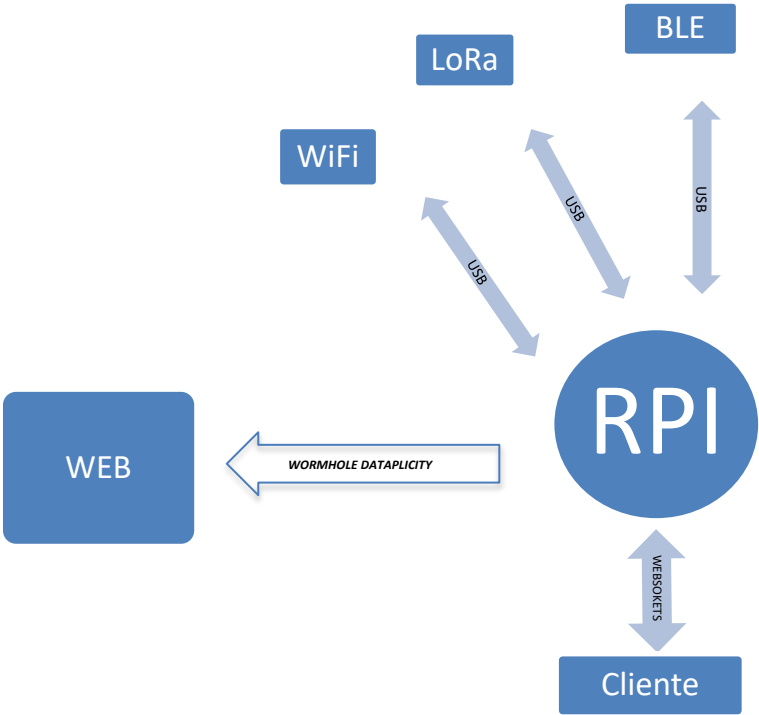


Figura 7 Diseño funcional.

## 5. IMPLEMENTACIÓN DE SISTEMA INTEROPERABLE.

### 5.1 INTERCONEXIÓN.

El primer paso para la implementación del sistema fue la selección del sistema de control de hardware principal, inicialmente se evaluaron varios microcontroladores buscando hallar el más adecuado para la solución, se realizaron pruebas de respuesta con microcontroladores de 8, 16 y 32 bits, lenguajes de programación para sistemas embebidos sobre los cuales se implementaron servidores web con HTML y CSS buscando el control y lectura de puertos sobre Java y C, en la búsqueda de una solución se encontró node.js de javascript que por su orientación a aplicaciones en tiempo real puede ser utilizado para IoT para optimizar el tiempo de entrega.

Buscando reducir las latencias se optó por una solución en la cual el *Backend* y el *Frontend* estén conectados continuamente, como resultado se implementó el requerimiento de modelo concurrente de un solo hilo que funciona en sistemas embebidos es decir las dos aplicaciones sobre el dispositivo, el modelo concurrente cliente servidor en el proceso de visualización presentó resultados positivos en cuanto a velocidad de visualización, sin embargo, en el momento de realizar el almacenamiento local se encontró que la solución presentó sobrecalentamiento en el procesador, se implementó un sistema de refrigeración para reducir la temperatura, pero se observó que la aplicación web de *Frontend* estaba presentado inestabilidad, recepción de datos inconsistentes y perdida de información a partir de aproximadamente siete minutos de operación, procesando, visualizando y almacenando los datos de los tres nodos sensores, realizando gestión de lecturas cada segundo. La búsqueda de la estabilidad en el sistema concibió la necesaria migración del *Frontend* y el *Backend* a la plataforma de Azure, configurando la *Raspberry* como un IoT hub, se realizó procesamiento y almacenamiento en la nube y se obtuvo estabilidad en la aplicación, la arquitectura se rediseñó y es ilustrada en la figura 53 del capítulo de resultados.

La conexión de hardware para el sistema de interoperabilidad entre dispositivos LoRa, *Bluetooth* 4.0 BLE y WiFi conectados a *Raspberry* Pi3, se puede visualizar en la Figura 8. En esta se detalla la implementación de acuerdo con las tecnologías propuestas

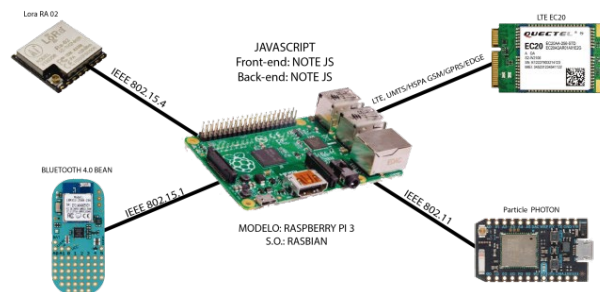


Figura 8 Conexión de sistema de interoperabilidad.

Un sitio web puede ser alojado en un servidor web y proporcionar servicios solicitados por los clientes, los servidores web constan de un sistema operativo, una página web y un gran espacio de memoria, es necesario, como en este caso un hardware especial. Los clientes pueden acceder a un servidor web a través de un enrutador LAN que proporcione una salida a Internet. La aplicación de tipo web consta de dos partes: el programa de servidor *Backend* y el programa de cliente *frontend*. El programa de servicio de *Backend* es principalmente responsable de gestionar los comandos del usuario, procesar la lógica de datos comerciales, comunicarse con el sistema integrado, además de enviar páginas de vista y recursos estáticos a los clientes o *frontend*. La aplicación *Frontend* que se ejecuta en el dispositivo o cliente proporciona la interfaz gráfica de usuario y el procesamiento simple de la lógica de negocios. La aplicación de *Backend* se está ejecutando en el servidor, mientras que la aplicación *Frontend* se analiza y se ejecuta en el navegador o en cualquier otro cliente de tipo *frontend*. (Allafi, 2017).

Los usuarios pueden llamar al localizador uniforme de recursos (URL) para enviar comandos a través de la solicitud HTTP mediante la aplicación web del cliente o *Frontend* que se ejecuta en el navegador con un método llamado JavaScript asincrónico y XML (AJAX). De esta manera después de recibir la solicitud HTTP del cliente *frontend*, la aplicación de servidor *Backend* se ocupará de esta solicitud que opera la base de datos o envía comandos al sistema integrado y origina la lectura de datos consecutivos desde el sistema integrado de acuerdo con la lógica comercial, varios clientes *Frontend* pueden conectarse al servidor al mismo tiempo, es decir, esta topología permite a múltiples usuarios operar una gran cantidad de sistemas integrados al mismo tiempo. (Zhang et al., 2018).

Para el caso de la solución propuesta se desarrolla el código de cliente servidor sobre la *Raspberry PI3*, para optimizar el intercambio de datos, sin embargo se reformuló la solución y la base de datos se gestiona por el motor de base de datos de *PostgreSQL* desde la plataforma de *heroku*, la app *Backend*, la app *Frontend* y el servicio de IoT Hub se implementaron finalmente en la plataforma *Azure* que para este caso tiene costo, la proyección, facturación y análisis de costos se pueden ver en los anexos 1 al 3.

En el proceso de muchos sistemas integrados la comunicación con la computadora o host es necesaria y esencial. En la actualidad y para nuestro caso el modo de comunicación principal es la interacción del puerto universal asíncrono de tipo serie como receptor/transmisor (UART) para la interacción de datos de red. Al mismo tiempo la mayoría de los métodos de desarrollo de software utilizan lenguajes de programación como JAVA, C #, C ++ o *LabVIEW* para crear aplicaciones de escritorio.(Zhang et al., 2018).

## 5.2 REDES CELULARES.

La interconexión a redes de largo alcance es necesaria para lograr la interacción con los dispositivos y los servidores de datos, tomando como referencia el estudio de (Wang et al., 2017), el cual tiene la intención de implementar una propuesta de Internet de las cosas de bajo costo para monitorear, analizar la circulación del tráfico, proporcionar soluciones para reducir los efectos negativos del tráfico vehicular en las ciudades de mayor densidad de población, la cantidad de accidentes y la violencia vial, el sistema de envío de información a la red se realiza vía GPRS, tomando como referencia este trabajo se diseñó y desarrolló una tarjeta que implementa la tecnología GSM y GPRS, la cual se muestra en la figura 9, opera vía USB y se gestiona por comandos AT para el envío de datos a la plataforma Web.

El diagrama de bloques planteado inicialmente fue de interconexión por cada puerto USB a un tipo de tecnología celular y se eligió el fabricante quectel, sin embargo, este fabricante lanzó al mercado el circuito integrado de referencia EC20 el cual implementa LTE 4G, pero también tecnologías de 3G y 2G, este chip que se muestra en la figura 10, hizo posible que la interfaz a LTE y demás tecnologías celulares se lograran por un solo puerto de datos USB del sistema embebido.



Figura 9 Tarjeta desarrollada para red celular 2g y 3g.

Quectel *Wireless Solutions*, es un proveedor global de módulos, soluciones y servicios GSM / GPRS, UMTS / HSPA / LTE y GNSS. Durante el desarrollo del proyecto se anunció el lanzamiento de los módulos EC20 y EC20 Mini PCIe. Estos módulos LTE adoptan el 3GPP Rel. Tecnología LTE, con

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

velocidades de datos de enlace descendente de 100Mbps y enlace ascendente de 50Mbps y ofrece FDD-LTE multibanda (B1 / B3 / B5 / B7 / B8 / B20) junto con HSPA y GPRS / EDGE.

Se incorpora al desarrollo la serie EC20 Mini PCIe-E 4G, pues es compatible con las redes GSM, GPRS y EDGE, teniendo en cuenta que en Colombia el despliegue de redes celulares de segunda y tercera generación es amplio.



Figura 10 Modulo EC20 fuente quectel.com

Los módulos LTE son compatibles con la tecnología MIMO (múltiple entrada - múltiple salida), también combinan la conectividad inalámbrica de alta velocidad con el receptor GPS + GLONASS de posicionamiento de alta sensibilidad integrado de múltiples constelaciones.(Quectel, 2012).



Figura 11 Tarjeta utilizada en desarrollo de comunicación celular.

El módulo seleccionado se puede ver en la figura 10 y La tarjeta electrónica seleccionada para el desarrollo del sistema finalmente está basada en el circuito integrado de referencia EC20 del fabricante Quectel el cual se muestra en la figura 11. Este tipo de módulos se configura por una interfaz de comunicación que diseñe el fabricante y las instrucciones se envían de acuerdo a su hoja de datos por comandos AT, para este caso se realizaron pruebas por la interfaz serial RS 232 y por la interfaz USB, encontrando más versátil el acceso al componente por USB. Específicamente para este desarrollo es necesario tener acceso a la red para enviar y recibir información por el protocolo TCP/IP, de esta manera y según el sistema de interoperabilidad propuesto es necesario crear inicialmente la interconexión del servidor alojado en la *Raspberry* y la red de internet para tener acceso a funciones de gestión remota o lectura de información.

El proceso de interoperabilidad debe ser versátil, portable y lo más aproximado a tiempo real, se implementa el sistema de control en un microprocesador de 32 bits, es versátil, cuenta con autonomía y es portable, se puede ver en los anexos fotográficos del 4 al 12 que muestran además los nodos sensores de tipo IoT para las tecnologías propuestas.

La conexión a la red IP se realiza por comandos AT desde un microcontrolador, en este caso se gestionó desde el sistema operativo raspbian un script que contiene los comandos AT necesarios para gestionar la conexión a la red celular, para el acceso de datos se realizaron pruebas con los operados locales Movistar, Tigo y Claro y la serie de comandos se acopla al operador, pues el apn es almacenado en un archivo de la carpeta etc este archivo consulta y se actualiza en el momento de ejecutar el *script connect*, basados en un repositorio ubicado en git hub (GitHub, 2019) el contenido del script se muestra en el anexo 14.

Luego de la ejecución de comandos por consola, el módulo quectel proporciona una dirección IP, se debe comprobar la respuesta de ping por parte del DNS para verificar el funcionamiento correcto. Los picos de corriente para el funcionamiento del módulo celular requieren que sea alimentada con una fuente adicional de 500 mA y requiere una antena de tipo celular para estabilizar la recepción y entrega de datos, incorpora un led RGB, el cual cambia de color a verde cuando la red está disponible para establecer una conexión, como se muestra en la figura 12

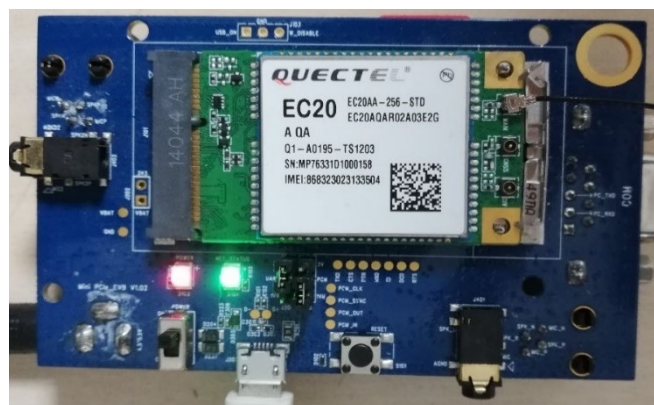


Figura 12 Red disponible.

### 5.3. REDES INALÁMBRICAS.

#### 5.3.1 ESTANDAR IEEE 802.15.1.

Para este estándar se iniciaron pruebas con módulos de implementación del estándar bluetooth de referencia HC05, KC21, HC06 y HM10, se estableció conectividad, tráfico de datos, control y lectura de dispositivos, pero se encontró un módulo de origen americano el cual durante la puesta en funcionamiento se mostró más robusto, estable y de muy bajo consumo de energía. El dispositivo seleccionado se muestra en la figura 13 y se conoce como *LightBlue Bean*.

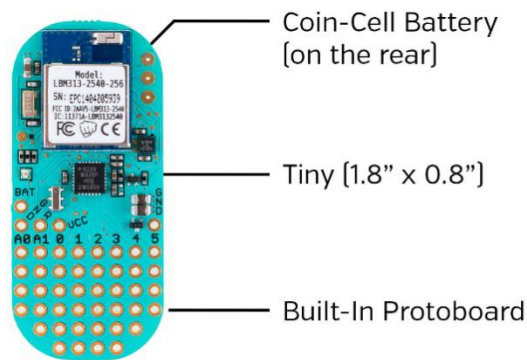


Figura 13 LightBlue Bean Fuente: [https://punchthrough.github.io/bean-docs/\\_assets/images/guides/getting-started/tech-specs/bean-diagram.jpg](https://punchthrough.github.io/bean-docs/_assets/images/guides/getting-started/tech-specs/bean-diagram.jpg)

Este desarrollo es adecuado para la construcción de algunos dispositivos portátiles y de carácter IoT, las características del *LightBlue Bean* y los recursos integrados se utilizaron para la solución de la implementación del estándar IEEE 802.15.1, en el sistema descrito en este texto, la figura 14 muestra la arquitectura completa de LightBlue Bean.

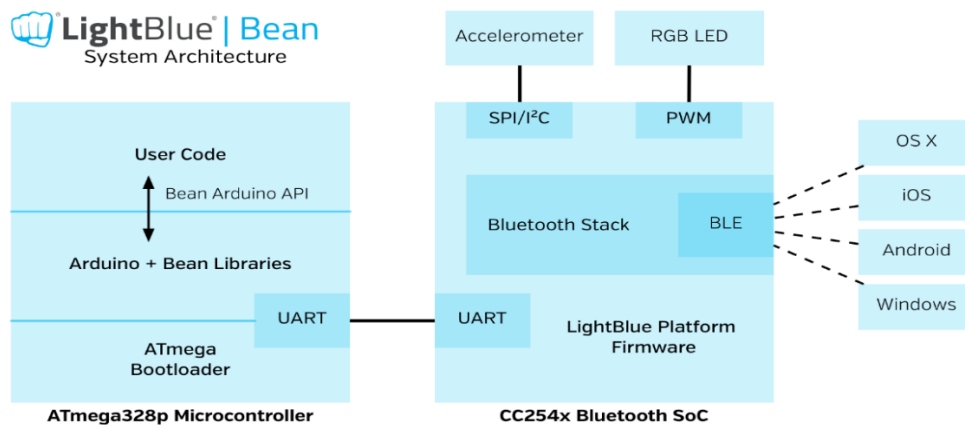


Figura 14 Arquitectura completa de LightBlue Bean, Fuente ([https://punchthrough.github.io/bean-docs/\\_assets/images/guides/getting-started/tech-specs/bean-block-diagram.png](https://punchthrough.github.io/bean-docs/_assets/images/guides/getting-started/tech-specs/bean-block-diagram.png)).



### 5.2.1. ESTANDAR IEEE 802.15.4.

Para la implementación de este protocolo se exploraron varias opciones, inicialmente se implementó la recepción y envío de datos mediante Zigbee, como conclusión de la implementación se tiene que es una tecnología robusta de muy bajo consumo pero de costo elevado, razón por la cual se implementa la tecnología LoRa que se muestra como una tecnología emergente para IoT, se utilizan los transceptores SX1278 teniendo en cuenta que funcionan en un rango espectral libre, manejan un bajo consumo de energía, el componente electrónico es muy económico y cuentan con el módem de largo alcance LoRa® que proporciona comunicación de espectro extendido de ultra largo alcance e inmunidad a la interferencia, según lo relaciona su fabricante *Semtech* en su hoja de datos.



**Figura 15** Circuito integrado SX1178, Fuente (<https://www.semtech.com>)

La tecnología se impulsa con el uso de la técnica de modulación LoRa patentada de Semtech, la cual puede lograr una sensibilidad de más de  $-148\text{dBm}$  utilizando un cristal y una lista de materiales de bajo costo. El módulo LoRa basado en el circuito integrado SX1178 que se implementó en este desarrollo se muestra en la figura 15. La alta sensibilidad combinada con el amplificador de potencia integrado de  $+20\text{dBm}$  ofrece un buen desempeño de enlace para diversas aplicaciones.

LoRa también proporciona ventajas significativas tanto en el bloqueo como en la selectividad sobre las técnicas de modulación convencionales, resolviendo el compromiso de diseño tradicional entre rango, inmunidad a la interferencia y consumo de energía, según lo describe su fabricante. En la figura 16 se muestra el diagrama de bloques del integrado seleccionado

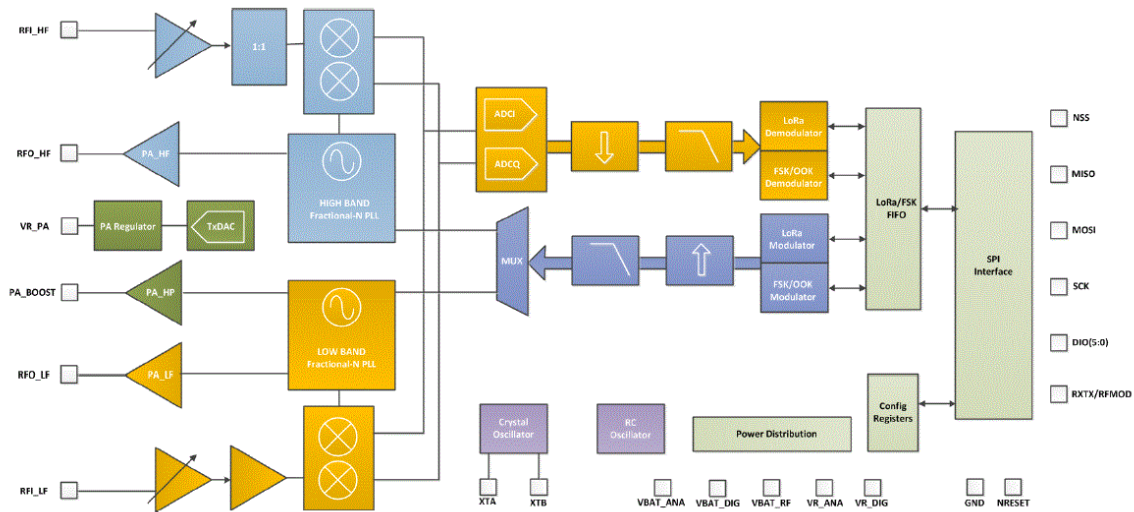


Figura 16 Diagrama de bloques transceptor LoRa Fuente: (<https://www.semtech.com/uploads/products/sx1276-bd.gif>)

## 5.2.2 ESTANDAR IEEE 802.11.b/g/n

Durante la implementación de este protocolo se realizaron pruebas de funcionamiento con módulos de comunicación inalámbricos como el ESP8266, ESP32, sobre los cuales se implementaron servicios web, lectura y envío de datos, durante la puesta en funcionamiento se experimentó la carga lenta e inestabilidad en el servicio web, pérdida de paquetes y latencias altas, razón por la cual se experimentó con un módulo especializado en IoT que cuenta con una plataforma web denominada *particle.io*. La plataforma ofrece un circuito integrado basado en la arquitectura WICED de *Cypress*, se denomina *Photon* comercialmente, se enmarca en la familia de *Particle Photon*, combina un microcontrolador STM32 ARM *Cortex* M3 con el chip Wi-Fi BCM43362, probado en la industria y orientado a productos IoT según el fabricante. En la figura 17 se observa la apariencia del hardware del componente electrónico.



Figura 17 Particle Photon nodo WiFi

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

El hardware de la familia Photon se puede utilizar de varias maneras según las necesidades de cada proyecto, en las pruebas de implementación se observó la estabilidad en la conectividad. Este hardware está orientado a la creación de prototipos para luego de una validación se proceda a implementar en diseños especializados propios para una fabricación masiva. La implementación de la comunicación con este dispositivo requiere de la configuración de las características propias del estándar IEEE 802.11 en el sistema operativo, para esta preparación se toma como base el trabajo de (Haines et al, 2013), la serie de instrucciones de configuración ejecutadas por consola sobre el sistema operativo en modo súper administrador en se muestran a continuación:

```
sudo cp /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant/wpa_supplicant.conf.sav
sudo cp /dev/null /etc/wpa_supplicant/wpa_supplicant.conf
```

Es necesario ingresar al archivo de configuración ubicado en:

/etc/wpa\_supplicant/wpa\_supplicant.conf, y se deben agregar las siguientes líneas de código:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
```

La configuración de la red implementada se describe a continuación y se almacena en el archivo conf, este archivo debe contener los parámetros necesarios para la configuración del estándar, el acceso a dispositivos de red, la cantidad de dispositivos y el tipo de seguridad a implementar, el identificador de red y la clave de acceso, los comandos requeridos y las acotaciones se muestran a continuación:

```
IP address: 192.168.0.1
DHCP range: 192.168.0.10 to 192.168.0.50
Security: WPA
Channel:11
SSID: Interop
Password: interop2019
Wireless Mode: 802.11.n
```

Para crear el punto de acceso se deben crear directorios desde el usuario raíz:

```
sudo mkdir /etc/raspap
sudo mv /var/www/html/raspap.php /etc/raspap/
sudo chown -R www-data:www-data /etc/raspap
sudo mkdir /etc/raspap/hostapd.
Se deben crear los permisos de control de scripts:
sudo mv /var/www/html/installers/*log.sh /etc/raspap/hostapd
sudo mv /var/www/html/installers/service*.sh /etc/raspap/hostapd
sudo chown -c root:www-data /etc/raspap/hostapd/*.sh
sudo chmod 750 /etc/raspap/hostapd/*.sh
```

Se requieren crear las tablas de enrutamiento IP de forma manual para direccionar el tráfico de la red celular hacia el servidor de red inalámbrica los comandos son:

```
echo 1 > /proc/sys/net/ipv4/ip_forward #RASPAP
iptables -t nat -A POSTROUTING -j MASQUERADE #RASPAP
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 ! -d 192.168.0.0/24 -j MASQUERADE #RASPAP
sudo systemctl restart rc-local.service
sudo systemctl daemon-reload
```

Se habilita el servicio de host para el acceso de dispositivos de red IP con el protocolo establecido en el estándar IEEE 802.11.

```
sudo systemctl unmask hostapd.service
sudo systemctl enable hostapd.service
sudo mv /var/www/html/installers/raspap.service /lib/systemd/system
sudo systemctl enable raspap.service
sudo reboot
```

### **5.3 SISTEMA DE INFORMACIÓN.**

El creciente interés en utilizar la Web como plataforma para compartir datos ha motivado la investigación sobre la publicación y el consumo de datos en la Web, si bien este tema está cobrando importancia, hasta ahora no hay muchos artículos académicos que revisen los enfoques para publicar y consumir datos en la Web. (H. D. A. do. Santos et al., 2018)

En este capítulo se describe la implementación del sistema de información, inicialmente es necesario describir los requerimientos funcionales y no funcionales:

- El sistema debe capturar e interpretar los datos de los nodos sensores
- El sistema debe incorporar la función de visualización es decir presentar el estado de las variables de manera gráfica
- El sistema debe permitir el almacenamiento de datos para su análisis.
- El sistema debe ser capaz de proporcionar un histórico de los datos arrojados por las pruebas de puesta en marcha y funcionamiento.
- El sistema debe tener la capacidad de funcionar de manera constante para la recolección de datos almacenamiento y visualización.
- Los datos deben poder ser actualizados de acuerdo a las necesidades del usuario final.
- El sistema debe hacer uso de una plataforma web.
- El motor de base de datos debe ser relacional.

## *Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

En el desarrollo se incorpora el sistema operativo raspbian inspirado en Linux debían, utilizando Node.js, y sockets.io, de forma similar a la implementada en el desarrollo descrito en (Nikolov & Nakov, 2019) el cual describe el diseño y la comunicación entre la aplicación móvil de Android, IoT Cloud y los sistemas integrados de IoT.

Para lograr la vinculación del sistema operativo con el dispositivo de comunicación BLE LightBlue Bean se revisaron los tutoriales del fabricante ubicadas en: <https://punchthrough.com/bean>. La librería para interactuar el LightBlue Bean usando Node.js quien se comporta como dispositivo esclavo está ubicada en el link [https://github.com/burrsutter/lightbluebean\\_node](https://github.com/burrsutter/lightbluebean_node). Se utiliza express para incorporar el servidor implementado sobre la *Raspberry*. Esta librería se puede ubicar en: <http://expressjs.com/es/>

La lectura de puertos seriales desde node.js es necesaria para la gestión de las tecnologías celulares y la tecnología LoRa, la librería para controlar el puerto serial desde la *Raspberry* se puede consultar en: <https://github.com/node-serialport/node-serialport>

En el desarrollo del *Frontend* es necesario incorporar la visualización del comportamiento de las variables entregadas por los nodos sensores. La librería para graficar los datos se puede consultar en: <http://www.chartjs.org/>

Para la realización de pruebas de latencia es necesario que los paquetes de envío y recepción de información tengan marca de tiempo, para esta función se debe incorporar la librería de control de timestamps ubicada en: <https://momentjs.com/>

Una solución de conectores web puede establecer una comunicación entre sistemas integrados IoT, IoT Cloud y aplicaciones móviles como en el trabajo de (Nikolov & Nakov, 2019) donde los sistemas integrados están conectados a la nube a través de una red WiFi, la aplicación móvil está conectada a la red 3G, IoT Cloud está conectado a Internet con una dirección IP real y estática. La aplicación móvil de Android sirve para controlar el sistema integrado IoT Node Mcu, se conecta al servidor Epoll en el puerto 3000 a través de sockets TCP. La tecnología utilizada para implementar la aplicación móvil es Android Studio 3.0, el lenguaje de programación es Java, la forma del protocolo de comunicación JSON. Con un concepto similar, en el trabajo descrito en este texto se utilizan conectores web, el formato tipo json, pero la aplicación se desarrolla en Javascript. El primer paso para acondicionar el sistema embebido es la instalación y actualización de npn, la vista desde el sistema embebido se puede ver en la figura 18. La comunicación por conectores web entre el cliente y servidor se realiza por medio de la librería ubicada en <https://socket.io/>, y comprende la comunicación a través de las interfaces del sistema embebido hasta la aplicación web.

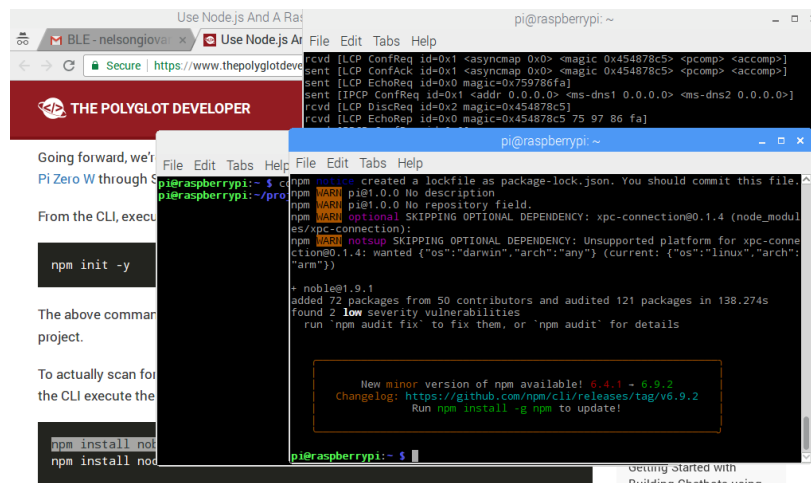


figura 18. Instalación de npm

Las aplicaciones de JavaScript se basan en un motor de JavaScript *JavaScript engine* para ejecutar su lógica central y en módulos de interfaz de usuario (UI) *user interface*, para representar su interfaz. El motor de JavaScript interactúa con el *host operating system* sistema operativo host a través de un módulo puente *javascript bridge module*. El módulo puente proporciona la API de JavaScript con recursos del sistema, también ofrece una manera de cumplir con el requisito de sistemas particulares que contienen el motor de JavaScript en un entorno aislado y controlado *sandbox*. Por ejemplo, en el navegador de Android, para evitar que las páginas web accedan a los recursos del sistema, esto ejecuta un componente *WebView* dentro del entorno limitado de *sandbox*. Cuando *WebView* se usa para ejecutar aplicaciones móviles híbridas, utiliza un marco de middleware basado en complementos como puente. Con este puente, el código JavaScript puede invocar código Java nativo para acceder a los recursos del sistema. (Mao et al., 2018). De igual forma en el desarrollo de este trabajo, se realiza el consumo del API propia del fabricante de dispositivo WiFi del estándar 802.11 y el API de geolocalización de google maps

Se debe desarrollar el *Backend* y *Frontend* en el mismo dispositivo, en este caso sobre una *Raspberry pi 3*, teniendo en cuenta que es necesario controlar de forma remota nuestro servidor instalado en el sistema embebido, se utilizó una aplicación denominada *Dataplicity*, orientada específicamente a controlar de forma remota nuestro sistema embebido. Esta aplicación necesita una configuración indicada para iniciar de manera remota o local los servicios incorporados en el sistema embebido, el acceso es similar al realizado por SSH e iniciado por HTTPS, en la figura 19 se observa la apariencia de la aplicación, accediendo al sistema embebido de forma remota, la documentación consultada para el desarrollo en *Dataplicity* se puede ubicar en: <https://docs.dataplicity.com/>. Se incorporó el concepto de agujero de gusano para la interfaz de visualización y puede consultarse en la página de *Dataplicity* en el enlace: <https://docs.dataplicity.com/docs/host-a-website-from-your-pi>

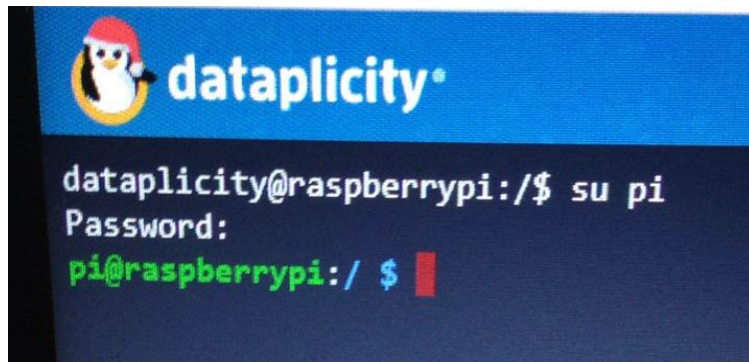


Figura 19 Apariencia de aplicación *Dataplicity*

Esta aplicación se incorpora al sistema embebido porque permite controlar, administrar y reparar la *Raspberry Pi* de forma remota, pero es necesario tener el dispositivo con salida y entrada de datos desde internet, el acceso funciona de manera similar a la conexión por SSH, pero la configuración se torna diferente especialmente para firewall y NAT.

De esta forma esta aplicación ya no utiliza una dirección ip de red local, como se realiza por SSH, sino que permite ingresar mediante una URL propia que incluye un identificador específico para cada dispositivo a controlar, además contiene un nuevo concepto denominado *wormhole* o agujero de gusano, sin embargo, es necesario tener habilitados en el sistema embebido los protocolos de SSH, SCP, VNC y demás protocolos de acceso remoto.

La apariencia inicial de la aplicación se muestra en la figura 20 y la respuesta en tiempo real de un sensor de temperatura utilizando la tecnología LoRa. En esta sección se desarrolló un menú de estándares propuestos para seleccionar la tecnología que se requiere visualizar, cuando el usuario selecciona la tecnología el software inicia un proceso de lectura y recepción de datos de los puertos del sistema embebido y de manera dinámica los vectoriza de acuerdo a su valor en unas coordenadas en plano cartesiano que forman una gráfica en tiempo real.



Figura 20 Vista de aplicación nodo sensor LoRa

En el archivo `index.js` se encuentra el código implementado sobre el sistema embebido *Raspberry Pi 3*, se debe iniciar desde consola con el comando “`sudo node index.js`”, el cual inicia la aplicación, el código desarrollado se muestra en el anexo 15

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

Este prototipo se realiza con fines académicos para permitir la evaluación y el rendimiento de algunas tecnologías emergentes como BLE, LoRa y WiFi en la figura 21 se observa la respuesta en tiempo real de un sensor de luz el cual se conecta al sistema embebido mediante la tecnología BLE bluetooth 4.0 *Low energy*, como se recomienda en la literatura, al seleccionar el menú que presenta la opción de varias tecnologías de forma simultánea la aplicación permite visualizar las tecnologías propuestas en una sola pantalla y seleccionando el botón "Todas".



## 6. RESULTADOS.

El creciente interés en utilizar la Web como plataforma para compartir información ha motivado la investigación sobre la publicación y el consumo de datos (H. D. A. do. Santos et al., 2018). El sistema de información implementado utiliza node.js sobre el sistema operativo raspbian, El transporte de datos desde los periféricos de hardware se logra mediante el uso de web sockets para la comunicación a través de periféricos, con la librería sockets.io, como en el trabajo (Nikolov & Nakov, 2019). El desarrollo de la plataforma se fundamenta en las aplicaciones basadas en un motor de JavaScript, se diseñan e implementan aplicaciones de *Backend* y de *Frontend* en la nube, se realiza el consumo de la API de particle para la tecnología WiFi y el API de Google maps para georreferenciación, como se visualiza en la figura 21.

Se diseñaron e implementaron tres nodos sensores IoT que interoperan en una plataforma web. Con el desarrollo se obtiene la visualización y almacenamiento proyectado en el diseño inicial. Durante la implementación se observa que un sitio web puede ser alojado en un servidor en la nube o en un sistema embebido y proporcionar servicios solicitados por los clientes.

Los servidores web constan de un sistema operativo, una página web y un espacio de memoria, los clientes pueden acceder a un servidor web a través de una salida a Internet (Allafi, 2017). Varios clientes *Frontend* pueden conectarse al servidor *Backend* al mismo tiempo con diversos sistemas integrados y vinculados (Zhang et al., 2018) Estas dos funciones fueron validadas en el servidor web implementado.

Se diseñó e implementó un *Backend* y *Frontend* compatible con dispositivos móviles como se visualiza en la figura 21. Para verificar el funcionamiento se ingresa al navegador y se ubica la aplicación web de *Frontend* situada en el dominio (<https://interopview.azurewebsites.net/>), los datos estáticos están cargados continuamente, En el momento que se inicia la aplicación los datos dinámicos se despliegan sobre la plataforma web, la vista de la aplicación gestionada desde el celular se ilustra en la figura 22



Figura 21 Incorporación de API Google maps para ubicación geográfica

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

Internet ha hecho posible la interacción con aplicaciones web desde cualquier lugar y utilizando cualquier dispositivo, como resultado, es posible usar una plataforma web para realizar tareas utilizando múltiples dispositivos de manera coordinada. La Web es una plataforma que ofrece muchos de los principios necesarios para gestionar con éxito las interacciones distribuidas en entornos de múltiples dispositivos. Estos principios ofrecen orientación para el diseño de aplicaciones web en las áreas de compatibilidad, utilidad, interoperabilidad y diseño universal, algunos elementos de la interfaz de usuario se pueden configurar para comunicarse con otros elementos y así determinar las interacciones que se están recibiendo. Con este enfoque, los usuarios pueden aprovechar las diferentes capacidades presentes en todos los dispositivos de su ecosistema para realizar sus tareas bajo el desarrollo web adaptativo o *responsive*. (Albertos-Marco et al., 2018). Este concepto es muy necesario en una aplicación de tipo web y se implementó una solución sobre node.js que busca la compatibilidad con dispositivos móviles como se muestra en la figura 22.



Figura 22 Vista del *Frontend* en equipo de cómputo y un dispositivo móvil.

Teniendo en cuenta que la tendencia de la descarga de aplicaciones se está viendo limitada por la gran cantidad de información que en la actualidad los usuarios manejan en sus dispositivos móviles y la necesidad de estar conectados a internet, hace que el consumo de datos esté creciendo de forma exponencial lo que apunta a desarrollar aplicaciones web que apunten a la compatibilidad con dispositivos móviles sin tener que realizar la descarga de aplicaciones, este concepto es abarcado en el diseño y desarrollo web adaptativo o *responsive*, se incorpora este concepto, luego de desarrollado para esta aplicación es verificado desde un dispositivo móvil en el cual luce como se muestra en la figura 22.

JavaScript fue diseñado para agregar interacciones a las páginas HTML, es un lenguaje de programación simple que se puede escribir directamente en documentos HTML para permitir una mayor interactividad con el usuario. La interactividad de la página web es una referencia al contenido "dinámico"; teniendo en cuenta que existen también datos estáticos y las capacidades de comunicación bidireccional entre el sitio web y el usuario final. Las características interactivas pueden adoptar una variedad de apariencias y comportamientos en un intento de cumplir con la intención del sitio web. (Luo, 2010).

En la implementación descrita en (Mangiatordi & Scenini, 2017) La accesibilidad es de tipo web porque utiliza HTML como lenguaje, admite CSS para el diseño y se puede usar JavaScript en menor grado para mejorar la experiencia del usuario, el uso real de las funciones de accesibilidad depende principalmente del emparejamiento de dispositivos y aplicaciones que adopte el usuario final, esto hace que la selección de dispositivos y aplicaciones sea crítica, creando un fuerte vínculo de dependencia entre el usuario y la tecnología. En este desarrollo se utilizó HTML y CSS para el escenario de usuario final que interactúa con el *Backend* realizado principalmente en javascript para interacciones con dispositivos, la verificación del funcionamiento de la aplicación de *Backend* se puede realizar desde postman como se visualiza en el anexo 16, el código incorporado se encuentra en los archivos raíz en la carpeta newproject y se muestra en el anexo 13.

## 7. PRUEBAS Y DISCUSION DE RESULTADOS.

---

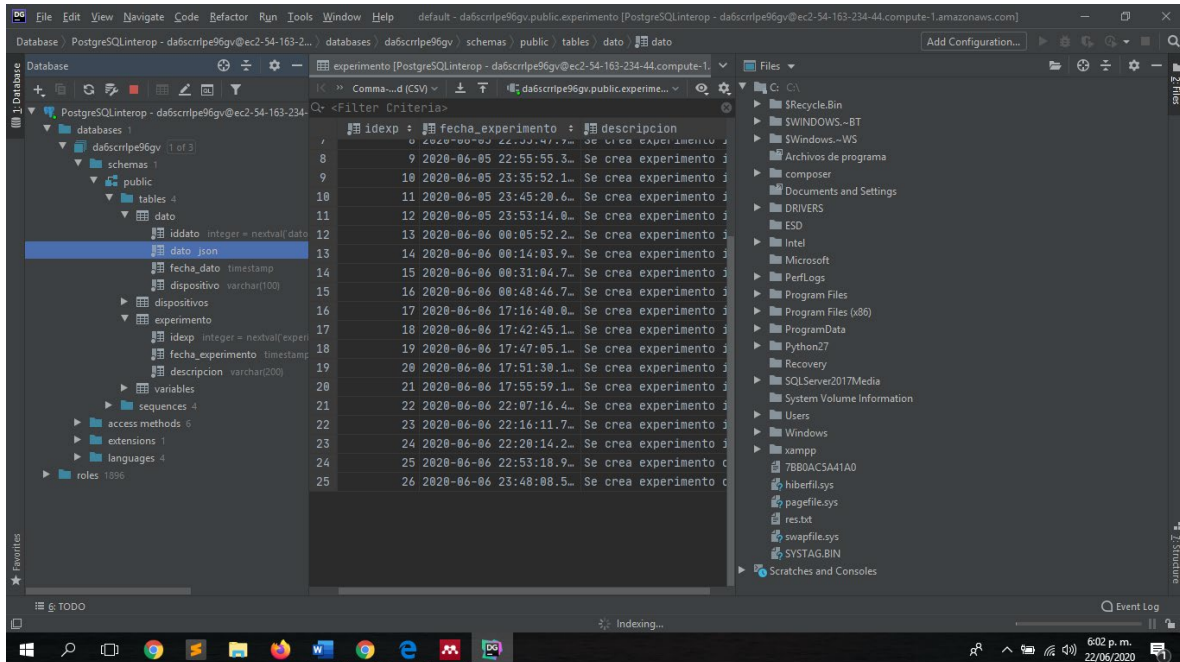
### 7.1 PRUEBAS DE FUNCIONAMIENTO.

Un criterio importante denominado (RSSI) "*Received Signal Strength Indicator*", o indicador de intensidad de señal recibida es muy significativo en sistemas de tipo IoT, Este parámetro puede tomar valores diferentes de acuerdo al fabricante y al dispositivo, en el documento de (Suganya & Pradeep, 2020) es utilizado en la simulación de un modelo matemático en Matlab para determinar la distancia entre el nodo sensor, en este caso móvil y nodo coordinador o maestro para detectar el nodo sensor en movimiento, utilizando la tecnología 6LoWPAN el valor está entre  $-10$  dBm y  $92$  dBm, donde  $-10$  dBm es la intensidad máxima de la señal recibida y  $92$  dBm es la intensidad mínima para 802.15.4. El RSSI identifica la distancia del nodo móvil desde un nodo coordinador, la fuerza de la señal recibida es alta cuando el nodo móvil está más cerca del nodo maestro y disminuye a medida que se aleja del nodo principal. La red 6LoWPAN descrita consta de dispositivos y sensores integrados que detectan datos ambientales y los entregan a una puerta de enlace o Gateway llamada 6LBR (*6LoWPAN Border Router*) que luego lo transfiere a un servidor de internet para su agregación y procesamiento. Los dispositivos receptores de tipo IoT se caracterizan por ser estáticos o móviles con recursos limitados, de capacidad de cómputo, memoria, batería y ancho de banda de comunicación generalmente estos dispositivos permiten al desarrollador consultar el nivel de RSSI (*Received Signal Strength Indicator*).

Por otro lado en el artículo de (Elgui et al., 2020) relacionan la geolocalización con el parámetro de RSSI para investigar los enfoques de aprendizaje automático que abordan el problema de la ubicación geográfica mediante algunos métodos de aprendizaje clásicos para construir un mapa de radio. Estos métodos se dividen en dos categorías: métodos basados en la probabilidad y métodos de huellas digitales. Los métodos propuestos se comparan en dos conjuntos de datos originados de la red *Sigfox* y un conjunto de datos en interiores realizado en un edificio de tres pisos. Los experimentos muestran el interés de los métodos propuestos, tanto en términos de rendimiento de estimación de ubicación, como en la capacidad de construir mapas de radio.

Las pruebas de funcionamiento planteadas para el sistema de interoperabilidad desarrollado, pretenden caracterizar el desempeño de las redes inalámbricas propuesta las cuales se caracterizan por compartir una misma puerta de enlace o Gateway, siendo el principio de interoperabilidad el principio fundamental de los sistemas de IoT y su operación en ambientes de usuario final son los espacios abiertos y edificaciones, los cuales se abordan durante el desarrollo de las pruebas. En la primera fase de pruebas se realizaron 25 tipos de experimentos que se almacenaron en el sistema de información ubicado en heroku, gestionado desde el *Backend* de tipo Web y que se puede consultar mediante la herramienta de consulta denominada *DataGrip*, como se muestra en la figura 23.

## Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11



The screenshot shows the DataGrip interface with a PostgreSQL database connected. The table 'dato' is selected, and its contents are displayed in a grid view. The table has columns for 'idexp', 'fecha\_experimento', and 'descripcion'. The data shows 26 rows of experiment records, all with the description 'Se crea experimento'. The dates range from 2020-06-05 to 2020-06-06.

idexp	fecha_experimento	descripcion
8	2020-06-05 22:55:47.9...	Se crea experimento
9	2020-06-05 22:55:55.3...	Se crea experimento
10	2020-06-05 23:35:52.1...	Se crea experimento
11	2020-06-05 23:45:20.6...	Se crea experimento
12	2020-06-05 23:53:14.0...	Se crea experimento
13	2020-06-06 00:05:52.2...	Se crea experimento
14	2020-06-06 00:14:03.9...	Se crea experimento
15	2020-06-06 00:31:04.7...	Se crea experimento
16	2020-06-06 00:48:46.7...	Se crea experimento
17	2020-06-06 17:16:40.0...	Se crea experimento
18	2020-06-06 17:42:45.1...	Se crea experimento
19	2020-06-06 17:47:05.1...	Se crea experimento
20	2020-06-06 17:51:30.1...	Se crea experimento
21	2020-06-06 17:55:59.1...	Se crea experimento
22	2020-06-06 22:07:16.4...	Se crea experimento
23	2020-06-06 22:16:11.7...	Se crea experimento
24	2020-06-06 22:28:14.2...	Se crea experimento
25	2020-06-06 22:53:10.9...	Se crea experimento
26	2020-06-06 23:48:08.5...	Se crea experimento

Figura 23 Vista en DataGrip de número de experimentos creados

DataGrip es una herramienta de desarrollo de bases de datos multiplataforma necesaria para facilitar el desarrollo, consulta y detección de errores entre plataformas. Esta herramienta está orientada a ofrecer la opción de usar las herramientas que se desean en diferentes plataformas de acuerdo a la elección realizada para cada uno de los escenarios requiriendo la instalación previa de Java. (Completo, 2017).

Existen 26 experimentos que contienen 2706 mensajes recopilados por el sistema de procesamiento principal en el formato de tipo json, que contienen la información de cada una de las tecnologías que interoperan en el sistema de información y que previamente fueron mensajes enviados por los nodos sensores, los registros de información de mensajería se pueden consultar mediante la herramienta de DataGrip, la cual proporciona una vista de la tabla general que reposa en heroku, como se muestra en la figura 24.

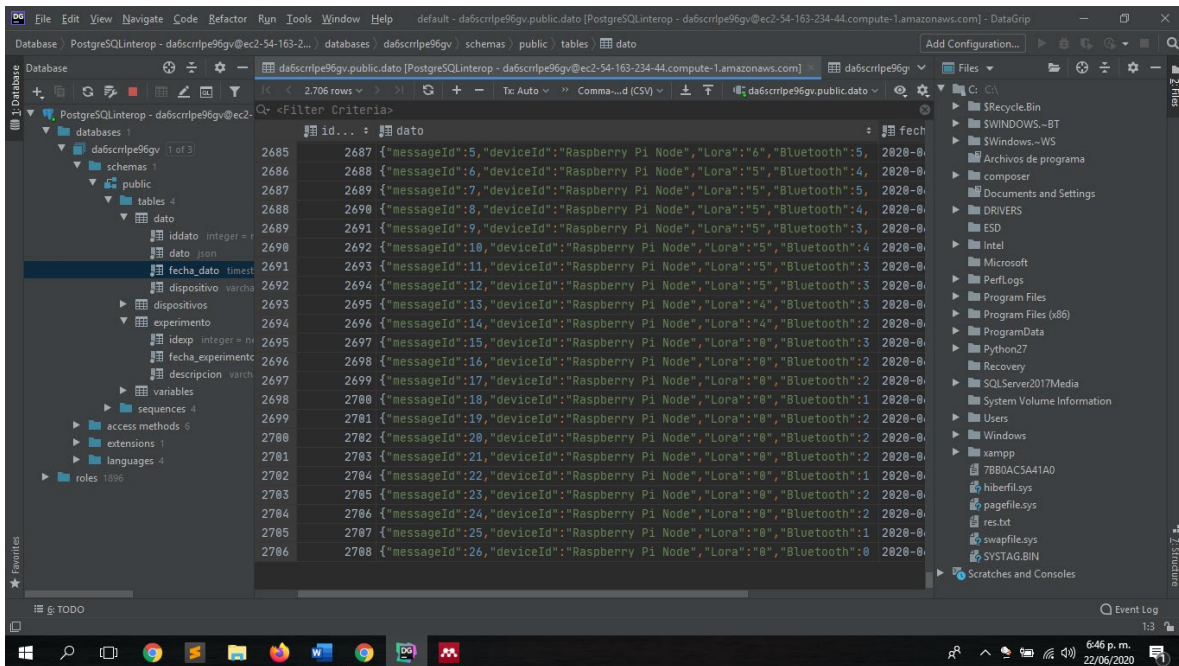


Figura 24 Vista en DataGrip de número de registros almacenados en Backend .

## 7.2 PRUEBAS DE DESEMPEÑO OUTDOOR.

### Descripción de la primera prueba:

Se propone una prueba de funcionamiento en espacio abierto en una zona urbana para analizar el comportamiento de la RSSI con respecto al distanciamiento progresivo de cada una de las tres tecnologías inalámbricas que interoperan con una tecnología celular, se tomaron 88 datos enmarcados en el experimento ID26, ubicados en los registros de datos 2706 al 2618, los datos relacionados contienen la información de las coordenadas de cada nodo sensor el cual pertenece a un estándar determinado y la RSSI de cada uno en cada punto. La ubicación geográfica de cada punto de envío de mensaje se puede ver en la figura 25 en los identificadores de color gris. La elevación del suelo del Gateway y de los nodos sensores de cada tecnología equivale a aproximadamente un metro de distancia tratando de mantener la línea de vista entre los nodos sensores y la puerta de enlace.

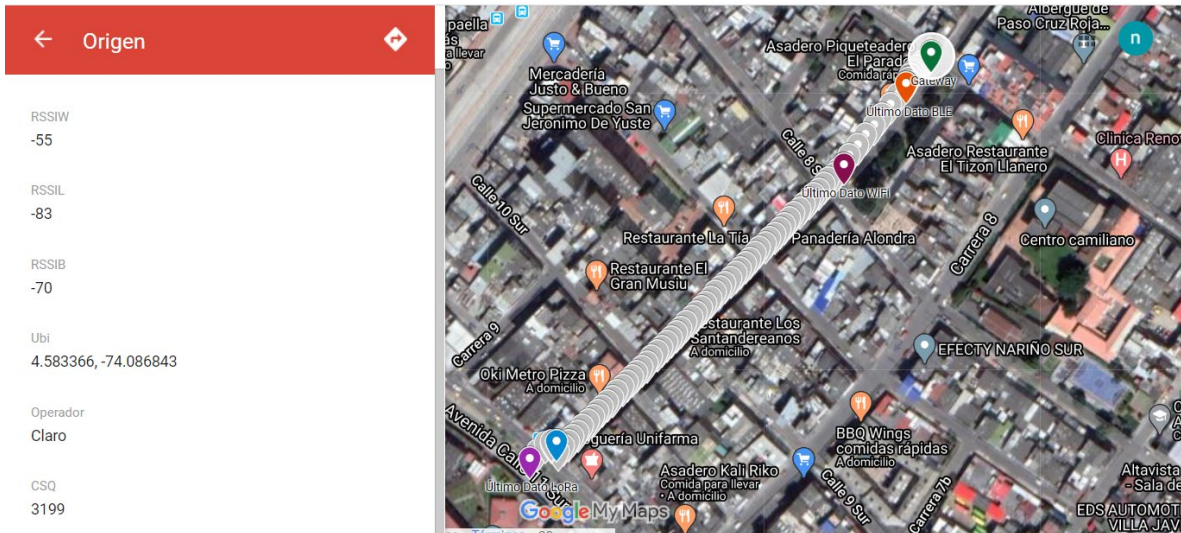


Figura 25 Vista de los puntos geográficos para la prueba de desempeño *outdoor*.

La figura 25 muestra los niveles de RSSI en dBm de cada tecnología en el punto de origen de la prueba (RSSI: WiFi: -55 dBm “marcador vino tinto”, LoRa: -83 dBm “marcador morado”, BLE: -70 dBm “marcador naranja”) separadas cada una 50 centímetros de la puerta de enlace o pasarela ubicada en el marcador verde, la puerta de enlace es la encargada entre otros de publicar los datos en tiempo real en la base de datos de tipo web, las condiciones de calidad de la señal del Gateway son: CSQ: 31.99 dBm, operador: claro, red celular conectada: LTE, como se detalla en la figura 26 la cual también muestra las coordenadas de longitud y latitud del punto.

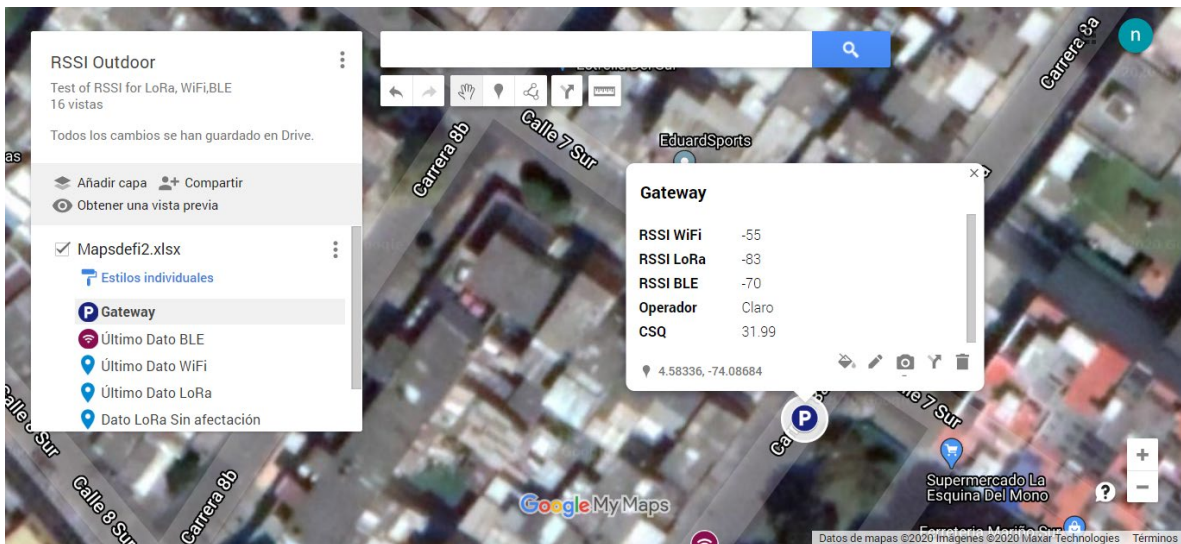


Figura 26 Datos de RSSI en el Gateway o pasarela.

El distanciamiento progresivo afecta los niveles de RSSI en las tres tecnologías. En la figura 27 se ubican los puntos de origen A y destino C en el cual se entregó el último mensaje de la tecnología

BLE, la separación de los puntos geográficos ubicados por longitud y latitud tienen una diferencia de 22 metros de distancia. La distancia se verificó durante la prueba con el instrumento de medida correspondiente, el cual arrojó 21 metros 90 centímetros.



Figura 27 Distancia máxima de envío BLE.

La figura 28 muestra los niveles de RSSI en dBm de cada tecnología y las coordenadas del punto ubicado a 22 metros del Gateway (RSSI: WiFi: -66 dBm, LoRa: -101 dBm, BLE: -91 dBm) en el cual se realizó el envío del último dato mediante la tecnología BLE durante la prueba.

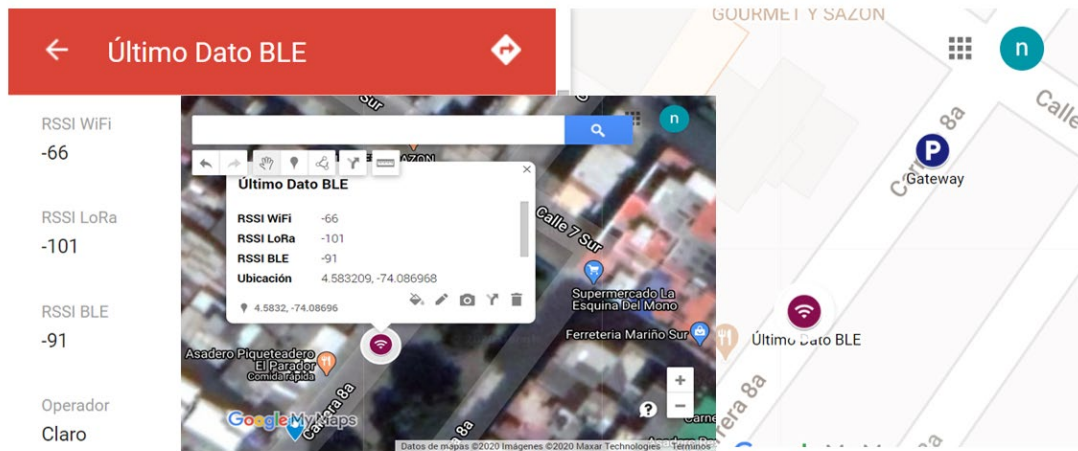


Figura 28 Niveles de RSSI a 22 metros del Gateway.

Durante el desarrollo de la prueba se prolonga el distanciamiento progresivo el cual continúa afectando los niveles de RSSI en las dos tecnologías IoT restantes, en la figura 29 se ubican los puntos de origen A y destino C en el cual se entregó el último mensaje de la tecnología WiFi, la separación de los puntos geográficos A y C tienen una diferencia de 44 metros.



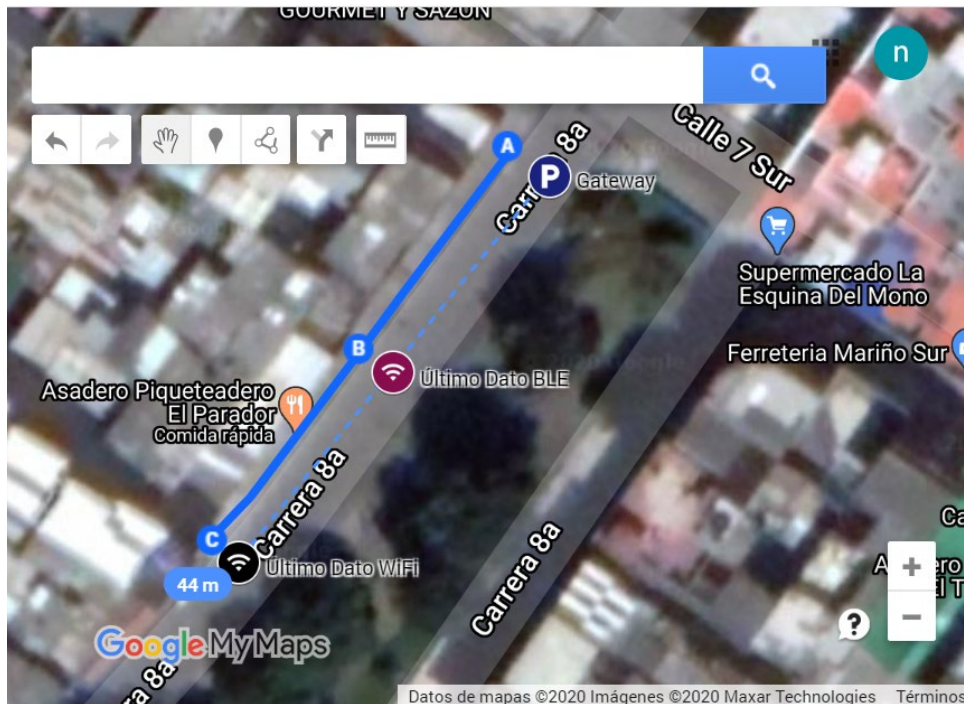


Figura 29 Distancia de último dato enviado por WiFi

En la figura 30 se muestran los niveles de RSSI en dBm de las tecnologías WiFi y LoRa en el punto ubicado a 44 metros del Gateway (RSSI: WiFi: -76 dBm, LoRa: -107 dBm).

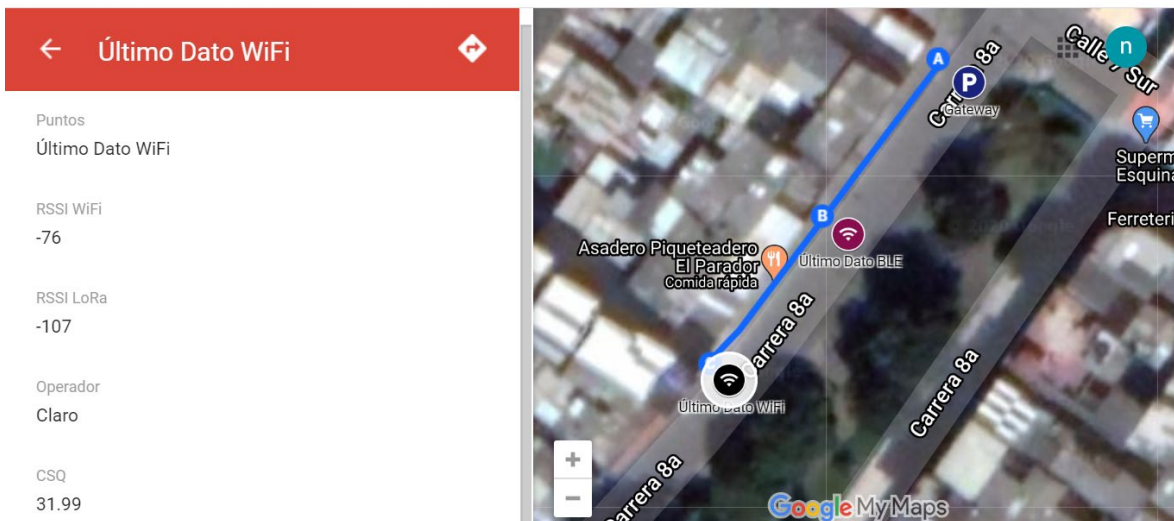


Figura 30 Parametros de RSSI a 44 metros de distancia.

La prueba finaliza analizando el alcance máximo de la tecnología LoRa, el desarrollo de la prueba evidenció que la tecnología LoRa entrega el último mensaje a una distancia de 318 metros de la pasarela como se muestra en la figura 31

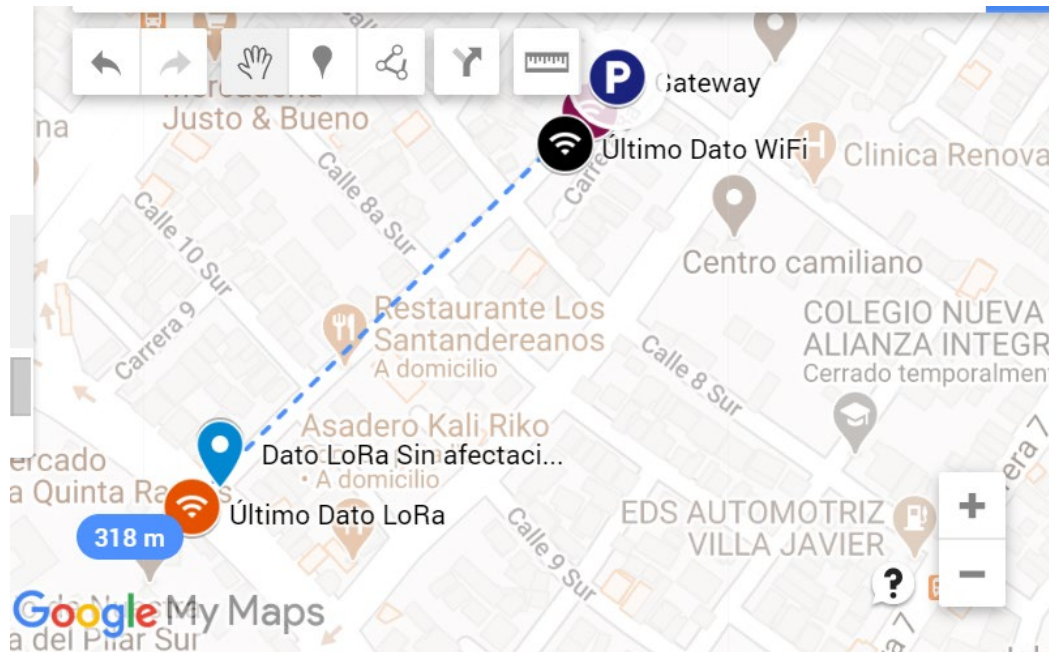


Figura 31. Último mensaje tecnología LoRa a 318 metros de distancia.

En el punto ubicado a 318 metros el nivel de RSSI para la tecnología LoRa es de -113 dBm, como se muestra en la figura 32

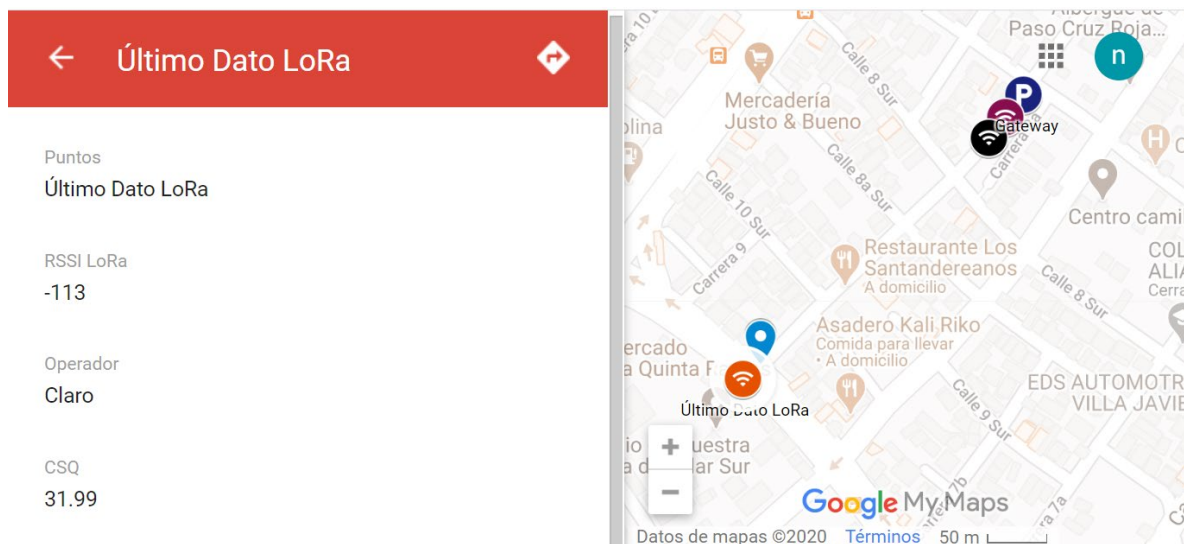
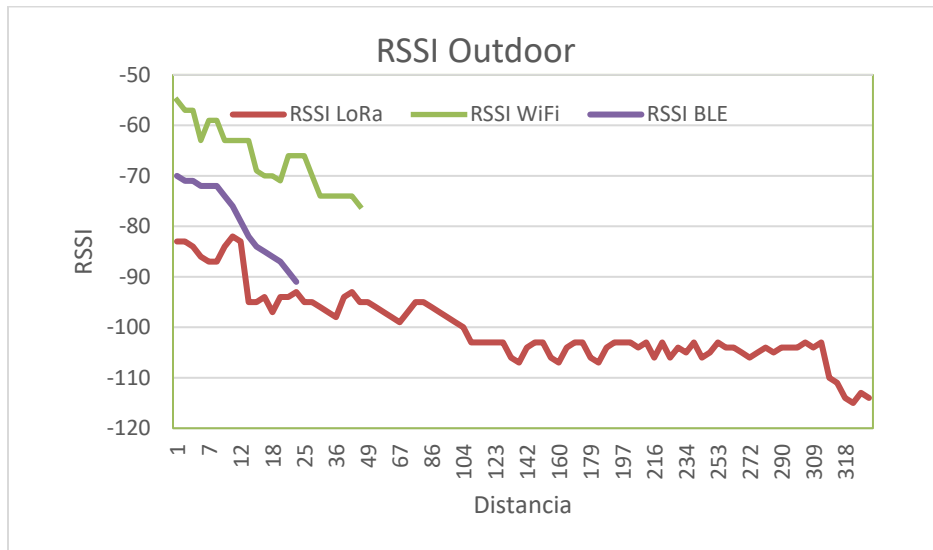


Figura 32 Parametros de RSSI a 318 metros de distancia.

El comportamiento de la RSSI por tecnologías con respecto a la distancia se puede verificar en la gráfica 1



Gráfica 1. Comportamiento de la RSSI respecto a la distancia.

Durante el desarrollo de la prueba se comprobó que la tecnología que proporciona un mayor alcance es la tecnología LoRa, sin embargo, el valor de RSSI en ese punto oscila entre -113 dBm y -115dBm. Estos valores de RSSI no son recomendados en los estándares de comunicaciones inalámbricas, extrayendo del sistema de información los registros entregados por la tecnología LoRa durante las pruebas se encontró un total de 34 datos enviados con una RSSI entre -113 dBm y -115 dBm como se muestra en la tabla 2.

Estado del mensaje			
RSSI	False	True	Total
<b>LoRa RSSI</b>	<b>7</b>	<b>27</b>	<b>34</b>
-113\ r	4	15	19
-114\ r	2	6	8
-115\ r	1	6	7
<b>Total</b>	<b>7</b>	<b>27</b>	<b>34</b>
<b>Porcentaje</b>	<b>20,59%</b>	<b>79,41%</b>	<b>100,00%</b>

Tabla 2 Análisis de datos falsos

De esta manera se evidencia que trabajando en estos valores de RSSI sobre la tecnología LoRa se proponen tres soluciones: La primera propuesta es ubicar un punto máximo de recepción de datos validos cercano a -103 dBm como el que se ubicó en la figura 33, a 313 metros de la puerta de enlace.

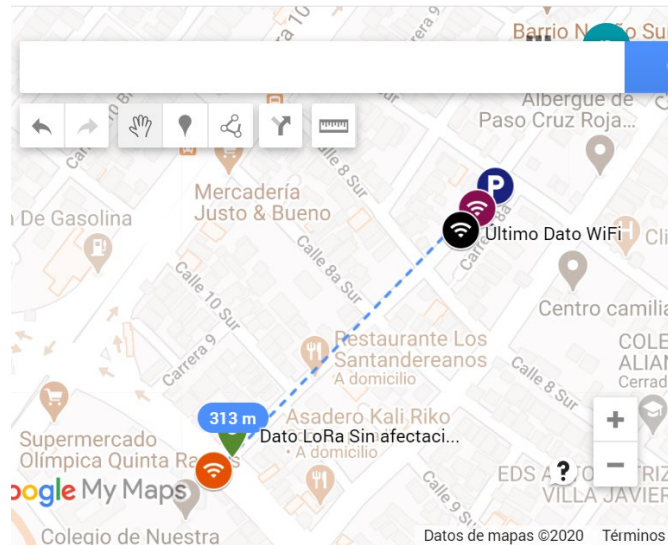


Figura 33. Último mensaje tecnología LoRa a 313 metros de distancia

Las coordenadas de punto máximo y el punto sin afectación para la tecnología LoRa están separados por aproximadamente 5 metros, pero esta separación corresponde al paso de una vía de dos sentidos como se muestra en la figura 34, durante la prueba la circulación de vehículos fue constante y la elevación se mantuvo en aproximadamente un metro, al igual que la pasarela la cual permaneció en la misma posición con un valor de CSQ de 31,99 dBm. El valor de RSSI de la tecnología LoRa para este punto es de -103 dBm identificado con el marcador de color verde.

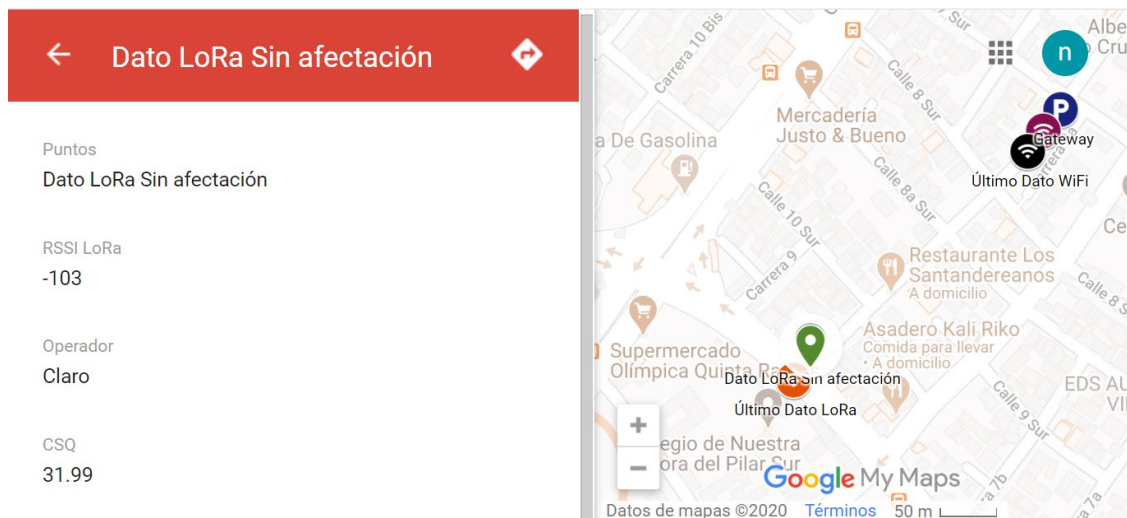
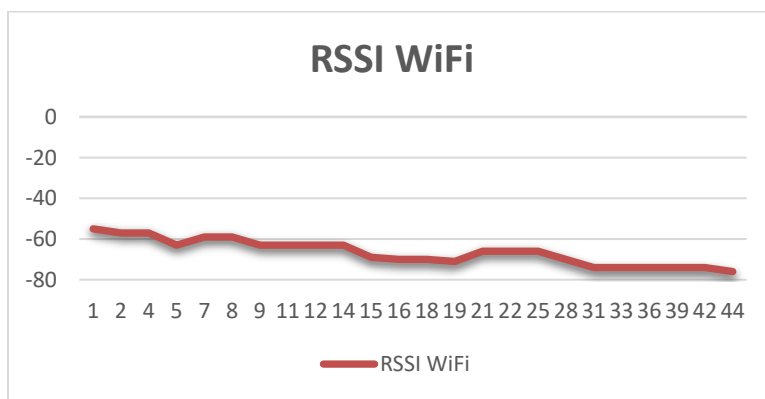


Figura 34. Último mensaje tecnología LoRa a 313 metros de distancia

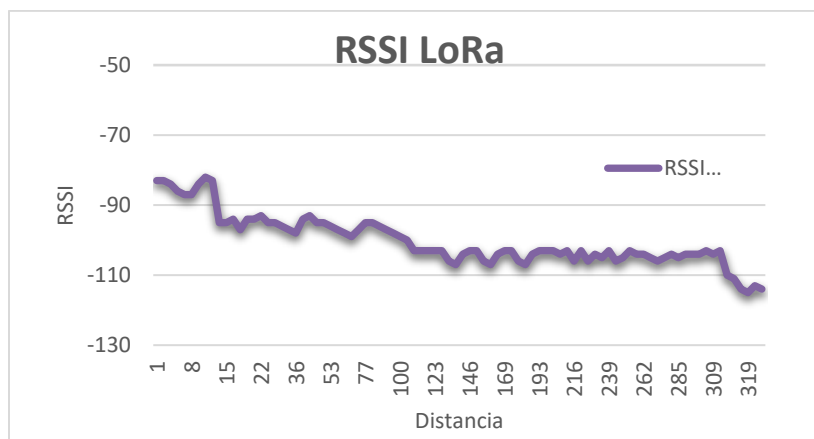
La segunda propuesta parte del porcentaje del 20,59% de datos que se recibieron como caracteres no válidos, basados en este valor se puede desarrollar un algoritmo de validación en el sistema de procesamiento de datos antes del envío de mensajes a la nube, aunado a un sistema de retransmisión de los datos perdidos en un *time stamp* de por lo menos  $T = T_0/5$ , apoyados en que el

carácter no valido corresponde a aproximadamente el 21 % de los datos, buscando consolidar el mensaje con datos validos en los tiempos requeridos por el sistema en el escenario de recepción de datos en el intervalo de -113 dBm y -115 dBm de RSSI, esta solución propuesta podría enmarcarse al concepto de *cognitive* IoT e implicaría aumentar el procesamiento en el nodo sensor y en el Gateway por lo tanto se incrementaría el consumo energético. La segunda propuesta es realizar el tratamiento mediante técnicas de clean data en la plataforma Web adoptando el concepto de procesamiento de tipo *big data* requiriendo grandes cantidades de información. La gráfica 2, identifica el comportamiento de la tecnología WiFi durante la prueba de distanciamiento progresivo, se observa que el valor de RSSI máximo se presenta en el punto inicial a una distancia de 0,5 metros y es de -55 dBm y el valor mínimo se encuentra a los 44 metros de distancia y corresponde a -76 dBm.



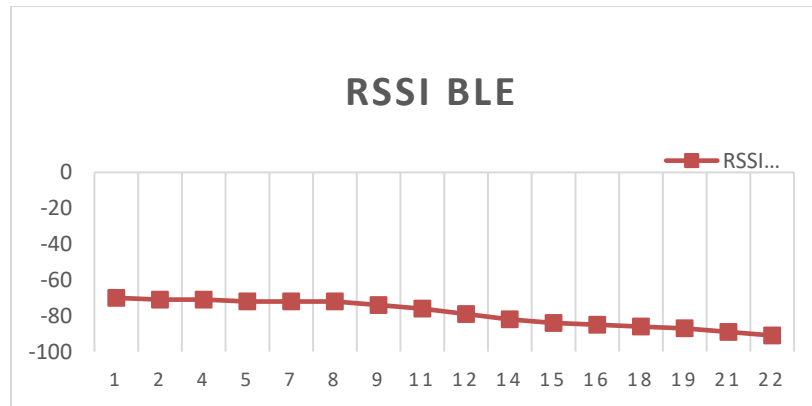
Gráfica 2. Comportamiento de la RSSI WiFi respecto a la distancia.

La gráfica 3, identifica el comportamiento de la tecnología LoRa en un alejamiento gradual del Gateway se observa que el valor de RSSI máximo se presenta en el punto inicial a una distancia de 0,5 metros y es de -83 dBm y el valor mínimo se encuentra a los 318 metros de distancia y corresponde a -113 dBm.



Gráfica 3. Comportamiento de la RSSI LoRa con respecto a la distancia.

La gráfica 4, identifica el comportamiento de la tecnología BLE el punto de máxima distancia que se encuentra a 22 metros de la puerta de enlace y en este punto obtiene el valor mínimo de RSSI correspondiente a -91 dBm y el valor máximo de RSSI se presenta en el punto inicial a una distancia de 0,5 metros y corresponde a -70 dBm.



Gráfica 4. Comportamiento de la RSSI BLE con respecto a la distancia.

#### Descripción de la segunda prueba:

Se propone una segunda prueba de funcionamiento en espacio abierto en una zona urbana para analizar el comportamiento de la RSSI con respecto al distanciamiento progresivo de cada una de las tres tecnologías inalámbricas que interoperan con una tecnología celular, pero teniendo en cuenta un escenario de obstáculos para analizar la cobertura perimetral de cada solución. La puerta de enlace se encuentra elevada a 7,7 metros indoor, separada del espacio abierto por un vidrio de 7 milímetros hacia el norte de la toma de datos, los nodos sensores están elevados del suelo aproximadamente un metro. La figura 35. muestra los niveles de RSSI en dBm de cada tecnología en el punto de origen de la prueba (RSSI: WiFi: -53 dBm, LoRa: -78 dBm, BLE: -67 dBm) separadas 0,3 metros de la puerta de enlace o pasarela.

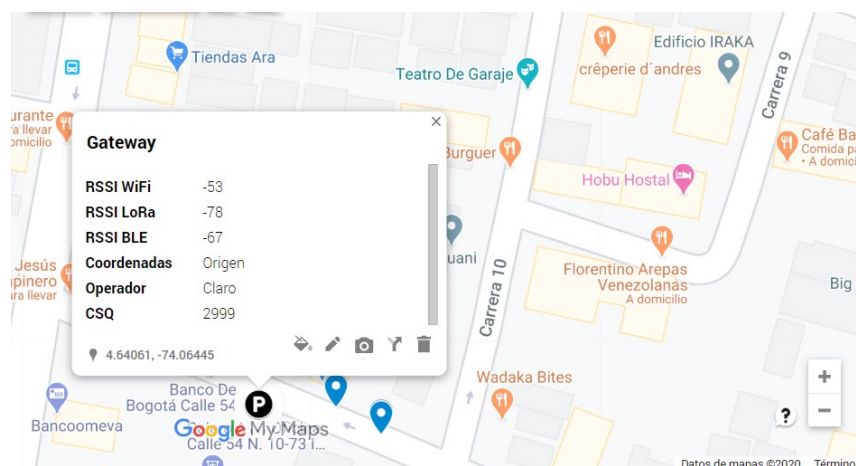


Figura 35. Ubicación de la pasarela

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

Las condiciones de calidad de la señal del Gateway son: CSQ: 29.99 dBm, operador: claro, red celular conectada: LTE, las coordenadas de longitud y latitud que detallan la ubicación geográfica del punto se pueden visualizar en la figura 35. Para esta prueba se tomaron 176 datos enmarcados en el experimento 1, ubicados en los registros de datos 2 al 176, almacenados de manera local. Los datos relacionados contienen la información de las coordenadas de cada nodo sensor el cual pertenece a un estándar determinado y la RSSI de cada uno en cada punto.

La ruta resultante de la realización de las pruebas se describe en la figura 36 en los puntos A y B, con los nodos sensores de cada tecnología distanciados del suelo aproximadamente un metro durante la duración de la prueba.

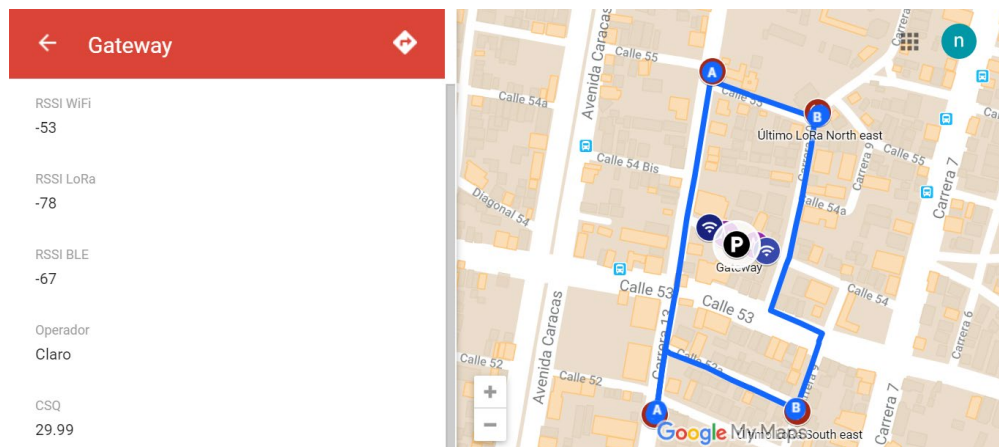


Figura 36. Ubicación e información de la puerta de enlace.

La dinámica inicial de la prueba consiste en ubicar los puntos más alejados para cada tecnología, en el caso de la tecnología BLE se encontraron los puntos de ultima transmisión de dato con el nodo sensor desplazado hacia el oriente y posteriormente al occidente, los puntos de máximo alcance se ubicaron en el mapa de la figura 37.

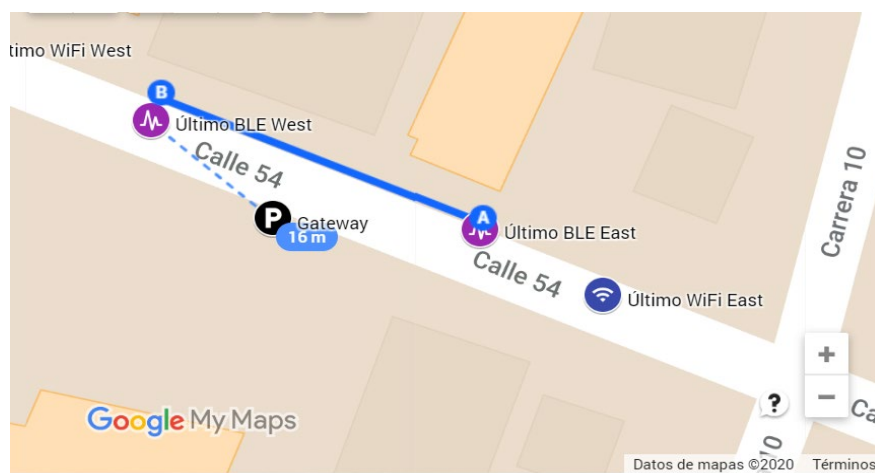


Figura 37. Ubicación de los puntos BLE.

Para la tecnología BLE el punto máximo al oriente fue de 20 metros y hacia occidente de 16 metros identificados en los puntos A y B. Los datos de RSSI para el punto de entrega del último dato al oriente son: (RSSI: WiFi: -66 dBm, LoRa: -94 dBm, BLE: -91 dBm), los cuales se relacionan en el mapa de la figura 38.

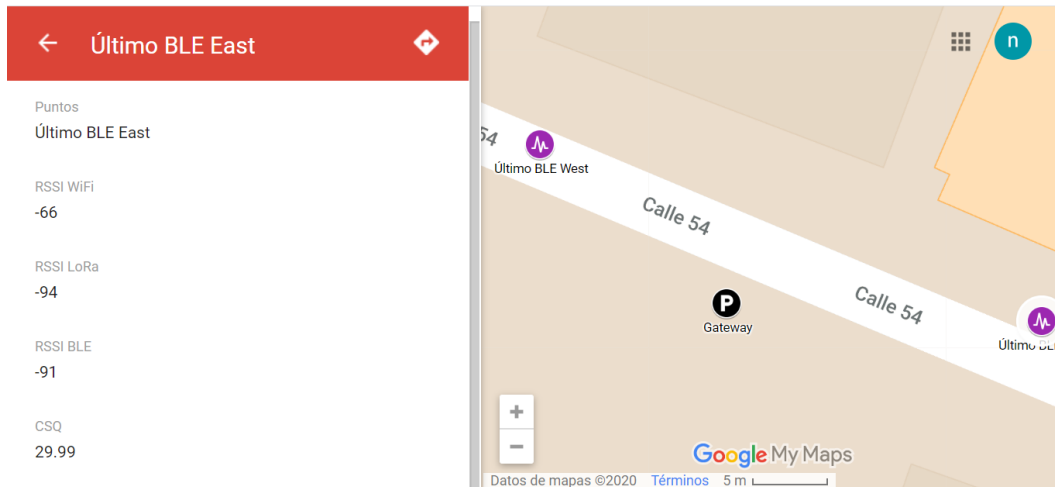


Figura 38. Ubicación de los puntos BLE.

La distancia relacionada en el mapa para el punto oriental de 20 metros y el punto occidental 16 metros, requiere realizar un ajuste debido a la elevación del Gateway, la diferencia en el eje z corresponde 6,7 metros, este ajuste se puede realizar con la ecuación 1

$$h^2 = a^2 + b^2$$

Ecuación 1 Calculo de distancia máxima de BLE

Para el cálculo de la distancia máxima del dispositivo BLE a representa la elevación del Gateway, b representa la distancia calculada desde las coordenadas satelitales y h la distancia aproximada del nodo sensor a la puerta de enlace de esta manera  $h^2 = (6,7m)^2 + (20m)^2$  para la distancia oriental, el resultado de la ecuación  $h=21,09 m$  y  $h^2 = (6,7m)^2 + (16m)^2$  para la distancia occidental, el resultado de la ecuación  $h= 17,33 m$ . Los ángulos se pueden calcular con las ecuaciones 2 y 3.

$$\varphi = \arcsin (a/h)$$

Ecuación 2 Calculo de distancia máxima de BLE

$$\theta = \arcsin (b/h)$$

Ecuación 3 Calculo de distancia máxima de BLE

Para el caso específico los ángulos corresponden a  $\varphi = 18,52^\circ$  y  $\theta = 71,50^\circ$  BLE Oriental y  $\varphi = 22,72^\circ$  y  $\theta = 67,52^\circ$  BLE occidental.



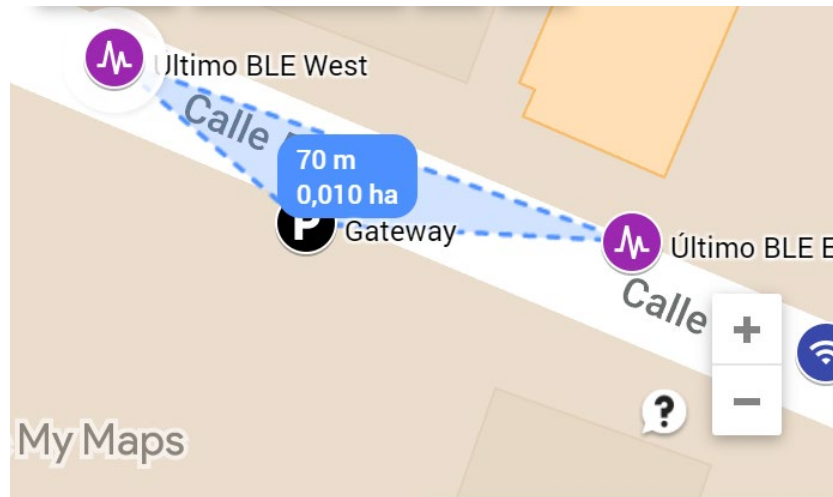


Figura 38. Ubicación de los puntos BLE.

$$P = a + b + c$$

Ecuación 4 Calculo de perímetro de cobertura máxima de BLE

$$A = \frac{ah}{2}$$

Ecuación 5 Calculo de Área de cobertura máxima de BLE

$$h = b \sin \varphi = c \sin \theta$$

Ecuación 6 Calculo de ángulos de Área de cobertura máxima de BLE

El perímetro y el área calculada para la tecnología BLE corresponde a  $P = 70 \text{ m}$ ,  $A = 99,87 \text{ m}^2$ , y los ángulos calculados de la zona de cobertura corresponden a  $\alpha = 141,3752^\circ$ ,  $\theta = 21,5422^\circ$   $\varphi = 17,0826^\circ$ .

Realizando los ajustes correspondientes a las longitudes calculadas por elevación de Gateway se obtienen valores diferentes de acuerdo a la ecuación 1 y la ecuación 2, para el perímetro, el área y los ángulos  $P = 72,42 \text{ m}$ ,  $A = 151,14 \text{ m}^2$ , y los ángulos calculados de la zona de cobertura corresponden a:  $\alpha = 124,2011^\circ$ ,  $\theta = 30,8656^\circ$ ,  $\varphi = 24,9333^\circ$ .

Para la tecnología WiFi el punto máximo al oriente fue de 33 metros y hacia occidente de 35 metros, identificados con los marcadores de color azul en la figura 39. Los datos de RSSI para el punto máximo se muestran en la figura 40. En este punto se entrega el último dato del nodo sensor de la tecnología WiFi y el RSSI corresponde a: WiFi: -74 dBm, LoRa: - 101 dBm y hacia occidente (RSSI: WiFi: -76 dBm, LoRa: - 102 dBm). En el desarrollo de la prueba se encontraron dos puntos de máximo alcance de la tecnología WiFi, sin embargo, deben ser recalculados.

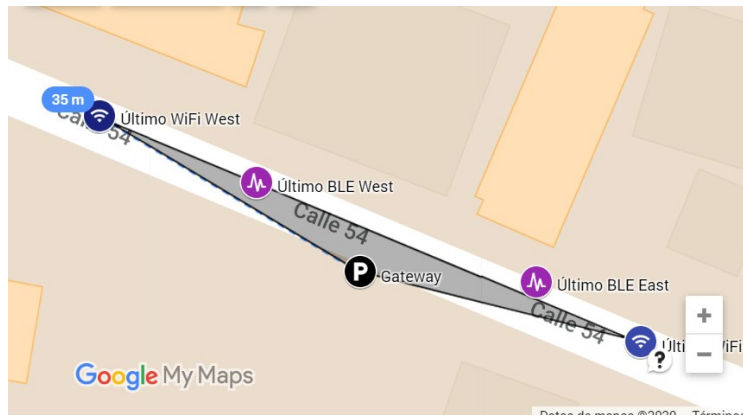


Figura 39. Ubicación de los puntos máximos de WiFi.

En el cálculo de la distancia máxima del nodo sensor de la tecnología WiFi,  $a$  representa la diferencia de elevación del Gateway,  $b$  representa la distancia calculada desde las coordenadas satelitales y  $h$  la distancia aproximada del nodo sensor al enrutador principal, de la ecuación 1 se resuelve que:  $h^2 = (6,7m)^2 + (33m)^2$  para la distancia oriental el resultado de la ecuación es:  $h= 33,67$  metros, y  $h^2 = (6,7m)^2 + (35m)^2$  para la distancia máxima del punto occidental, el resultado de la ecuación es  $h= 35,64$  metros.



De la ecuación 4 y la ecuación 5, se calculó el perímetro total y el área de cobertura para la tecnología WiFi, estos valores corresponden a  $P = 135 m$ ,  $A = 194,53m^2$  y los ángulos calculados de la zona de cobertura corresponden a  $\alpha = 160,3149^\circ$ ,  $\beta = 10,1349^\circ$   $\varphi = 9,5502^\circ$ . El valor del perímetro medido en el mapa se muestra en la figura 41.

Realizando los ajustes correspondientes a las longitudes recalculadas por la diferencia en el eje z entre el nodo sensor y el Gateway se obtienen valores diferentes basados en la ecuación 4 y la ecuación 5, para el perímetro, el área y los ángulos:  $P = 136,31m$ ,  $A = 297,10m^2$ , y los ángulos calculados de la zona de cobertura corresponden a  $\alpha = 14,4087^\circ$ ,  $\beta = 15,2717^\circ$ ,  $\varphi = 150,3197^\circ$ .

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

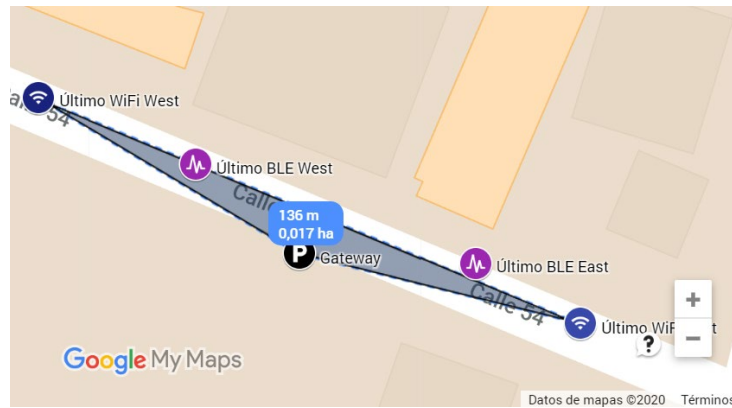
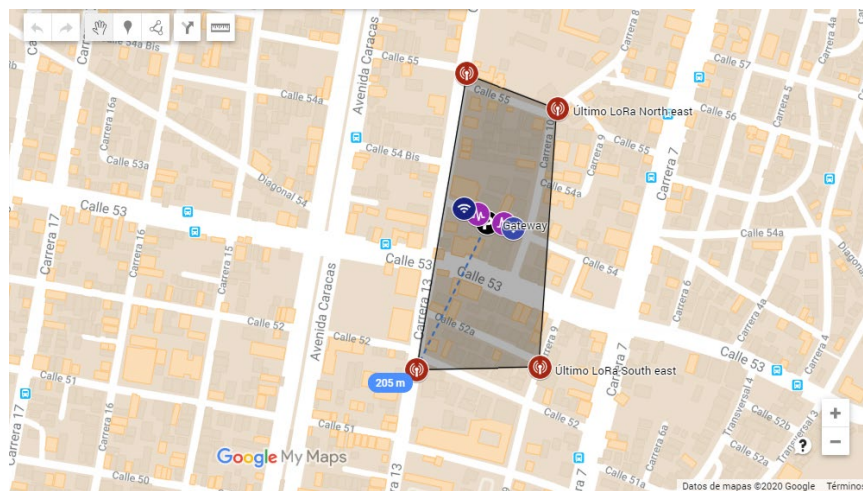


Figura 41. Ubicacion de los puntos WiFi.

La última parte de la documentación de la prueba 2 comprende el análisis de la tecnología LoRa para espacios urbanos con obstáculos. La tecnología LoRa presentó cuatro puntos de máxima transmisión desde la pasarela, el primer punto se ubica al noroccidente a 189 metros, el segundo punto se ubica al nororiente a 168 metros, el tercer punto se encuentra 193 metros al suroriente y finalmente a 205 metros al suroccidente el último punto, como se muestra en la figura 42.



Los datos de RSSI para los puntos de entrega de último dato de la tecnología LoRa son: al noroccidente RSSI- 113 dBm, nororiente RSSI- 112 dBm, suroriente RSSI- 113 dBm y suroccidente RSSI- 114 dBm.

La figura obtenida del área de cobertura para la tecnología LoRa es un rectángulo de lados no uniformes, para el cálculo de perímetro y determinación de área de cobertura es posible separar la forma en dos triángulos para calcular el área mediante las coordenadas, sin embargo, para incluir la diferencia de alturas entre el nodo sensor y la pasarela es necesario separar la forma en cuatro triángulos distribuidos desde el origen hacia el norte, sur, oriente y occidente.

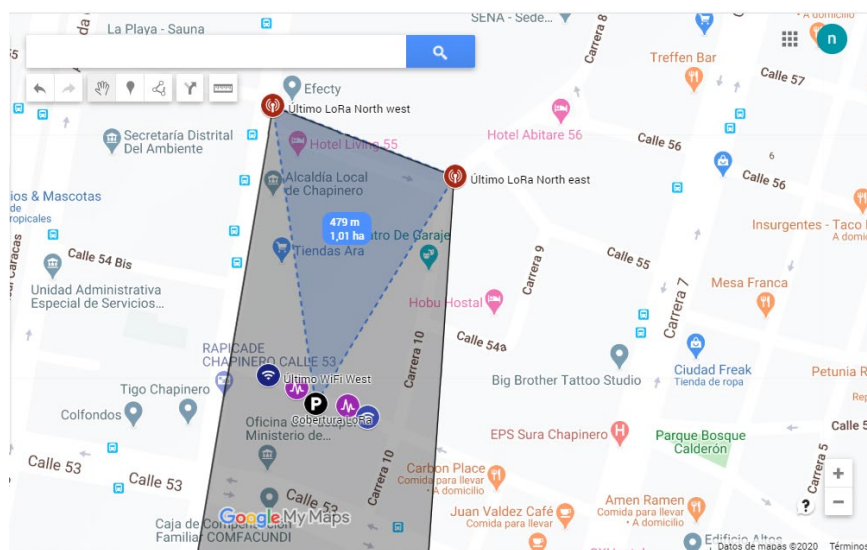


Figura 43. Cobertura hacia el norte tecnología LoRa.

El triángulo formado hacia el norte se puede ver en la figura 43 y presenta un perímetro medido por coordenadas de 479 metros. El cálculo de ajuste para cada forma y los ángulos formados para cada triángulo se muestran en la tabla 3

Punto	Distancia	Ajuste	Área:	Perímetro	$\alpha$	$\beta$
Noroccidente	189	189,12	633,15	384,82	2,03	87,96
Nororient	168	168,13	562,8	342,83	2,28	87,75
Surorient	193	193,12	646,55	392,82	1,99	87,98
Suroccidente	205	205,11	686,75	416,81	1,87	88,12

Tabla 3 Distancias y ángulos calculados para formas de cobertura.

En la figura 44 se muestra el área de cobertura total de la tecnología LoRa. La figura 45 muestra un perímetro de 979 metros medido por las coordenadas y la RSSI en el punto sur oeste.

Teniendo en cuenta la diferencia de alturas existente entre el sistema de control principal y el nodo sensor LoRa, es necesario realizar los ajustes a las formas mediante las ecuaciones 1 y 6, los ajustes para cálculo de área basados en las ecuaciones: 1, 4 y 5 y el cálculo de ángulos formados mediante la ecuación 2, 3 y 6, para cada punto, los resultados se presentan en la tabla 4.

Cobertura	Base (m)	A (m)	b (m)	Perímetro (m)	Área (m <sup>2</sup> )	$\alpha^\circ$	$B^\circ$	$\gamma^\circ$
Norte	122	189,12	168,13	479,25	10088,37	39,3869	79,6287	60,9844
Sur	155	193,12	205,11	553,23	14171,96	45,6891	63,0675	71,2434
Oriente	325	193,12	168,13	686,25	12777,06	128,0917	24,0257	27,8827
Occidente	377	205,11	189,12	771,23	10854,88	145,9671	17,7276	16,3052

Tabla 4 Distancias y ángulos calculados para cada zona de cobertura.

El perímetro total de la cobertura de la tecnología LoRa medido por las coordenadas satelitales en el mapa de la figura 44 muestra un resultado de 979 metros y una RSSI de -114 dBm para el punto máximo de cobertura.

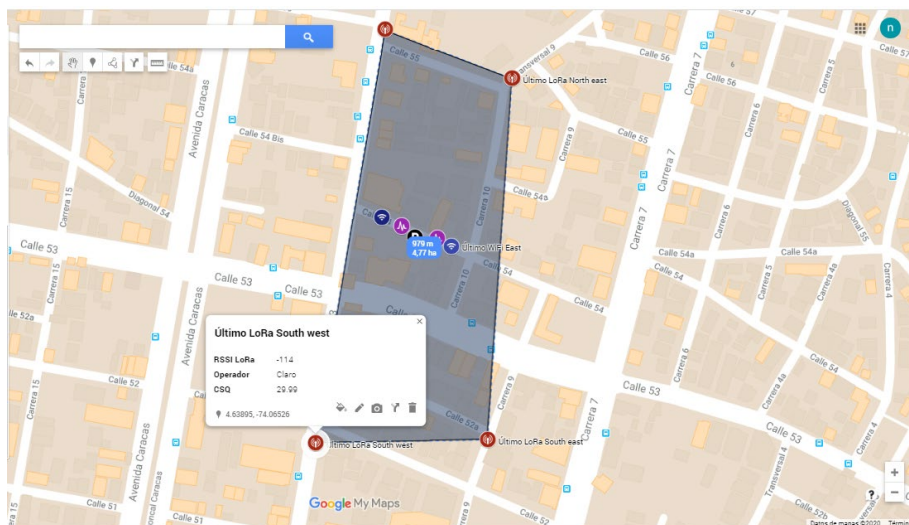


Figura 44. Zona de cobertura tecnología LoRa.

### 7.3 PRUEBAS DE DESEMPEÑO INDOOR.

Para la realización de pruebas en ambientes cerrados se propone ubicar un edificio de 5 pisos que cuente con un punto físico en común en cada piso. Este punto físico en común debe tener la posibilidad de la ubicación de los nodos sensores de las tecnologías propuestas con el objetivo de analizar la cobertura de cada una basados en el comportamiento de la RSSI frente a los obstáculos horizontales.

Se ubican inicialmente los nodos sensores separados 0.2 metros cada uno de la puerta de enlace y se coloca en funcionamiento el servicio web de *frontend*. Este procedimiento se realiza de forma remota para este caso desde la aplicación putty que permite ingresar al dispositivo controlador principal *Raspberry pi 3* mediante una sesión por SSH por el puerto 22 apuntando a la dirección IP local 192.168.0.1 como se muestra en la figura 45, posteriormente se realiza el proceso de autenticación de usuario y se ejecuta el comando desde la consola: `sudo node index.js`, desde la carpeta de archivos como se muestra en la figura 46.

## Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11

Para verificar el funcionamiento se ingresa al navegador y se ubica la aplicación web situada en el dominio (<https://interopview.azurewebsites.net/>), En el caso particular del desarrollo de esta prueba la aplicación muestra los niveles de iluminación medido de forma paralela en las tres tecnologías y la apariencia muestra en la figura 47,

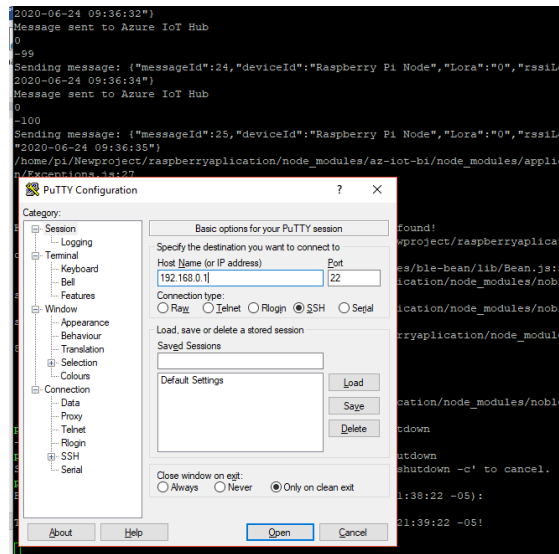


Figura 45. Ingreso remoto al controlador principal.

El gestor de base de datos seleccionado para el almacenamiento es PostgreSQL, también denominado Postgres, este motor de base de datos se implementó en la aplicación por su versatilidad, por ser estructurado y multiplataforma

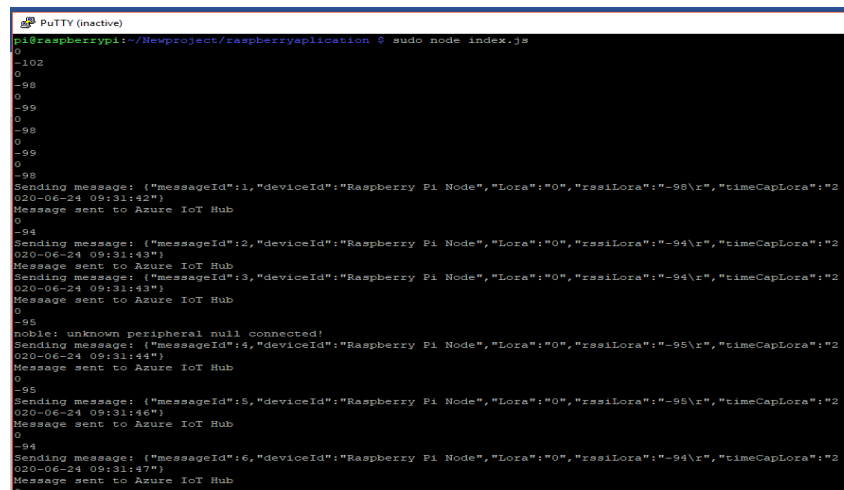


Figura 46. Puesta en marcha de la aplicación.

Los registros en la base de datos se encuentran específicamente ubicados en las posiciones 2806 hasta el 3289 como se muestra en la figura 48. Las consultas se pueden realizar por el *Frontend* o se

## Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11

logran consultar y descargar la base de datos completa desde la herramienta Data Grip para el análisis de datos

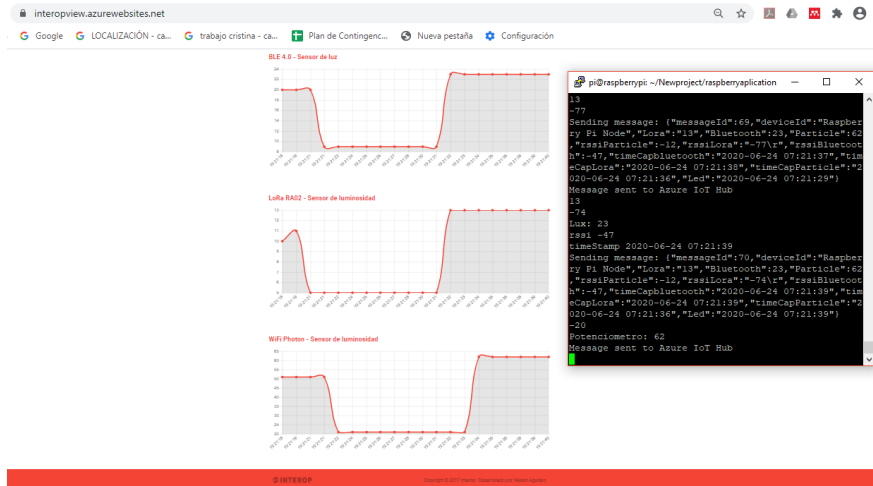


Figura 47. Arranque remoto de la aplicación.

Se realiza seguimiento de la aplicación en tiempo real desde la consola para monitorear el comportamiento de los niveles de RSSI de cada tecnología, frente al distanciamiento de los nodos sensores, sin embargo el almacenamiento se realiza en tiempo real y está disponible en una base de datos de tipo web para el análisis posterior, para el caso particular de las pruebas *indoor* de alcance y latencia se utilizaron 483 mensajes de tipo json que contienen la información de cada tecnología y sus características.

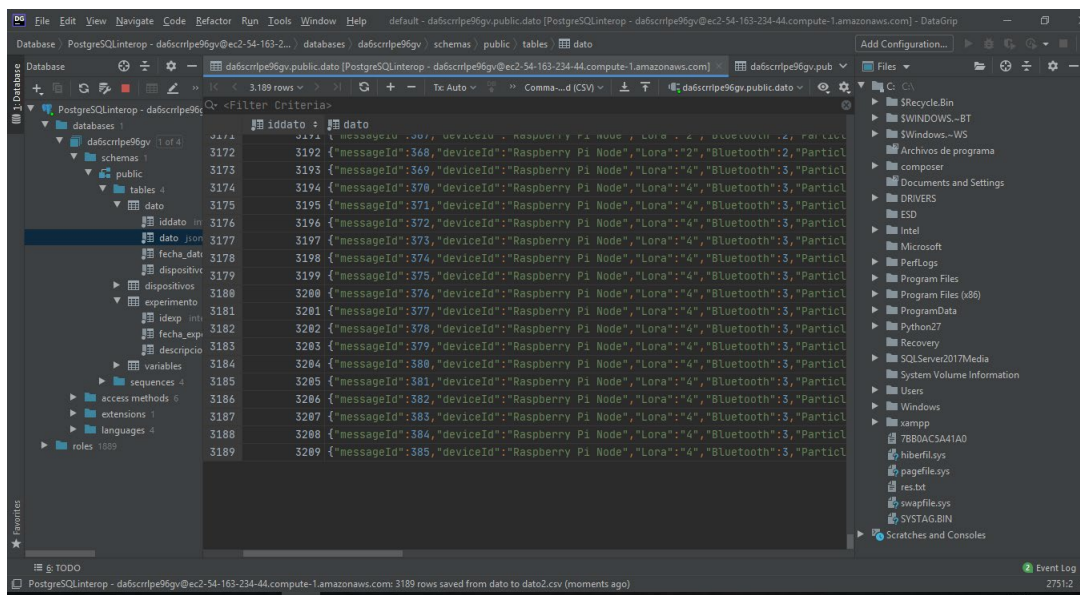
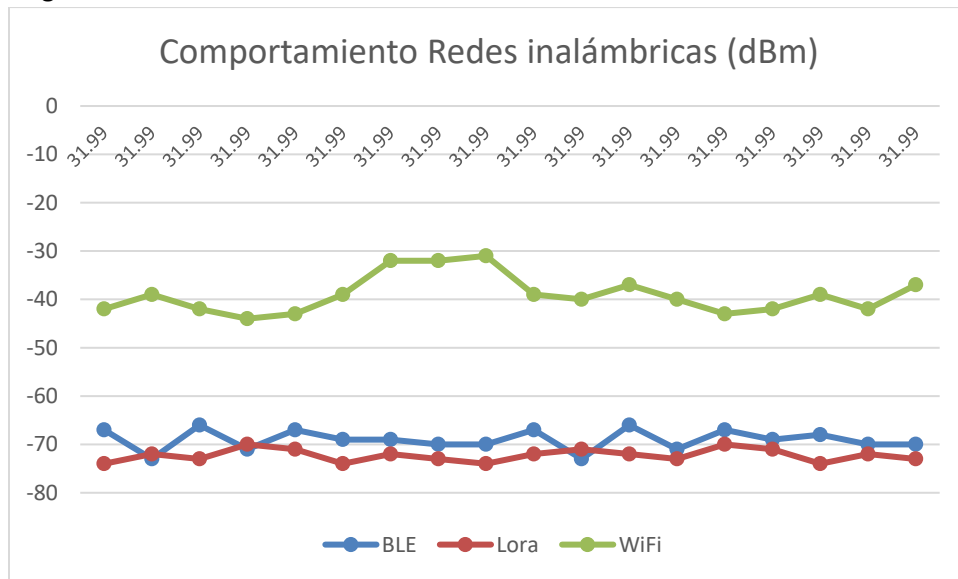


Figura 48. Visualización de base de datos desde Data Grip.

consultando los datos obtenidos durante el desarrollo de la prueba en el sistema de información se tienen los siguientes resultados:



Gráfica 5. Comportamiento de las redes inalámbricas en espacios cerrados.

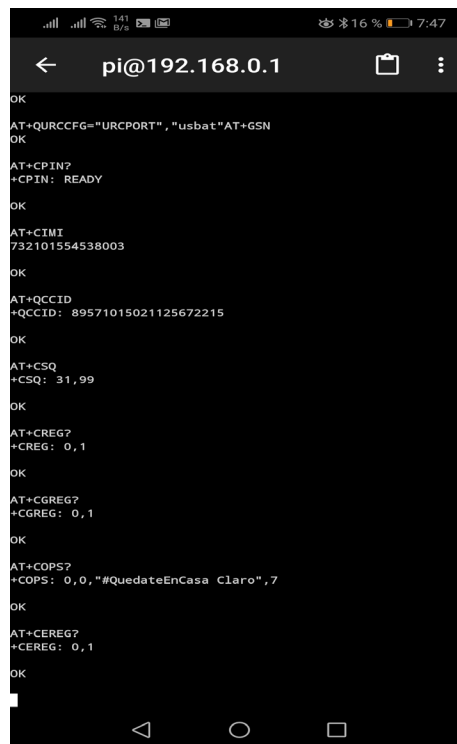
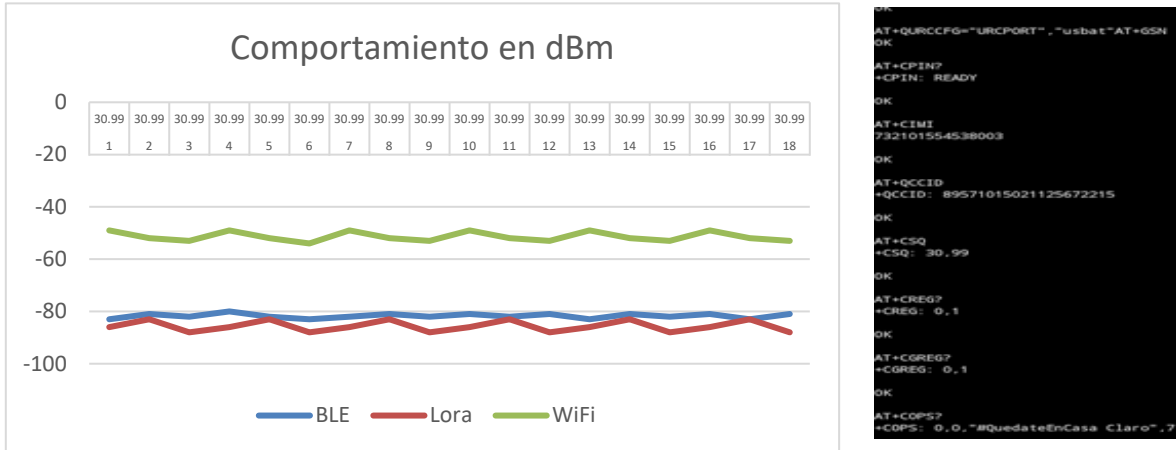


Figura 49 Módulo de consulta para redes celulares.

Para esta prueba se tomaron 18 registros encontrando que la CSQ =31.99dBm de la señal celular se mantiene constante y conectada a la red LTE del operador claro como se muestra en el módulo de

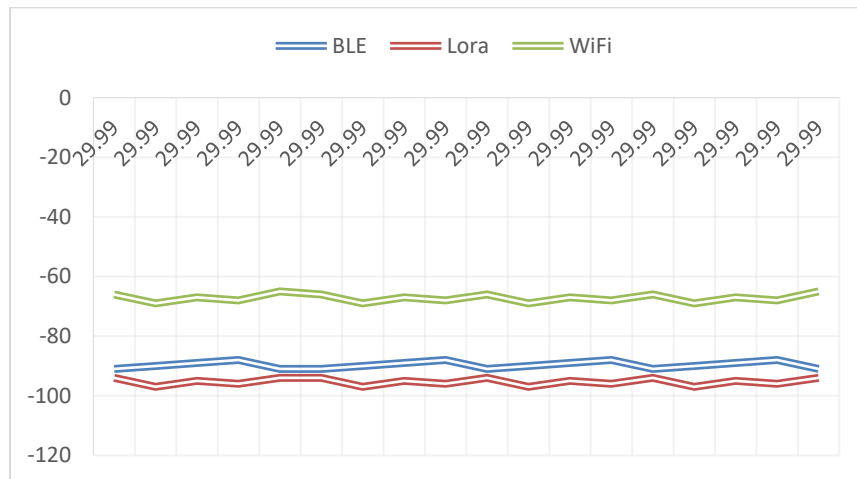


consulta de la figura 49, la prueba fue efectuada en el piso 5 del edificio sin obstáculos a 0,6 metros de distancia entre la puerta de enlace y los nodos sensores. El comportamiento de las señales en dBm se puede ver en la gráfica 5.



Gráfica 6. Comportamiento de las redes inalámbricas en espacios cerrados

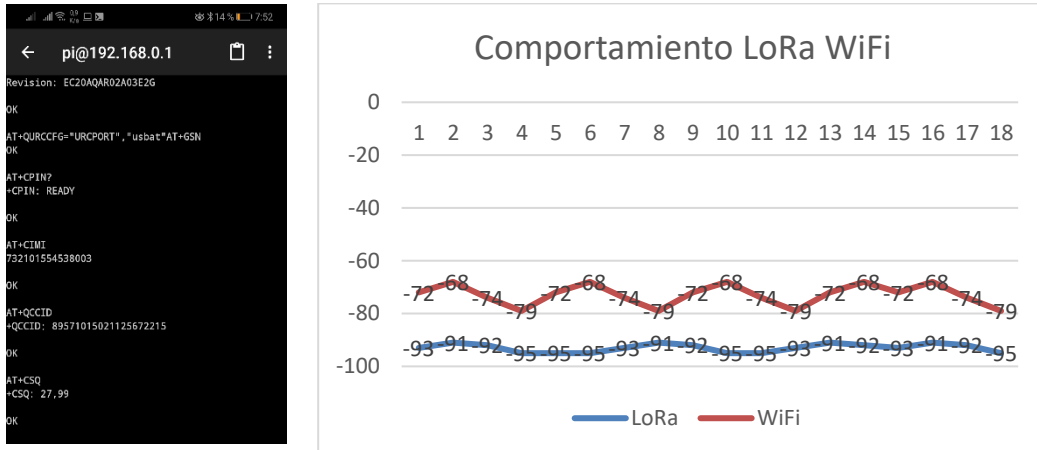
Posteriormente los nodos sensores permanecen fijos en el piso 5 a una elevación de un metro y se procede a desplazar el Gateway, la distancia total de separación corresponde a 317 cm, con un obstáculo de 21 cm representado en una placa de concreto de separación de niveles, el comportamiento de las señales se muestra en la gráfica 6 encontrando la CSQ de la tecnología LTE constante conectada al operador de claro con una CSQ de 30.99 dBm.



Gráfica 7. Comportamiento de RSSI y CSQ en (dBm).

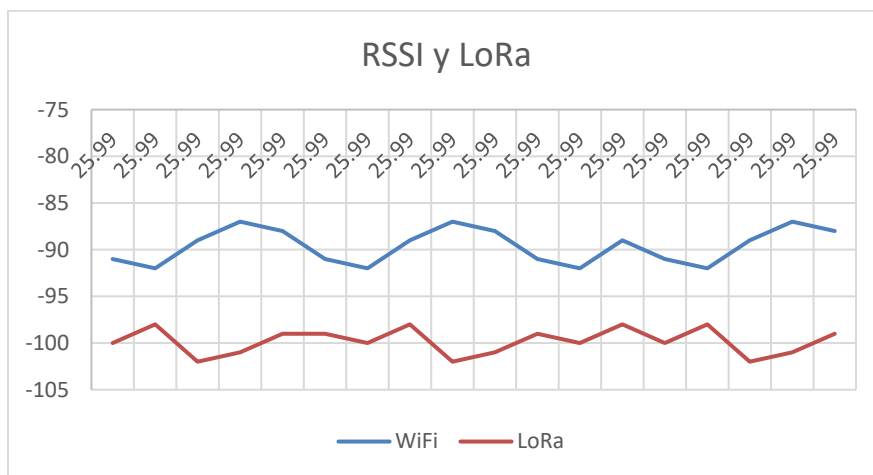
La tercera parte de la prueba consiste en alejar el Gateway a una distancia de 457 cm, y agregar un segundo obstáculo vertical de 21 cm correspondientes al grosor de una segunda placa de concreto, el comportamiento de las tecnologías inalámbricas se puede visualizar en la gráfica 7, en la cual se recibe el último dato de la tecnología BLE.

Conservando el obstáculo se procede a distanciar el Gateway a 634 cm, el resultado de las mediciones se muestra en la gráfica 8, la CSQ de la red celular es de 27.99 dBm, conectada al operador claro



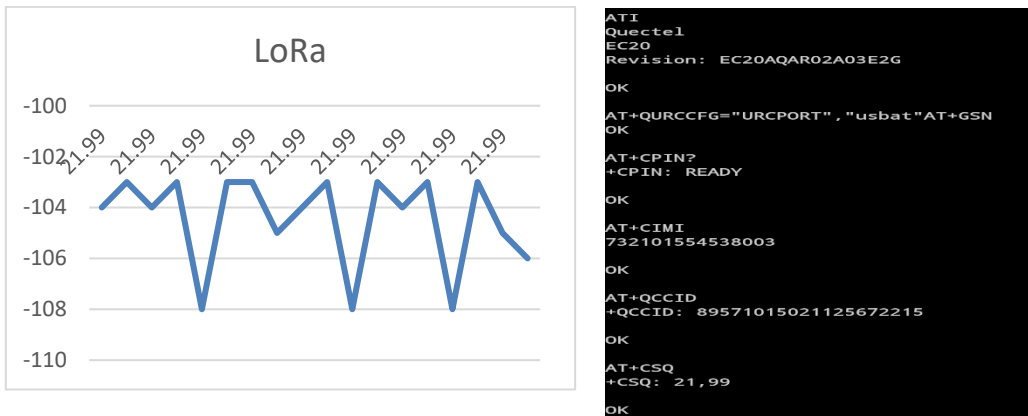
Gráfica 8. Comportamiento de RSSI y CSQ en (dBm).

Se procede a alejar el Gateway 785 e introducir un nuevo obstáculo de 21 cm en esta distancia se enviaron 43 registros desde el nodo sensor y se procesaron y almacenaron en la base de datos 18 registros, lo que indica que se perdieron el 58 % de los datos, En esta distancia del Gateway se recibió el último dato del nodo sensor de la tecnología WiFi



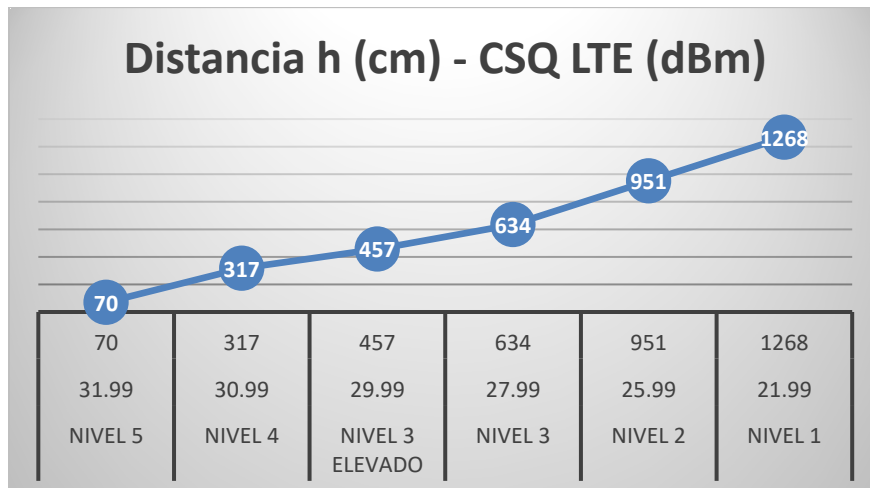
Gráfica 8. Comportamiento de RSSI y CSQ en (dBm).

La última etapa de la prueba consiste en colocar un obstáculo adicional completando 1268 cm con cuatro obstáculos en total, el comportamiento en este punto se muestra en la gráfica 9, la interoperabilidad en este punto se mantiene por la tecnología LTE que envía el dato al servidor y mantiene un CSQ de 21.99dBm y la tecnología LoRa que se mantiene en niveles que oscilan entre -103dBm y -108 dBm.



Gráfica 9. Comportamiento de RSSI y CSQ en (dBm).

El comportamiento durante la prueba de la tecnología LTE en el parámetro de CSQ se muestra en la gráfica 10, los resultados graficados son los indicadores obtenidos en cada etapa contra la distancia sobre el eje vertical.



Gráfica 10. Comportamiento de RSSI y CSQ en (dBm).

## 7.4 PRUEBAS DE LATENCIA.

### Descripción de la prueba

Se pretende calcular cuál de las tecnologías presenta un menor tiempo de respuesta, interoperando con las herramientas Web de visualización y almacenamiento. Para esta prueba se capturaron y almacenaron 490 datos.

Se propone un escenario como el ilustrado en la figura 50, y se puede ver en anexos fotografía 1 el escenario antes de ser aislado de luz externa, está compuesto por los siguientes elementos:

- Un dispositivo controlado por la aplicación Web que gestione un led emisor de luz RGB.
- Tres sensores de iluminación cada uno conectado a una tecnología con protocolo de comunicación I2C.
- Un espacio aislado de iluminación.

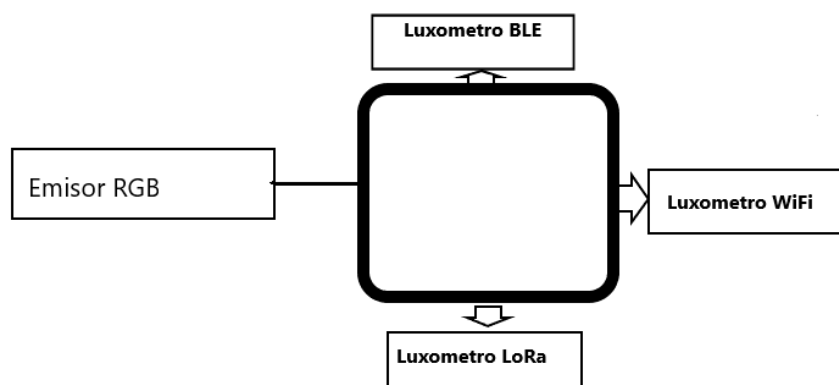


Figura 50 Escenario Propuesto pruebas de latencia.

Esta prueba pretende medir el tiempo empleado para gestionar un dispositivo actuador desde la aplicación Web en este caso un led RGB que cambia aleatoriamente de color cada 10 segundos y tres sensores de iluminación enfrentados al emisor de luz para detectar los cambios en los niveles de iluminación de cada color. El tiempo empleado por cada uno de los dispositivos es almacenado en la variable “*timecap*” de cada tecnología, de esta manera se conoce el tiempo de envío del mensaje al actuador y el tiempo de llegada del mensaje a la aplicación web como se muestra en la figura 51.

La respuesta de cada tecnología a los cambios de color se almacenó en 45 registros los cuales responden al cambio de color emitido por el actuador en 45 ocasiones los tiempos de respuesta por tecnología se muestran en la figura 12.

# Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11

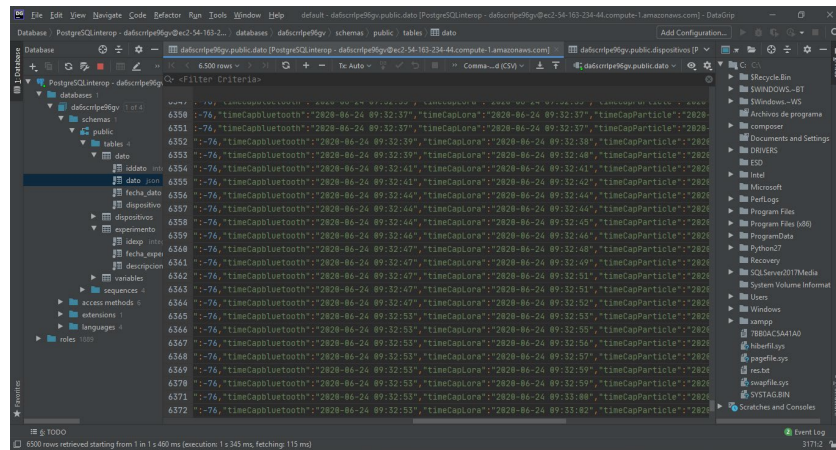
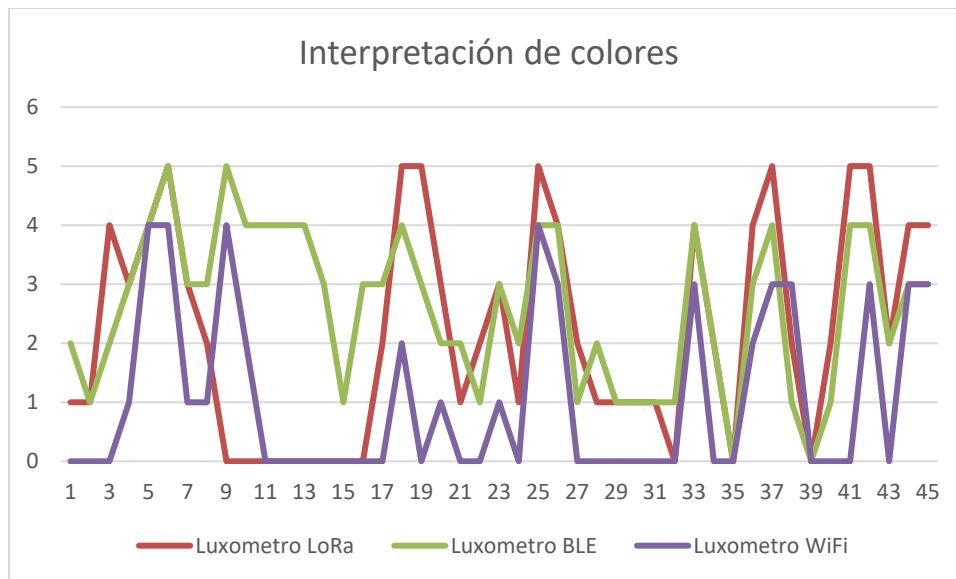
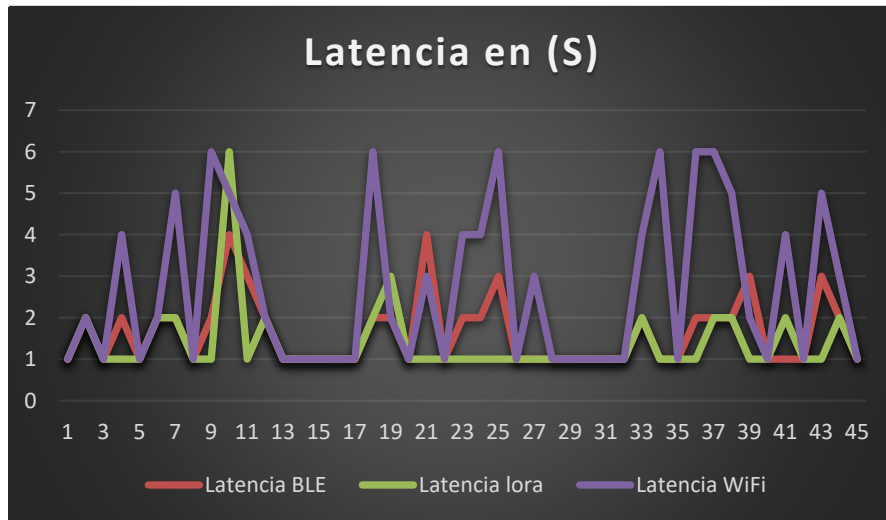


Figura 51 Datos almacenados para pruebas de latencia.

De acuerdo a los registros analizados, los patrones de reconocimiento al cambio de colores en el tiempo  $T = t_0 + 1$  se muestran en la gráfica 11.



Gráfica 11 Respuesta al cambio de colores por tecnología



Gráfica 12 Latencias por tecnología

La desviación estándar para la latencia representada en el registro tiempo de captura de cada tecnología se presenta en la tabla 5

Latencia por tecnología	Promedio de Latencia BLE	Promedio de Latencia LoRa	Promedio de Latencia WiFi
timeCapbluetooth	1,644444444	1,377777778	2,666666667
timeCapLoRa	1,644444444	1,377777778	2,666666667
timeCapParticle	1,644444444	1,377777778	2,666666667
<b>Total general</b>	<b>1,644444444</b>	<b>1,377777778</b>	<b>2,666666667</b>
Desviación por tecnología	Latencia BLE	Latencia LoRa	Latencia WiFi
timeCapbluetooth	0,829993307	0,860467338	1,906925178
timeCapLoRa	0,829993307	0,860467338	1,906925178
timeCapParticle	0,829993307	0,860467338	1,906925178
<b>Total general</b>	<b>0,829993307</b>	<b>0,860467338</b>	<b>1,906925178</b>

Tabla 5 Calculo de desviación por tecnología

En la primera fase de la prueba se observa una latencia mayor en la tecnología WiFi, el componente que se utiliza está diseñado y fabricado por *particle* y tiene la particularidad de tener una plataforma de gestión e interoperar con plataformas en la nube por medio de un API que se implementó en el *Backend* de la aplicación, se eligió por la robustez del componente comparado con los existentes en el mercado nacional, pero la API implica que el componente reporta los datos en un servidor la arquitectura de interoperabilidad entre plataformas se puede representar en el diagrama de la figura 52.

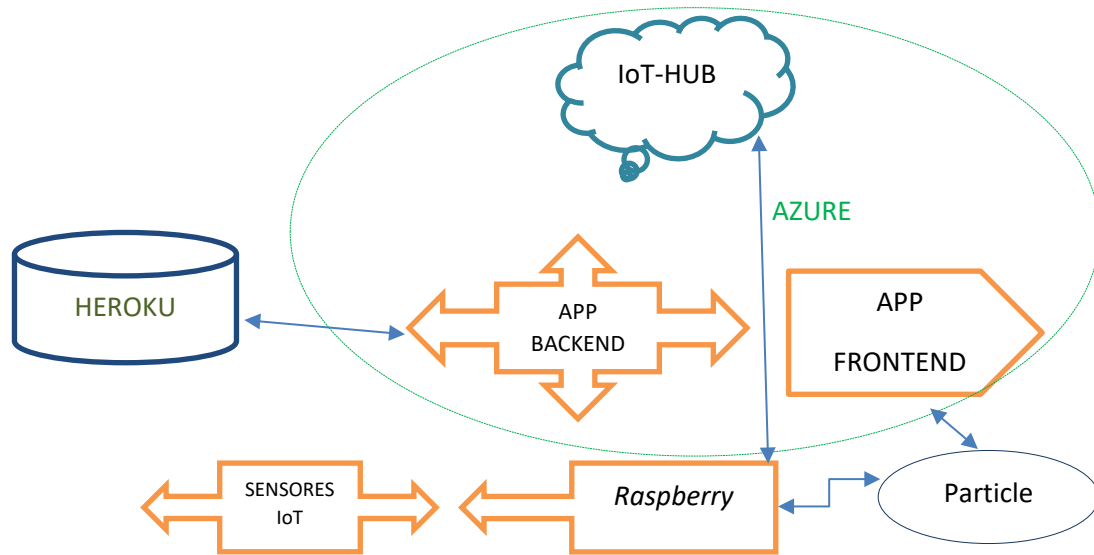
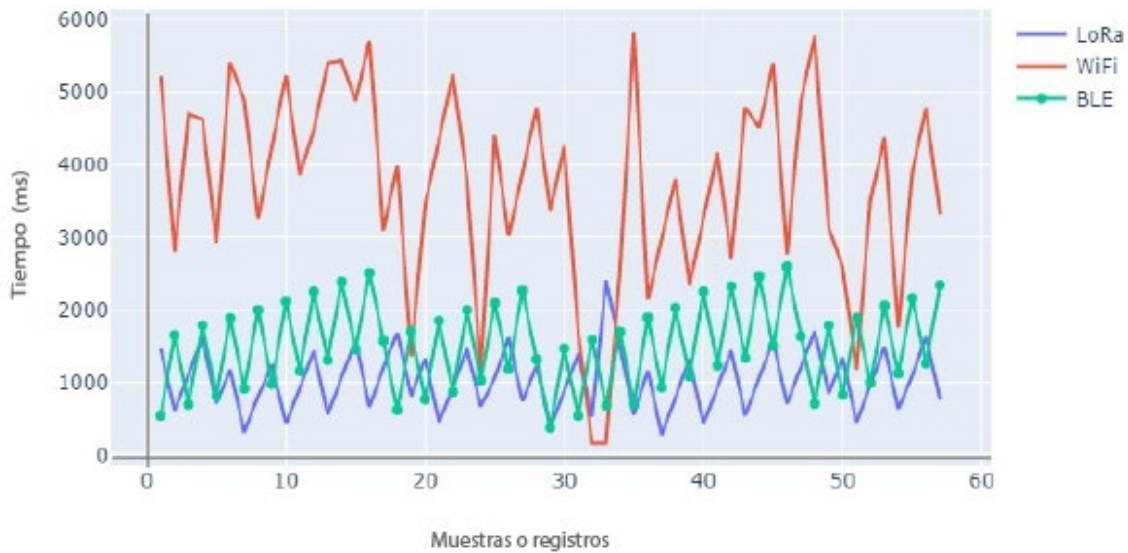


Figura 53 Arquitectura de Aplicación.

La diferencia en el procesamiento de la información radica en que las tecnologías son gestionadas por el controlador de hardware principal en este caso, la *Raspberry PI*, recibe los datos directamente de las tecnologías BLE por el la conectividad directa con su interfaz de bluetooth, la tecnología LoRa que se gestiona por un puerto USB al igual que las tecnologías celulares y el nodo sensor de WiFi gestionado por la interfaz WiFi, pero con la singularidad de tener que enviar a la plataforma de IoT de particle lo que representa una ruta adicional. Vale la pena esta ruta adicional pues, aunque la latencia aumente, la ruta adicional proporciona una mayor seguridad en el nodo sensor y una mayor versatilidad en las aplicaciones en la nube, en el anexo 12 se muestra una vista de los componentes del sistema de información.

En la segunda fase se pretende realizar una prueba de mayor precisión de latencias por tecnología, de esta manera, se procede a efectuar el ajuste de la aplicación para realizar la captura del tiempo timestamp tanto del envío del comando de activación al actuador, como del dato recibido en la plataforma en mili segundos. Se analizaron un total de 249 datos que contienen la información de tiempo de cada tecnología en 57 registros de cada nodo sensor, el comportamiento de la latencia de cada tecnología se describe en la gráfica 13



Gráfica 13 Latencias por tecnología medida en milisegundos

En la primera y la segunda fase de la prueba se observa una latencia mayor en el nodo sensor de la tecnología WiFi, el nodo sensor utiliza un componente que está diseñado y fabricado por particle que cuenta con una plataforma propia de gestión que se configura con el id del dispositivo y se consulta mediante el API dispuesta por el fabricante para interoperar con plataformas en la nube, esto significa que el dato debe consultarse del servidor de particle mediante el API key y el id del dispositivo estos dos parámetros proporcionan al dispositivo un nivel alto de seguridad frente a los demás nodos sensores, posteriormente el dato debe gestionarse mediante la *Raspberry Pi* y publicarse en la aplicación web.

En la tercera fase de las pruebas, se propuso realizar un ajuste en la aplicación desarrollada que consiste en ejecutar la consulta del dato entre las plataformas web, esto implica que el Gateway solo proporciona el acceso a internet del nodo sensor WiFi pero el dato no se gestiona en la puerta de enlace, es decir, el dato se consume directamente desde la plataforma web del fabricante y se publica y almacena en la aplicación de *frontend*. Esta modificación implica perder la centralización del nodo sensor WiFi, pues no es necesario iniciar el IoT hub en la *Raspberry Pi*, cuando el nodo sensor esté energizado inicia el envío de datos a la plataforma web comportándose como un dispositivo independiente, pero interoperando con el sistema mediante el API de integración web. Esta modificación reduce significativamente la latencia del nodo sensor WiFi, a pesar de estar consultando el dispositivo de dos formas diferentes, la integración entre plataformas se realiza desde la plataforma de particle y se puede visualizar en el anexo 16. Los resultados obtenidos de 2433 registros de dos dispositivos IoT y condensados en 154 datos. El tiempo de respuesta se muestra en la gráfica 14. El resumen de resultados de las tres fases de las pruebas realizadas se detalla en la tabla 6

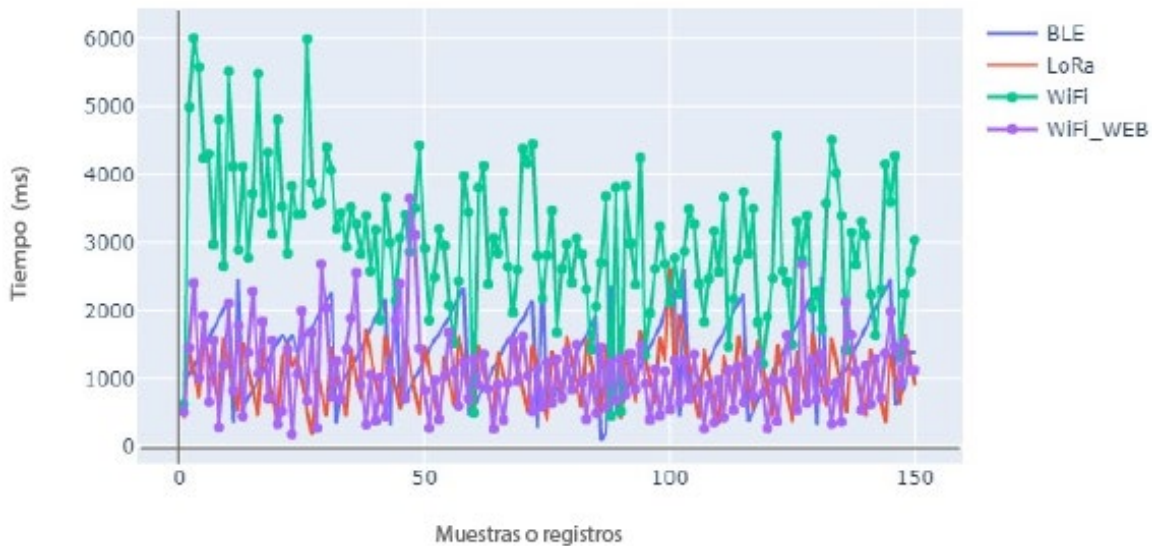


*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

TABLA 6  
PROMEDIO MÁXIMOS MÍNIMOS Y DESVIACIÓN POR TECNOLOGÍAS

FASES	Latencia BLE	Latencia LoRa	Latencia WiFi	Latencia WiFi Aj
<i>Promedio f1</i>	1,64444	1,37777	2,66666	
<i>Minimo f1</i>	1	1	1	
<i>Máximo f1</i>	4	6	6	
<i>Desviación f1</i>	0'8299	0'8604	1,90592	
<i>Promedio f2</i>	1493,684	1033,280	3708,457	
<i>Minimo f2</i>	375	279	164	
<i>Máximo f2</i>	2405	2405	5820	
<i>Desviación f2</i>	447,8982	607,2598	1369,795	
<i>Promedio f3</i>	1391	1038	3039	1126
<i>Minimo f3</i>	101	169	456	183
<i>Máximo f3</i>	2603	2627	6012	4314
<i>Desviación f3</i>	596,8865	427,4004	1042,956	703,4622

Tabla 6 Resumen pruebas de latencias por nodo sensor



Gráfica 14 Latencias por tecnología en mili segundos con dos dispositivos IoT hub.

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

La interoperabilidad de tecnologías se implementó para la realización de pruebas de latencia en la fase 1 y fase 2, mientras que la interoperabilidad de plataformas se ejecutó y validó en la fase 3 para mejorar la latencia de la tecnología WiFi, el resumen del proceso de diseño implementación y evaluación del sistema se visualiza en la figura 54

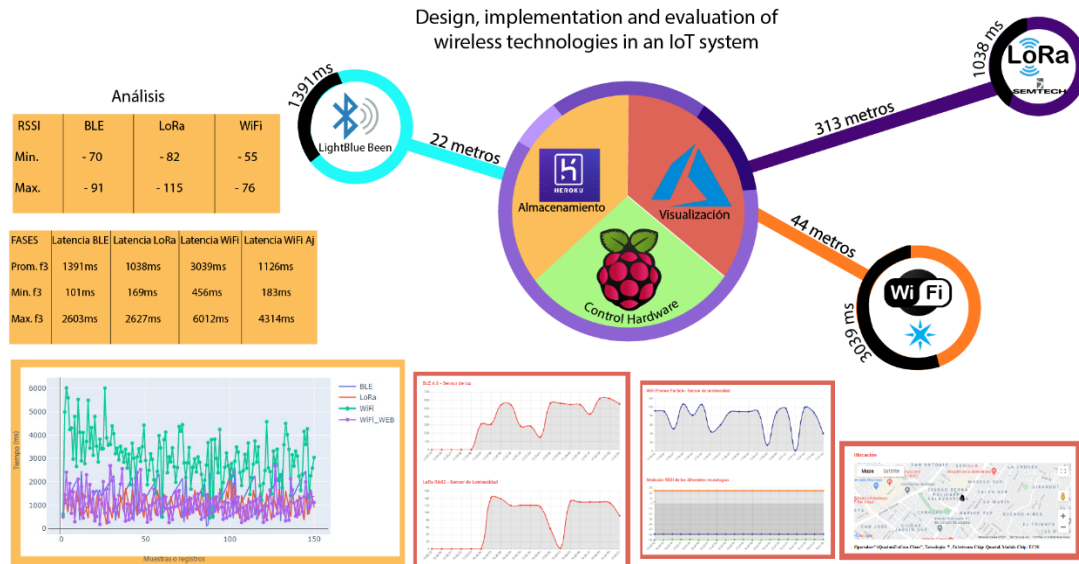


Figura 54 Resumen general diseño implementación y evaluación.

El lenguaje de programación más representativo en la implementación fue node.js, se eligió por ser un lenguaje modular que permite un fácil escalamiento y reutilización de código, adicionalmente tiene las prestaciones de otros lenguajes de programación como php para la gestión de bases de datos. Ventajas que fueron necesarias en la migración de todos los servicios a la nube.

## 8. CONCLUSIONES Y FUTUROS TRABAJOS.

---

### 8.1 CONCLUSIONES.

Las tecnologías analizadas BLE, LoRa y WiFi presentaron una latencia promedio medida en milisegundos de 1391ms, 1038ms y 3039ms, respectivamente. Consumiendo el dato directamente de la plataforma web del fabricante por el *Backend* desarrollado, con esta variación se encontró una latencia promedio para el nodo sensor WiFi de 1126ms representando una mejora de 1913ms.

De los resultados se concluye que no hay una tecnología mejor que otra, se debe emplear la más apropiada de acuerdo con la necesidad puntual frente a los factores de distancia, velocidad, calidad y cantidad de información.

- Las soluciones de tipo *cloud computing* son muy apropiadas para este tipo de desarrollo, la escogencia de esta arquitectura, en este desarrollo fue inicialmente su alta eficiencia, su excelente escalabilidad y la relación costo-beneficio, el costo de la implementación de todos los servicios tuvo un valor de 10 USD siendo una aplicación con fines académicos.
- El lenguaje de programación más representativo en la implementación fue node.js inicialmente para el modelo de interoperabilidad propuesto en tiempo real, y por ser un lenguaje modular que permite un fácil escalamiento y reutilización de código para ser usado en otro tipo de aplicaciones, adicionalmente tiene las prestaciones de otros lenguajes de programación como php para la gestión de bases de datos, en el desarrollo de la implementación se aplicaron estos conceptos en la etapa en que la aplicación presentaba inestabilidad a partir de los 7 minutos de funcionamiento siendo necesaria la migración de servicios a la nube.
- Aunque la latencia del sistema LoRa es aproximadamente 50% menor que WiFi, solo la tecnología WiFi implementa protocolos de seguridad adicionales proporcionados por la plataforma *particle*.
- Realizar aplicaciones con nuevas características implica combinar varias tecnologías y diversos fabricantes, adoptar una única tecnología generalmente no permite crear conceptos diferenciados.
- Las plataformas de *cloud computing* como *Azure* permiten una gestión eficiente de las arquitecturas IoT y la interconexión de múltiples tecnologías.
- Las redes LoRa ofrecen flexibilidad, bajo costo (5 usd) aproximadamente y un muy bajo consumo energético, siendo esta la razón principal de la elección de esta tecnología para este sistema.
- La solución presentada puede acoplarse a otro tipo de sistemas que requiera visualización y gestión en tiempo real, características propias de IoT.
- Las tecnologías propuestas permiten su implementación en áreas rurales y urbanas y la adopción de nuevas tecnologías, servicios y plataformas sin necesidad de implementar una infraestructura compleja que permita la comunicación de los diferentes periféricos del sistema.

- Los sistemas de tipo IoT son soluciones que evolucionan de acuerdo a las necesidades de usuario final de acuerdo a los resultados no se concluye una tecnología mejor que la otra, se debe emplear la más apropiada de acuerdo a la distancia calidad y cantidad de información necesaria.
- El tiempo de sincronización de la red implementada sobre la tecnología BLE basada en el estándar IEEE 802.15.4, oscila entre 200 y 300 milisegundos lo que puede representar una pérdida de datos, en el momento en el que el sistema deba reiniciarse.
- En los sistemas de gestión de información es importante la lectura de variables de interés en tiempo real, por lo cual IoT, de acuerdo con los resultados se debe manejar el concepto tiempo real entre 1 y 6 segundos.
- La tendencia en los modelos de negocio de las empresas que diseñan e implementan sistemas orientados a IoT y las empresas de software relacionadas al análisis de datos, está migrando a prestación de servicios de almacenamiento y análisis de datos en la nube.
- El sistema de interoperabilidad generado entre las redes de corto y las redes de largo alcance, repercuten en las latencias de entrega de datos, así como en la pérdida de paquetes.
- El despliegue de la computación en la nube, permite que los servidores de IoT se implementen en sistemas embebidos, de tamaño reducido y versátiles basados en robustos microprocesadores.
- Los servicios web permiten al usuario desarrollar aplicaciones de consumo web, mediante *Frontend* propio y *Backend* contratado.
- Los sistemas IoT permiten la captura de información en tiempo real, esto potencia la posibilidad de implementación de Big Data, minería de datos y machine learning

Uno de los aportes mas significativos, fue

En los sistemas de procesamiento, gestión y visualización de información, es importante la lectura de variables en tiempo real. El concepto tiempo real para el sistema se encuentra entre 1 y 6 segundos, y 101 y 6012 milisegundos

En los sistemas de procesamiento, gestión y visualización de información, es importante la lectura de variables en tiempo real. El concepto tiempo real para el sistema se encuentra entre 1 y 6 segundos, y 101 y 6012 milisegundos

## **8.2 FUTUROS TRABAJOS.**

En este trabajo se diseñó e implementó un sistema de interoperabilidad para redes inalámbricas el cual apunta a desarrollos de tipo IoT los dispositivos implementados cuentan con la seguridad ofrecida por el fabricante, pero en despliegues que requieran una gran cantidad de nodos sensores y combinación de grandes redes de diversas tecnologías se debe analizar el desarrollo de métodos de detección de malware como lo plantea el trabajo de (Ngo et al., 2020) el cual indica que los dispositivos IoT se basan en arquitecturas de CPU, incluso en hardware con recursos limitados, como

los sistemas operativos basados en Unix. Junto con este cambio, los dispositivos IoT se están convirtiendo en un objetivo favorito de los atacantes debido a la falta de diseño o implementación de seguridad. El malware IoT escanea el puerto abierto de los servicios IoT como FTP, SSH o Telnet y realiza un ataque de fuerza bruta para obtener acceso a dispositivos IoT. La detección de malware de IoT se está convirtiendo en un problema cada vez más urgente para garantizar la seguridad del sistema de Internet y los datos privados, de esta forma los métodos de detección de malware IoT se pueden dividir en dos grupos: Los no basados en gráficos pueden lograr un buen resultado al detectar malware simple y directo sin ofuscación o personalización, pero potencialmente pierde precisión al detectar malware invisible y los basados en gráficos muestran ventajas al analizar el flujo de control del malware IoT, por lo que tienen el potencial de detectar con precisión códigos maliciosos no vistos o complicados. De igual forma en el trabajo de (A. L. Santos et al., 2019) presenta la detección (THATCHI), un nuevo sistema de detección de intrusos IDS orientado a enrutamiento en redes de IoT masivas y móviles, THISACHI muestra estrategias de seguimiento detección y efectividad. La evaluación de desempeño bajo un escenario urbano muestra que el QUEACHI puede realizar con una tasa de detección del 99%.

La aplicaciones de IoT se han incluido en aeronaves no tripuladas en el trabajo de (Khosravi & Samadi, 2019) se propone un sistema de IoT móvil interoperables y sobre el sistema se desarrolla una investigación en la cual, se propuso un nuevo método de agregación de datos basado en una transformación discreta del coseno y una inyección de bits casi dispersa para redes de sensores Visar habilitados del sistema IoT para mejorar la capacidad de inclusión (o el rendimiento de agregación) se analiza la comunicación de carga útil de detección remota de alto rendimiento por ser un problema vital en los sistemas de vigilancia aéreos y espaciales. Entre los diferentes sistemas de imágenes de teledetección, el radar de video sintético de apertura (Visar) es una nueva tecnología con muchos datos principales y administrativos que deben comprimirse, agregarse y comunicarse desde una plataforma de radar (o una red de radares) a una estación terrestre a través de enlaces inalámbricos que requieren sistemas interoperables. De esta forma se proponen trabajos relacionados a diseñar e implementar técnicas para mejorar el ancho de banda, la sensibilidad y la potencia de las señales inalámbricas que interactúan en dispositivos de tipo IoT.

La infraestructura del Internet de las cosas (IoT) se ha amplia constantemente para incluir interacciones de usuario más inteligentes y efectivas. Conjuntos individuales de datos IoT pueden ser combinados con contenidos adicionales de manera distribuida y eficiente para ser utilizados en aplicaciones de usuario final. El documento de (Jo & Kim, 2019) propone una arquitectura para combinar la interfaz AR con IoT para una experiencia de compra mejorada. Esta distribución permite escalar y personalizar técnicas de interacción como la realidad aumentada (AR). La arquitectura propuesta se distribuye, por lo tanto, es escalable de modo que se puede acceder a cualquier producto IoT en el lugar localmente sin ninguna restricción del servidor y proporcionar visualización e interacción intuitiva basada en AR. En el trabajo se identifican tres componentes arquitectónicos clave necesarios para admitir un servicio de AR y una experiencia escalable y sin problemas para IoT: (1) Gestión y visualización de datos, (2) Mecanismos para acceder, controlar e interactuar con el objeto, (3) interoperabilidad de intercambio de contenidos. De esta manera es posible generar

nuevas aplicaciones sobre IoT combinando nuevos conceptos, se pueden proponer trabajos futuros orientados a realidad virtual, aumentada y mixta.

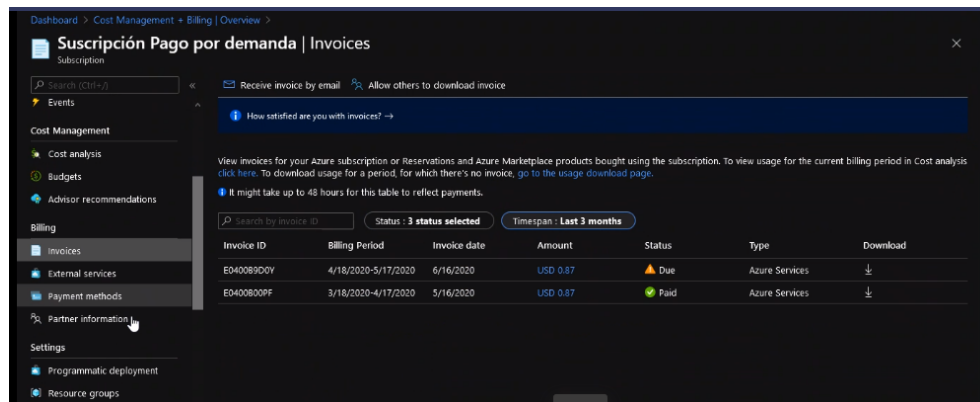
Durante el desarrollo del sistema se pudo observar la importancia de los radiadores para sistemas IoT en el trabajo de (Salucci et al., 2019) se propone un diseño eficiente de antenas compactas que funcionan en múltiples bandas adecuadas para Internet de las cosas (IoT) se aborda mediante una instancia del paradigma de sistema por diseño (SbD). Más específicamente, se propone una estrategia iterativa que combina diferentes módulos de software para la exploración del espacio de búsqueda, el modelado físico rápido de los radiadores y la evaluación de la calidad de las soluciones calculadas. Para habilitar tal instancia de SbD, una estrategia innovadora que explota un esquema de matriz ortogonal (OA) para determinar el conjunto de entrenamiento de un algoritmo de aprendizaje por ejemplo (LBE) basado en un regresor de vectores de soporte (SVR) para introducir el modelado físico eficiente del diseño que se optimizará. Las características y las potencialidades del enfoque metodológico propuesto se evalúan en diferentes escenarios aplicativos al considerar ejemplos representativos de validación numérica y experimental. Los diseños optimizados se han fabricado con una tecnología de circuito de impresión fotolitográfica que utiliza un sustrato Arlon con  $\epsilon_r = 3.38$ ,  $\tan \delta = 0.0025$  y  $h = 7.6 \times 10^{-4}$  [m], y se mide sucesivamente en una cámara anecoica. El caso de una antena de doble banda que resuena en las bandas WCDMA-1500 ( $f_1 = 1.470$  GHz) y LTE-2600 ( $f_2 = 2.595$  GHz). De acuerdo a la necesidad de interconexión de tecnologías que proponen los sistemas interoperables es muy interesante trabajar en el diseño y la implementación de antenas que resuenen en varias bandas apuntando a la versatilidad de los sistemas IoT.

En este desarrollo se seleccionaron estándares de comunicación inalámbrica de bajo consumo energético para sensores de tipo LAN de tipo autónomo y se configuró el tiempo de muestreo en una variable administrada por el usuario de acuerdo al tipo de necesidad, sin embargo el consumo energético podría reducirse para aprovechar mejor la carga de la batería en la forma en que se hizo en el trabajo de (Regulagadda et al., 2018) el cual implementó un receptor único de conversión directa de 2.4 GHz para la tecnología Zigbee y BLE, reduce el consumo de energía usando un CMOS de 180 nm con el acople QVCO y finalmente la técnica de reutilización de corriente. Estos tres conceptos combinados mejoran las características de consumo de energético en sistemas IoT.

El Internet de las cosas (IoT) se desarrolla rápidamente y se despliega ampliamente en los últimos años, lo que hace que los sistemas de almacenamiento de datos sensoriales generados por los sistemas IoT colapsen. La gran cantidad de datos sensoriales generados por algunos sistemas IoT ya ha excedido las capacidades de almacenamiento, transmisión y computación. Sin embargo, los datos relevantes que están altamente relacionados con una consulta en un sistema IoT son relativamente pequeños. Los datos que están altamente relacionados con una consulta forman el conjunto de datos de relativo. La recuperación de datos en el conjunto de datos del núcleo relativo de una consulta en lugar de los datos sensoriales brutos reducirá las grandes cargas de un sistema IoT. Se proponen algoritmos para estimar las respuestas de consultas continuas basadas en sus conjuntos de datos relativos. Se realizan simulaciones exhaustivas para evaluar el rendimiento de los algoritmos propuestos (Zhu et al., 2019). Este trabajo menciona la necesidad de solucionar la problemática de procesamiento y almacenamiento de datos que presentaran los sistemas IoT que apuntan a conceptos de *cognitive IoT*, limpieza de datos (*Data cleaning*), minería de datos (*data*

mining) entre otros, implementados sobre sistemas y dispositivos IoT. La siguiente generación de comunicaciones celulares 5G aunado al despliegue masivo de (IoT) Internet de las Cosas, inicia a ser vulnerable a ataques de enrutamiento debido a su infraestructura dinámica, será necesarios nuevos recursos informáticos y la heterogeneidad de los sensores móviles.

## 9. ANEXOS Y COMPLEMENTOS

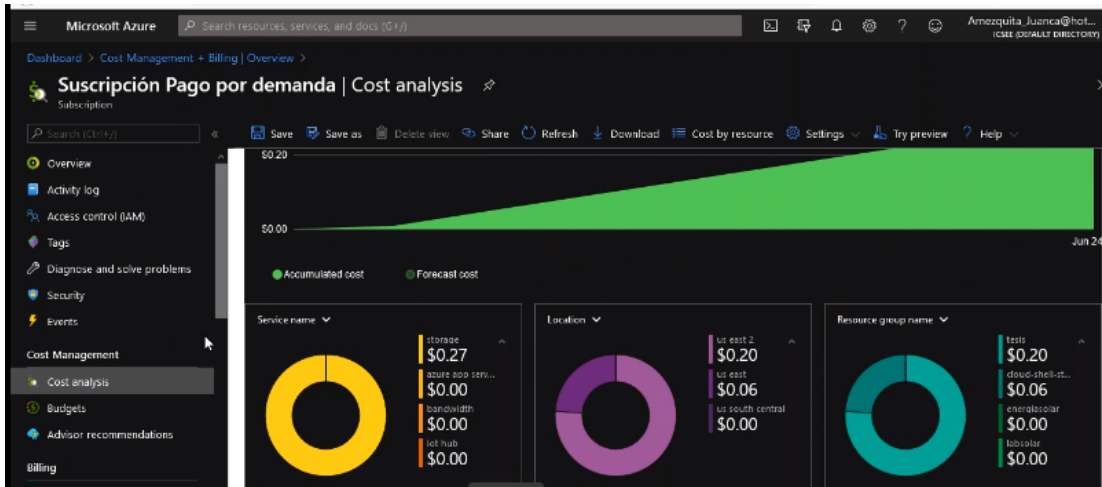


Anexo 1. Vista de costos de almacenamiento



Anexo 2 proyección de costos de almacenamiento

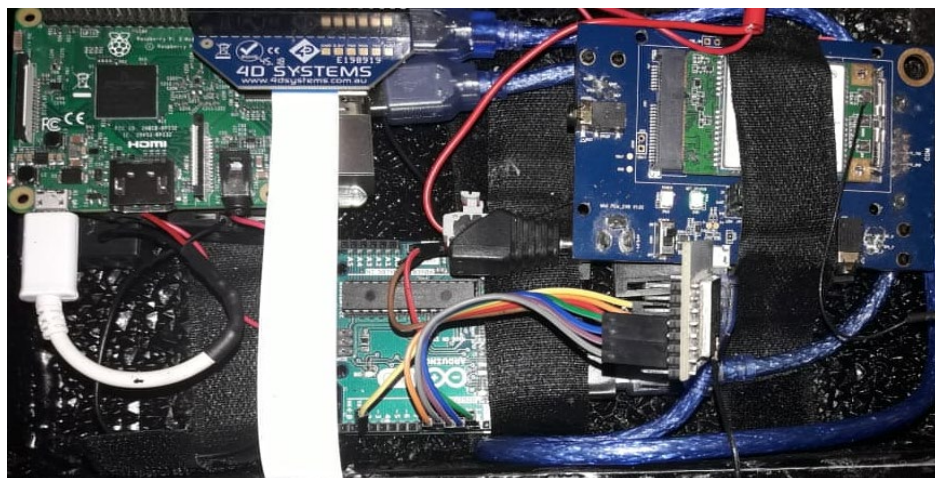
*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*



Anexo 3 Facturación de costos de procesamiento y almacenamiento



Anexo 4 Puerto de enlace para tecnologías propuestas



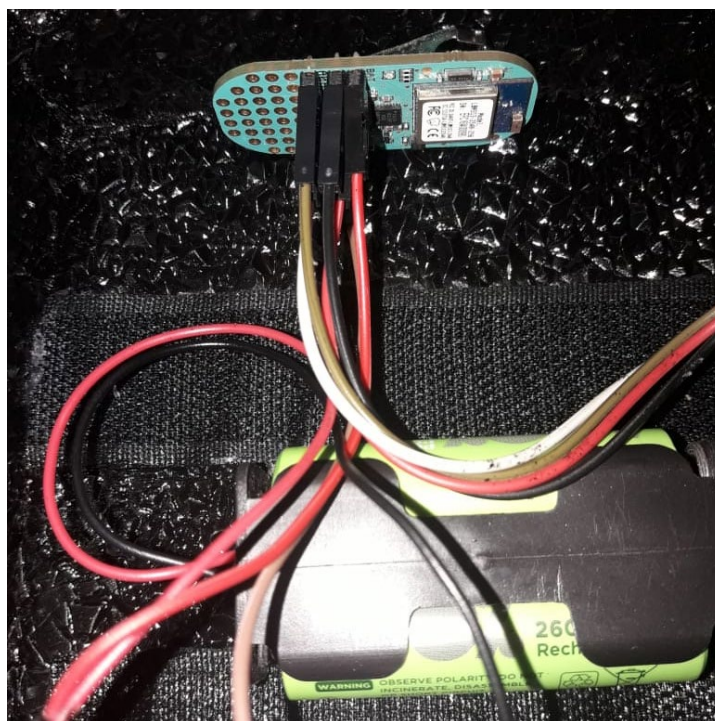
Anexo 5 Vista interna de Gateway



*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*



Anexo 6 Nodo sensor de tecnología BLE

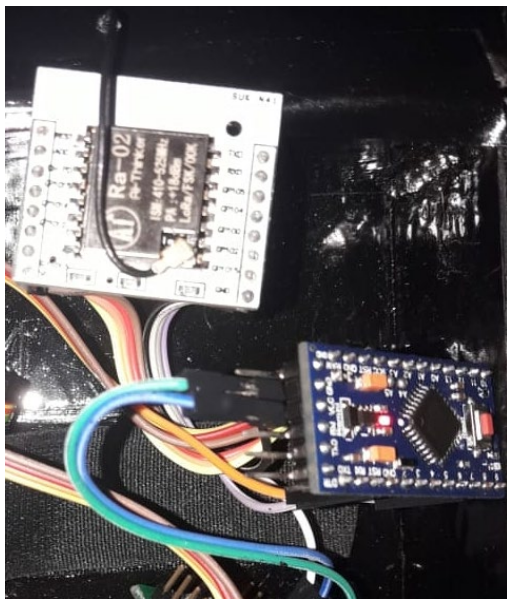


Anexo 7 Nodo sensor de tecnología BLE vista interna

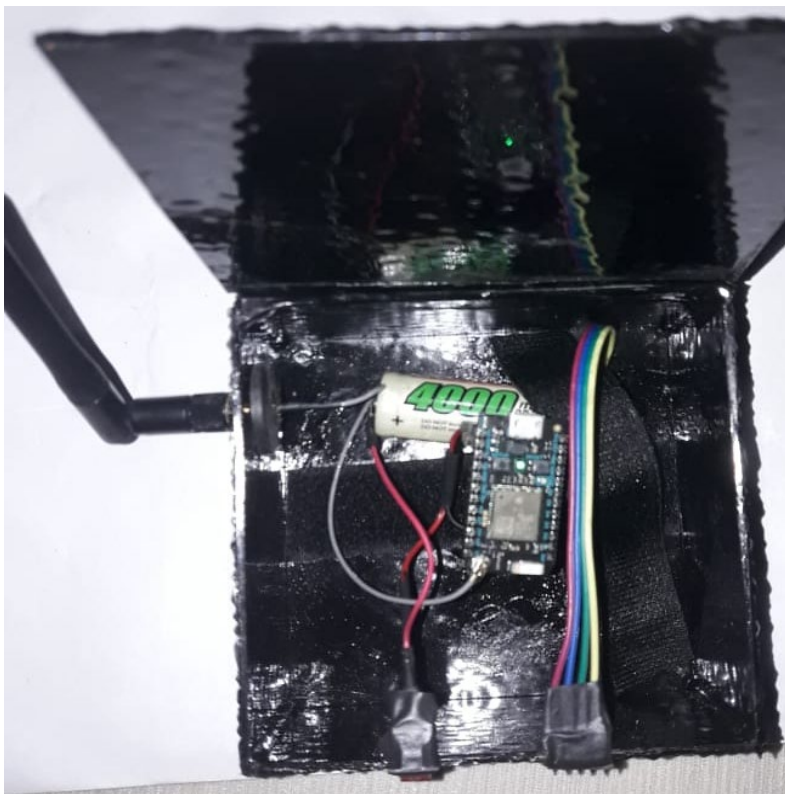
*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*



Anexo 8 Nodo sensor de tecnología LoRa



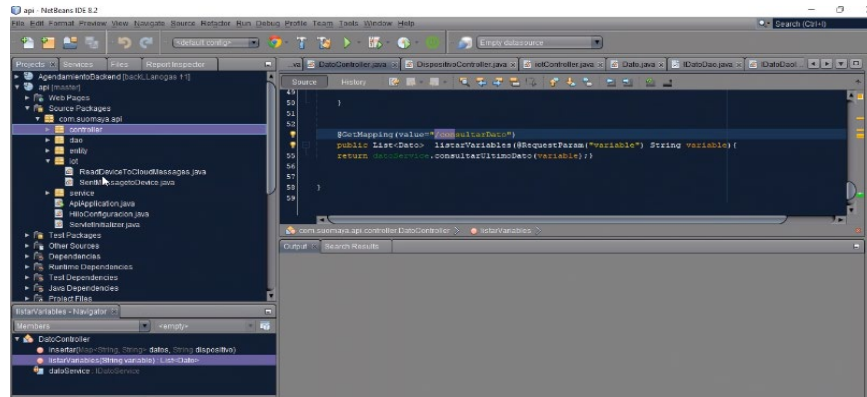
Anexo 9 Nodo sensor de tecnología WiFi y LoRa vista interna



Anexo 10 Nodo sensor de tecnología WiFi vista interna



Anexo 11 Escenario para pruebas de latencia antes de cerrarse.



Anexo 12 Componentes del sistema de almacenamiento

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <meta http-equiv="X-UA-Compatible" content="ie=edge">  
    <link rel="stylesheet" href="css/estilos.css">  
    <link href="https://fonts.googleapis.com/css?family=Roboto:400,700" rel="stylesheet">  
    <title>Tesis Nelson</title>  
    <link rel="stylesheet" href="/css/estilos.css">  
  </head>  
  <body>  
    <div id="overlay" class="overlay"> </div>  
    <header class="header">  
      <div class="fake-header">  
        <nav class="navbar contenedor">  
          <div class="logo">  
              
          </div>  
          <div id="hamburger" class="hamburger-menu">  
            <div id="barras">  
              <div id="bar1" class="bar"></div>  
              <div id="bar2" class="bar"></div>  
              <div id="bar3" class="bar"></div>  
            </div>  
          </div>  
          <div id="menu" class="menu">  
            <div class="menu-foto">  
                
            </div>  
            <div class="menu-text">
```

```
<p class="text">Nelson Agudelo. <br><br>Diseñador y Desarrollador. </p>
<h3 class="menu-t">Contacto</h3>
<p class="text2">contacto@interop.com</p>
</div>
<div class="menu-iconos">
  
  
</div>
</div>
</nav>
</div>
</header>
<div class="contenido contenedor">
  <div class="portada">
    <h1>Interoperabilidad en tus manos</h1>
    <p>Interop pone a tu alcance el monitoreo en tiempo real de diversos sensores electrónicos. Gracias a la interoperabilidad de tecnologías celulares e inalámbricas, puedes ver el estado de tus sistemas sin importar donde te encuentres y sin preocuparte por adaptar tu proyecto a una tecnología en particular. Interop conecta tecnologías inalámbricas como BLE, LoRa y WiFi con la red celular para poner tu información en la nube.
    <br>
    Este prototipo, creado con fines académicos, permite evaluar el rendimiento de diversas tecnologías de comunicación inalámbricas y la integración con la red celular para llevar a todo el mundo los datos capturados.
    </p>
  </div>
  <h2>Selecciona una tecnología</h2>
  <div class="botones">
    <div class="b-col1">
      <div class="boton1">
        <button onclick="ble()"class="boton">Bluetooth</button>
      </div>
      <div class="boton2">
        <button onclick="LoRa()"class="boton">LoRa</button>
      </div>
    </div>
    <div class="b-col2">
      <div class="boton3">
        <button onclick="xbee()"class="boton">X-Bee</button>
      </div>
      <div class="boton4">
        <button onclick="todos ()" class="boton">Todos</button>
      </div>
    </div>
  </div>
</div>
```

```
</div>
</div>
<div class="cuadros">
  <div id="ble1" class="c-LoRa">
    <h3>BLE - Sensor de luz</h3>
    <canvas id="chart1" width="400" height="150"></canvas>
  </div>
  <div id="LoRa1" class="c-ble">
    <h3>LoRa - Sensor de temperatura</h3>
    <canvas id="chart2" width="400" height="150"></canvas>
  </div>
  <div id="xbee1" class="c-xbee">
    <h3>Photon - Sensor de humedad</h3>
    <canvas id="chart3" width="400" height="150"></canvas>
  </div>
</div>
</div>
<div>
<footer class="footer">
  <div class="footer-caja">
    <div class="logo-footer">
      
    </div>
    <div class="letras-footer">
      <p class="pfooter">Copyright © 2017 Interop - Desarrollado por Nelson
Agudelo</p>
    </div>
  </div>
</div>
</footer>

<script src="/socket.io/socket.io.js"></script>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.0/Chart.bundle.min.js"></script>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.18.1/moment.min.js"></script>
<script type="text/javascript">
const socket = io()
let parquets = 0
let start = new Date()
var ctx = document.getElementById('chart1').getContext('2d')
var data = {
  labels: [0],
  datasets: [{
    data: [0],
    label: 'Temperature',
```

```
    // backgroundColor: '#F44436',
    borderColor: '#F44436',
    pointBackgroundColor: '#F44436'
  }]
}
var optionsAnimations = {
  animation: false,
  legend: {
    display: false
  },
  responsive: true
}
var chart1 = new Chart(ctx, {
  type: 'line',
  data: data,
  options: optionsAnimations
})

var ctx2 = document.getElementById('chart2').getContext('2d')
var data2 = {
  labels: [0],
  datasets: [{
    data: [0],
    label: 'Temperature',
    // backgroundColor: '#ff6600'
  }]
}
var chart2 = new Chart(ctx2, {
  type: 'line',
  data: data2,
  options: optionsAnimations
})

//chart3
var ctx3 = document.getElementById('chart3').getContext('2d')
var data3 = {
  labels: [0],
  datasets: [{
    data: [0],
    label: 'Potenciometro',
    // backgroundColor: '#ff6600'
  }]
}
var chart3 = new Chart(ctx3, {
```

```
    type: 'line',
    data: data3,
    options: optionsAnimations
  })

socket.on('getLux', function(value, timeStamp){
  // console.log(timeStamp);
  paquetes++
  const arrivalTime = new Date()
  const arrivalMs = arrivalTime.getTime()
  const latencia = arrivalMs - timeStamp
  console.log("El dato " + value + " tardó " + latencia + " ms en llegar");
  const elapsed = arrivalMs - start
  if (elapsed >= 60000) {
    console.log("Se recibieron " + paquetes + "/60 paquetes");
    paquetes = 0
    start = new Date()
  }
  var length = data.labels.length
  if (length >= 20) {
    data.datasets[0].data.shift()
    data.labels.shift()
  }

  data.labels.push(moment().format('HH:mm:ss'))
  data.datasets[0].data.push(value)

  chart1.update()
});

socket.on('getNum', function(value){
  var length = data2.labels.length
  if (length >= 20) {
    data2.datasets[0].data.shift()
    data2.labels.shift()
  }

  data2.labels.push(moment().format('HH:mm:ss'))
  data2.datasets[0].data.push(value)

  chart2.update()
});
```



```
pi@Raspberry pi:~ $ sudo ./ppp-creator.sh internet.comcel.com.co ttyUSB3
install ppp
Reading package lists... Done
Building dependency tree
Reading state information... Done
ppp is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 118 not upgraded.
creating directories
creating script file: /etc/chatscripts/quectel-chat-connect
creating script file: /etc/chatscripts/quectel-chat-disconnect
creating script file: /etc/ppp/peers/gprs
```

Use "sudo pppd call gprs" command and Surf

```
pi@Raspberry pi: ~ $ sudo pppd call gprs
pppd options in effect:
debug # (from /etc/ppp/peers/gprs)
nodetach # (from /etc/ppp/peers/gprs)
dump # (from /etc/ppp/peers/gprs)
noauth # (from /etc/ppp/peers/gprs)
remotename 3gppp # (from /etc/ppp/peers/gprs)
/dev/ttyUSB3 # (from /etc/ppp/peers/gprs)
115200 # (from /etc/ppp/peers/gprs)
lock # (from /etc/ppp/peers/gprs)
connect chat -s -v -f /etc/chatscripts/quectel-chat-connect -T internet.comcel.com.co #
(from /etc/ppp/peers/gprs)
disconnect chat -s -v -f /etc/chatscripts/quectel-chat-disconnect # (from /etc/ppp/peers/gprs)
nocrtscts # (from /etc/ppp/peers/gprs)
modem # (from /etc/ppp/peers/gprs)
asynmap 0 # (from /etc/ppp/options)
lcp-echo-failure 4 # (from /etc/ppp/options)
lcp-echo-interval 30 # (from /etc/ppp/options)
hide-password # (from /etc/ppp/peers/gprs)
novj # (from /etc/ppp/peers/gprs)
novjccomp # (from /etc/ppp/peers/gprs)
ipcp-accept-local # (from /etc/ppp/peers/gprs)
ipcp-accept-remote # (from /etc/ppp/peers/gprs)
ipparam 3gppp # (from /etc/ppp/peers/gprs)
noipdefault # (from /etc/ppp/peers/gprs)
ipcp-max-failure 30 # (from /etc/ppp/peers/gprs)
defaultroute # (from /etc/ppp/peers/gprs)
usepeerdns # (from /etc/ppp/peers/gprs)
noccp # (from /etc/ppp/peers/gprs)
```

noipx # (from /etc/ppp/options)

abort on (BUSY)

abort on (NO CARRIER)

abort on (NO DIALTONE)

abort on (ERROR)

abort on (NO ANSWER)

timeout set to 30 seconds

send (AT^M)

expect (OK)

^M

OK

-- got it

send (ATE0^M)

expect (OK)

^M

^M

OK

-- got it

send (ATI;+CSUB;+CSQ;+CPIN?;+COPS?;+CGREG?;&D2^M)

expect (OK)

^M

^M

Quectel^M

EC20^M

Revision: EC20AQR02A03E2G^M

^M

SubEdition: V02^M

^M

+CSQ: 24,99^M

^M

+CPIN: READY^M

^M

+COPS: 0,0,"Claro Claro",2^M

^M

+CGREG: 0,1^M

^M

OK

-- got it

send (AT+CGDCONT=1,"IP","[internet.comcel.com.co](http://internet.comcel.com.co)","",0,0^M)

expect (OK)

^M

```
^M
OK
-- got it
```

```
send (ATD*99#^M)
expect (CONNECT)
^M
^M
CONNECT
-- got it
```

```
Script chat -s -v -f /etc/chatscripts/quectel-chat-connect -T internet.comcel.com.co finished
(pid 1101), status = 0x0
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB3
sent [LCP ConfReq id=0x1 <asynmap 0x0> <magic 0x8d7e245a> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1b <asynmap 0x0> <auth chap MD5> <magic 0x7e6a0ea1>
<pcomp> <accomp>]
No auth is possible
sent [LCP ConfRej id=0x1b <auth chap MD5>]
rcvd [LCP ConfAck id=0x1 <asynmap 0x0> <magic 0x8d7e245a> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1c <asynmap 0x0> <magic 0x7e6a0ea1> <pcomp> <accomp>]
sent [LCP ConfAck id=0x1c <asynmap 0x0> <magic 0x7e6a0ea1> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0x8d7e245a]
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [LCP DiscReq id=0x1d magic=0x7e6a0ea1]
rcvd [LCP EchoRep id=0x0 magic=0x7e6a0ea1 8d 7e 24 5a]
rcvd [IPCP ConfNak id=0x1 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14> <ms-wins
10.11.12.13> <ms-wins 10.11.12.14>]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
rcvd [IPCP ConfNak id=0x2 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14> <ms-wins
10.11.12.13> <ms-wins 10.11.12.14>]
sent [IPCP ConfReq id=0x3 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
rcvd [IPCP ConfNak id=0x3 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14> <ms-wins
10.11.12.13> <ms-wins 10.11.12.14>]
sent [IPCP ConfReq id=0x4 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
rcvd [IPCP ConfReq id=0xe]
sent [IPCP ConfNak id=0xe <addr 0.0.0.0>]
rcvd [IPCP ConfRej id=0x4 <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
sent [IPCP ConfReq id=0x5 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
```

```
10.11.12.14>]
rcvd [IPCP ConfReq id=0xf]
sent [IPCP ConfAck id=0xf]
rcvd [IPCP ConfNak id=0x5 <addr 10.191.197.163> <ms-dns1 190.157.8.89> <ms-dns2
8.8.8.8>]
sent [IPCP ConfReq id=0x6 <addr 10.191.197.163> <ms-dns1 190.157.8.89> <ms-dns2
8.8.8.8>]
rcvd [IPCP ConfAck id=0x6 <addr 10.191.197.163> <ms-dns1 190.157.8.89> <ms-dns2
8.8.8.8>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing default route to eth0 [10.73.0.129]
local IP address 10.191.197.163
remote IP address 10.64.64.64
primary DNS address 190.157.8.89
secondary DNS address 8.8.8.8
Script /etc/ppp/ip-up started (pid 1107)
Script /etc/ppp/ip-up finished (pid 1107), status = 0x0
```

---

#### Anexo 14 Script para conexión a redes celulares

---

```
"use strict";

const express = require('express'); // librería para adquisición de datos
const app = express();
const http = require('http').Server(app)
const io = require('socket.io')(http) // Entrega de datos al servidor
const Bean = require('ble-bean') // dispositivo Blue tooth
let intervalId, connectedBean

const SerialPort = require('serialport'); // Lectura de puertos
const Readline = SerialPort.parsers.Readline;
const port = new SerialPort('/dev/ttyACM0');
const parser = new Readline();

const Particle = require('particle-api-js') // api para conexión de dispositivo
const particle = new Particle() // Dispositivo WiFi
const token = '77228bcaa34360fce83a2b69d3a739772232bcb5'
const deviceID = '450028000851363136363935'
app.get('/', function(req, res){
res.sendFile(__dirname + '/index.html')
})
```

```
app.use(express.static('public'));
io.on('connection', function(socket){
  console.log('User connected'); // Se visualiza si un usuario está conectado

  async function fetchFromParticle (variable){
    const response = await particle.getVariable({ deviceId: deviceID, name: variable, auth:
    token })
    return response.body.result
  }

  async function getData(){
    try {
      const sensor = await fetchFromParticle(` analogvalue `)
      const rssi = await fetchFromParticle(` rssi `)
      console.log(sensor)
      console.log(rssi)
      io.emit('getPote', sensor)
      console.log('Potenciómetro: ' + sensor)
    } catch (error) {
      console.log('There was an error: ' + error)
    }
  }

  setInterval(getData, 3000) // Dispositivo LoRa

  port.pipe(parser);
  parser.on('data', function (data) {
    const res = data.split(" ")
    const temp = res[1]
    const rssi = res[3]
    console.log(temp)
    console.log(rssi)
    io.emit('getNum', temp)
  });

  Bean.discover(function(bean){
    connectedBean = bean;
    process.on('SIGINT', exitHandler.bind(this));
    bean.on("serial", function(data, valid){
      const lux = parseInt(data.toString());
      const timeStamp = new Date()
      const ms = timeStamp.getTime()
      console.log('Lux: ' + lux)
      io.emit('getLux', lux, ms)
    });
  });
});
```

```
});
bean.on("disconnect", function(){
  process.exit();
});
bean.connectAndSetup(function(){
});
});

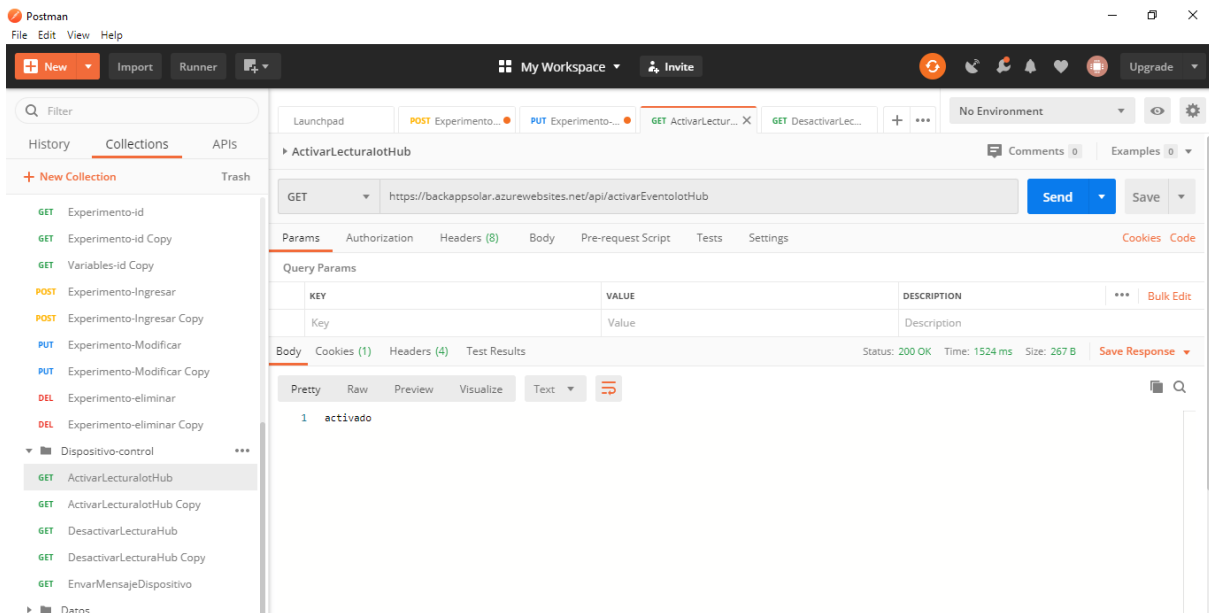
process.stdin.resume();
var triedToExit = false;
var exitHandler = function exitHandler() {
  var self = this;
  if (connectedBean && !triedToExit) {
    triedToExit = true;
    console.log('Turning off led...');
    clearInterval(intervalId);
    connectedBean.setColor(new Buffer([0x0,0x0,0x0]), function(){}); // No se puede
    conocer si el bean se conecta exitosamente se debe esperar cerca de 3 segundos.

    console.log('Disconnecting from Device...');
    setTimeout(connectedBean.disconnect.bind(connectedBean, function(){}), 2000);
  } else {
    process.exit();
  }
};
})
http.listen(80, function(){
  console.log('listening on *:80')
})
```

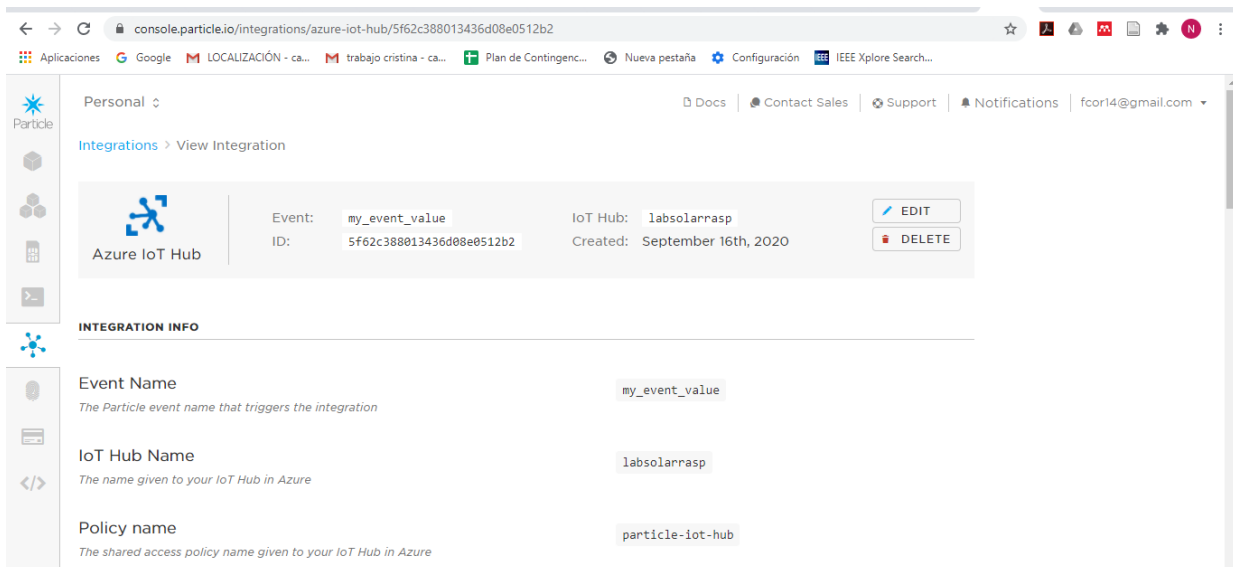
---

Anexo 15 Código de la aplicación contenido en el index.js

## Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11



Anexo 16 Pruebas de aplicación de *Backend*



Anexo 17 Integración de aplicación Azure y particle

## 10. REFERENCIAS

---

- Ahmad, K. A., Salleh, M. S., Segaran, J. D., & Hashim, F. R. (2018). Impact of foliage on LoRa 433MHz propagation in tropical environment. *AIP Conference Proceedings*, 1930(February), 1–7. <https://doi.org/10.1063/1.5022903>
- Albertos-Marco, F., Gallud, J. A., & Penichet, V. M. R. (2018). A study on how users perceive distributed interactions on web applications. *IEEE Access*, 6, 57389–57399. <https://doi.org/10.1109/ACCESS.2018.2873678>
- Allafi, I. (2017). *Design and Implementation of a Low Cost Web Server Using ESP32 for Real-Time Photovoltaic System Monitoring*. 1–5.
- Ayele, E. D., Das, K., Meratnia, N., & Havinga, P. J. M. (n.d.). Leveraging BLE and LoRa in IoT Network for Wildlife Monitoring System ( WMS ). *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 342–348. <https://doi.org/10.1109/WF-IoT.2018.8355223>
- Cheng, Y., Xu, X., Du, Y., Guan, P., Liu, S., & Zhao, L. (2019). Design of Air Quality Monitoring System Based on NB-IoT. *2019 IEEE International Conference on Power, Intelligent Computing and Systems, ICPICS 2019*, 385–388. <https://doi.org/10.1109/ICPICS47731.2019.8942533>
- Completo, T. (2017). *Microsoft Previews Cross Platform Tool for Working with SQL Server*. 1–2. <https://doi.org/https://search-proquest-com.bdigital.sena.edu.co/docview/1980921029769636634319PQ/6?accountid=31491>
- Dambal, V. A., Mohadikar, S., Kumbhar, A., & Guvenc, I. (2019). Improving LoRa Signal Coverage in Urban and Sub-Urban Environments with UAVs. *2019 International Workshop on Antenna Technology, IWAT 2019*, 210–213. <https://doi.org/10.1109/IWAT.2019.8730598>
- Duangsuwan, S., Takarn, A., Nujankaew, R., & Jamjareegulgarn, P. (2018). A Study of Air Pollution Smart Sensors LPWAN via NB-IoT for Thailand Smart Cities 4.0. *2018 10th International Conference on Knowledge and Smart Technology: Cybernetics in the Next Decades, KST 2018*, 206–209. <https://doi.org/10.1109/KST.2018.8426195>
- Electrical Engineering/Electronics, C., IEEE Thailand Section, IEEE Computational Intelligence Society. Thailand Chapter, & Institute of Electrical and Electronics Engineers. (2018). ISGIT 2018 : the 18th International Symposium on Communication and Information Technology : September 26-29, 2018, Sukosol Hotel, Bangkok, Thailand. *2018 18th International Symposium on Communications and Information Technologies (ISCIT), Iscit*, 313–317.
- Elgui, K., Bianchi, P., Portier, F., & Isson, O. (2020). Learning methods for RSSI-based geolocation: A comparative study. *Pervasive and Mobile Computing*, 192, 101199. <https://doi.org/10.1016/j.pmcj.2020.101199>
- Fredericks, J., & Botts, M. (2018). Promoting the capture of sensor data provenance: a role-based approach to enable data quality assessment, sensor management and interoperability. *Open Geospatial Data, Software and Standards*, 3(1). <https://doi.org/10.1186/s40965-018-0048-5>



- Garrido-Hidalgo, C., Hortelano, D., Roda-Sanchez, L., Olivares, T., Ruiz, M. C., & Lopez, V. (2018). IoT Heterogeneous Mesh Network Deployment for Human-in-the-Loop Challenges Towards a Social and Sustainable Industry 4.0. *IEEE Access*, 6, 28417–28437. <https://doi.org/10.1109/ACCESS.2018.2836677>
- Georgi, N., Corvol, A., Le, R., & Jeannès, B. (2018). Middleware Architecture for Health Sensors Interoperability. *IEEE Access*, 6, 26283–26291. <https://doi.org/10.1109/ACCESS.2018.2835644>
- Geraci A, Katki F, McMonegal L, Meyer B, Lane J, Wilson P, Radatz J, Yee M, Porteous H, S. F. (1991). *610-1990 - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. IEEE. <https://doi.org/10.1109/IEEESTD.1991.106963>
- GitHub. (2019). *GitHub*. [https://github.com/sixfab/Sixfab\\_RPi\\_3G-4G-LTE\\_Base\\_HAT](https://github.com/sixfab/Sixfab_RPi_3G-4G-LTE_Base_HAT)
- Haines et al, 2019. (2013). *Sixfab*. *Journal of Chemical Information and Modeling*. <https://doi.org/10.1017/CBO9781107415324.004>
- Hlaing, W., Thepphaeng, S., Nontaboot, V., Tangsunantham, N., Sangsuwan, T., & Pira, C. (2017). Implementation of WiFi-Based single phase smart meter for internet of things (IoT). *2017 International Electrical Engineering Congress, IEECON 2017, March*, 8–10. <https://doi.org/10.1109/IEECON.2017.8075793>
- Issa, S., Robles, E., & González, F. (2019). *Cultura De Innovación Turística: Una Apuesta Para El Cierre De Brechas Sociales Y Tecnológicas* (Vol. 53, Issue 9). <https://doi.org/10.1017/CBO9781107415324.004>
- Jayanthiladevi, a., Premlatha, H. M., & Nawaz, G. M. K. (2013). Analysis study of seamless integration and intelligent solution in any situation by the future advanced mobile universal systems 4G - (FAMOUS 4G). *2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System, ICEVENT 2013, Famous 40*. <https://doi.org/10.1109/ICEVENT.2013.6496551>
- Jo, D., & Kim, G. J. (2019). IoT + AR: pervasive and augmented environments for “Digi-log” shopping experience. *Human-Centric Computing and Information Sciences*, 9(1), 1–17. <https://doi.org/10.1186/s13673-018-0162-5>
- Jörke, P., Böcker, S., Liedmann, F., & Wietfeld, C. (2018). Urban channel models for smart city IoT-networks based on empirical measurements of LoRa-lmks at 433 and 868 MHz. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2017-Octob(Cn I)*, 1–6. <https://doi.org/10.1109/PIMRC.2017.8292708>
- Jung, H. (2015). A thing ID-based IoT internetworking framework. *International Conference on ICT Convergence 2015: Innovations Toward the IoT, 5G, and Smart Media Era, ICTC 2015*, 375–379. <https://doi.org/10.1109/ICTC.2015.7354567>
- Kang, H., & Kim, C. (2016). ISO / IEEE 11073-based Healthcare Services over IoT Platform using 6LoWPAN and BLE : Architecture and Experimentation. *2016 International Conference on Networking and Network Applications (NaNA)*, 313–318. <https://doi.org/10.1109/NaNA.2016.26>

- Keramidas, G., Voros, N., & Hübner, M. (2016). Components and services for IoT platforms: Paving the way for IoT standards. *Components and Services for IoT Platforms: Paving the Way for IoT Standards*, 1–383. <https://doi.org/10.1007/978-3-319-42304-3>
- Khalil, A., & Hamaydeh, N. (2013). *Tradeoff Analysis for Vertical Handover Between UMTS-WiFi Enhanced by IEEE802 . 21. Mcdm*, 150–155.
- Khan, A. H., Qadeer, M. A., Ansari, J. A., & Waheed, S. (2009). 4G as a Next Generation Wireless Network. *2009 International Conference on Future Computer and Communication*, 334–338. <https://doi.org/10.1109/ICFCC.2009.108>
- Khosravi, M. R., & Samadi, S. (2019). Efficient payload communications for IoT-enabled ViSAR vehicles using discrete cosine transform-based quasi-sparse bit injection. *Eurasip Journal on Wireless Communications and Networking*, 2019(1). <https://doi.org/10.1186/s13638-019-1572-4>
- Kim, J. Y. (2019). LoRa-based Mesh Network for IoT Applications. *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 524–527.
- Kirpes, B., Danner, P., Basmadjian, R., Meer, H. de, & Becker, C. (2019). E-Mobility Systems Architecture: a model-based framework for managing complexity and interoperability. *Energy Informatics*, 2(1). <https://doi.org/10.1186/s42162-019-0072-4>
- Liu, L., Liu, P., Wen, F., Zhang, C., Zhao, R., Yan, M., & Yu, X. (2018). Information collection system of duck products based on IoT. *Eurasip Journal on Wireless Communications and Networking*, 2018(1). <https://doi.org/10.1186/s13638-018-1144-z>
- Liu, T., & Wen, Y. (2019). WIFI security certification through device information. *Proceedings - 2018 International Conference on Sensor Networks and Signal Processing, SNSP 2018*, 302–305. <https://doi.org/10.1109/SNSP.2018.00065>
- Luo, Y. W. (2010). The effect of using JavaScript and HTML to implement an E-learning web sites as a remedial teaching aid on English underachievers at Universities of Science and Technology. *40th International Conference on Computers and Industrial Engineering: Soft Computing Techniques for Advanced Manufacturing and Service Systems, CIE40 2010*, 1–7. <https://doi.org/10.1109/ICCIE.2010.5668450>
- Mahmoud, M. S., & Mohamad, A. A. H. (2016). A Study of Efficient Power Consumption Wireless Communication Techniques/ Modules for Internet of Things (IoT) Applications. *Advances in Internet of Things*, 06(02), 19–29. <https://doi.org/10.4236/ait.2016.62002>
- Mangiatordi, A., & Scenini, F. (2017). Improving EPUB3 ebooks accessibility through Javascript and CSS. *2017 14th IEEE Annual Consumer Communications and Networking Conference, CCNC 2017*, 1073–1076. <https://doi.org/10.1109/CCNC.2017.7983288>
- Mao, J., Bian, J., Bai, G., Wang, R., Chen, Y., Xiao, Y., & Liang, Z. (2018). Detecting Malicious Behaviors in JavaScript Applications. *IEEE Access*, 6, 12284–12294. <https://doi.org/10.1109/ACCESS.2018.2795383>
- Mehaseb, M. A., Gadallah, Y., & El-Hennawy, H. (2013). WSN application traffic characterization

- for integration within the internet of things. *Proceedings - IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2013*, 318–323.  
<https://doi.org/10.1109/MSN.2013.20>
- Mishra, D., Alexandropoulos, G. C., & De, S. (2018). Energy Sustainable IoT with Individual QoS Constraints Through MISO SWIPT Multicasting. *IEEE Internet of Things Journal*, 4662(May), 1–12. <https://doi.org/10.1109/JIOT.2018.2842150>
- Morais, C. M. d., Sadok, D., & Kelner, J. (2019). An IoT sensor and scenario survey for data researchers. *Journal of the Brazilian Computer Society*, 25(1).  
<https://doi.org/10.1186/s13173-019-0085-7>
- Nasiri, H., Nasehi, S., & Goudarzi, M. (2019). Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities. *Journal of Big Data*, 6(1).  
<https://doi.org/10.1186/s40537-019-0215-2>
- Ngo, Q.-D., Nguyen, H.-T., Nguyen, L.-C., & Nguyen, D.-H. (2020). A survey of IoT malware and detection methods based on static features. *ICT Express*, xxxx.  
<https://doi.org/10.1016/j.icte.2020.04.005>
- Nikolov, N., & Nakov, O. (2019). Research of Communication Between IoT Cloud Structure, Android Application and IoT Device Using TCP Sockets. *2019 X National Conference with International Participation (ELECTRONICA)*, 1–4.  
<https://doi.org/10.1109/electronica.2019.8825568>
- Probability, O., & Networks, H. (2013). *Probabilidad de Degradación por Interferencia en Redes Heterogéneas*. 2–7.
- Quectel. (2012). *Quectel EC20*. עלון הנוטע.  
<https://www.quectel.com/product/ec20r21minipcle.htm>
- Rahman, T., & Chakraborty, S. K. (2018). Provisioning technical interoperability within ZigBee and BLE in IoT environment. *2018 2nd International Conference on Electronics, Materials Engineering and Nano-Technology, IEMENTech 2018*, 1–4.  
<https://doi.org/10.1109/IEMENTECH.2018.8465272>
- Rappaport, T. S., Sun, S., Mayzus, R., Zhao, H., Azar, Y., Wang, K., Wong, G. N., Schulz, J. K., Samimi, M., & Gutierrez, F. (2013). Millimeter wave mobile communications for 5G cellular: It will work! *IEEE Access*, 1, 335–349. <https://doi.org/10.1109/ACCESS.2013.2260813>
- Regulagadda, S. S., Nagaveni, S., & Dutta, A. (2018). A 550-  $\mu$ W, 2.4-GHz ZigBee / BLE Receiver Front End for IoT applications in 180-nm CMOS. *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)*, 48–52.
- Río Jalón, D., Antonio Chuquín Vasco, D., Javier Paredes Arquíola, D., & Néstor Lerma Elvira, D. (2016). *Trabajo Fin de Máster. Documento 1*, 2011–2012.
- Rivero, E., & Bolaño, F. (2018). Gestión Del Conocimiento Y Alianzas Estratégicas En Los Procesos De Innovación Tecnológica. In *Innovación, Desarrollo Tecnológico Y Gestión: Una Construcción Desde La Investigación*. <https://repositorio.sena.edu.co/handle/11404/5016>

- Ruiz, M., Alvarez, E., Serrano, A., & Garcia, E. (2016). The Convergence between Wireless Sensor Networks and the Internet of Things; Challenges and Perspectives: a Survey. *IEEE Latin America Transactions*, 14(10), 4249–4254. <https://doi.org/10.1109/TLA.2016.7786301>
- Salucci, M., Anselmi, N., Goudos, S., & Massa, A. (2019). Fast design of multiband fractal antennas through a system-by-design approach for NB-IoT applications. *Eurasip Journal on Wireless Communications and Networking*, 2019(1). <https://doi.org/10.1186/s13638-019-1386-4>
- Santos, A. L., Cervantes, C. A. V., Nogueira, M., & Kantarci, B. (2019). Clustering and reliability-driven mitigation of routing attacks in massive IoT systems. *Journal of Internet Services and Applications*, 10(1). <https://doi.org/10.1186/s13174-019-0117-8>
- Santos, H. D. A. do., Oliveira, M. I. S., Lima, G. de F. A. B., da Silva, K. M., Rayelle, R. I. V. C., & Lóscio, B. F. (2018). Investigations into data published and consumed on the Web: a systematic mapping study. *Journal of the Brazilian Computer Society*, 24(1). <https://doi.org/10.1186/s13173-018-0077-z>
- Savaux, V. (2020). DFT-based low-complexity optimal cell ID estimation in NB-IoT. *Eurasip Journal on Advances in Signal Processing*, 2020(1). <https://doi.org/10.1186/s13634-020-00677-4>
- Simo-Reigadas, J., Municio, E., Morgado, E., Castro, E. M., Martinez, A., Solorzano, L. F., & Prieto-Egido, I. (2015). Sharing low-cost wireless infrastructures with telecommunications operators to bring 3G services to rural communities. *Computer Networks*. <https://doi.org/10.1016/j.comnet.2015.09.006>
- Suganya, P., & Pradeep, P. R. (2020). LNR-PP: Leaf Node Count and RSSI Based Parent Prediction Scheme to Support QoS in Presence of Mobility in 6LoWPAN. *Computer Communications*, 150(November 2019), 472–487. <https://doi.org/10.1016/j.comcom.2019.12.012>
- Wan, J., A. A. H. Al-awlaqi, M., Li, M. S., O’Grady, M., Gu, X., Wang, J., & Cao, N. (2018). Wearable IoT enabled real-time health monitoring system. *Eurasip Journal on Wireless Communications and Networking*, 2018(1). <https://doi.org/10.1186/s13638-018-1308-x>
- Wang, S., Hou, Y., Gao, F., & Ji, X. (2017). A novel IoT access architecture for vehicle monitoring system. *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*, 639–642. <https://doi.org/10.1109/WF-IoT.2016.7845396>
- Wu, F., & Christoph, R. (n.d.). *WE-Safe : A Wearable IoT Sensor Node for Safety Applications via LoRa*. 144–148. <https://doi.org/10.1109/WF-IoT.2018.8355234>
- Zhang, D., Lin, S., Fu, Y., & Huang, S. (2018). The communication system between web application host computers and embedded systems based on Node.JS. *Proceedings - 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2017, 2018-Janua*, 1–5. <https://doi.org/10.1109/CISP-BMEI.2017.8302325>
- Zhu, T., Wang, J., Cheng, S., Li, Y., & Li, J. (2019). Retrieving the relative kernel dataset from big sensory data for continuous queries in IoT systems. *Eurasip Journal on Wireless Communications and Networking*, 2019(1), 1–14. <https://doi.org/10.1186/s13638-019-1467-4>

*Sistema de interoperabilidad para tecnologías celulares e inalámbricas establecidas en los estándares IEEE 802.15.1, IEEE 802.15.4, IEEE 802.11*

Zourmand, A. (2019). Internet of Things ( IoT ) using LoRa technology. *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, June, 324–330.