



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO ELECTRÓNICO**

**TEMA:
SISTEMA DE DETECCIÓN DE NIVELES DE DIÓXIDO DE CARBONO (CO₂)
DENTRO DE LA CABINA SIMULADA DE UN VEHÍCULO UTILIZANDO
COMUNICACIÓN LoRaWAN**

**AUTORES:
GABRIEL REYNALDO GALARZA ROMERO
MARÍA GUADALUPE LARA PONCE**

**TUTOR:
ING. RAFAEL ENRIQUE PEREZ ORDOÑEZ MSc.**

Guayaquil, Octubre del 2020

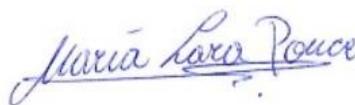
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Las definiciones analizadas y desarrolladas en el presente proyecto de titulación, el desarrollo del tema, la implementación y sus respectivas conclusiones, corresponden exclusivamente a MARIA GUADALUPE LARA PONCE con cédula de identidad 0928823806 y GABRIEL REYNALDO GALARZA ROMERO con cédula de identidad 0931883078 y el patrimonio intelectual del mismo a la Universidad Politécnica Salesiana.

Guayaquil, Octubre del 2020



GABRIEL REYNALDO GALARZA ROMERO
CI: 0931883078



MARÍA GUADALUPE LARA PONCE
CI: 0928823806

CESIÓN DE DERECHOS DE AUTOR

Nosotros, María Guadalupe Lara Ponce, con documento de identidad N° 0928823806, y Gabriel Reynaldo Galarza Romero, con documento de identidad N° 093188307-8, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: SISTEMA DE DETECCIÓN DE NIVELES DE DIÓXIDO DE CARBONO (CO₂) DENTRO DE LA CABINA SIMULADA DE UN VEHÍCULO UTILIZANDO COMUNICACIÓN LoRaWAN, mismo que ha sido desarrollado para optar por el título de INGENIEROS ELECTRÓNICOS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

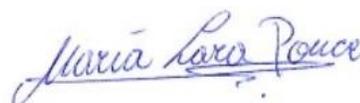
En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, Octubre del 2020



GABRIEL REYNALDO GALARZA ROMERO
CI: 0931883078



MARÍA GUADALUPE LARA PONCE
CI: 0928823806

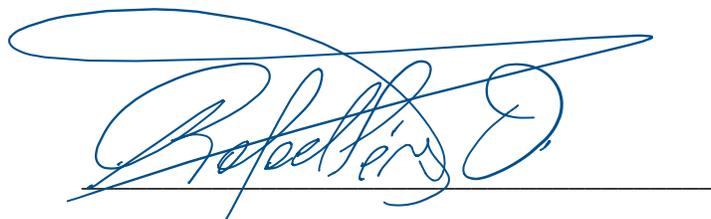
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, RAFAEL ENRIQUE PEREZ ORDOÑEZ declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: SISTEMA DE DETECCIÓN DE NIVELES DE DIÓXIDO DE CARBONO (CO₂) DENTRO DE LA CABINA SIMULADA DE UN VEHÍCULO UTILIZANDO COMUNICACIÓN LoRaWAN, realizado por: María Guadalupe Lara Ponce y Gabriel Reynaldo Galarza Romero, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Por medio de la presente certifico que el documento cumple con los requisitos establecidos en el Instructivo para el Estructura y Desarrollo de Trabajos de Titulación para pregrado de la Universidad Politécnica Salesiana.

En virtud de lo anterior, autorizo su presentación y aceptación como una obra auténtica y de valor académico

Guayaquil, Octubre del 2020

A handwritten signature in blue ink, reading 'Rafael Enrique Perez Ordoñez', is written over a horizontal line. The signature is stylized and cursive.

Tutor del Trabajo de Titulación

Ing. RAFAEL ENRIQUE PEREZ ORDOÑEZ MSc.

DEDICATORIA

Dedico este proyecto a mi Padre Celestial por la fuerza que me ha dado para seguir adelante cada vez que siento caer, definitivamente sin Él no soy nadie, ¡Gracias Papito Dios!

A mi familia, a mis padres César y Merlis, mi hermano Toni, mis hermosos sobrinos César y Sebas que con su gran amor me han motivado en los momentos más difíciles y que de una u otra manera me han apoyado para que continúe con todo lo que me he propuesto.

A la familia Gamboa Alabuella, por ser una gran segunda familia para mí, que también han sido parte de este proceso y de mi vida universitaria.

A mis grandes amigos que siempre los llevo en mi corazón.

María Guadalupe Lara Ponce

DEDICATORIA

En primer lugar, a Dios por darme mucha fuerza, para seguir adelante y no dejarme renunciar a mis sueños, aconsejándome en cada uno de los tropiezos en mi vida y sobre todo no dejándome solo.

En segundo lugar, quiero agradecer a mi madre María Dolores Romero Romero quien siempre me ha dado su apoyo incondicional, que con su amor y consejos haciéndome un hombre de bien. A mi padre Fulvio Oswaldo Galarza Ramírez que con su claro ejemplo ha marcado cada uno de los pasos que doy, lo cual me ha permitido ser quien soy.

A mis abuelitas, Blanca y Esther quienes siempre han marcado mi vida, que, con su amor y un fuerte abrazo, dándome ganas de seguir adelante.

A mis hermanos quienes siempre me han apoyado desde pequeño de una u otra manera con sus consejos.

A la Virgencita del “El Cisne”, por guiarme y cuidarme en mi vida cotidiana.

A mis compañeros con quienes he compartido experiencias memorables y satisfactorias a lo largo de la carrera ingeniería electrónica.

Gabriel Reynaldo Galarza Romero

AGRADECIMIENTO

Agradezco a mi querida Universidad Politécnica Salesiana por haber sido un segundo hogar para mí, por haberme formado como una profesional llena de valores, donde indudablemente no solo se adquieren conocimientos, sino que conoces a las mejores personas de las cuales de cada una te llevas las más lindas enseñanzas y experiencias.

A nuestro tutor Ing. Rafael Pérez por su guía y total apoyo en el desarrollo de este proyecto.

A mi compañero de trabajo de titulación Gabriel Galarza, gracias por haber confiado en mí.

A mis amigos, Jhon, Maite, Dennys que sin duda hicieron divertida mi vida dentro de la U.

A mi gran amigo Jonathan Choez que ahora se encuentra en el cielo, sé que también esperabas esto tanto como yo, el poder graduarnos, sé que desde el cielo has de estar feliz al saber de qué por fin se acaba esto, como siempre me lo sabias decir.

A mis profesores y compañeros de clases tanto en la sede de Guayaquil como en la sede de Quito siempre fueron los mejores.

A mi amor Jonathan Gamboa, por ser esa personita especial, por haberme apoyado tanto en esta carrera, por haberme brindado una mano cuando siempre lo necesitaba.

María Guadalupe Lara Ponce

AGRADECIMIENTO

Agradezco a la Universidad Politécnica Salesiana por brindarme los conocimientos necesarios para afrontar la vida de forma profesional, a los docentes que de una u otra manera me enseñaron mi vida universitaria dándome habilidades y soluciones a cada dificultad.

Al Ing. Rafael Pérez por su apoyo, consejo y tiempo en el desarrollo de este proyecto.

A su vez agradezco al Club de Robótica y Club de Telecomunicaciones por formarme valores y espíritu competitivo.

A mi compañera de trabajo de titulación María Guadalupe Lara Ponce por la sinergia de trabajo en equipo y sobresalir en los momentos más difíciles durante esta pandemia del Covid-19, mientras duró el proceso de elaboración de nuestro proyecto.

Finalmente, pero no menos importante agradezco a mi amigo José Boderó por todo el ánimo y apoyo que me ha brindado desde el pre-universitario.

Gabriel Reynaldo Galarza Romero

RESUMEN

AÑO	TÍTULO	ALUMNOS	DIRECTOR DE PROYECTO	TEMA DE PROYECTO DE TITULACIÓN
2020	Ingeniero Electrónico	María Lara Ponce. Gabriel Galarza Romero.	Ing. Rafael Pérez MSc.	Sistema de detección de niveles de dióxido de carbono (CO ₂) dentro de la cabina simulada de un vehículo utilizando comunicación LoRaWAN

El presente proyecto de titulación tiene por objetivo diseñar e implementar un sistema de detección de niveles de CO₂ dentro de la cabina simulada de un vehículo utilizando comunicación LoRaWAN, con el fin de evitar que las personas inhalen gases contaminantes. El prototipo es capaz de detectar niveles altos de concentración en tiempo real generando una notificación que será emitida por una aplicación web al usuario.

Existen varias tecnologías que van de la mano con el internet de las cosas que tiene como funcionalidad una transmisión de largo alcance, grandes cantidades de conexiones inalámbricas para implementarse en un área determinada, por tal razón se demuestra que la tecnología LoRa/LoRaWAN tienen una gran ventaja dentro de las tecnologías inalámbricas.

Mediante un Gateway se pretende recopilar toda la información adquirida mediante el sensor de CO₂ colocado dentro del automotor en conjunto a la tarjeta Lora Shield encargada de la transmisión de datos hacia el servidor proporcionando al usuario una plataforma de visualización y supervisión de los valores adquiridos mediante la implementación de estos equipos.

El desarrollo del proyecto de investigación permitió generar un prototipo que servirá como base de desarrollo de alternativas para prevenir la inhalación del gas CO₂ en concentraciones altas y al mismo tiempo puede ser usado como instrumento de estudio para los alumnos que se encuentren cursando materias como redes inalámbricas, comunicaciones digitales y redes de comunicaciones ya que al ser una nueva tecnología se podrá plasmar en un prototipo todos los conocimientos teóricos obtenidos en las mismas.

ABSTRACT

YEAR	TITLE	STUDENTS	TECHNICAL PROJECT MANAGER	ITEM OF PROJECT OF TITULATION
2020	Ingeniero Electrónico	María Lara Ponce. Gabriel Galarza Romero.	Ing. Rafael Pérez MSc.	Carbon dioxide (CO ₂) level detection system inside a simulated vehicle cabin using LoRaWAN communication

The objective of this project is to design and implement a CO₂ level detection system inside the simulated cabin of a vehicle using LoRaWAN communication, in order to prevent people from inhaling polluting gases. The prototype is capable of detecting high levels of concentration in real time, generating a notification that will be emitted by a web application to the user.

There are several technologies that work in first-hand with the internet of things that have a long-range transmission as a functionality, large amounts of wireless connections to be implemented in a certain area, for this reason it is shown that LoRa/LoRaWAN technology has a great advantage within wireless technologies.

Through a Gateway, it is intended to collect all the information acquired by means of the CO₂ sensor placed inside the car in conjunction with the Lora Shield card in charge of transmitting data to the server, providing the user with a platform for viewing and monitoring the values acquired through the implementation of these equipment's.

The development of the research project allowed the generation of a prototype that will serve as a basis for the development of alternatives to prevent the inhalation of CO₂ gas in high concentrations and at the same time it can be used as a study instrument for students who are studying subjects such as Wireless Networks , Digital Communications and Communications Networks since being a new technology it will be possible to translate into a prototype all the theoretical knowledge obtained in them.

ABREVIATURAS

WIFI	Wireless Fidelity
LTE	Long Term Evolution
TCP/IP	Protocolo de control de transmisión/Protocolo de Internet
MQTT	Message Queue Telemetry Transport
IoT	Internet of Things
DHCP	Dynamic Host Configuration
LAN	Local Area Network
TTN	The Things Network
GPIO	General Purpose Input/Output
LPWAN	Low Power Wide Area Network
CO ₂	Dióxido de Carbono

ÍNDICE GENERAL

INTRODUCCIÓN	1
1.1 DESCRIPCIÓN DEL PROBLEMA.....	1
1.2 ANTECEDENTES	2
1.3 IMPORTANCIA Y ALCANCE	2
1.4 DELIMITACIÓN.....	3
1.4.1. DELIMITACIÓN TEMPORAL.....	3
1.4.2. DELIMITACIÓN ESPACIAL	3
1.4.3. DELIMITACIÓN ACADÉMICA	3
1.5 OBJETIVOS	3
1.5.1. OBJETIVO GENERAL	3
1.5.2. OBJETIVO ESPECÍFICO.....	3
1.6 DESCRIPCIÓN DE LA PROPUESTA	4
1.7 BENEFICIARIO DE LA PROPUESTA.....	4
1.8 IMPACTO.....	4
ESTADO DEL ARTE	5
2.1 CONTAMINACIÓN.....	5
2.1.1 DEFINICIÓN DE CONTAMINACIÓN	5
2.1.2 Dióxido de Carbono (CO ₂)	5
2.1.2.1 EMISIONES DE CO ₂ DENTRO DE UN VEHÍCULO	6
2.1.2.2 NIVELES DE CONCENTRACIÓN DEL CO ₂	6
2.2 INTERNET DE LAS COSAS (IoT).....	7
2.2.1 CAPA DE APLICACIONES	8
2.2.2 CAPA DE CENTRO DE DATOS	9
2.2.3 CAPA DE RED.....	10

2.2.4	CAPA DE GATEWAY	11
2.2.5	CAPA DE SEGURIDAD	11
2.2.6	CAPA DE DISPOSITIVOS	12
2.2.7	CAPA DE GESTIÓN	13
2.3	LoRa.....	13
2.4	LoRaWAN	14
2.4.1	ARQUITECTURA LORAWAN.....	15
2.4.2	CLASES DE LoRaWAN	16
	MARCO METODOLÓGICO	18
3.1	DISEÑO E IMPLEMENTACION	18
3.2	TECNOLOGÍAS A IMPLEMENTAR	20
3.3.1	PLATAFORMA ARDUINO	20
	3.3.1.1 HARDWARE ARDUINO.....	20
	3.3.1.2 SOFTWARE ARDUINO	21
3.3.2	ESPECIFICACIONES LORA	22
3.3	DETECCION Y MONITORIZACION	23
3.4.1	SENSOR DE DIOXIDO DE CARBONO SEN0219.....	23
3.4.2	ARDUINO MEGA 2560	25
3.4	COMUNICACION LORA	26
3.5.1	GATEWAY DRAGINO LG02.....	27
3.5.2	CONFIGURACIÓN DE GATEWAY DRAGINO LG02	28
3.5.3	DRAGINO LORA SHIELD.....	31
3.5	THE THINGS NETWORK	32
3.6.1	REGISTRO DE UN GATEWAY	34
3.6.2	REGRISTRO DE UN NODO FINAL.....	35

3.6	MY DEVICES CAYENNE	37
3.7.1	REGISTRO DE SENSOR EN CAYENNE	38
3.7	DIAGRAMA ELECTRICO.....	39
	PRUEBAS Y ANALISIS DE RESULTADOS	40
4.1	PRUEBAS DE FUNCIONAMIENTO.....	40
4.1.1	ETAPA DE OBTENCION DE DATOS	40
4.1.2	ETAPA DE VISUALIZACIÓN	41
4.1.3	ETAPA DE ENVÍO DE NOTIFICACIONES	42
4.2	PRUEBAS Y ANALISIS DE RESULTADOS.....	43
4.2.1	VEHÍCULO EN MOVIMIENTO CON VENTANAS ABIERTAS	43
4.2.2	VEHÍCULO ENCENDIDO Y ESTACIONADO CON VENTANAS CERRADAS 45	
4.2.3	PROTOTIPO COLOCADO CERCA DEL TUBO DE ESCAPE	46
	CONCLUSIONES	48
	RECOMENDACIONES	49
	BIBLIOGRAFÍA Y REFERENCIAS	50
	ANEXOS	52
	ANEXO A. LISTADO DE MATERIALES Y PRESUPUESTO	52
	ANEXO B. PROGRAMACIÓN IMPLEMENTADA EN LA TARJETA ARDUINO	53
	ANEXO C. ESPECIFICACIONES DEL GATEWAY LG02, SENSOR CO ₂ Y LORA SHIELD	58
	ANEXO D. ESPECIFICACIONES DE ARDUINO MEGA 2560	63

ÍNDICE DE FIGURAS

Figura 2.1 Capas de la arquitectura IoT.....	8
Figura 2.2 Aplicaciones de la IoT.....	9
Figura 2.3 Las principales funciones de la capa de gateway.....	11
Figura 2.4 Función de la Capa de Dispositivos.....	12
Figura 2.5 Parámetros de LoRa.....	14
Figura 2.6 Arquitectura LoRaWAN.....	16
Figura 2.7 Clases de Dispositivos LoRaWAN.....	17
Figura 3.1 Proceso en diagrama de bloques.....	18
Figura 3.2 Componentes del prototipo.....	19
Figura 3.3 Tarjetas Arduino.....	21
Figura 3.4 Interfaz de Programación.....	22
Figura 3.5 Red LoRa.....	23
Figura 3.6 Distribución de pines.....	24
Figura 3.7 Esquema de conexión Lora Shield y SEN0219.....	25
Figura 3.8 Arduino Mega 2560.....	26
Figura 3.9 Esquema de conexión Arduino Mega 2560 y Lora Shield.....	26
Figura 3.10 Gateway Dragino LG02.....	28
Figura 3.11 Ingreso a configuración Gateway.....	28
Figura 3.12 Configuración Wireless.....	29
Figura 3.13 Configuración de servidor.....	29
Figura 3.14 Etiqueta de equipo.....	30
Figura 3.15 Configuración de transmisión.....	30
Figura 3.16 Lora Shield.....	32
Figura 3.17 Ubicación de Gateways TTN.....	32

Figura 3.18 Arquitectura de Red.	33
Figura 3.19 Cuenta en TTN.	34
Figura 3.20 Plataforma TTN.....	34
Figura 3.21 Registro de Gateway.....	35
Figura 3.22 Gateway registrado.....	35
Figura 3.23 Registro de la aplicación.....	36
Figura 3.24 Registro de dispositivo	36
Figura 3.25 Dispositivo registrado.....	37
Figura 3.26 Selección de dispositivo en Cayenne	38
Figura 3.27 Registro de dispositivo en Cayenne.....	38
Figura 4.1 Obtención de datos del sensor de CO2	40
Figura 4.2 Visualización de datos en aplicativo móvil	41
Figura 4.3 Ejemplos de señal aceptable	42
Figura 4.4 Registro de correos electrónicos.....	43
Figura 4.5 Prototipo dentro de vehículo	44
Figura 4.6 Valor obtenido con vehículo en movimiento.....	44
Figura 4.6 Vehículo estacionado y con vidrios elevados.....	45
Figura 4.7 Valor obtenido con vehículo estacionado y ventanas cerradas	45
Figura 4.8 Valor obtenido con el prototipo cerca del tubo de escape	46
Figura 4.9 Niveles promedios de concentración de CO2.....	46
Figura I. Características del Gateway LG02.	58
Figura II. Aplicaciones del Gateway LG02.	58
Figura III. Especificaciones técnicas del Gateway LG02.....	59
Figura IV. Características y Especificaciones técnicas del Sensor CO ₂	60
Figura V. Descripción general del Sensor CO ₂	60

Figura VII. Asignación de pines del Lora Shield.....	61
Figura VIII. Especificaciones técnicas del Lora Shield.....	61
Figura IX. Características del Lora Shield.....	62
Figura XI. Especificaciones técnicas del Arduino Mega 2560.....	63
Figura XII. Mapeo pines Arduino y los puertos de Arduino Mega 2560.	64
Figura XIII. Asignación de pines del Arduino Mega 2560.....	65

ÍNDICE DE TABLAS

Tabla 2.1 Niveles de concentración de CO ₂	6
Tabla 2.2 Tabla comparativa de los diferentes protocolos de IoT.....	10
Tabla 2.3 Tabla de Frecuencias LoRa.	13

INTRODUCCIÓN

El presente proyecto de titulación consiste en la implementación de un prototipo del cual monitorea el dióxido de carbono (CO₂) en un ambiente cerrado como es la cabina de un vehículo. Además, con el fin de evitar que las personas inhalen este tipo de gas, el cual puede ser perjudicial para la salud, el prototipo es capaz de generar alertas en mensajes de texto cuando exista un nivel de concentración que supere los rangos establecidos.

El dióxido de carbono presenta varios efectos negativos en la respiración de los individuos dentro de los vehículos, debidos a los niveles de concentración, provocando fatiga pudiendo llevar a una situación de asfixia cuando se encuentra en altas concentraciones.

Con el propósito de prevenir la inhalación de dicho gas nocivo se requiere implementar un modelo que ayude a detectar los niveles de CO₂, mediante la colaboración de LoRaWAN que brinda seguridad, recepción y transmisión de datos, bajo consumo de energía.

Para cumplir con los objetivos propuestos se ha dividido en los siguientes cuatro capítulos.

En el capítulo I, se expone de manera general la descripción del problema, importancia y objetivos.

En el capítulo II, se explica el marco teórico donde se abordan temas conceptuales importantes para la investigación como lo es la contaminación, dióxido de carbono, niveles de concentración del CO₂, IOT y LoRaWAN.

En el capítulo III, se elaboró el marco metodológico se detalla las herramientas necesarias de programación y hardware requeridos para el diseño del prototipo de detección de CO₂.

En el capítulo IV, se explica el procedimiento de evaluación y análisis del prototipo, adicionalmente se muestra los costos del prototipo implementar.

1.1 DESCRIPCION DEL PROBLEMA

Esta investigación se orienta a prevenir una de las muchas causas de muertes en el país y en el mundo como es la inhalación del dióxido de carbono (CO₂) ya sea que el sujeto se encuentre dentro del vehículo del cual este estuviera en marcha o reposo.

Por lo tanto, cada vez que una persona muere por la inhalación de dióxido de carbono, se vuelve un riesgo permanente al que se expone los pasajeros de un vehículo cuando no se detectan a tiempo o no se hace un cuidado preventivo del sistema de escape de los vehículos. Aunque el desarrollo tecnológico de los

automóviles ha hecho que los sistemas de escape sean más amables con el medio ambiente gracias a implementaciones de catalizadores, esto no significa que no sean nocivos para la salud humana. El dióxido de carbono es un gas no irritante, no emite olor por lo que su presencia pasa desapercibida y se origina en la combustión incompleta de los combustibles orgánicos.

Ciertas personas tienen un mal hábito de dormir dentro de su vehículo después de pasar una jornada de largas horas sin descanso, lo cual es un error que muchos cometemos, debido a los gases que produce la combustión de la gasolina mientras esta encendido el motor del auto y uno de ellos es el dióxido de carbono. Sin embargo, cuando se inhala una mínima cantidad de CO₂ este se combina con la hemoglobina de la sangre, la hemoglobina es la que transporta el oxígeno por todo el organismo la cual puede llegar a ser mortal y a si ha sido para muchos conductores en el mundo.

Dentro del vehículo, existen varios factores en el cual el ser humano no puede omitir, uno de ellos debes ser el correcto mantenimiento y posibles fallas de circulación del aire en espacios pequeños. Al momento de inhalar este gas puede producir varios síntomas mortales a la persona, una de ellas es llamado “la muerte dulce”, por ser muy rápida y por no producir angustia, es decir que el individuo, a medida que va inhalando el dióxido de carbono, siente una especie de mareo, que es el resultado de una hipoxia, es decir el cerebro se va quedando sin oxígeno, por lo que en ningún momento tiene sensación de que va a morir.

1.2 ANTECEDENTES

En las últimas décadas, se han producido múltiples muertes a nivel mundial debido a una larga estadía del conductor dentro del vehículo , ya sea que se encuentre en movimiento o en reposo, por medio de las emisiones causadas por el motor de combustión interna de gasolina de los automóviles, esto ha ocasionado un aumento en los niveles de contaminación en el interior de la cabina del vehículo.

Al ser este, un detonante de asfixio simple, el cual se da por el desplazamiento del oxígeno que al tener elevadas concentraciones puede causar ciertas afectaciones puede causar dolor de cabeza, mareos, somnolencia y problemas respiratorios, dependiendo de la concentración y de la duración de la exposición.

1.3 IMPORTANCIA Y ALCANCE

Esta investigación es importante, ya que da a conocer a los seres humanos el efecto negativo que provoca inhalar concentraciones altas de CO₂.

De acuerdo a lo expuesto, se plantea como solución a este problema que ha traído consigo varias muertes a lo largo de los años, la utilización de la tecnología LoRaWAN, que tiene como ventaja la implementación de dispositivos de bajos

costos y bajo consumos de energía en un esquema IoT (Internet de las cosas), a través de una red LPWAN (Low Power Wide Area Network).

De este modo, se obtendrá en tiempo real los datos que ingresen de CO₂ dentro de la cabina del vehículo cuando se encuentre en reposo o movimiento, tanto para obtener valores exactos de CO₂ dentro del vehículo.

1.4 DELIMITACIÓN

1.4.1. DELIMITACIÓN TEMPORAL

La información que serán considerados para la implementación de este proyecto está prevista a realizarse en el 2019 hasta el 2020 en la ciudad de Guayaquil.

1.4.2. DELIMITACIÓN ESPACIAL

El proyecto propuesto fue desarrollado en la residencia de la estudiante María Lara Ponce, la cual está ubicada vía Daule, dado a los motivos recientes que el presente año 2020 ha presentado.

1.4.3. DELIMITACIÓN ACADÉMICA

Académicamente la investigación se basa en el área de las telecomunicaciones donde se aplican los conocimientos de redes inalámbricas, comunicaciones digitales y medios de transmisión.

1.5 OBJETIVOS

Diseñar e implementar un servidor IoT de bajo costo utilizando LoRaWAN, Arduino, Wifi y una app Android para monitorear el ingreso de dióxido de carbono dentro de la cabina de un vehículo.

1.5.1. OBJETIVO GENERAL

Implementar un sistema de detección de niveles de dióxido de carbono (CO₂) dentro de la cabina de vehículos utilizando comunicación LoRaWAN.

1.5.2. OBJETIVO ESPECÍFICO

- Diseño del sistema para la integración electrónica entre LoRa.
- Diseño del diagrama de comunicación para el uso de LoRaWAN.
- Diseño de un sistema con software libre para la interfaz de monitoreo.
- Diseño de las interfaces en tiempo real.

1.6 DESCRIPCIÓN DE LA PROPUESTA

La propuesta consiste en usar tecnología IoT y LoRaWAN para lograr el monitoreo de las emisiones de CO₂, con el objetivo de prevalecer la vida del ser humano al momento de que son expulsadas dichas toxinas e implementar un sistema de detección que permita alertar cuando los rangos de CO₂ se elevan.

1.7 BENEFICIARIO DE LA PROPUESTA

Este proyecto tendrá como primeros beneficiarios a Militares, Policías, ATM, Fuerzas Armadas y al resto de las personas, así como un gran crecimiento tecnológico, orientado a salvar miles de vidas que día a día usan vehículos para poder transportarse a diferentes lugares ya sea trabajo, compras, viajes, o simplemente descansar en sus vehículos. Logrando así un aprendizaje fundamentado en la práctica, otorgando una formación técnica – profesional importante en el campo laboral.

Este proyecto aportará en la enseñanza a los alumnos sobre los sistemas de detección de niveles de dióxido de carbono (CO₂), utilizando comunicación LoRa WAN, también ayudará a miles de personas que realizan diferentes actividades en su diario vivir.

1.8 IMPACTO

El desarrollo de la investigación ayudará a estudiantes a fortalecer los conocimientos adquiridos en las distintas materias estudiadas a lo largo de la carrera. A través del Sistema LoRaWAN cuyo único fin es reducir la tasa de muertes en el país causadas por la inhalación del dióxido de carbono dentro de la cabina del vehículo.

ESTADO DEL ARTE

2.1 CONTAMINACIÓN

La contaminación es un problema global que perjudica la salud de las personas y al medio ambiente, una de las principales consecuencias que ocasiona es el calentamiento global y de las cuales existe una diversidad de tipos de contaminación dependiendo del ámbito en que se produzcan las alteraciones de las condiciones normales para la vida.

La contaminación con el pasar de los años se ha intensificado debido al crecimiento tecnológico que se ha presentado en el mundo, mucho de esto es gracias al consumo de combustibles por parte de las industrias químicas y de procesos industriales o a las emisiones de gases que causan el efecto invernadero.

Aquello ha producido que el ser humano tenga un mejor estilo de vida, pero perjudicando al planeta, lo cual ha provocado niveles de contaminación elevados en país desarrollados y de menor proporción en país en vías de desarrollo.

2.1.1 DEFINICIÓN DE CONTAMINACIÓN

La contaminación se puede definir como la presencia en el ambiente de cualquier compuesto químico, físico o biológico que pueden causar daños, riesgos o molestias para la salud y bienestar de las personas, de los animales y de las plantas. (Inspiration, 2020).

La contaminación es la existencia o acumulación de sustancias en el medio ambiente que afectan negativamente el entorno y las condiciones de vida. El contacto con contaminantes tóxicos puede darse durante procesos de producción, distribución o utilización de productos, o bien cuando éstos son desechados al ambiente.

Según la Organización Mundial de la Salud (OMS, 2020), la contaminación del aire “Representa un importante riesgo medioambiental para la salud. Mediante la disminución de los niveles de contaminación del aire los países pueden reducir la carga de morbilidad derivada de accidentes cerebrovasculares, cánceres de pulmón y neuropatías crónicas y agudas, entre ellas el asma”.

2.1.2 Dióxido de Carbono (CO₂)

El dióxido de carbono es un gas incoloro e inodoro. La emisión de dióxido de carbono en la respiración humana está ligada a la de otros productos procedentes del metabolismo humano (agua, aerosoles biológicos, partículas, alcoholes, aldehídos, etc.). El dióxido de carbono es un asfixiante simple que actúa básicamente por desplazamiento del oxígeno a elevadas concentraciones.

“Desde el inicio de la Industrialización, la concentración de CO₂ en la atmósfera ha aumentado considerablemente”, afirma Friedrich-Wilhelm Gerstengarbe. (Cheung, 2012). Por ese motivo es el principal culpable del calentamiento global durante las últimas décadas del cual conlleva un incremento de la temperatura en todo el planeta y como consecuencia a largo plazo se pueden dar un incremento del nivel del mar, la extinción de varias especies, la decadencia de los recursos naturales, entre otros.

Las personas también expulsamos CO₂ del aire que respiramos, lo que quiere decir que somos una fuente de este gas. Por ello cuando hay lugares cerrados o donde el aire no fluye con facilidad, pero existe una conglomeración de personas puede producir el bajo rendimiento de concentración y rendimiento.

2.1.2.1 EMISIONES DE CO₂ DENTRO DE UN VEHÍCULO

Las emisiones de CO₂ se producen por la quema del combustible y son expulsadas a través del tubo de escape. La cantidad de CO₂ emitida, depende de la cantidad de energía necesaria para circular y de la eficiencia del motor. No todos los vehículos emiten los distintos tipos de contaminantes en la misma proporción, ya que depende del tipo de motor que lleven y si consumen gasolina o diésel.

El correcto mantenimiento del vehículo da como resultado una óptima calidad y reducción de CO₂. Y una adecuada ventilación del coche previene de que estos gases ingresen al interior de la cabina.

2.1.2.2 NIVELES DE CONCENTRACIÓN DEL CO₂

A continuación, se presenta una tabla indicando los niveles de concentración de dióxido de carbono en espacios cerrados, del cual muestra el comportamiento del cuerpo humano cuando inhala este gas.

Tabla 2.1 Niveles de concentración de CO₂.

Nivel de CO ₂ (ppm)	Resultado en Humanos
350-450	Concentracion atmosferica tipica
600-800	Calidad del aire interno aceptable
1000	Calidad del aire interno tolerable
2000-3000	Limite promedio de exposicion en un periodo de 8 horas
3-8%	Incremento de la frecuencia respiratoria, dolor de cabeza
>10%	Nauseas, vomitos, perdida de conocimiento
>20%	Perdida de conocimiento repentina, muerte

Fuente (Vaisala, 2013)

2.1.2.3 EFECTOS SOBRE LA SALUD DEL GAS CO₂

El principal efecto que produce el CO₂ es la asfixia por desplazamiento del oxígeno, pero esto se debe a que la concentración de este gas debe llegar cerca de los 1000 ppm. En concentraciones cercanas a la 800 ppm, puede causar dolores de cabeza, falta de concentración, somnolencia, mareos y problemas respiratorios. (AxioMET, 2020). También se encuentran de forma natural en el aire, concentraciones que varían entre 300 ppm a 550 ppm dependiendo de los entornos rurales o urbanos. En entornos laborales, como oficinas, se comienzan a tener quejas de olores a partir de las 800-1000 ppm. (Geoambiental, s.f.)

2.2 INTERNET DE LAS COSAS (IoT)

Es el conjunto e interconexión de dispositivos y objetos a través de una red IP, donde cada uno de ellos debe ser visibles e interaccionar, usando diversas tecnologías las cuales tienen capacidades de conexión e interacción con el entorno, dichas capacidades que les permiten tomar decisiones y comunicarse con el mundo. También se puede decir que el IoT es un sistema de radiofrecuencia que nos sirve de gran ayuda para identificar, rastrear y localizar recursos de forma única, precisa y automática por medio de ondas de radio inalámbricas.

Cada vez está más cerca las redes móviles 5G, ella abre a un mundo de posibilidades tecnológicas y así como una comunicación más rápida y confiable porque a medida que los celulares y otros dispositivos se vuelven más inteligentes y excesivos, las múltiples empresas a nivel internacional necesitan envolver las necesidades de sus clientes por ello están implementando este tipo de red móvil para brindar seguridad, fiabilidad y rapidez a los usuarios.

Con la llegada de la red móvil 5G las compañías tienen la necesidad de crear una plataforma de IoT porque cada vez las personas tienen en sus hogares o trabajos uno o varios de estos dispositivos inteligentes, por ejemplo: teléfonos digitales, relojes o smartwatch, luz inteligente, computadoras, sensores, electrodomésticos, entre otros.

También podemos decir que *“Uno de los mayores beneficios de IoT es la eficiencia que puede ofrecer. Muchas empresas la usan para automatizar los negocios y los procesos de fabricación, monitorear y controlar de manera remota las operaciones, optimizar las cadenas de suministro y conservar los recursos”*. (SAP, 2019)

Hoy en día los numerosos avances tecnológicos han permitido la conectividad del Internet a cualquier lugar que uno desee, esto trae por consecuencia la cantidad de datos generados sea cada vez mayor, por lo tanto, se puedan emplear de manera segura la captación de información útil que uno envía o recibe. A través de los años los aparatos electrónicos han sufrido un cambio o adaptación al medio que uno está acostumbrado, haciéndose muy prácticos y fáciles de usar por la necesidad del usuario. Una parte muy importante que da el IoT es poder visualizar toda la

información obtenida a partir de los datos censados por los dispositivos y poder guardarlos en la nube que el usuario a autorizado.

Una de las cuestiones más importante que genera el IoT es la seguridad y privacidad, debido al gran volumen de tráfico que se genera en Internet por lo que transporta muchos datos personales, los cuales deben ser protegidos para que no exista la manipulación o acceso no autorizado a los dispositivos que se encuentran conectados en el hogar o lugar de trabajo.

Uno de los mayores principios que representa la arquitectura IoT es conceder escalabilidad, ampliación de capacidades y soporte de nuevos estándares, por ello, debe ser horizontal, para que pueda aceptar la integración de diferentes soluciones IoT y así servir de modelo para la creación de arquitecturas más específicas.



Figura 2.1 Capas de la arquitectura IoT. *Fuente:* (Benitez, 2016).

A continuación, hablaremos la importancia de cada de las capas que conforman la arquitectura del IoT para poder entender mejor su funcionamiento en este tipo de tecnología que cada vez va avanzando en la actualidad. (Benitez, 2016).

2.2.1 CAPA DE APLICACIONES

Es aquella que se dedica en las aplicaciones del usuario, desde lo más básico hasta lo más avanzado, desde las aplicaciones domesticas hasta las aplicaciones para las empresas, que optimicen los recursos y su tiempo. Desde esta capa se puede encontrar sensores, actuadores y el hardware necesario para notificar el mundo físico con el virtual, de esta manera se puede manejar y visualizar la información que necesitamos para poder ser amigables al ser humano, también estas aplicaciones tendrán la capacidad de modificar los parámetros para que los sistemas se comporten de una manera pre-determinada y así accediendo no solo a la visualización sino también al manejo.



Figura 2.2 Aplicaciones de la IoT. Fuente: (CIC, 2019)

2.2.2 CAPA DE CENTRO DE DATOS

Esta capa de centro de datos debe estar directamente vinculadas con varios de los objetivos y necesidades del IoT y a su vez debe contar con la capacidad de manejar los dispositivos que se encuentren conectados en el hogar o empresa donde requiera el usuario en tiempo real y soportar los protocolos que se puedan aplicar en los mismos.

Estos dispositivos están insistentemente enviando datos, por lo cual ocasionan un tráfico de información para ello se requiere que los sistemas de almacenaje cuenten con una gran capacidad de escalabilidad para poder mantener los diferentes tipos y grandes volúmenes de datos y esto en tiempo real. Además de la escalabilidad, otro de los requerimientos más importantes para esta arquitectura del IoT es que pueda contar con el soporte necesario para la integración de pequeños o grandes dispositivos, y así, también desarrollar una infraestructura Cloud.

Una vez que el sistema IoT haya recolectado, almacenado y analizado la información de los dispositivos electrónicos, este debe ser capaz de estudiar estos datos y poder lanzar alertas basados en reglas según sus estándares que tengan dichos dispositivos conectados, por ejemplo, si en un lugar existe un sensor de temperatura, este sensor debe ser capaz de medir o detectar una variación de temperatura que se encuentra en el ambiente para transfórmalo a señal eléctrica para que pueda llegar la información al gateway o dispositivo que controle este sensor. Este sensor está formado por unos materiales metálicos de diferentes naturalezas que juntos mediante uniones se los llaman unión fría y unión caliente. En base a esto dicho dispositivo debe ser capaz de tomar la decisión de que temperatura ambiental ya sea calefacción o el aire acondicionado. Ahora con el sistema IoT ya no dependerá simplemente de efectuar esa disposición porque también se podrá añadir parámetros como el número de personas que hay en el lugar para obtener una temperatura eficaz y agradable a las personas.

2.2.3 CAPA DE RED

En esta capa de red se mencionan las diferentes redes que brindan soporte a los sistemas IoT, como bien su nombre lo indica “Internet de las cosas”, aquí podemos apreciar lo importante que va a hacer la arquitectura de red porque en ella va a recaer o soportar todo el ecosistema IoT. (Benitez, 2016).

Una manera más simple de decir es que esta capa se va a encargar de conectar los dispositivos electrónicos a los dispositivos de red o servidores. También debe disponer de herramientas necesarias para transmitir datos entre dispositivos y así poder realizar cierto margen de procesamiento de los mismos.

Por ello para poder conectar ambos mundos como lo es el mundo físico y el mundo virtual se deberá abrir un canal de comunicación a través de interfaces hardware porque es donde hay una compleja arquitectura donde la información fluye en ambos sentidos.

Para este tipo de capa se necesita protocolos para la comunicación entre dispositivos y el Cloud, una de las más conocidas es el HTTP (Hypertext Transfer Protocol) porque hay variedades de librerías que pueden soportar y puede ofrecer una conectividad unificada y uniforme, pero a su vez también presenta unos problemas, uno de ellas es el tamaño de memoria del programa y esto puede producir una pequeña inconformidad en dispositivos pequeños, y otro pequeño problema es que el protocolo HTTP tiene una mayor demanda de energía, por lo tanto, necesitaremos un protocolo binario pequeño y sencillo para que a su vez pueda superar con habilidad los firewalls.

Debemos tomar en cuentas que no todos los dispositivos que existen se van a conectar directamente a la red, otros lo harán por medio del gateway, pero a su vez no se puede olvidar que dichos dispositivos van a necesitar dos protocolos adicionales para que así uno se pueda conectar al gateway y otro conectarse del gateway a la nube. No solamente existe el protocolo HTTP, ya que hay una gran variedad de protocolos, pero las conocidas son MQTT de ZigBee. Profinet de PROFIBUS, CoAP (Constrained application protocol) de IEFT (Internet Engineering Task Force), entre otros.

Tabla 2.2 Tabla comparativa de los diferentes protocolos de IoT.

PROTOCOLO	TRANSPORTE	MENSAJERIA	RED 2G, 3G, 4G	SEGURIDAD	BAJO PODER Y PERDIDAS
CoAP	UDP	Solicitar/Responder	Excelente	Media	Excelente
HTTP	TCP	Solicitar/Responder	Excelente	Baja	Justa/Al limite
MQTT	TCP	Solicitar/Responder	Excelente	Media	Buena
SNMP	UDP	Solicitar/Responder	Excelente	Alta	Justa/Al limite

Fuente: (Galarza, Lara).

2.2.4 CAPA DE GATEWAY

Es aquella que está formada por todos los dispositivos que intervienen en el lugar que se está empleando la arquitectura del IoT, su principal funcionalidad puede darse tanto en software como en hardware según como desee el usuario, en el mercado existen varios dispositivos electrónicos que usa este tipo de sistema y cada uno de ellas tiene un protocolo diferente, es decir, que se van a necesitar diferentes tipos de gateway para poder interconectar los diferentes dispositivos.

Por ende, si se llega a aplicar un gateway compatible con un solo protocolo de comunicación, los dispositivos deberían comunicarse con el protocolo que se está usando, por eso es muy difícil la comunicación de varios dispositivos de diferentes fabricantes. (Benitez, 2016).

Se concluye que la manera más eficaz para este tipo de problema es utilizar un gateway que permita la comunicación de varios protocolos y así se pueda integrar diferentes dispositivos de diferentes marcas de IoT para poder garantizar la interoperabilidad en el ecosistema o entorno.



Figura 2.3 Las principales funciones de la capa de gateway. Elaborado por: (Galarza, Lara).

2.2.5 CAPA DE SEGURIDAD

La capa de seguridad es uno de los principales y más importantes requisitos que debe haber para poder hacer uso del sistema IoT, debido a la cantidad de datos e información que recolectan de los dispositivos. (Benítez, 2016).

Por lo tanto, en esta capa debe constar algunas funciones que nos permitan a nosotros los usuarios poder contar con la seguridad necesaria para no evitar robo de informaciones personales. Algunas de las funciones deberían de ser: la autorización y validación de cara a los dispositivos, datos y usuarios; un control de acceso para la autorización y validación; un mantenimiento de actualizaciones; un antivirus para poder garantizar la seguridad de los diferentes dispositivos que

conforman el sistema IoT y, por último, no menos importante es la protección de hardware de dichos dispositivos. Dicho esto, aun en la actualidad nadie puede asegurar el 100% de la seguridad informática, pero si podemos poner algunas trabas para que el delincuente no tenga un camino fácil hacia el robo de información.

Uno de los ejemplos que se puede dar es como un banco monetario tiene más de 2 firewalls para poder evitar el robo de información y dinero, estos tipos de sistemas tienen diferentes comandos, así, si el delincuente logra evadir un firewall solo tendrá pocos minutos de acceso antes de que el sistema de protección detecte una irregularidad, entonces al pasar el primer nivel de seguridad inmediatamente deberá superar el otro tipo de nivel de cortafuegos de diferente empresa, esto hará más difícil el acceso por ello es casi imposible el robo de información en un banco. Entonces si la cerradura de tu casa trabaja con un sistema IoT y este es hackeado a través de un móvil podrían entrar a tu casa, por ello esta capa debe darnos diferentes funciones para generar obstáculos al delincuente.

2.2.6 CAPA DE DISPOSITIVOS

Esta capa es la base de toda la arquitectura y se encarga de múltiples funciones sobre todo con el medio físico que se encuentra el sistema IoT, cada dispositivo es capaz de obtener datos de una variedad de sensores que rastrean los parámetros más críticos. Estos dispositivos también se pueden usar para poder controlar el estado del equipo, es decir, que esta capa proporciona la conectividad de la última milla al equipo y de poder controlarlos a distancia. Por este motivo la capa de dispositivos es la interfaz entre el mundo físico y el mundo de la información.

Esta capa de dispositivos debe cumplir unas funciones especiales para el fácil uso del IoT; como, por ejemplo, debe generar y responder las peticiones del usuario, sensar para obtener información del medio físico, enviar constantemente reportes de manera automática del estado de los equipos, actuar de manera autónoma si hay un problema sobre el medio físico y por ultimo un modo de energía cuando no se estén usando los equipos que hay en el lugar.

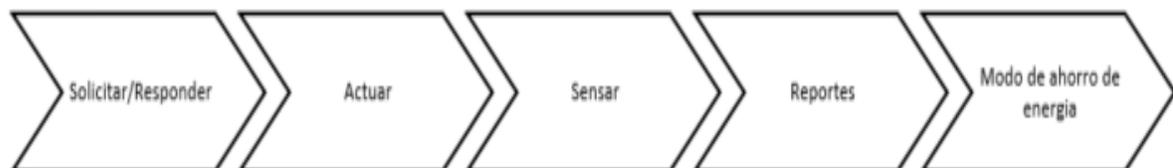


Figura 2.4 Función de la Capa de Dispositivos. Elaborado por: (Galarza, Lara).

2.2.7 CAPA DE GESTIÓN

La capa de gestión es la principal de todas las capas ya mencionadas anteriormente, porque ella es la encargada del correcto funcionamiento del sistema IoT, por el cual nos permite tanto a los usuarios como a los operadores, una variedad de herramientas para poder controlar y observar el correcto funcionamiento de dicho sistema. (Benitez, 2016).

La capa de gestión en conjunto con las demás capas da la posibilidad de desconectar un dispositivo robado, la habilidad de actualizar un software y credenciales de seguridad, localizar dispositivos perdidos, reconfigurar parámetros de Wi-Fi, GPS u otras redes, limpiar o restablecer la información de un equipo, autorización de entrada a los equipos, gestión del tráfico y la congestión, gestión de la topología de red local. Las funciones dichas con anterioridad no serán necesarias en todas las arquitecturas porque habrá casos en las que no se las vayan a necesitar o tengan alguna utilidad para nosotros.

Esta capa también está diseñada para prevenir posibles problemas, si hay una probabilidad de un ataque y que impacto tendría en la información del usuario.

2.3 LoRa

LoRa significa long range (largo alcance), esta es un tipo de red LPWAN (Low Power Wide Area Network). Esta tecnología proporciona que los dispositivos IoT conectados intercambien pequeñas cantidades de datos a baja velocidad con un largo alcance y bajo consumo de energía. Se utiliza la modulación en radiofrecuencia, como la AM o la FM.

Tabla 2.3 Tabla de Frecuencias LoRa.

	Europa	America	Asia
Banda de Frecuencia	867-869 MHz	902-928 MHz	470-510 MHz
Canal de Ancho de Banda	125/250 KHz	125/500 KHz	En definición por comité técnico
Poder de Transmision	14 dBm	20/30 dBm	
Factor de Dispersion	7-12 SF	7-10 SF	
Date Rate	250 bps - 50 Kbps	980 bps - 21.9 Kbps	
Presupuesto de Enlace	155 dB	157 dB	

Fuente: (Galarza, Lara).

Esta tecnología de modulación se llama CSS (*Chirp Spread Spectrum*), y se usa en comunicaciones militares y espaciales desde hace varias décadas. La gran ventaja de la misma es que puede lograr comunicaciones a largas distancias (kilómetros) y tiene gran firmeza frente a las interferencias.

Es una tecnología que permite enviar datos de manera inalámbrica entre dispositivos, con capacidad de enviar y recibir información. Las frecuencias de comunicaciones que LoRa usa están primordialmente en las bandas ISM (industrial, científica y médicas); es decir, este tipo de tecnología puede operar fácilmente en la frecuencia menor a 1 GHz. (LoRaWAN, 2020). Esta característica puede ser muy útil para localizar y posicionar activos de valor en áreas de difícil acceso, en especial cuando se tiene pensado para aplicaciones industriales.

Para el uso de este tipo de frecuencia se debe respetar el valor de emisión, y así cualquier persona o empresa puede hacer el uso de dicha tecnología si necesidad de licencia alguna. LoRa suele operar en las bandas de 433 MHz (Asia), 868 MHz (Europa) y 915 MHz (América), dependiendo el país en el que se esté utilizando. Unos de los factores por el cual se usa este tipo de tecnología es por el gran alcance que posee (hasta 10 Km), conexión punto a punto, baja transferencia de datos (hasta 255 bytes), alta sensibilidad para recibir datos (-168dB), poco consumo de energía y seguridad, aplicando algoritmos para proteger los datos que se transmiten, pero no garantiza la confidencialidad de la información.



Figura 2.5 Parámetros de LoRa. Elaborado por: (Galarza, Lara).

Las señales emitidas por LoRa pueden atravesar edificios, pero también llegar a salas profundas como garajes y sótanos. Esta característica puede ser muy útil para localizar y posicionar activos de valor en áreas de difícil acceso. (Ferrer, 2019).

2.4 LoRaWAN

Es un protocolo de control de acceso a medios (MAC) para una red en estrella que usa la tecnología LoRa para comunicar y administrar dispositivos LoRa, LoRaWAN puede asignar a la segunda y tercera capa del modelo OSI, en la que se compone de 2 partes: Gateways y Nodos.

Los protocolos LoRaWAN son definidos por LoRa Alliance y se implementa sobre a modulación LoRa o FSK en bandas de radio industriales, científicas y médicas (ISM). (Lora Alliance, 2020). Permite variar dinámicamente el bitrate de los equipos que están en la red, desde 0.3 Kbps hasta 37.5 Kbps, dependiendo el chip y parámetros de modulación.

La red LoRaWAN permite la interconexión entre objetos inteligentes sin la necesidad de instalaciones complejas, y además dispone una amplia libertad al usuario final, al desarrollador y a las empresas que quieran instalar su propia red IoT. El protocolo y la arquitectura de la red LoRaWAN determina de forma fácil la vida útil de la batería de un nodo, la capacidad de red, la calidad de servicio, la seguridad, y la variedad de aplicaciones de la red. LoRaWAN apunta a los requisitos clave del IoT tales como los servicios de comunicación de una manera segura.

La comunicación entre nodos y gateways se realiza mediante diferentes canales de frecuencia y velocidades de datos. Mediante el uso de la técnica del espectro ensanchado, las comunicaciones con diferentes velocidades de datos no interfieren entre sí, para poder crear un conjunto de canales virtuales que aumentan la capacidad del gateway.

Los nodos puedes transmitir por cualquier cana disponible en cualquier momento y utilizando la tasa de datos disponibles siempre y cuando cumplan las siguientes normas:

- El nodo respeta el máximo ciclo de trabajo de transmisión en relación a la sub-banda utilizada y a las regulaciones locales.
- El nodo cambia de canal de forma pseudo-aleatoria para cada transmisión.
- El nodo respeta el tiempo de duración máxima de transmisión en relación a la sub-banda y a las regulaciones locales.

2.4.1 ARQUITECTURA LORAWAN

La arquitectura de red LoRaWAN se implementa en una topología estrella de estrellas, en la que las puertas de enlace retransmiten mensajes entre los dispositivos finales y un servidor de red central. Las puertas de enlace están enlazadas al servidor de red a través de conexiones IP y actúan como un puente transparente, simplemente convirtiendo paquetes RF en paquetes IP y viceversa.

Los dispositivos LoRaWAN pueden ser dos formas: en modo punto-a-punto (P2P) como en las redes LoRa, y en modo híbrido. En modo P2P que permite enviar mensajes utilizando redes LoRaWAN. Este modo requiere licencia LoRaWAN. Y en el modo híbrido, se utiliza una combinación de los modos LoRaWAN.

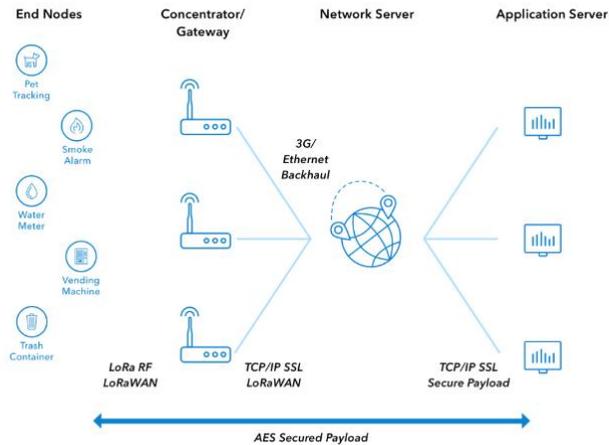


Figura 2.6 Arquitectura LoRaWAN. Fuente: (THE THINGS NETWORK, 2020)

A continuación, las diferentes áreas de la arquitectura LoRaWAN: (THE THINGS NETWORK, 2020)

- Dispositivo final: Un objeto con un dispositivo de comunicación integrada de baja potencia.
- Gateway: Antenas que reciben transmisiones de dispositivos finales y envían datos de regreso a dispositivos finales.
- Servidor de red: Servidores que enrutan mensajes desde dispositivos finales a la aplicación correcta y viceversa.
- Aplicación: Un software que se ejecuta en un servidor.

2.4.2 CLASES DE LoRaWAN

Existen tres clases de dispositivos LoRaWAN, denominadas Clase A, Clase B y Clase C. Todos los dispositivos implementan al menos la clase A. Pero, pueden implementar las demás clases como es la clase B y clase C.

Clase A.- Este tipo de clase es la más soportada y utilizada en casi todos los dispositivos IoT, porque nos permite una comunicación bidireccional, este tipo de clase ofrece el mayor ahorro de energía debido a la limitante de que solo puede recibir datos si han enviado antes un paquete. Esto se debe a que cada vez que un dispositivo envía un paquete e instantáneamente se abren dos ventanas de recepción, la primera un segundo después de la transmisión, y la segunda un segundo después de abrir la primera ventana, para que con ello se pueda dar la oportunidad de recibir el paquete de vuelta. (Sabas, 2017).

Clase B.- Este tipo de clase nos permite añadir la capacidad de recibir datos sin necesidad de haber enviado un paquete con base a tiempos predeterminados con el gateway. Pero a diferencia de la clase A, esta clase consume mayor energía de la batería o una fuente externa. (Sabas, 2017).

Clase C.- Este tipo de clase nos ofrece un mayor ahorro de energía en la batería porque siempre están permanentemente escuchando; es decir, en modo recepción, y cuando es necesario en modo de transmisión. Esta clase nos da los mejores tiempos de respuestas y capacidad de envío desde el servidor a los dispositivos, a costa de un mayor consumo de energía con respecto a las otras dos clases, por eso es recomendado que al usar este tipo de clase siempre cuenten con una fuente externa de alimentación. (Sabas, 2017)

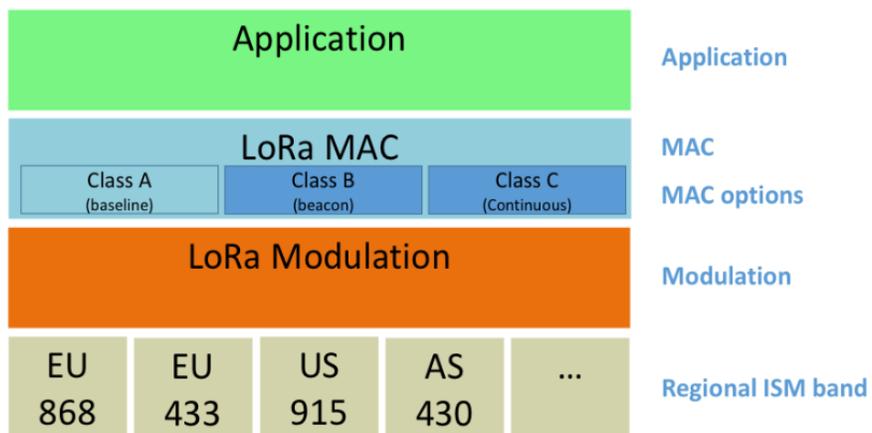


Figura 2.7 Clases de Dispositivos LoRaWAN. Fuente: (Sabas, 2017).

MARCO METODOLÓGICO

3.1 DISEÑO E IMPLEMENTACION

El prototipo mostrado en este proyecto capta los niveles de CO₂ dentro de una cabina simulada de un vehículo y muestra los datos en un servidor web asociado a los componentes utilizados para esta implementación; además, será capaz de detectar y notificar los niveles altos de concentración del gas dentro de ese ambiente.

En el presente capítulo se detalla las herramientas necesarias de programación y hardware requeridos para el diseño del prototipo de detección de CO₂.

En el siguiente diagrama de bloques se visualiza el proceso a seguir para cumplir con la implementación del prototipo.

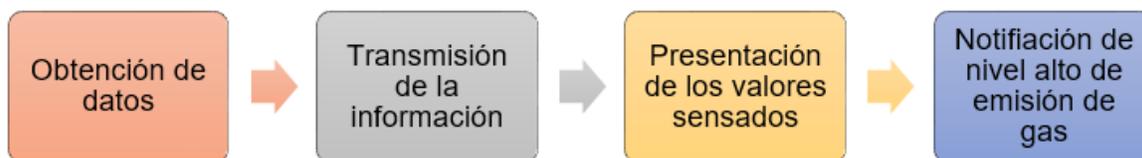


Figura 3.1 Proceso en diagrama de bloques. (Galarza, Lara)

Los requerimientos a cumplir son las siguientes:

- Recopilar los valores de concentración del gas CO₂ dentro de la cabina simulada de un vehículo.
- Interpretar los datos obtenidos por el sensor.
- Transmitir los datos hacia el servidor web.
- Presentar los datos en la aplicación web.
- Generar notificación cuando existen concentraciones en niveles elevados de gas.

En la Figura 3.1 se puede observar al detalle los dispositivos que forman parte del diseño e implementación del prototipo las cuales cumplen con los requerimientos mencionados anteriormente.

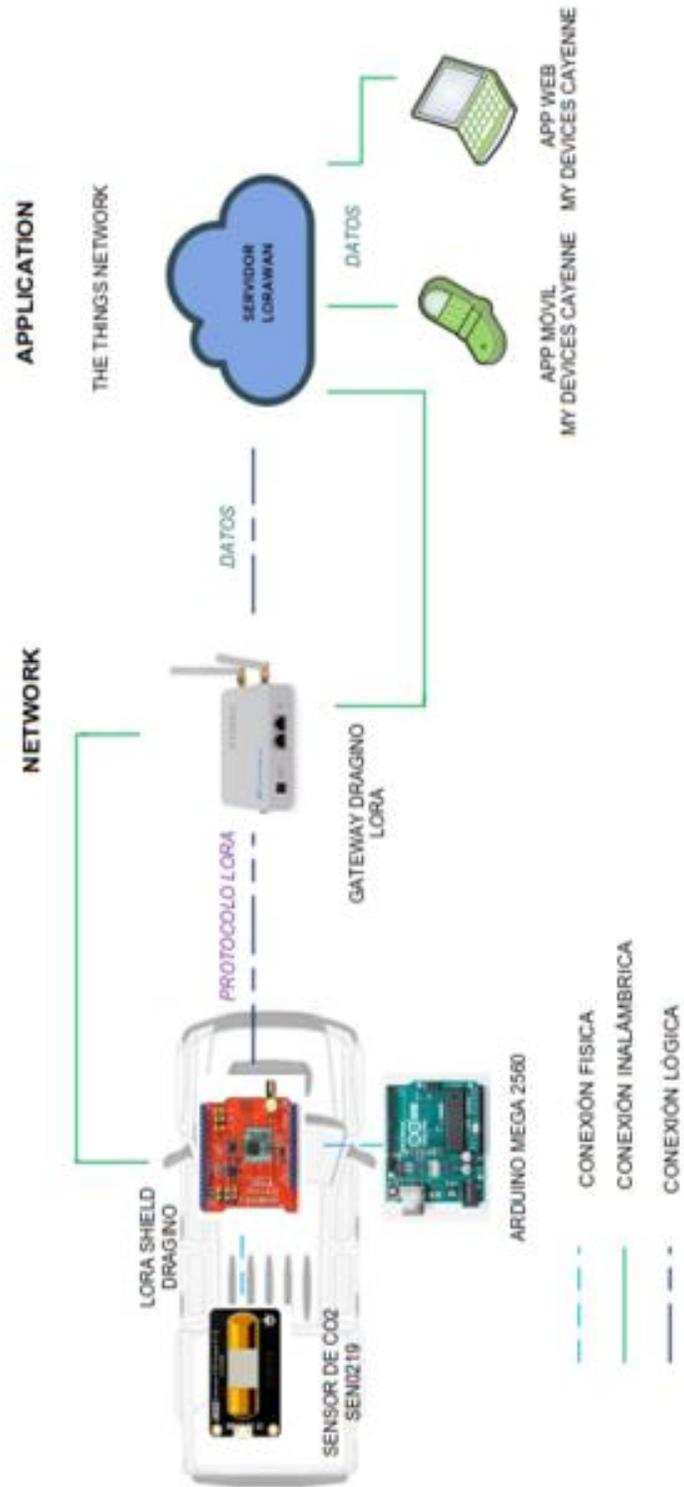


Figura 3.2 Componentes del prototipo. (Galarza, Lara)

3.2 TECNOLOGÍAS A IMPLEMENTAR

Dentro del ámbito de un proyecto académico es útil y de importancia recurrir a tecnologías libres como elementos principales para el diseño de cualquier sistema, ya que nos permite desarrollar servicios que satisfagan las diferentes necesidades humanas para un bien común. A continuación, se presenta las tecnologías libres a utilizar: Arduino y LoRa.

3.3.1 PLATAFORMA ARDUINO

Arduino, es un hardware de código abierto, programable a través de una plataforma de software propio, donde los contenidos creados por los usuarios se pueden compartir en una comunidad global (Gonzalez & Rocío, 2018).

Arduino permite realizar diferentes proyectos electrónicos interactivos desde lo más simple como es encender un led, hasta proyectos más complejos como el desarrollo de un brazo robótico.

Para el diseño del prototipo se escogió la plataforma Arduino por los siguientes motivos:

- Puede ser utilizado en varias plataformas como: Windows, Mac OSX y Linux
- Programación de fácil uso y manejo
- Permite realizar proyectos de diferente índole.
- Código abierto
- Bajo costo en las placas

3.3.1.1 HARDWARE ARDUINO

Arduino pone en manifiesto una gran variedad de tarjetas existentes, sin entrar en detalles técnicos, se menciona que en función de los requisitos de diseño e implementación será necesario el empleo de una placa u otra, ya que las tarjetas tienen un comportamiento similar, lo que las diferencia son su número de entradas y salidas, precio, tamaño, microcontrolador, comunicación, etc. Por tanto, cada aplicación puede variar y emplear bastantes placas sin mayor problema.

En la Figura 3.3, se muestran algunas placas existentes, estas están catalogadas dentro del grupo de tarjetas con características avanzadas.



Figura 3.3 Tarjetas Arduino. (Arduino, 2020)

Las placas están conformadas por entradas y salidas digitales o analógicas y su programación está enfocada para un fácil aprendizaje del usuario.

3.3.1.2 SOFTWARE ARDUINO

El IDE (Integrated Development Environment, Entorno de Desarrollo Integrado) (Gonzalez & Rocío, 2018) utiliza un lenguaje de programación similar a C++ donde en el denominado sketch (interfaz para programar) se puede introducir las diferentes líneas de comandos distribuidas en dos bloques principales:

- **Void setup:** Se realiza la configuración del equipo, también se asignan los diferentes pines de entradas y salidas necesarias para la programación de la placa.

- **Void loop:** Bucle que contiene el programa principal, se incluye todas las sentencias de programación necesarias para la funcionalidad de la implantación.

Mediante cable USB se puede cargar el programa en la tarjeta Arduino.

```

ttn-otaa-prueba Arduino 1.8.2
Archivo Editar Programa Herramientas Ayuda
ttn-otaa-prueba
28 * Do not forget to define the radio type correctly in config.h.
29 *
30 *****/
31
32 #include <mic.h>
33 #include <hal/hal.h>
34 #include <SPI.h>
35
36 // This EUI must be in little-endian format, so least-significant-byte
37 // first. When copying an EUI from ttnctl output, this means to reverse
38 // the bytes. For TTN issued EUIs the last bytes should be 0xD5, 0xB3,
39 // 0x70.
40 static const ul_t PROGMEM APPEUI[8]={ 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x02, 0x36, 0x39
41 void os_getArtEui (ul_t* buf) { memcpy_P(buf, APPEUI, 8);}
42
43 // This should also be in little endian format, see above.
44 static const ul_t PROGMEM DEVEUI[8]={ 0x00, 0xA9, 0x45, 0x5F, 0x86, 0xED, 0x6B, 0xCD
45 void os_getDevEui (ul_t* buf) { memcpy_P(buf, DEVEUI, 8);}
46
47 // This has to be in big-endian format for pipes (see note below)

```

Figura 3.4 Interfaz de Programación. (Galarza, Lara)

Se destaca que existe gran variedad de librerías creadas por diferentes usuarios dentro de la comunidad Arduino y que son de código abierto que se pueden utilizar en cualquier proyecto con la finalidad de reutilizar códigos o más bien mejorarlos para optimizar el trabajo del programador y simplificarlo, de esta forma se reduce el número de líneas de código, ya que simplemente se necesitaría de importar una librería y definir variables.

3.3.2 ESPECIFICACIONES LORA

Aunque los detalles técnicos fueron expuestos en el capítulo anterior, cabe recalcar las especificaciones necesarias de LoRa que se muestran dentro de la implementación del proyecto.

Se necesita de tres dispositivos principales para poder formar una red LoRa, se detalla a continuación:

- **Servidores:** Recibe, procesa y gestiona toda la información enviada por el Gateway.

- **Gateway:** Envía toda la información a la red de Internet, también, recibe los múltiples datos de varios dispositivos finales. Se considera las estaciones base Lora.
- **Dispositivos Finales:** Son los objetos como sensores que se conectan a la red LoRa, transmitiendo la información hacia los Gateways.

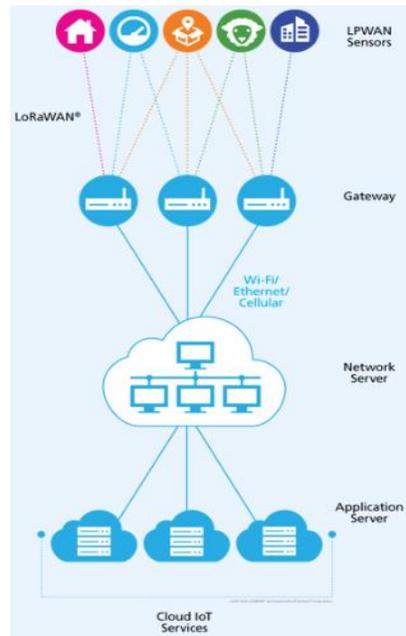


Figura 3.5 Red LoRa. (SEMTECH, 2020)

Como se puede ver en la Figura 3.5 cada Gateway recibe datos múltiples de diferentes dispositivos finales y esos datos son enviadas a los servidores, esto se representa como una configuración en estrella, cabe recalcar que una gran ventaja que posee esta red consiste en que varios Gateways pueden recibir datos del mismo dispositivo final, de forma en que se puede proveer un servicio de posicionamiento basado únicamente en LoRa.

3.3 DETECCION Y MONITORIZACION

Dentro del diseño, para la detección y monitorización del gas CO₂ se describe a continuación los siguientes componentes:

3.4.1 SENSOR DE DIOXIDO DE CARBONO SEN0219

El sensor SEN0219 fabricado por DFRobot, es un sensor de gas de dióxido de carbono de alta precisión con su rango de detección de 0 ppm a 5000 ppm, está

basado en la tecnología infrarrojo no dispersivo (NDIR) y tiene buena selectividad y dependencia libre de oxígeno.

Integra mecanismo de compensación de temperatura que permiten obtener valores confiables de medición, además, este producto es un sensor de alto rendimiento que combina la tecnología de detección de gases de absorción infrarroja con diseño de circuito óptico de precisión, así como diseño de circuito sofisticado. (DRFOBOT, s.f.)

Sus principales especificaciones técnicas se muestran a continuación

- Voltaje de funcionamiento: 4.5 – 5.5 VDC
- Señal de salida analógica: 0.4 ~ 2 V
- Rango de medición: 0 ~ 5000 ppm
- Precisión: ± 50 ppm
- Tiempo de respuesta: 120 segundos.
- Vida de servicio: > 5 años

Principales características:

- Impermeable
- Alta sensibilidad
- Bajo consumo de energía
- Compensación de temperatura



Figura 3.6 Distribución de pines. (DRFOBOT, s.f.)

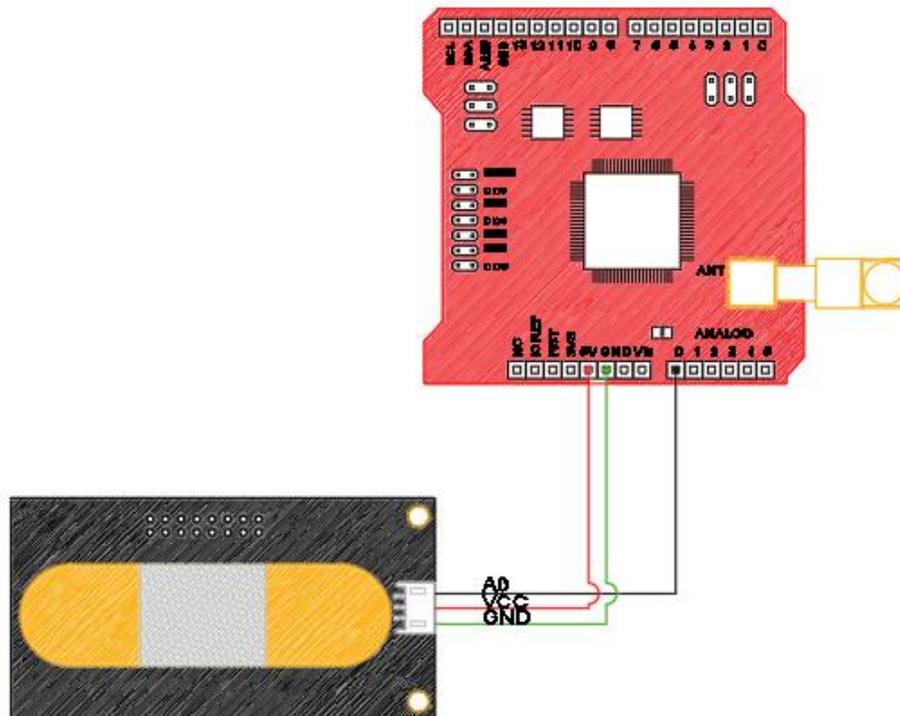


Figura 3.7 Esquema de conexión Lora Shield y SEN0219. (Galarza, Lara)

3.4.2 ARDUINO MEGA 2560

El Arduino Mega 2560 es una placa de microcontrolador actualizada de la versión Arduino Mega, esta posee 54 pines de entradas y salidas digitales, 16 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio. (Arduino, 2020)

Sus principales especificaciones técnicas se muestran a continuación: (Arduino, 2020)

- Voltaje de funcionamiento: 5VDC
- Voltaje de entrada recomendado: 7 – 12 VDC
- Voltaje de entrada límite: 6 – 20 VDC
- Memoria Flash: 256 KB, 8KB para gestor de arranque



Figura 3.8 Arduino Mega 2560. (Arduino, 2020)

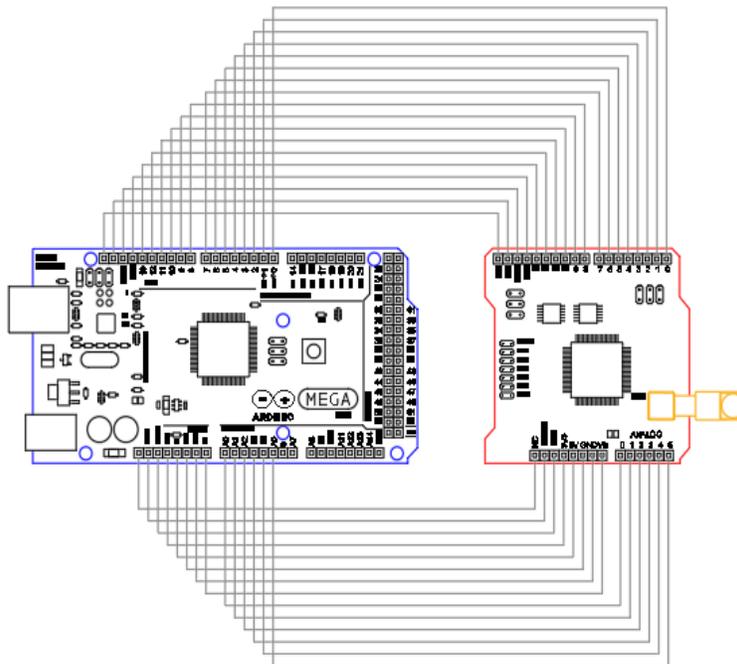


Figura 3.9 Esquema de conexión Arduino Mega 2560 y Lora Shield (Galarza, Lara)

3.4 COMUNICACION LORA

A continuación, se detallan los componentes a usar para que exista una correcta comunicación LoRa.

3.5.1 GATEWAY DRAGINO LG02

LG02 es una puerta de enlace LoRa de doble canal de código abierto. Permite conectar la red inalámbrica LoRa a una red IP a través de WiFi, Ethernet o celular 3G / 4G a través del módulo LTE opcional. (Dragino, 2019)

La conexión inalámbrica de este dispositivo permite a los usuarios enviar datos en rangos de alcance extremadamente largos proporcionando también una comunicación con inmunidad a alta interferencia admitiendo así una solución inalámbrica de bajo costo para admitir más de 50 nodos de sensores.

El Gateway en modo LoRaWAN puede admitir diferentes modos de funcionamiento como: modo cliente o servidor TCP/IP, modo MQTT, etc., para adaptarse a sinnúmero de funcionalidades aplicables para una conexión IoT.

Sus principales especificaciones técnicas se muestran a continuación (Dragino, 2019):

- Procesador: 400MHz, 24K MIPS
- Flash: 16 MB; RAM: 64 MB
- Puertos RJ45 10M / 100M x 2
- WiFi: 802.11 b / g / n
- LoRa Wireless
- Entrada de energía: 12V DC
- Conector host USB 2.0 x 1

Dentro de sus principales características se obtiene (Dragino, 2019):

- Bajo consumo de energía
- Actualización de firmware vía web
- Software actualizable a través de la red
- Servidor web incorporado
- Conexión a Internet a través de LAN, WiFi, o 3G / 4G a través del módulo LTE opcional
- Dos canales de recepción y un canal de transmisión.
- Banda LoRa disponible a 433/868/915/920 MHz
- Alcance máximo en LoRa: 5 ~ 10 km. Área de densidad:> 500 m



Figura 3.10 Gateway Dragino LG02. (Dragino, 2019)

3.5.2 CONFIGURACIÓN DE GATEWAY DRAGINO LG02

Para su configuración se accede mediante un navegador web a través de la dirección ip 10.130.1.1.

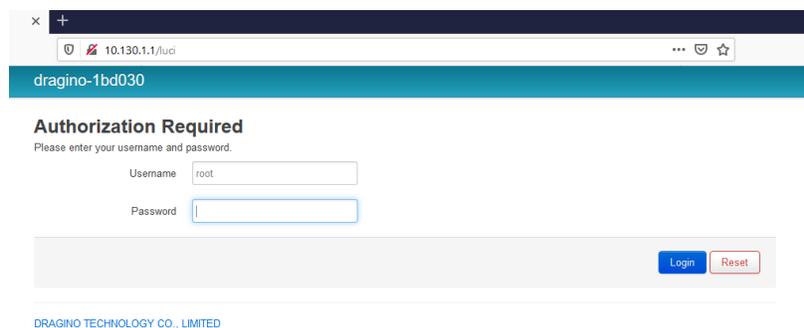


Figura 3.11 Ingreso a configuración Gateway. (Galarza, Lara)

Una vez que se haya accedido, la página web solicita una contraseña, esta viene por defecto de fábrica: “Dragino”, a partir de esto se tiene acceso a todos los parámetros a configurar del equipo.

Como segundo paso, se configura el método de conexión a internet que utilizará el gateway para su transmisión de datos mediante el menú: Network -> Wireless

dragino-1bd030 Status System Network Service Logout AUTO REFRESH ON

radio0: Master "dragino-1bd030" radio1: Client "LaraPonce"

Wireless Overview

radio0	Generic MAC80211 802.11bgn Channel: 11 (2.462 GHz) Bitrate: 19.5 Mbit/s	Restart Scan Add
0%	SSID: dragino-1bd030 Mode: Master Wireless is disabled	Enable Edit Remove
60%	SSID: LaraPonce Mode: Client BSSID: 08:BD:43:C3:7F:F9 Encryption: WPA2 PSK (CCMP)	Disable Edit Remove

Associated Stations

Network	MAC-Address	Host	Signal / Noise	RX Rate / TX Rate
Client: "LaraPonce" (wlan0)	08:BD:43:C3:7F:F9	192.168.1.1	-61 / -90 dBm	58.5 Mbit/s, 20MHz, MCS 6 39.0 Mbit/s, 20MHz, MCS 4

DRAGINO TECHNOLOGY CO., LIMITED

Figura 3.12 Configuración Wireless. (Galarza, Lara)

Para la implementación, se configura el gateway en modo “Wifi Client Mode”, Dragino actúa como cliente Wifi y obtiene mediante DHCP del router el enlace para conectarse a la red internet, esta también permite compartir dicha red con su puerto LAN para otros dispositivos.

En este caso se configuró la red wifi visible denominada “LaraPonce” como se muestra en la Figura 3.12.

En la siguiente configuración se ingresa los datos que será de utilidad para el registro de la aplicación en el servidor web. Network -> Gateway

dragino-1bd030 Status System Network Service Logout

LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

LoRaWAN Server Settings

IoT Service	LoRaWan/RAW forwarder
Debug Level	No debug
Service Provider	The Things Network
Server Address	ttn-router-us-west
Server Port	1700
Gateway ID	a840411bd0304150
Mail Address	dragino-1bd030@dragino.com
Latitude	22.73
Longitude	114.23
RadioMode	A for RX, B for TX

Figura 3.13 Configuración de servidor. (Galarza, Lara)

Tomar en cuenta los principales parámetros a configurar:

- Services Provider: seleccionando The Things Network, este será el servidor web encargada de recopilar todos los datos enviados y,
- Gateway ID: Identificación única del dispositivo, se encuentra en la etiqueta colocada en la parte de atrás del equipo.



Figura 3.14 Etiqueta de equipo. (Galarza, Lara)

Después, se configura los parámetros de transmisión del equipo dentro de la misma ruta.

Channel 1 Radio Settings

Radio settings for Channel 1

RadioA Frequency (Unit:Hz)	<input type="text" value="915000000"/>
RadioA Spreading Factor	<input type="text" value="SF7"/>
RadioA Coding Rate	<input type="text" value="4/5"/>
RadioA Signal Bandwidth	<input type="text" value="125 kHz"/>
RadioA Preamble Length	<input type="text" value="8"/> <small>Length range: 6 ~ 65536</small>
RadioA LoRa Sync Word	<input type="text" value="52"/> <small>Value 52(0x34) for LoRaWAN</small>
Encryption Key	<input type="text" value="Encryption Key"/>

Figura 3.15 Configuración de transmisión. (Galarza, Lara)

LoRaWAN utiliza una configuración diferente de frecuencias, factores de propagación y anchos de banda dependiendo de donde se encuentre localizado el equipo, para este caso, 915 kHz de frecuencia y como factor de propagación SF7 con ancho de banda 125 kHz, todo esto permitido dentro de América.

3.5.3 DRAGINO LORA SHIELD

Dragino Lora Shield es un Arduino Shield con Tecnología LoRa gracias a su chip incorporado Semtech SX1276 que es lo que proporciona la transmisión. Esta tarjeta permite enviar baja tasas de datos alcanzando rangos extremadamente largos, además, posee una alta inmunidad a la interferencia y comunicación de espectro extendido de rango ultra bajo con un consumo mínimo de corriente.

Es una tarjeta que posee todo lo necesario para que sea simple y eficaz su conexión con la placa Arduino, siendo sus principales características (Dragino, Dragino Lora Shield, 2020):

- Banda de frecuencia pre-configurado en fábrica: 915MHz / 868 MHz / 433 MHz
- Antena externa
- Fuente de alimentación compatible con las placas de Arduino

También, sus especificaciones técnicas (Dragino, Dragino Lora Shield, 2020):

- Alta sensibilidad: hasta -148 dBm
- Sintetizador totalmente integrado con una resolución de 61 Hz
- Modulación FSK, GFSK, MSK, GMSK, LoRaTM y OOK
- 127 dB de rango dinámico RSSI
- Sensor de temperatura incorporado e indicador de batería baja
- Velocidad de bits programable de hasta 300 kbps
- Sincronizador de bits incorporado para recuperación de reloj
- Fuente de alimentación compatible con Arduino: 3.3 VDC o 5 VDC

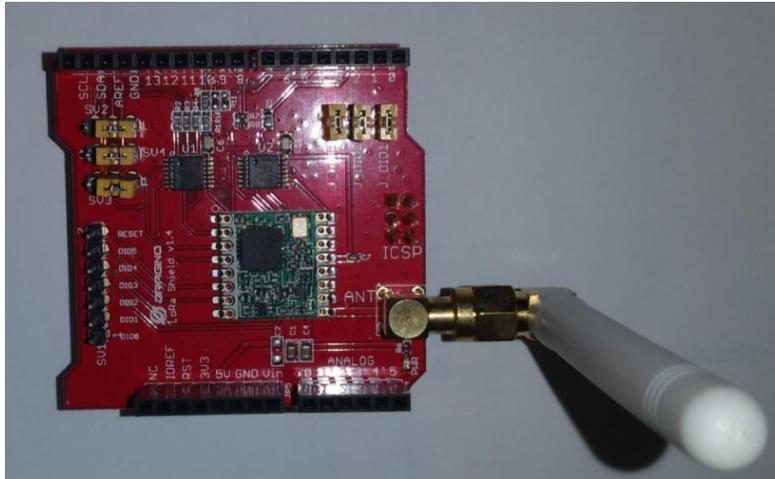


Figura 3.16 Lora Shield. (Galarza, Lara)

3.5 THE THINGS NETWORK

THE THINGS NETWORK permite que los dispositivos de baja potencia utilicen gateways de largo alcance para conectarse a una red descentralizada de código abierto para intercambiar datos con aplicaciones (THE THINGS NETWORK, 2020).

TTN empezó como una iniciativa pensada en la comunidad para formar una red global IoT dentro del contexto LoRaWAN. La iniciativa fue lanzada por Wienke Giezeman en 2015 y actualmente cuenta con 12675 gateways en 150 países a nivel mundial.



Figura 3.17 Ubicación de Gateways TTN. (THE THINGS NETWORK, 2020)

TTN es un entorno adecuado para monitorear los datos enviados por el sensor que emplea LoRaWAN desde cualquier sitio con acceso a la red internet y poder tratar esos mismos datos como mejor convenga. TTN necesita de una arquitectura red para poder realizar la gestión de la información recibida desde el sensor, las partes fundamentales son:

- **Sensores:** nodos finales, se conectan con los gateways empleando la radio LoRa.
- **Gateways:** Reciben la información de los sensores mediante LoRa y envían a Internet mediante conexión Wifi.
- **Servidor Red:** En este caso es el servidor TTN quien recibe y almacena la información enviada por el gateway
- **Servidores de aplicaciones:** Permiten implementar aplicaciones para trabajar con los datos recibido.

El objetivo de TTN es facilitar las opciones de implementación al usuario.

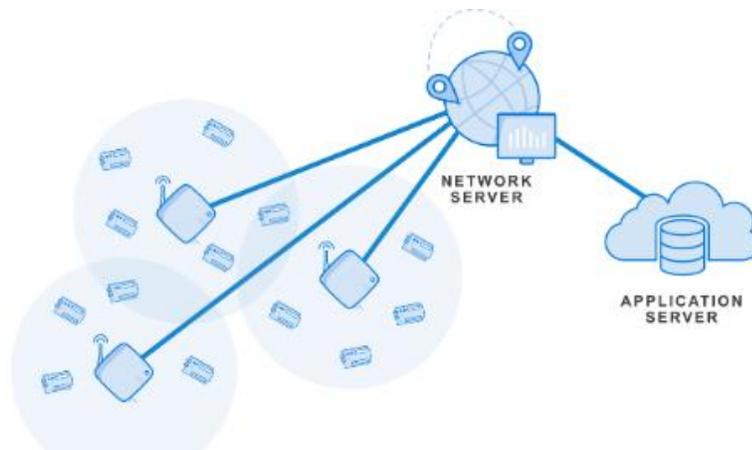


Figura 3.18 Arquitectura de Red. (Network Architecture, s.f.)

Para que el servidor TTN pueda recibir los datos censados es necesario registrar el gateway y el nodo final creando una aplicación en la plataforma, para esto se requiere tener una cuenta en la misma.

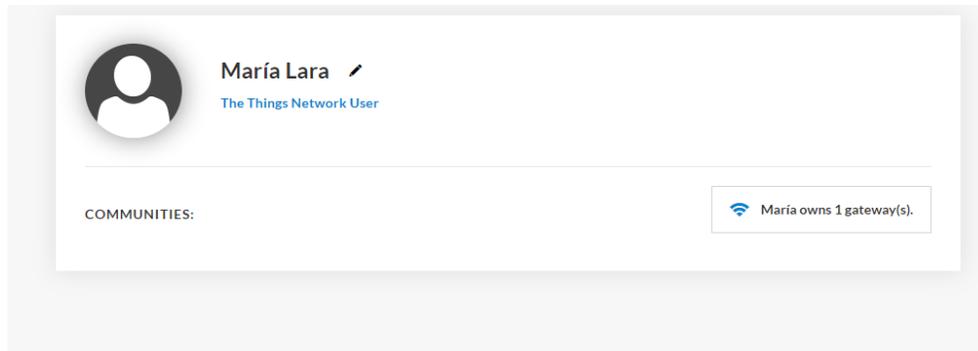


Figura 3.19 Cuenta en TTN. (Galarza, Lara)

Para crear la cuenta se utilizó el correo institucional de la Universidad Politécnica Salesiana: “mlarap@est.ups.edu.ec”

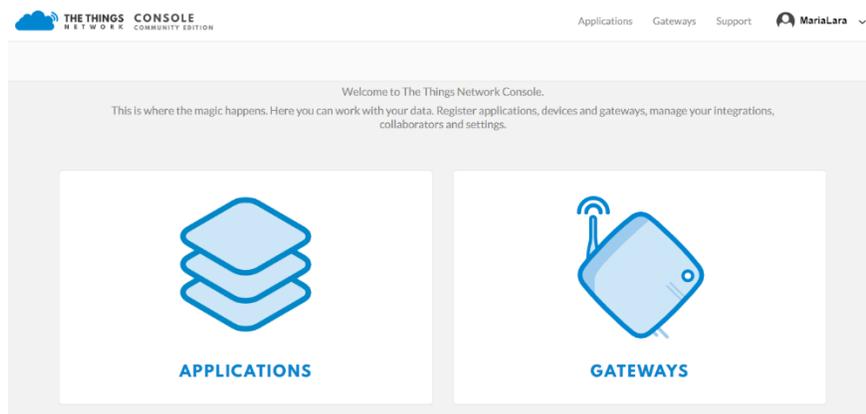


Figura 3.20 Plataforma TTN. (Galarza, Lara)

3.6.1 REGISTRO DE UN GATEWAY

Desde “Gateways”, se encuentra la opción “Register Gateway”, se rellenan los campos solicitados, en esto incluye:

- **Gateway ID:** Cualquier nombre para identificar al dispositivo.
- **Description:** Pequeña descripción del dispositivo, queda a libre criterio del usuario.
- **Frecuency Plan:** 915MHz.
- **Router:** Puerta de enlace, se selecciona un enrutador que se encuentre cerca de la región para reducir la latencia.
- **Antenna Placement:** Colocación de la antena, si esta va a ser interior o exterior

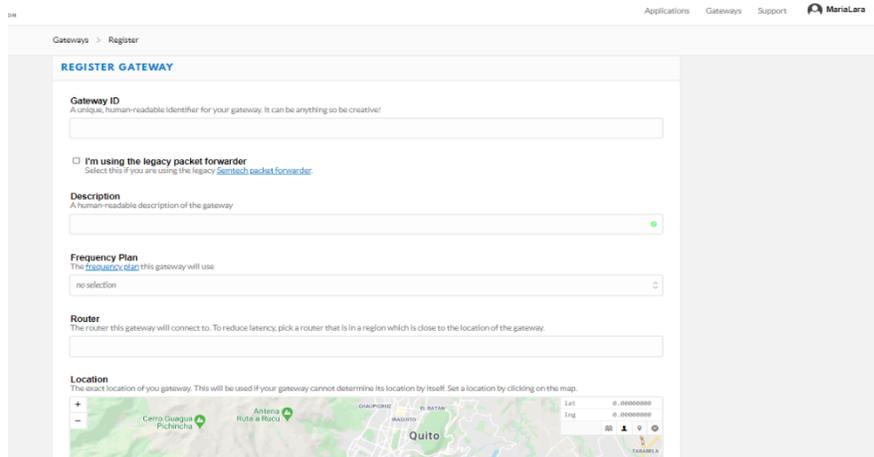


Figura 3.21 Registro de Gateway. (Galarza, Lara)

Finalizado del proceso de registro se pueden observar las características y empezar a utilizarlo. Es posible volver a editar los parámetros antes mencionados para que estos puedan ser visibles o no.

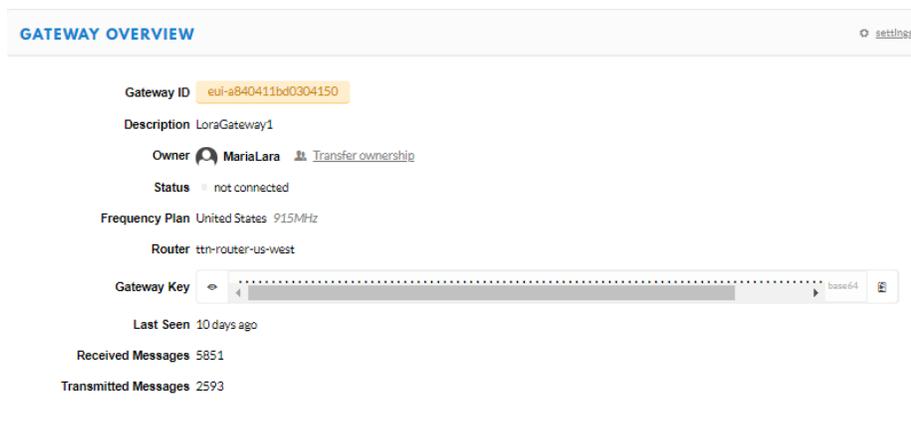


Figura 3.22 Gateway registrado. (Galarza, Lara)

3.6.2 REGISTRO DE UN NODO FINAL

Desde "Applications", se encuentra la opción "add application", se rellenan los campos solicitados, en esto incluye:

- **Application ID:** Nombre único de la aplicación.
- **Description:** Pequeña descripción de la aplicación.
- **Handler registration:** Seleccionar el enrutador al que se desea registrar la aplicación, se selecciona el mismo enrutador elegido en el registro del gateway.

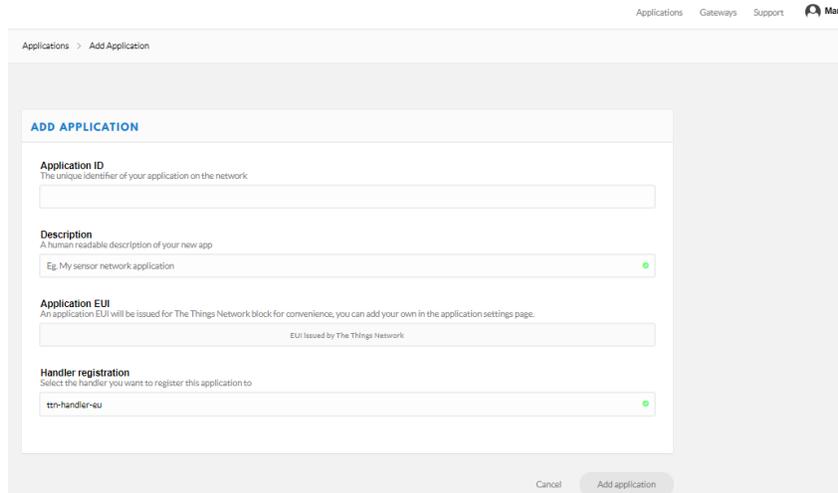


Figura 3.23 Registro de la aplicación. (Galarza, Lara)

Se registra el dispositivo una vez añadida la aplicación, de modo que se elige la opción “register device”

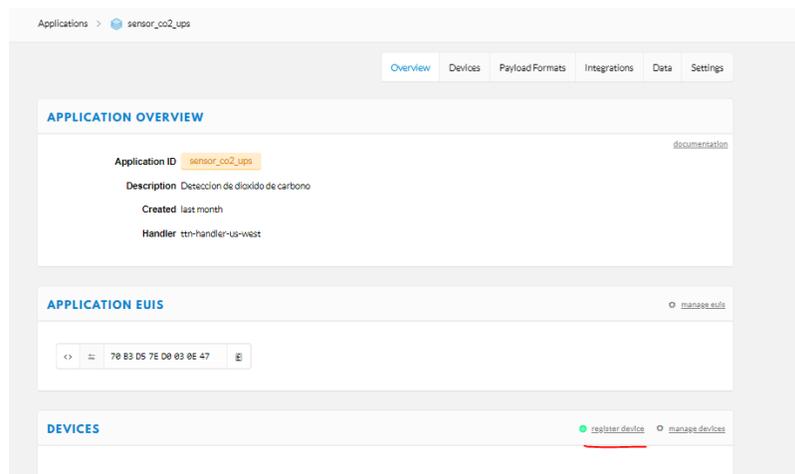


Figura 3.24 Registro de dispositivo (Galarza, Lara)

Dentro de la selección, se llenan los siguientes parámetros:

- Device ID: Nombre alfanumérico que identifica el dispositivo de manera única e inequívocamente.

El resto de los parámetros se generan de manera automática.

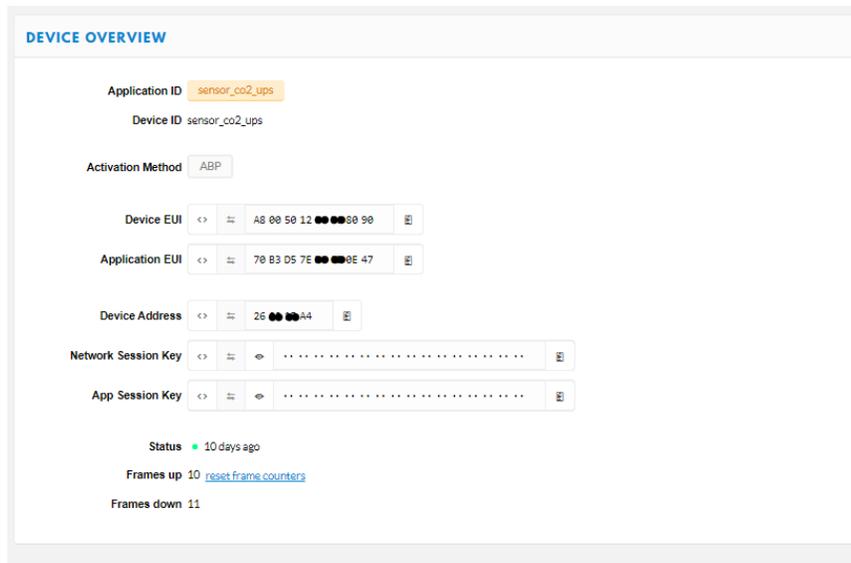


Figura 3.25 Dispositivo registrado. (Galarza, Lara)

Para que el dispositivo pueda conectarse y enviar datos es necesario que se incluyan los siguientes parámetros: Network Session Key, App Session Key y Device Address en el sketch de Arduino de forma que el dispositivo pueda identificarse ante la red.

3.6 MY DEVICES CAYENNE

My Devices Cayenne es una plataforma que brinda soluciones de IoT permitiendo a los usuarios conectar dispositivos de manera eficiente, visualizar datos y aplicar reglas sofisticadas a los datos mostrados en el mismo.

Cayenne sirve como herramienta en este proyecto para visualizar datos históricos y en tiempo real que fueron enviados a través de The Things Network.

Sus principales características se muestran a continuación:

- Aplicación móvil para configurar, monitorear y controlar dispositivos.
- Programación de luces, motores, sensores y actuadores.
- Acceso remoto instantáneo desde la computadora o un teléfono.
- Control GPIO configurable
- Paneles de control personalizables.

3.7.1 REGISTRO DE SENSOR EN CAYENNE

Para realizar el registro del dispositivo en Cayenne, será necesario como primer punto crear una cuenta en la plataforma para poder conectar, en este caso, la tarjeta Lora Shield registrada previamente en The Things Network.

Para crear la cuenta se utilizó el correo institucional de la Universidad Politécnica Salesiana: “mlarap@est.ups.edu.ec”

Dentro de “Dashboard” en la plataforma, se selecciona la opción “New Devices/Widgets” y se procede a llenar los campos con la información requerida.

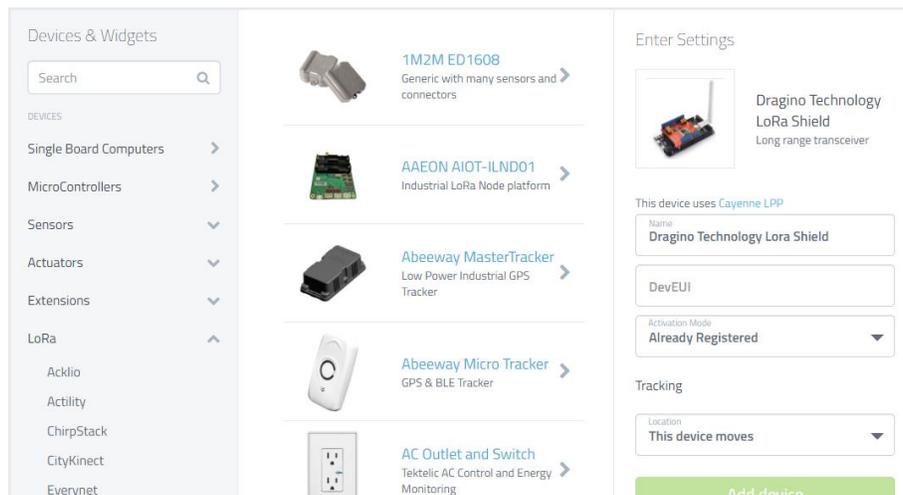


Figura 3.26 Selección de dispositivo en Cayenne. (Galarza, Lara)

En el campo DevEUI se incluye el registro realizado en TTN en el apartado anterior.

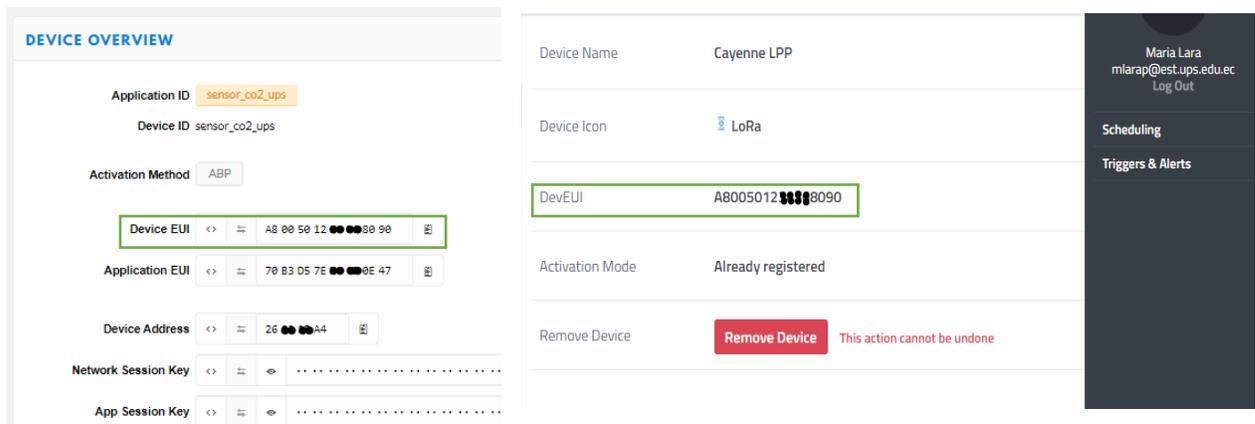
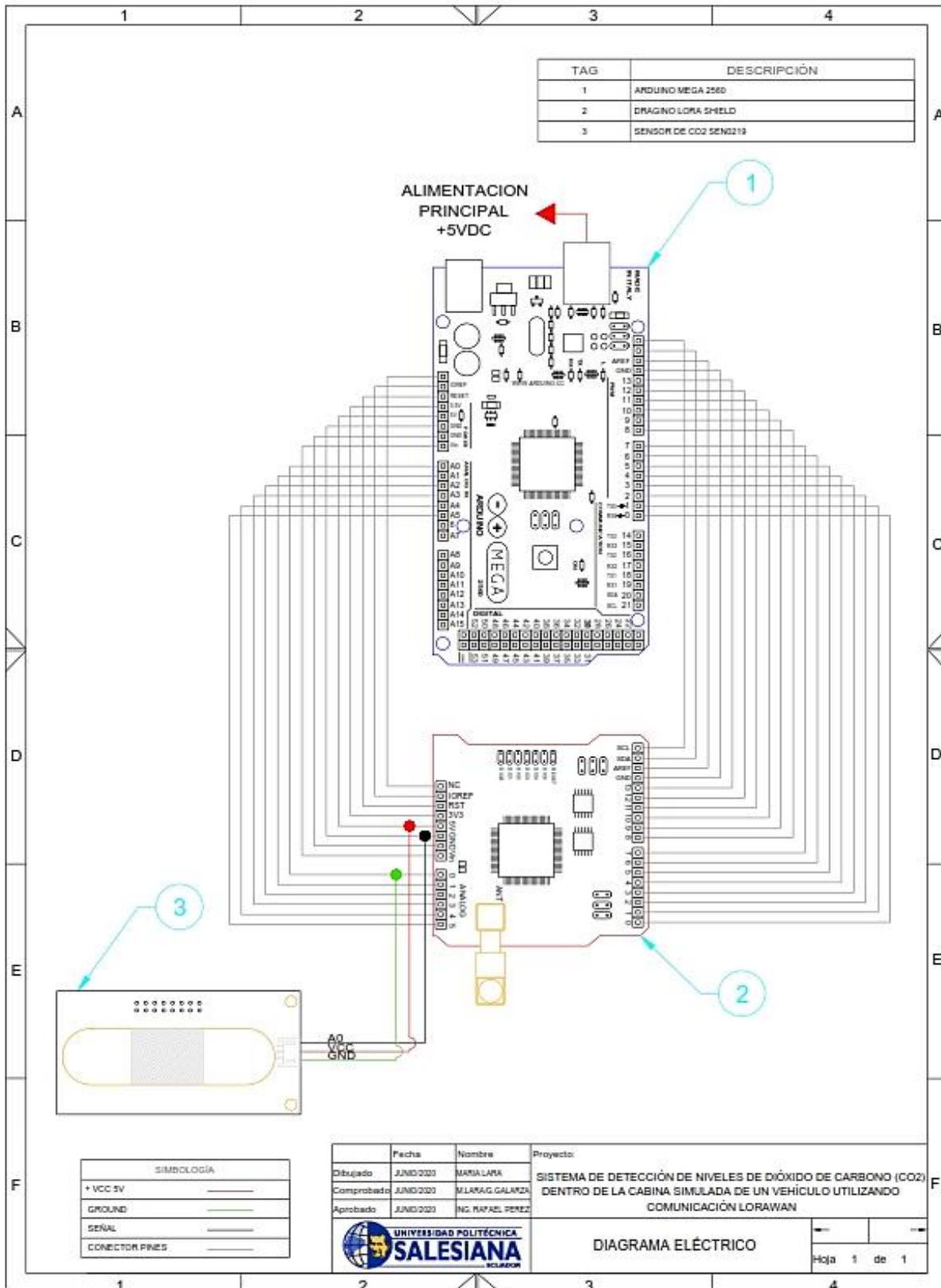


Figura 3.27 Registro de dispositivo en Cayenne. (Galarza, Lara)

3.7 DIAGRAMA ELECTRICO



PRUEBAS Y ANALISIS DE RESULTADOS

4.1 PRUEBAS DE FUNCIONAMIENTO

Para evidenciar el buen funcionamiento del equipo, se ha dividido las pruebas en tres etapas:

- Obtención de datos
- Visualización
- Envío de notificaciones

4.1.1 ETAPA DE OBTENCION DE DATOS

En esta etapa se observa la obtención de los datos emitidos por el sensor de CO₂. Para visualizar y comprobar de que las mediciones del sensor sean las adecuadas, se verifica a través de la plataforma Arduino mediante el Monitor Serial.

```
voltage: 522mv
Concentracion: 381ppm
9493478: engineUpdate, opmode=0x908
9493864: TXMODE, freq=915000000, len=17, SF=7, BW=125, CR=4/5, IH=0
Packet queued
9555378: RXMODE_SINGLE, freq=915000000, SF=7, BW=500, CR=4/5, IH=0, rxsyms=255
9588237: RXMODE_SINGLE, freq=915000000, SF=9, BW=125, CR=4/5, IH=0, rxsyms=255
9652643: EV_INCOMPLETE (includes waiting for RX windows)
9652690: engineUpdate, opmode=0x900
voltage: 522mv
Concentracion: 381ppm
10283983: engineUpdate, opmode=0x908
10284371: TXMODE, freq=915000000, len=17, SF=7, BW=125, CR=4/5, IH=0
Packet queued
10345853: RXMODE_SINGLE, freq=915000000, SF=7, BW=500, CR=4/5, IH=0, rxsyms=255
10378650: RXMODE_SINGLE, freq=915000000, SF=9, BW=125, CR=4/5, IH=0, rxsyms=255
10442992: EV_INCOMPLETE (includes waiting for RX windows)
10443041: engineUpdate, opmode=0x900
voltage: 522mv
Concentracion: 381ppm
```

Figura 4.1 Obtención de datos del sensor de CO₂ (Galarza, Lara)

En la Figura 4.1 se puede observar en el monitor serial el despliegue de los datos del cual corresponden a: Voltaje recibido en milivoltios, concentración en partes por millón, la frecuencia por el cual es emitida y el ancho de banda.

Por tanto, se ha comprobado que las mediciones de CO₂ llegan hasta la placa de Arduino.

4.1.2 ETAPA DE VISUALIZACIÓN

Se contempla en esta etapa la visualización de los valores medidos por el receptor, en este caso la visualización de los datos mostrados en la etapa anterior será a través de una página web y de un aplicativo que es de descarga libre para el celular.

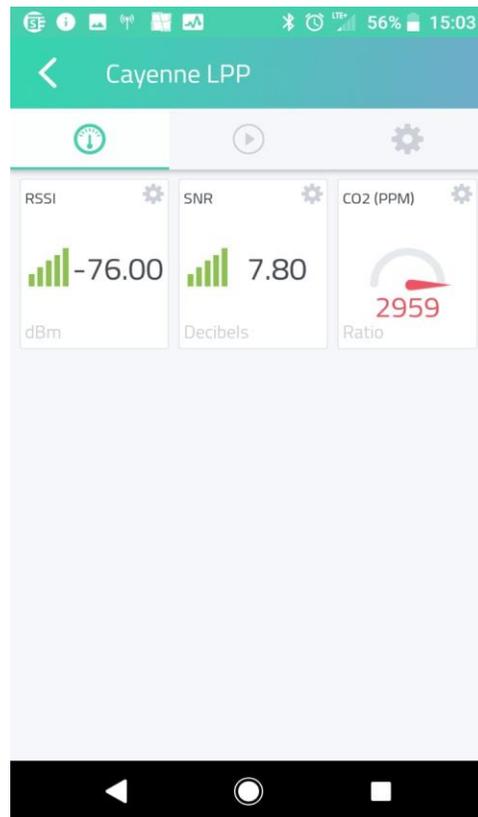


Figura 4.2 Visualización de datos en aplicativo móvil (Galarza, Lara)

Los datos que se muestran tanto en página web y aplicativo son las siguientes:

- **RSSI:** Es el indicador de intensidad de señal recibida (RSSI) es una medida estimada de lo bien que un dispositivo puede oír, detectar y recibir señales de cualquier punto de acceso o de un router específico (Lipfert, s.f.) ayuda a determinar y conocer si la señal es óptima en este caso el prototipo con el Gateway. Se mide en valores negativos, un valor que esté más cerca del cero representa una mejor señal.

Intensidad de la señal (dBm)	Calificación
-30 dBm	Asombroso
-67 dBm	Muy bueno
-70 dBm	De acuerdo
-80 dBm	No es bueno
-90 dBm	Inutilizable

Figura 4.3 Ejemplos de señal aceptable (Lipfert, s.f.)

- SNR: Es la relación señal-ruido (SNR) es la relación entre la señal de potencia recibida y el nivel de potencia de ruido.
 - Si el SNR es mayor que 0, la señal recibida opera por encima del ruido.
 - Si el SNR es menor que 0, la señal recibida opera por debajo del ruido.
- CO2 (PPM): Es la concentración del gas medido dentro de la cabina de un vehículo en partes por millón.

4.1.3 ETAPA DE ENVÍO DE NOTIFICACIONES

Se contempla en esta etapa el envío de notificaciones desde la app Cayenne hacia al receptor cuando este exista una alta concentración fuera de los rangos permitidos de CO2 vistos en el vehículo que se está monitoreando.

Si dentro del vehículo presenta altas concentraciones se genera una notificación para indicar que el nivel permitido de concentración ha sobrepasado el límite establecido, por lo que el equipo comienza a enviar correos electrónicos y mensajes de textos al usuario registrado.

Se realizó las respectivas pruebas para evidenciar si se generan las alertas cuando existan altas concentraciones de CO2, para esto se colocó el prototipo directo al tubo de escape del vehículo con el objetivo de demostrar mejor su funcionamiento.

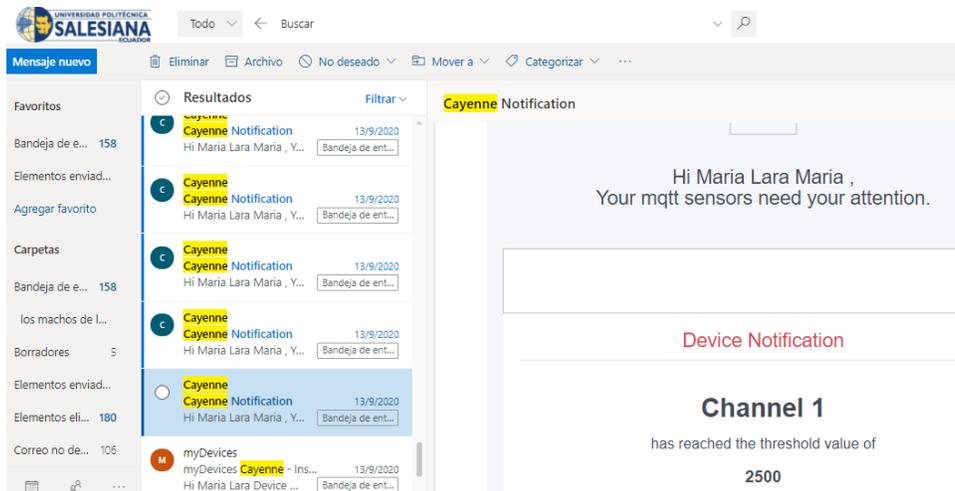


Figura 4.4 Registro de correos electrónicos (Galarza, Lara)

En la Figura 4.4 se comprueba la recepción del mismo al correo registrado, en este caso al correo institucional mlarap@est.ups.edu.ec.

4.2 PRUEBAS Y ANALISIS DE RESULTADOS

En esta sección se procede a realizar las pruebas en el vehículo, pero en diferentes escenarios, con el objetivo de comprobar y analizar su funcionamiento.

Las pruebas del sistema se realizaron con los siguientes escenarios:

- Vehículo en movimiento con ventanas abiertas
- Vehículo encendido y estacionado con ventanas cerradas
- Prototipo colocado cerca del tubo de escape

Estos escenarios fueron seleccionados al realizar el debido estudio de cuando se presentan altas concentraciones de CO₂ perjudicando a los usuarios dentro del vehículo.

Los valores medidos que se muestran a continuación fueron monitoreados tanto a través de la página de Cayenne como en la página de The Things Network para verificar que coincidan los datos.

4.2.1 VEHÍCULO EN MOVIMIENTO CON VENTANAS ABIERTAS

Como primera prueba, se ubicó el prototipo dentro del vehículo, para ser exactos, entre el conductor y el acompañante.

El vehículo es un Chery Van del año 2011 que posee como características principales el uso de gasolina como combustible, una distribución de cuatro válvulas por cilindro y una potencia máxima de 83 caballos de fuerza.



Figura 4.5 Prototipo dentro de vehículo (Galarza, Lara)

Se realizó las pruebas con el prototipo durante un lapso de una hora por una semana para este escenario, dando como resultado un nivel de concentración entre 451 ppm y 600 ppm, esto indica que aún se encuentra dentro de los valores permitidos y que no habría riesgo para la persona que se encuentra dentro del vehículo.

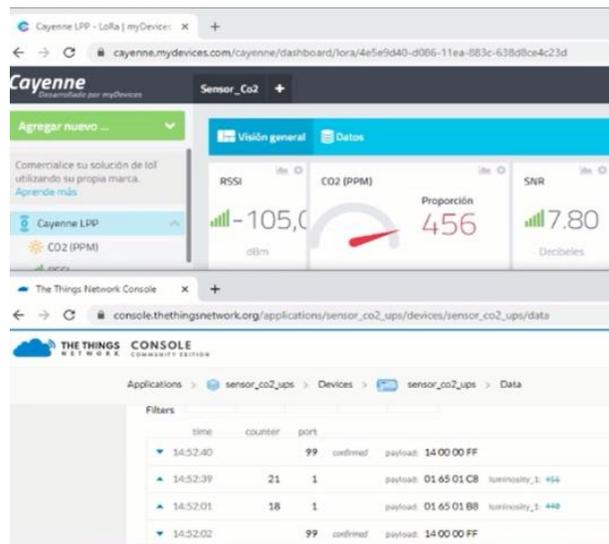


Figura 4.6 Valor obtenido con vehículo en movimiento (Galarza, Lara)

4.2.2 VEHÍCULO ENCENDIDO Y ESTACIONADO CON VENTANAS CERRADAS

Las pruebas se las realizó después de que el vehículo estuviera en movimiento en un lapso de 3 horas diarias durante una semana, en el horario de 2 pm hasta las 5 pm, donde también el clima de la ciudad de Guayaquil llega a los 30 C°, dando como resultado una concentración de 1.500 a 2.700 ppm siendo estos valores no permitidos a exposición de una persona por un período prolongado de tiempo ya que esto sería perjudicial para la salud.



Figura 4.6 Vehículo estacionado y con vidrios elevados (Galarza, Lara)

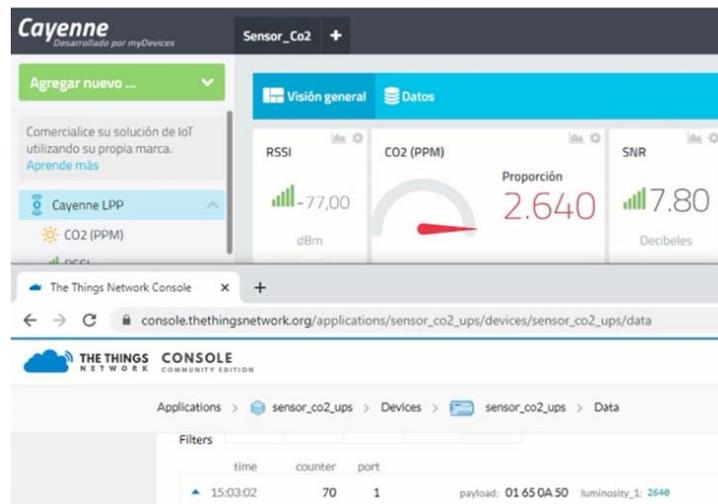


Figura 4.7 Valor obtenido con vehículo estacionado y ventanas cerradas (Galarza, Lara)

4.2.3 PROTOTIPO COLOCADO CERCA DEL TUBO DE ESCAPE

Esta prueba se la realizó para comprobar la funcionalidad total del prototipo en un período corto de media hora por una semana dando como resultados una concentración de CO2 alta demasiado considerable entre 3.000 hasta los 4.800 ppm. Se observa la gran contaminación expulsada por el vehículo que a la larga afecta a la salud de las personas como del medio ambiente.

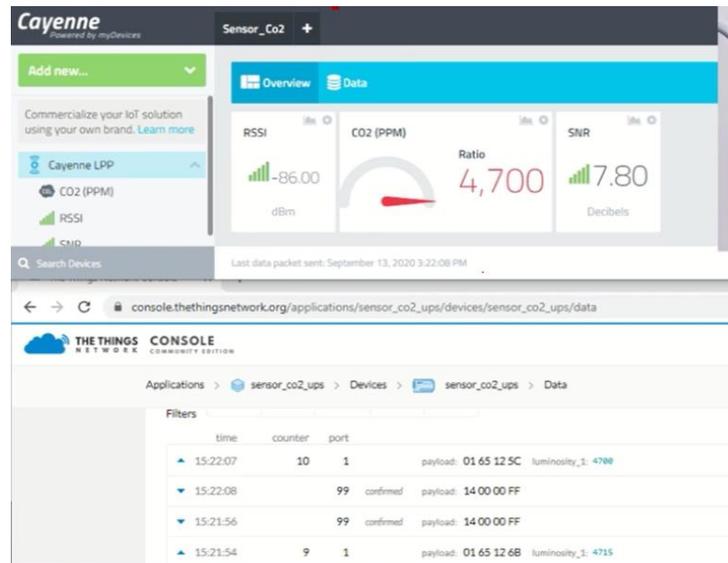


Figura 4.8 Valor obtenido con el prototipo cerca del tubo de escape (Galarza, Lara)

Después de las pruebas realizadas, se obtuvo el promedio por día de cada análisis para observar su comparativa en la Figura 4.8.

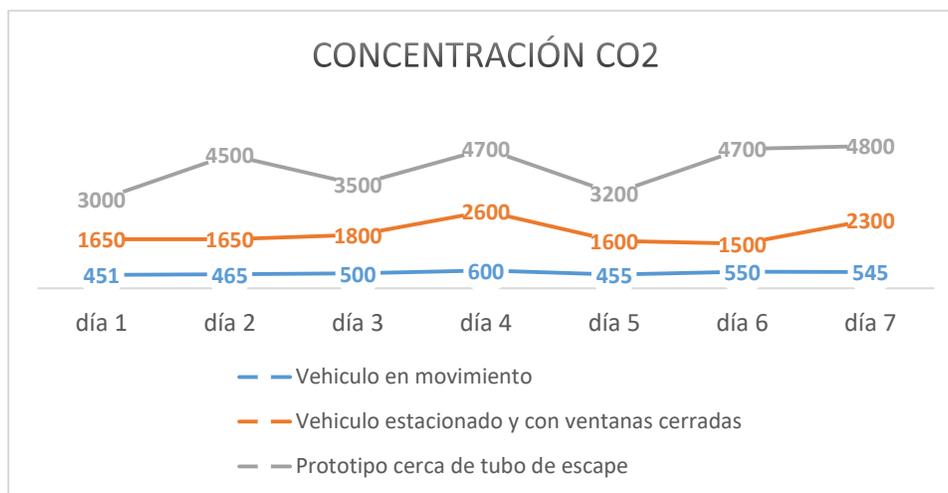


Figura 4.9 Niveles promedios de concentración de CO2 (Galarza, Lara)

Como se puede observar en la gráfica los niveles de concentraciones altas se emitieron en dos escenarios que se muestran en color “naranja” y “gris” estos causarían afectaciones a la salud si el conductor prolonga su permanencia dentro del vehículo como ya se había indicado en apartados anteriores.

CONCLUSIONES

- El desarrollo del equipo con tecnología LoRa/LoRaWAN representa una red de baja potencia con alto rango de cobertura que cumple con el objetivo de entregar valores reales al usuario mediante una nueva tecnología.
- La utilidad del prototipo para la obtención de los altos niveles de concentración del gas CO₂ a través del prototipo es de mucha importancia, ya que se puede prevenir accidentes o incluso la muerte de conductores o pasajeros con tiempos prolongados de viaje dentro de un mismo vehículo.
- La visualización de los datos mediante plataformas virtuales ayuda a la obtención de los datos en cualquier momento a través del internet o una red segura orientada al usuario
- Cayenne y The Thing Network tienen la ventaja de ser plataformas gratuitas con acceso público o privado a los datos y sus conexiones.
- La aplicación de Cayenne instalada en el celular permitió monitorear los datos de las concentraciones del gas CO₂ con actualizaciones constantes.
- Al tener acceso gratuito en Cayenne que es un ambiente virtual, se tiene como limitante de que cada dato sea transmitido y actualizado cada 15 segundos aproximadamente, dependiendo de qué tan fuerte sea el enlace, la única manera de que esto sea modificado es mediante código de programación en Arduino.
- La programación tuvo que ser desarrollada en base Arduino, ya que todos los equipos implementados y los ambientes virtuales son compatibles con la misma, existen varias librerías ya creadas que facilitan la programación y su configuración.
- En las pruebas realizadas tanto en espacios cerrados, abiertos o en línea de vista donde no se presenta ningún obstáculo que interrumpa el enlace se obtiene una recepción de datos mucho más rápido, del cual solo se demora en procesar la información en la web alrededor de los 15 segundos aproximadamente, en comparación cuando el enlace no es tan fuerte en potencia esto alcanzaría de los 30 a los 40 segundos aproximados.
- Se realizó pruebas en diferentes escenarios con el prototipo implantado para verificar su correcto funcionamiento y se comprobó mediante un valor umbral, 2000 ppm, si este sobrepasa los límites permitidos de inhalación según las organizaciones internacionales de salud.

RECOMENDACIONES

- Predefinir la utilización de los diferentes equipos LoRa, ya que estos poseen una gran variedad de tipos según el fabricante.
- Implementar soluciones a nivel de IoT para así tener una red a mayor escala.
- Se recomienda dejar que transcurra un tiempo aproximado de 5 a 10 minutos (valor recomendado por el fabricante) de precalentado antes de las mediciones respectivas.
- Para una mejora del prototipo, se recomienda el uso de baterías externas, y no del vehículo para que este sea un sistema autónomo.
- Manejar sensores que poseen radiación infrarroja permite obtener valores con más precisión en comparación a otros sensores como por ejemplo los químicos, además poseen un mayor tiempo de vida útil.

BIBLIOGRAFÍA Y REFERENCIAS

1. (s.f.). Obtenido de <https://www.redciteco.org/institucional/el-conocimiento/tecnologias-libres-y-abiertas>
2. Arduino. (2020). *Arduino*. Obtenido de <https://www.arduino.cc/en/main/products>
3. AxioMET. (2020). *Monitoreo de CO2*. Obtenido de <https://axiomet.eu/es/es/page/1954/Monitoreo-de-la-calidad-del-aire-en-los-espacios-cerrados/>
4. Benitez, D. (Noviembre de 2016). *Internet de las cosas*. Obtenido de https://www.researchgate.net/publication/320353907_Propuesta_de_arquitectura_para_Internet_de_las_Cosas
5. Berenguer, J., & Bernal, F. (2000). *Dioxido de carbono de la calidad del aire interior*. Obtenido de https://www.insst.es/documents/94886/327064/ntp_549.pdf/e9364a82-6f1b-4590-90e0-1d08b22e1074
6. Cheung, P. K. (Marzo de 2012). *Dioxido de carbono*. Obtenido de <https://www.dw.com/es/di%C3%B3xido-de-carbono-bendici%C3%B3n-y-maldici%C3%B3n/a-15119911>
7. CIC. (2019). *Aplicaciones de la IoT*. Obtenido de <https://www.cic.es/iot-sus-aplicaciones/>
8. de la Cámara, R. A. (12 de Junio de 2019). Diseño de un sistema de monitorización remota de un depósito de agua mediante LoRa.
9. Dragino. (31 de Octubre de 2019). Obtenido de <https://www.dragino.com/products/lora-lorawan-gateway/item/135-lg02.html>
10. Dragino. (1 de Agosto de 2020). *Dragino Lora Shield*. Obtenido de <https://www.dragino.com/products/lora/item/102-lora-shield.html>
11. DRFOBOT. (s.f.). Obtenido de https://wiki.dfrobot.com/Gravity__Analog_Infrared_CO2_Sensor_For_Arduino_SK_U__SEN0219
12. Ferrer, V. (2019). *Que es LoRa*. Obtenido de <https://vicentferrer.com/lora-lorawan/>
13. Geoambiental, S. (s.f.). *Dioxido de Carbono*. Obtenido de <https://www.saludgeoambiental.org/dioxido-carbono-co2>
14. Gonzalez, A., & Rocío, G. (15 de Octubre de 2018). *Plataforma Arduino*. Obtenido de http://revistas.educa.jcyl.es/revista_digital/index.php?option=com_content&view=article&id=3879&catid=84&Itemid=87
15. Inspiration. (2020). *Contaminacion ambiental*. Obtenido de <https://www.inspiration.org/cambio-climatico/contaminacion>
16. Lipfert, D. F. (s.f.). *SPEEDCHECK*. Obtenido de <https://www.speedcheck.org/es/wiki/rssi/>

17. Lora Alliance. (2020). *Que es LoRaWAN*. Obtenido de <https://loralliance.org/about-lorawan>
18. LoRaWAN. (2020). *Que es LoRa y LoRaWAN*. Obtenido de <https://lorawan.es/>
19. *Network Architecture*. (s.f.). Obtenido de <https://www.thethingsnetwork.org/docs/network/architecture.html>
20. OMS. (2020). *Contaminacion de aire*. Obtenido de https://www.who.int/topics/air_pollution/es/
21. Sabas. (Octubre de 2017). *Tipos y Clases de Nodos*. Obtenido de <https://medium.com/beelan/haciendo-iot-con-lora-capitulo-2-tipos-y-clases-de-nodos-3856aba0e5be>
22. SAP. (2019). *Internet de las cosas*. Obtenido de <https://www.sap.com/latinamerica/trends/internet-of-things.html>
23. SEMTECH. (2020). *SEMTECH*. Obtenido de <https://www.semtech.com/lora/what-is-lora>
24. *THE THINGS NETWORK*. (2020). Obtenido de <https://www.thethingsnetwork.org/docs/>

ANEXOS

ANEXO A. LISTADO DE MATERIALES Y PRESUPUESTO

ITEM	DESCRIPCION	CANT	VALOR TOTAL
1	ARDUINO UNO	1	\$ 30,00
1	LORA SHIELD RFM95W	1	\$ 57,00
1	SENSOR DE CO2 SEN0219	1	\$ 100,00
1	GATEWAY LORAWAN	1	\$ 150,00
1	CONECTORES Y CABLES	1	\$ 30,00
1	MATERIALES VARIOS	1	\$ 50,00
TOTAL			\$ 417,00

ANEXO B. PROGRAMACIÓN IMPLEMENTADA EN LA TARJETA ARDUINO

```
#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>
#define PIN_A A0

int sensorIn = A0; //Pin analógico para el sensor de CO2

unsigned int count = 1;

String datastring1="";
String datastring2="";
String datastring3="";

/*Definición de parámetros para el registro del dispositivo a la nube TTN*/

static uint8_t mydata[5] = {0x01,0x7D,0x00,0x00,0x00};

static const PROGMEM ul_t NWSKEY[16] = { 0x34, 0xA0, 0x83, 0x04, 0x12, 0xB0, 0x41, 0x2D, 0x33, 0x08, 0xAC, 0x89, 0x9C, 0xAE, 0x5F, 0xB3 } ;

/*La clave APPSKEY debe estar en formato big edian, esto lo genera automaticamente
TTN para poder registrar el equipo, es una clave predeterminada de semtech.*/

static const ul_t PROGMEM APPSKEY[16] = { 0xFA, 0xAF, 0x38, 0xDE, 0xAF, 0x26, 0xA6, 0xCF, 0x19, 0x36, 0x2D, 0x96, 0xCC, 0x10, 0xBF, 0x94 } ;

/*
LoRaWAN end-device address (DevAddr)
See http://thethingsnetwork.org/wiki/AddressSpace

ttn*/
static const u4_t DEVADDR = 0x260215A4;

/* These callbacks are only used in over-the-air activation, so they are
left empty here (we cannot leave them out completely unless
DISABLE_JOIN is set in config.h, otherwise the linker will complain).*/
void os_getArtEui (ul_t* buf) { }
void os_getDevEui (ul_t* buf) { }
void os_getDevKey (ul_t* buf) { }

static osjob_t initjob,sendjob,blinkjob;

/*Para programar el envio de datos (TX) cada ciertos segundos.*/

const unsigned TX_INTERVAL = 10;

// Pin mapping
/*Declaración de los pines de las tarjeta LoRa, sin esta configuración no se podría
enviar datos a la nube TTN*/
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {2, 6, 7},
};
```

```

void do_send(osjob_t* j){
    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println("OP_TXRXPEND, not sending");
    } else {

        sensor(); // llamada
        // Prepare upstream data transmission at the next possible time.
        // LMIC_setTxData2(1,datasend,sizeof(datasend)-1,0);
        LMIC_setTxData2(1, mydata, sizeof(mydata), 0);
        Serial.println("Packet queued");
        Serial.print("LMIC.freq:");
        Serial.println(LMIC.freq);
        Serial.println("Receive data:");

    }
    // Next TX is scheduled after TX_COMPLETE event.
}

void onEvent (ev_t ev) {
    Serial.print(os_getTime());
    Serial.print(": ");
    Serial.println(ev);
    switch(ev) {
        case EV_SCAN_TIMEOUT:
            Serial.println(F("EV_SCAN_TIMEOUT"));

            break;
        case EV_BEACON_FOUND:
            Serial.println(F("EV_BEACON_FOUND"));
            break;
        case EV_BEACON_MISSED:
            Serial.println(F("EV_BEACON_MISSED"));
            break;
        case EV_BEACON_TRACKED:
            Serial.println(F("EV_BEACON_TRACKED"));
            break;
        case EV_JOINING:
            Serial.println(F("EV_JOINING"));
            break;
        case EV_JOINED:
            Serial.println(F("EV_JOINED"));
            break;
        case EV_RFU1:
            Serial.println(F("EV_RFU1"));
            break;
        case EV_JOIN_FAILED:
            Serial.println(F("EV_JOIN_FAILED"));
            break;
        case EV_REJOIN_FAILED:
            Serial.println(F("EV_REJOIN_FAILED"));
            break;
        case EV_TXCOMPLETE:
            Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));

```

```

        if(LMIC.dataLen) {
            // data received in rx slot after tx
            Serial.print(F("Data Received: "));
            Serial.write(LMIC.frame+LMIC.dataBeg, LMIC.dataLen);
            Serial.println();
        }
        // Schedule next transmission
        os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
        break;
    case EV_LOST_TSYNC:
        Serial.println(F("EV_LOST_TSYNC"));
        break;
    case EV_RESET:
        Serial.println(F("EV_RESET"));
        break;
    case EV_RXCOMPLETE:
        // data received in ping slot
        Serial.println(F("EV_RXCOMPLETE"));
        break;
    case EV_LINK_DEAD:
        Serial.println(F("EV_LINK_DEAD"));
        break;
    case EV_LINK_ALIVE:
        Serial.println(F("EV_LINK_ALIVE"));
        break;
    default:
        Serial.println(F("Unknown event"));
    }

    break;
}
}

void setup() {
    // initialize digital pin  as an output.

    Serial.begin(9600); //Configuración de la velocidad serial del arduino
    while(!Serial);
    Serial.println("Connect to TTN:");

    #ifdef VCC_ENABLE

    pinMode(VCC_ENABLE, OUTPUT);
    digitalWrite(VCC_ENABLE, HIGH);
    delay(1000);
    #endif

    // LMIC init
    os_init();
    // Reset the MAC state. Session and pending data transfers will be discarded.
    LMIC_reset();
    /*LMIC_setClockError(MAX_CLOCK_ERROR * 1/100);
    Set static session parameters. Instead of dynamically establishing a session
    by joining the network, precomputed session parameters are be provided.*/
    #ifdef PROGMEM
    /* On AVR, these values are stored in flash and only copied to RAM
    at boot time. On other architectures, they are stored in RAM at boot time.
    */

```

```

    once. Copy them to a temporary buffer here, LMIC_setSession will
    copy them into a buffer of its own again.*/
uint8_t appskey[sizeof(APPSKEY)];
uint8_t nwkskey[sizeof(NWKSKEY)];
memcpy_P(appskey, APPSKEY, sizeof(APPSKEY));
memcpy_P(nwkskey, NWKSKEY, sizeof(NWKSKEY));
LMIC_setSession (0x1, DEVADDR, nwkskey, appskey);
#else
// If not running an AVR with PROGMEM, just use the arrays directly
LMIC_setSession (0x1, DEVADDR, NWKSKEY, APPSKEY);
#endif

// Disable link check validation
LMIC_setLinkCheckMode (0);

// TTN uses SF9 for its RX2 window.
LMIC.dn2Dr = DR_SF9;

/*Ajuste de la velocidad de datos y la potencia de transmisión*/

LMIC_setDrTxpow(DR_SF7,14);

// Start job
do_send(&sendjob);
}

void sensor(){

```

```

int sensorValue = analogRead(sensorIn);

/*Arreglo para envio de valores en ppm hacia la nube*/
// The analog signal is converted to a voltage
float voltage = sensorValue*(5000/1024.0);
if(voltage == 0)
{
    Serial.println("Fault");
}
else if(voltage < 400)
{
    Serial.println("preheating");
}
else
{
    int16_t co2; //Variable que almacena el valor del sensor
    int voltage_diference=voltage-400;
    float concentration=voltage_diference*50.0/16.0;
    // Print Voltage
    /******Lectura de CO2*****
    Serial.print("voltage:");
    Serial.print(voltage);
    Serial.println("mv");
    //Print CO2 concentration
    Serial.print(concentration);
    Serial.println("ppm");

```

```
    co2=concentration;

    mydata[2]=co2>>8;
    mydata[3]=co2;

}
delay(100);

}
void loop() {
    os_runloop_once();
}
}
```

ANEXO C. ESPECIFICACIONES DEL GATEWAY LG02, SENSOR CO₂ Y LORA SHIELD

Features:

- Open Source OpenWrt system
- Low power consumption
- Firmware upgrade via Web
- Software upgradable via network
- Flexible protocol to connect to IoT servers
- Auto-Provisioning
- Built-in web server
- Managed by Web GUI, SSH via LAN or WiFi
- Internet connection via LAN, WiFi, Or 3G/4G via optional LTE module
- Failsafe design provides robustly system
- 2 x SX1276/SX1278 LoRa modules
- Full-duplex LoRa transceiver
- Two receive channels, and one transmit channel
- Limited support in LoRaWAN/ Support Private LoRa protocol
- Support upto 300 nodes
- LoRa band available at 433/868/915/920 Mhz
- Max range in LoRa: 5~10 km. Density Area:>500m

Figura I. Características del Gateway LG02.

Applications:

- Wireless Alarm and Security Systems
- Home and Building Automation
- Automated Meter Reading
- Industrial Monitoring and Control
- Long range Irrigation Systems
- GPS tracker,etc
- LoRaWAN Reapeter

Figura II. Aplicaciones del Gateway LG02.

Specifications:

Linux Side:

- Processor: 400MHz, 24K MIPS
- Flash: 16MB ; RAM: 64MB

Cellular 4G LTE (optional):

- Quectel EC25 LTE module (Click the link to find the right model for your region)
- Micro SIM Slot
- Internal 4G Antenna + External 4G Sticker Antenna.
- Up to 100Mbps downlink and 50Mbps uplink data rates
- Worldwide LTE,UMTS/HSPA+ and GSM/GPRS/EDGE coverage
- MIMO technology meets demands for data rate and link reliability in modem wireless communication systems

Interfaces:

- 10M/100M RJ45 Ports x 2
- WiFi : 802.11 b/g/n
- LoRa Wireless
- Power Input: 12V DC
- USB 2.0 host connector x 1
- USB 2.0 host internal interface x 1
- 2 x LoRa Interfaces

Figura III. Especificaciones técnicas del Gateway LG02.

Feature	
•	Waterproof and anti-corrosion
•	High sensitivity
•	Low power consumption
•	Excellent stability
•	Temperature compensation
•	Excellent linear output
•	High cycle life
•	Anti-water vapor interference
•	No poisoning
Specification	
•	Gas Detection: Carbon Dioxide (CO ₂)
•	Operating Voltage: 4.5 ~ 5.5V DC
•	Average Current: <60mA @ 5V
•	Peak Current: 150mA @ 5V
•	Output Signal: Analog output (0.4 ~ 2V)
•	Measuring Range: 0 ~ 5000ppm
•	Accuracy: ± (50ppm + 3% reading)
•	Preheating Time: 3min
•	Response Time: 120s
•	Operating Temperature: 0 ~ 50 °C
•	Operating Humidity: 0 ~ 95% RH (no condensation)
•	Service Life: >5 years
•	Size: 37mm * 69mm

Figura IV. Características y Especificaciones técnicas del Sensor CO₂.

Num	Label	Description
1	Signal	Analog Output (0.4~2V)
2	VCC	VCC (4.5~5.5V)
3	GND	GND

Figura V. Descripción general del Sensor CO₂.

Pin Mapping For LoRa

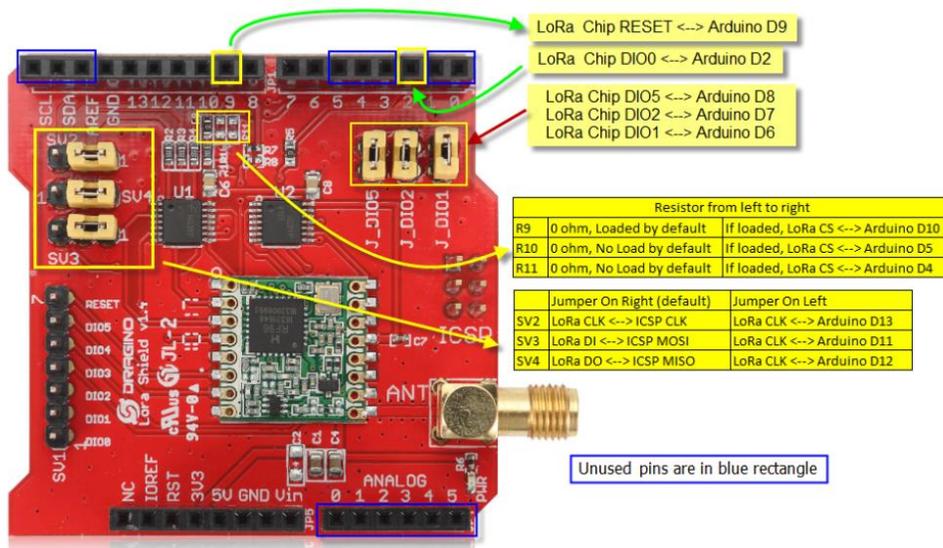


Figura VII. Asignación de pines del Lora Shield.

Wireless Specifications of RFM95W

- 168 dB maximum link budget.
- +20 dBm - 100 mW constant RF output vs.
- +14 dBm high efficiency PA.
- Programmable bit rate up to 300 kbps.
- High sensitivity: down to -148 dBm.
- Bullet-proof front end: IIP3 = -12.5 dBm.
- Excellent blocking immunity.
- Low RX current of 10.3 mA, 200 nA register retention.
- Fully integrated synthesizer with a resolution of 61 Hz.
- FSK, GFSK, MSK, GMSK, LoRaTM and OOK modulation.
- Built-in bit synchronizer for clock recovery.
- Preamble detection.
- 127 dB Dynamic Range RSSI.
- Automatic RF Sense and CAD with ultra-fast AFC.
- Packet engine up to 256 bytes with CRC.
- Built-in temperature sensor and low battery indicator.

Figura VIII. Especificaciones técnicas del Lora Shield.

Features

- Compatible with 3.3v or 5v I/O Arduino Board.
- Frequency Band: 915 MHZ/868 MHZ/433 MHZ (Pre-configure in factory)
- Low power consumption
- Compatible with **Arduino Leonardo, Uno, Mega, DUE**
- External Antenna via I-Pex connector

Figura IX. Características del Lora Shield.

ANEXO D. ESPECIFICACIONES DE ARDUINO MEGA 2560

Microcontrolador	ATmega2560
Tensión de trabajo	5V
Tensión de entrada (recomendada)	7-12V
Tensión de entrada (límite)	6-20V
Pines Digitales I/O	54 (de los cuales 15 proporcionan salida PWM)
Pines de entradas Analógicas	16
DC Corriente por Pin I/O	20 mA
DC Corriente por Pin 3.3V	50 mA
Memoria Flash	256 KB de los cuales 8 KB se usan por el bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz
Largo	101.52 mm
Anchu	53.3 mm
Peso	37 g

Figura XI. Especificaciones técnicas del Arduino Mega 2560.

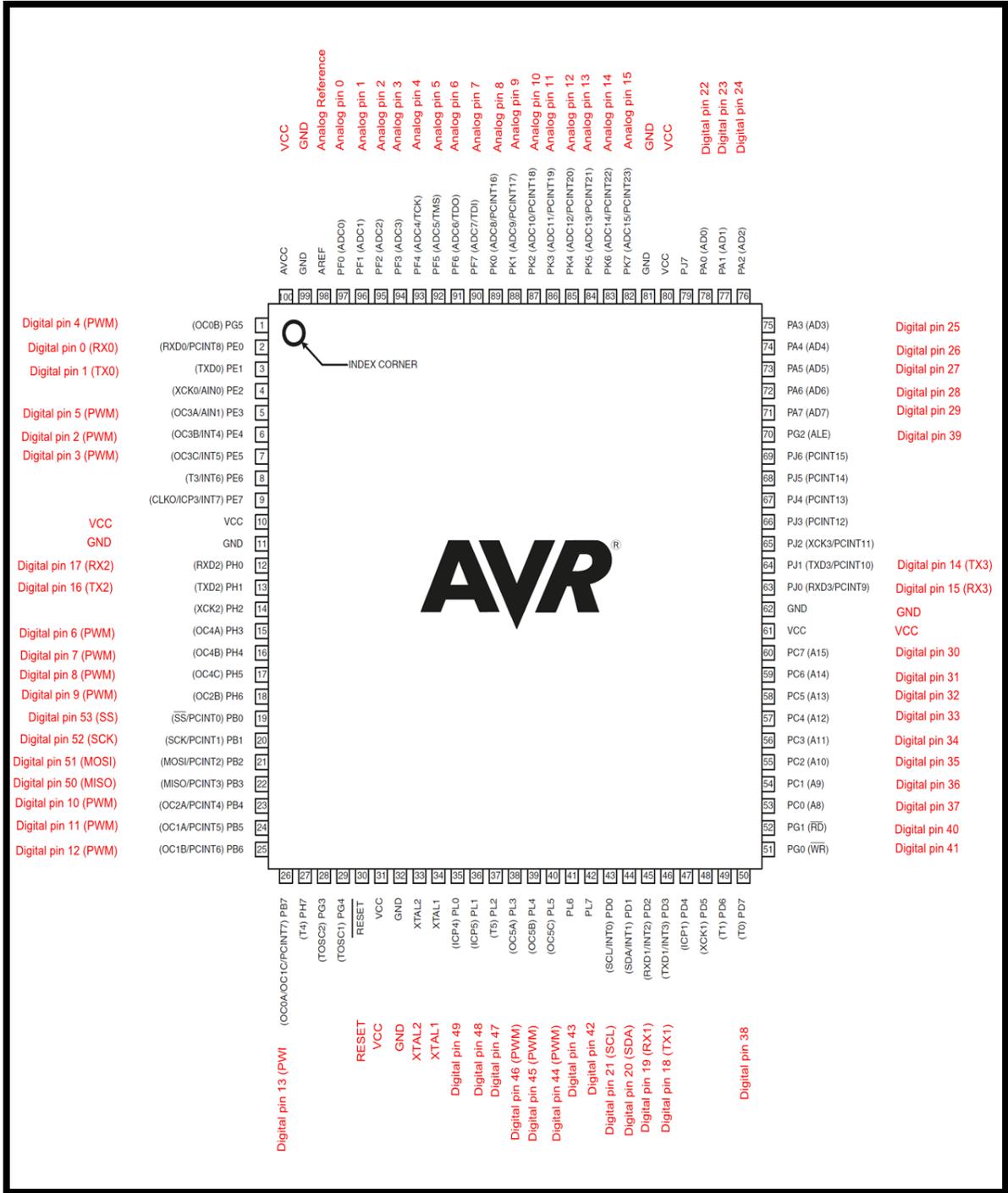
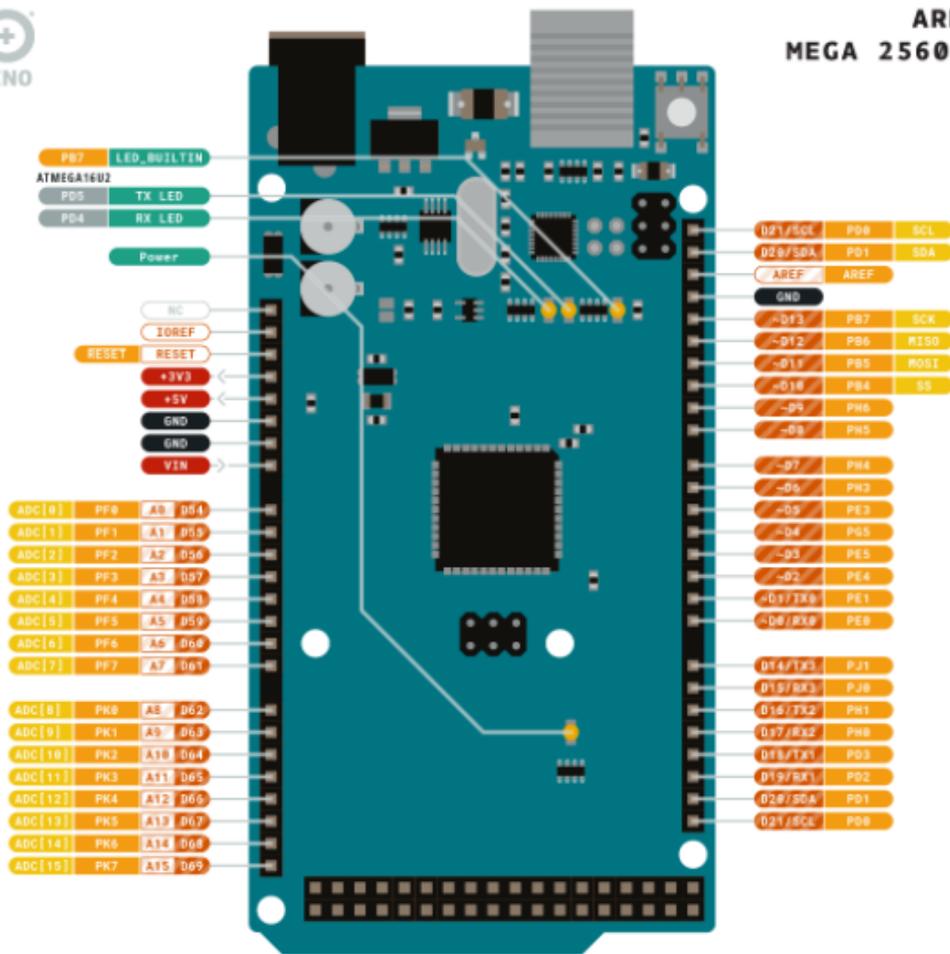


Figura XII. Mapeo pines Arduino y los puertos de Arduino Mega 2560.



ARDUINO MEGA 2560 REV3



Ground	Internal Pin	Digital Pin	Microcontroller's Port
Power	SWD Pin	Analog Pin	
LED	Other Pin	Default	

ARDUINO . CC

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1888, Mountain View, CA 94042, USA.

Figura XIII. Asignación de pines del Arduino Mega 2560.