

Universidad Andrés Bello
Facultad de Ingeniería
Magíster en Ingeniería Informática



**OBSERVATORIO MEDIOAMBIENTAL GEOREFERENCIADO
PARA MONITOREO DE JARDINES INFANTILES BASADO
EN INTERNET DE LAS COSAS Y WEBSOCKETS**

**Trabajo de título para optar al grado de
Magíster en Ingeniería Informática**

HÉCTOR EUGENIO REYES DÍAZ

**PROFESOR GUÍA
DAVID RUETE ZUÑIGA**

**SANTIAGO-CHILE
MARZO-2018**

Universidad Andrés Bello
Facultad de Ingeniería
Magíster en Ingeniería Informática



DECLARACIÓN DE ORIGINALIDAD Y PROPIEDAD

Yo, **Héctor Reyes Díaz**, declaro que este documento no incorpora material de otros autores sin identificar debidamente la fuente.

Santiago, 5 de Marzo del 2018

Firma del alumno

Agradecimientos

Agradezco a todas las personas que de uno u otro modo colaboraron en la realización de este proyecto de investigación especialmente a mi profesor guía David Ruete Zúñiga, por su paciencia, sus conocimientos y constante apoyo en el desarrollo de esta tesis, que gracias a sus feedback pude desarrollar esta tesis en función de su ayuda y conocimientos.

A mi familia más cercana por apoyarme siempre y no permitirme nunca dudar de mis capacidades, además de mi prima Liz que me ayudo a realizar las mediciones en el Jardín Vitamina.

Índice de Contenidos

Resumen.....	8
I. Introducción.....	9
1.1 Introducción al Tema.....	9
1.2 Motivación.....	10
1.3 Contexto del Trabajo.....	14
II. Pregunta o Problema a Solucionar.....	15
2.1 Diagrama de Ishikawa.....	15
2.2 Selección de causas a trabajar.....	16
III. Objetivos.....	17
3.1 Objetivos específicos.....	17
3.2 Objetivo general.....	17
3.3 Hipótesis.....	17
3.4 Alcance.....	17
3.4.1 Definición de Alcance.....	17
3.4.2 Limitación al Alcance.....	19
3.4.3 Supuestos.....	21
IV. Marco Teórico.....	22
4.1 Internet de las Cosas.....	22
4.1.1 Arquitecturas de Internet de las Cosas.....	24
4.1.2 Plataformas de hardware y software IoT.....	27
4.1.2.1 Dispositivos de Procesamiento.....	27
4.1.2.2 Sensores.....	27
4.1.2.3 Protocolos de comunicación.....	29
4.1.2.4 Middleware y gateway.....	32
4.1.2.5 Plataformas de analítica.....	35
4.1.2.6 Plataformas de despliegue.....	36
V. Estado del Arte.....	38
5.1 Medición de Variables ambientales.....	39
5.2 Arquitectura de sistemas de medición remota.....	47
5.3 Discusión.....	56
VI. Enfoque metodológico.....	58
6.1 Tipo de proyecto.....	58
6.2 Metodología de investigación cuantitativa.....	59
6.2.1 Proceso de Obtención De Datos.....	61
VII. Plan de tesis.....	62
7.1 Definición de tareas.....	62
7.2 Relación de dependencia de tareas.....	62
7.3 Asignación de recursos a tareas.....	64
7.4 Hitos.....	65
VIII. Desarrollo del Diseño de Investigación y Proyecto.....	67
8.1 Especificación de Requerimientos.....	67
8.1.1 Introducción.....	67
8.1.1.1 Propósito.....	67
8.1.1.2 Alcance.....	67
8.1.1.3 Personal Involucrado.....	67
8.1.2 Descripción General.....	68
8.1.2.1 Funcionalidad del Producto.....	68
8.1.2.2 Características de los Usuarios.....	66

8.1.2.3 Restricciones.....	68
8.1.2.4 Suposiciones y dependencias.....	68
8.1.3 Requisitos Específicos.....	68
8.1.3.1 Requisitos comunes de las interfaces.....	69
8.1.3.1.1 Interfaces de Usuario.....	69
8.1.3.1.2 Interfaces de Hardware.....	69
8.1.3.1.3 Interfaces de Software.....	69
8.1.3.1.4 Interfaces de Comunicación.....	69
8.1.4 Requerimientos del Negocio o Dominio.....	69
8.1.5 Requerimientos Funcionales.....	70
8.1.5.1 Casos de Uso.....	71
8.1.6 Requerimientos no funcionales.....	74
8.1.7 Matrices de Trazabilidad.....	76
8.1.8 Atributos de Calidad y Métricas.....	76
8.2 Análisis y Diseño de Arquitectura.....	79
8.2.1 Introducción.....	79
8.2.2 Arquitectura de Sistemas.....	79
8.2.2.1 Descripción de capas del sistema.....	80
8.2.2.2 Filosofía de interacción del sistema.....	81
8.2.3 Arquitectura de Software.....	82
8.2.3.1 Vistas Arquitectónicas.....	84
8.2.3.2 Descripción de Componentes.....	89
8.2.3.3 Descripción de conectores.....	91
8.2.3.4 Matriz de Trazabilidad.....	92
8.2.3.5 Detalles de Implementación.....	93
8.2.4 Arquitectura de Hardware.....	90
8.2.3.1 Descripción de Arquitectura.....	90
8.2.4.2 Descripción de Componentes.....	91
8.3 Desarrollo Plataforma.....	99
8.3.1 Plataforma de Medición.....	100
8.3.1.1 Montaje Nodo de Medición.....	100
8.3.1.2 Implementación módulo de medición.....	102
8.3.2 Plataforma de Monitoreo.....	104
8.3.2.1 Implementación módulo servidor web.....	104
8.3.2.2 Implementación módulo comunicación TR.....	106
8.3.2.3 Implementación módulo persistencia de datos.....	107
8.3.2.4 Implementación de módulo de Mapas.....	110
8.3.2.5 Implementación módulo de Gráficos.....	113
8.3.2.6 Implementación módulo de alerta Twitter.....	115
8.3.2.7 Implementación de Interfaces gráficas y vistas.....	116
8.4 Pruebas.....	117
8.4.1 Set de pruebas.....	118
IX. Experimentación.....	133
9.1 Selección de población y Muestra.....	133
9.2 Recolección de datos.....	135
9.3 Análisis de datos.....	142
X. Conclusiones.....	143
XI. Referencias Bibliográficas.....	146
XII. Anexos.....	151

Índice de figuras

FIGURA 1: Mapa de calidad del aire región metropolitana	12
FIGURA 2: Mapa de contaminación acústica región metropolitana.....	13
FIGURA 3: Diagrama Ishikawa.....	15
FIGURA 4: Convergencia del Protocolo de Internet IP.....	23
FIGURA 5: Arquitectura de capas o layer.....	25
FIGURA 6: Modelo de arquitectura genérica para una aplicación de IoT.....	26
FIGURA 7: Esquema de interacción de sensores y actuadores físicos y virtuales	28
FIGURA 8: Tecnologías de telecomunicación inalámbricas.....	30
FIGURA 9: Esquema de funcionalidad de Gateway.....	33
FIGURA 10: Middleware como infraestructura de sistemas distribuidos.....	33
FIGURA 11: Esquema de WebSockets.....	35
FIGURA 12: Sensor Plantower PMS3003.....	39
FIGURA 13: Sensores comerciales de bajo costo.....	40
FIGURA 14: Sensores Profesionales.....	40
FIGURA 15: Sensor comerciales mediano costo AQMesh.....	41
FIGURA 16: Esquema de arquitectura y nodo sensor.....	42
FIGURA 17: Imagen nodo sensor Libelium Waspote Plug & Sense.....	43
FIGURA 18: Imagen sensores de audio de alta gama.....	44
FIGURA 19: Imagen sensor de sonido fabricación académica.....	46
FIGURA 20: Modelo arquitectura genérica para IoT.....	47
FIGURA 21: Imagen de interfaz aplicación OpenSense.....	49
FIGURA 22: Imagen de interfaz aplicación Citi-Sense-MOB.....	50
FIGURA 23: Interfaz monitoreo Jardines Infantiles, basada sensor AQMesh.....	52
FIGURA 24: Interfaz plataforma SINCA.....	53
FIGURA 25: Interfaz monitoreo SINCA.....	54
FIGURA 26: Infograma estación monitoreo SINCA.....	55
FIGURA 27: Esquema de sensor Thermo 48i utilizado por SINCA.....	55
FIGURA 28: Interfaz plataforma monitoreo aqicn.org.....	56
FIGURA 29: Esquema de proceso cuantitativo.....	58
FIGURA 30: Esquema de actividades del proyecto EDT.....	62
FIGURA 31: Esquema de carta Gantt del proyecto.....	63
FIGURA 32: Estimación de recursos del proyecto.....	65
FIGURA 33: Esquema Arquitectura del Proyecto.....	80
FIGURA 34: Filosofía Cliente-Servidor.....	81
FIGURA 35: Esquema arquitectura de componentes de sistema.....	82
FIGURA 36: Esquema Modelos de “4+1” vistas.....	83
FIGURA 37: Esquema vista Escenarios.....	84
FIGURA 38: Esquema vista Lógica.....	85
FIGURA 39: Esquema vista Desarrollo.....	86
FIGURA 40: Esquema vista despliegue.....	87
FIGURA 41: Esquema vista de Procesos.....	88
FIGURA 42: Esquema arquitectura de hardware.....	94

FIGURA 43: Imagen de controlador Arduino Uno.....	95
FIGURA 44: Imagen de unidad de comunicación DTK Zigbee.....	95
FIGURA 45: Imagen Sensor MQ5.....	97
FIGURA 46: Imagen Sensor GPS Neo-6M.....	98
FIGURA 47: Imagen Sensor Sonido LM-393.....	99
FIGURA 48: Esquema conexiones nodo medición.....	100
FIGURA 49: Fotografías de montaje lo módulo de medición.....	101
FIGURA 50: Fotografía módulo de medición y envió de datos.....	101
FIGURA 51: Esquema implementación módulo de captura y envió de datos.....	103
FIGURA 52: Esquema modulo servidor web.....	104
FIGURA 53: Esquema código modulo servidor web.....	105
FIGURA 54: Esquema modulo comunicación tiempo real.....	106
FIGURA 55: Esquema código modulo comunicación tiempo real.....	107
FIGURA 56: Esquema modulo persistencia de datos.....	108
FIGURA 57: Esquema modelo de datos.....	109
FIGURA 58: Esquema código modulo persistencia de datos.....	110
FIGURA 59: Esquema módulo mapas.....	111
FIGURA 60: Esquema código modulo mapas.....	113
FIGURA 61: Esquema código modulo Gráficos.....	114
FIGURA 62: Esquema módulo Twitter.....	115
FIGURA 63: Esquema código modulo Twitter.....	116
FIGURA 64: Esquema de vistas.....	116
FIGURA 65: Esquema de Consola monitor IDE Arduino.....	118
FIGURA 66: Esquema de Consola servidor NodeJS.....	119
FIGURA 67: Esquema de Consola servidor MongoDB.....	120
FIGURA 68: Esquema de Consola cliente Web.....	121
FIGURA 69: Esquema de Dashboard.....	122
FIGURA 70: Esquema de Interfaz de Twitter.....	123
FIGURA 71: Esquema de zona de pruebas de alcance de red sensores.....	127
FIGURA 72: Fotografías de instalación de nodos de zona de pruebas.....	128
FIGURA 73: Fotografías estación monitoreo y nodo de medición.....	128
FIGURA 74: Esquema configuración red de sensores zigbee.....	129
FIGURA 75: Fotografía sensor Thermo 48i utilizado por SINCA, Puente Alto.....	134
FIGURA 76: Esquema de ubicación zona de experimentación, Puente Alto.....	134
FIGURA 77: Fotografías de estaciones de medición en Jardines Vitamina.....	135
FIGURA 78: Captura de plataforma de monitoreo de data en Jardines Vitaminas.....	136

Índice de Tablas

TABLA 1: Tabla de efectos de contaminación acústica.....	11
TABLA 2: Tabla de causas basales.....	16
TABLA 3: Matriz de definición de objetivos específicos.....	17
TABLA 4: Matriz de definición de métricas de objetivos específicos.....	17
TABLA 5: Matriz de trazabilidad de objetivos específicos medibles vs causas.....	18

TABLA 6: Descripción de alcance según los objetivos específicos medibles.....	19
TABLA 7: Limitaciones del alcance del proyecto.....	20
TABLA 8: Parámetros de búsqueda sistemática del estado del arte.....	38
TABLA 9: Definición de hitos del proyecto.....	66
TABLA 10: Matriz de Interesados del Proyecto.....	67
TABLA 11: Componentes de software.....	91
TABLA 12: Conectores de software.....	92
TABLA 13: Matriz de Trazabilidad Módulos SW vs Requisitos RN, RF, RNF.....	92
TABLA 14: Matriz de Trazabilidad Nodos HW vs Módulos SW.....	92
TABLA 15: Promedios de mediciones primer día.....	136
TABLA 16: Volumen de datos capturados primer día.....	138
TABLA 17: Tabla de Validación de Objetivos Especifico Medibles.....	144
TABLA 18: Extracción datos desde curva logarítmica.....	152
TABLA 19: Tabla resumen de plataformas web de monitoreo ambiental.....	156
TABLA 20: Tabla mediciones adicionales en zonas focalizadas.....	156

Índice de Gráficos

GRAFICO 1: Integridad de datos versus distancia de envió.....	129
GRAFICO 2: Integridad de datos versus tiempo de medición.....	130
GRAFICO 3: Consumo de batería de nodos de medición datos versus tiempo....	130
GRAFICO 4: Promedios de mediciones primer día.....	137
GRAFICO 5: Promedios de mediciones segundo día.....	138
GRAFICO 6: Promedios de mediciones primera semana.....	139
GRAFICO 7: Resumen de Porcentaje Promedio semanales de datos capturados	139
GRAFICO 8: Promedios de lectura analógica sensor sonido primera día.....	140
GRAFICO 9: Resumen de Promedios semanales de lectura sensor MQ.....	140
GRAFICO 10: Resumen de Promedios semanales de lectura sensor sonido.....	141
GRAFICO 11: Promedios de porcentaje de datos capturados primera semana....	141
GRAFICO 12: Curva comportamiento logarítmica Sensor de Gas MQ.....	151
GRAFICO 13: Curva tendencia comportamiento lineal Sensor de Gas MQ.....	152

Resumen

El presente trabajo corresponde a un proyecto de diseño e implementación de una plataforma de medición georeferenciada que permite desplegar en tiempo real vía web, información medioambiental focalizada de Jardines Infantiles mediante la utilización de tecnología de Internet de las Cosas y protocolo de WebSockets.

La plataforma desarrollada tiene como objetivo principal aumentar el nivel de información medioambiental de zonas de interés social mediante un prototipo de software y hardware de fácil implementación y escalado utilizando nodos de medición de bajo costo y de fácil instalación e integración. La plataforma permite desplegar información en un mapa georeferenciado el comportamiento de variables como el nivel de ruido o calidad del aire, además proporciona interacción con los usuarios de forma activa mediante avisos de alarma vía Twitter.

El diseño de la solución fue desarrollado en base a una arquitectura basada en capas con una filosofía servidor-cliente implementada mediante el paradigma push basado en WebSockets, lo que proporciona una solución ideal para la transmisión de datos en tiempo real entre servidores y clientes. En cuanto a las tecnologías utilizadas por el proyecto, se contempló como ambiente de desarrollo el stack MEN (Mongodb, Express.js, Node.js), lo que permitió disponer de una solución modular, liviana y auto contenida, adecuada para proyectos de internet de las cosas. En cuanto a la arquitectura de hardware implementada, se desarrollaron dos estaciones de medición basada en Arduino intercomunicadas a través de una red ZigBee de largo alcance, lo cual permitió independencia a la hora de la interconexión de dispositivos en terreno. Respecto a los resultados de las pruebas y experimentos, se puede mencionar que se realizaron mediciones durante el mes de Enero en dos Jardines Infantiles ubicados en una zona urbana de la comuna de Puente alto. Estas mediciones permitieron comprobar las capacidades de capturar y envió datos al servidor central. Este envió fue exitoso lográndose tasas de efectividad cercana al 92% con distancias superiores a los 800 metros. También los experimentos permitieron comprobar que los datos obtenidos por sensores MQ detectaron diferencias en las condiciones ambientales de los dos jardines, pudiendo establecer un patrón de comportamiento primitivo de las condiciones ambientales en función de la ubicación del Jardín. Finalmente se comprueba que la plataforma desarrollada, es útil para aumentar el nivel de información medioambiental de zonas focalizadas, permitiendo apoyar a los padres y apoderados a la hora de seleccionar Jardines infantiles para sus hijos.

I. Introducción

1.1 Introducción al Tema

En los últimos años, ha surgido un nuevo paradigma tecnológico llamado internet de las cosas y en particular telemetría como servicio, que se ha convertido en una posibilidad real de capturar información de interés en zonas que antes era imposible monitorear, debido a la dificultad de accesibilidad, alto costo del equipamiento o simplemente por desconocimiento de la importancia que representa la medición de diferentes magnitudes presentes en el medioambiente.

Según **(Fabregas, 2016)**, el termino de internet de las cosas (IoT por sus siglas en inglés), postula que en un futuro cercano millones de dispositivos podrán conectarse a internet, abriendo un nuevo abanico de aplicaciones y servicios que enriquecerán la forma en que los seres humanos interactúan con su entorno. La clave en la adopción y desarrollo del Internet de las Cosas se fundamenta en la interconexión de objetos y servicios, por lo que necesariamente se requiere el uso de protocolos y estándares, a través de los cuales los objetos puedan interactuar con los usuarios, e incluso entre ellos. Sin embargo, para que las aplicaciones y productos del Internet de las Cosas sean escalables requieren también que sean sostenibles, lo que se traduce en equipos capaces de operar años, utilizando únicamente baterías u obteniendo energía de su entorno. Así mismo, un requisito fundamental es la implementación de tecnologías y comunicación inalámbrica, que permita la conexión remota de miles de dispositivos de manera ubicua, sin restricciones de infraestructura o instalación.

Además, este paradigma incorpora metodología, arquitectura, y sistemas que permiten integrar mediante software, diferentes plataformas y desplegar información a miles de usuario a través de ambientes web y redes sociales en forma fácil y rápida, lo que permite desarrollar aplicaciones que pueden ayudar a mejorar la calidad de vida de los habitantes de una ciudad.

De acuerdo con esto, la idea de esta investigación es justamente aprovechar las capacidades, postulados y plataformas que proporciona el internet de las cosas, con el fin de mejorar el bienestar y las condiciones de salud de los habitantes de una ciudad, en particular preservar la salud de niños y jóvenes.

Es por esta razón, que este proyecto, se enmarca en la llamada telemetría como servicio, específicamente la captura en forma remota de variables medioambientales

(observatorio) presentes en zonas de interés social. Esta información, permitiría a la población, seleccionar mejores lugares donde vivir, estudiar o trabajar, reduciendo el riesgo de exposición prolongada.

1.2 Motivación

La medición de variables medioambientales en entornos de interés social, como colegios y jardines infantiles, se torna cada vez más relevante a la hora de preservar la salud y bienestar de niños y jóvenes. Por este motivo la sociedad en su conjunto debe preocuparse por salvaguardar en cada momento la salud de nuestros menores de los efectos de la contaminación del aire y acústica.

De acuerdo con esto, resulta de gran interés, determinar el real impacto que poseen las variables medioambientales y los episodios de contaminación en la salud de los habitantes de una comunidad. Según **(Gauderman, 2004)**, existe evidencia de que la contaminación del aire por monóxido de carbono está asociada con enfermedades respiratorias y sucesos de intoxicación especialmente en niños menores de 5 años. Los niños que viven en áreas altamente contaminadas son más propensos a buscar atención médica por síntomas respiratorios y crisis de asma. También hay evidencia de que la contaminación del aire especialmente por monóxido de carbono disminuye la función pulmonar, produce exacerbaciones de crisis de asma y aumenta las tasas de hospitalización por esta causa **(Eder, 2006)**.

Los estudios usualmente determinan como indicador de exposición el promedio diario de material monóxido de carbono respirable, lo cual según la OMS (Organización mundial de la salud) constituye la guía a corto plazo, y las medias anuales corresponden a las guías a largo plazo. Entonces la OMS establece las guías de calidad de aire para monóxido de carbono Co: hasta 15 ppm (partículas por millón) media de 8 horas. Estas guías de calidad de aire se fundamentan en la evidencia de estudios epidemiológicos los cuales muestran que a partir de estos valores los efectos en salud aumentan significativamente.

Los efectos sobre la salud se dividen en: a) agudos, como irritación de ojos y nariz, aumento de las infecciones respiratorias agudas altas y bajas, exacerbaciones del asma bronquial, de enfermedad pulmonar obstructiva crónica, de cardiopatías y aumento de las tasas de mortalidad **(Ostro, 1996)**, **(Román, 2004)**, y b) efectos crónicos o diferidos, como el cáncer pulmonar **(Rosales, 2001)**.

Con respecto a la contaminación acústica se pueden mencionar que, según la “European Commission - Environment”, la contaminación acústica ambiental se refiere al ruido causado por carretera, ferrocarril y tráfico del aeropuerto, la industria, la construcción, así como algunas otras actividades al aire libre.

“El ruido excesivo perjudica seriamente la salud humana e interfiere con las actividades diarias de las personas en la escuela, en el trabajo, en casa y en el tiempo libre. Se puede perturbar el sueño, causar efectos cardiovasculares y psicofisiológicos, reducir el rendimiento y provocar respuestas molestia y los cambios en el comportamiento social” **(Sonko, 2013)**.

En base a varios estudios experimentales, la relación entre la exposición del ruido nocturno y efectos la salud se puede resumir de la siguiente manera **(Teixeira, 2015)**.

Rango	Efectos en la salud
Hasta 30 dB.	Aunque las sensibilidades y las circunstancias individuales pueden diferir, parece que hasta este nivel no se observan efectos biológicos importantes.
De 30 dB a 40 dB	Una serie de efectos sobre el sueño se observan desde este rango: movimientos corporales, despertar, molestias, etc. La intensidad del efecto depende de la naturaleza de la fuente y el número de eventos. Los grupos más vulnerables son los niños, los enfermos crónicos, etc.
De 40 dB a 55 dB	Se observan efectos de salud perjudiciales entre la población expuesta. Muchas personas tienen que adaptar sus vidas para hacer frente al ruido por la noche. Los grupos vulnerables están más severamente afectados
Por encima de 55 dB	La situación se considera cada vez más peligrosa para la salud pública. Suceden efectos adversos para la salud con frecuencia, como la perturbación del sueño. Además, hay pruebas de que el riesgo de enfermedades cardiovasculares aumenta.

TABLA 1: Tabla de efectos de contaminación acústica (Fuente: Teixeira, 2015)

En Chile, el ministerio de medio ambiente y ministerio de salud se encargan de recolectar información y alertar periódicamente de las condiciones medioambientales

que existen en la ciudad y la evolución de las atenciones médicas debido a condiciones desfavorables.

Actualmente, se disponen de dos mapas informativos de las condiciones medioambientales y acústicas presentes en la región metropolitana. El primero se basa en las informaciones obtenidas mediante once estaciones distribuidas en la ciudad y que capturan continuamente los índices de material particulados y otras variables de interés. Estas estaciones monitorean las condiciones medioambientales y emiten informes que permiten determinar eventos de alerta, preemergencias y emergencias ambiental a la comunidad **(Ministerio Medio Ambiente, 2017)**.

A continuación, se muestra el mapa georeferenciado con la ubicación de las estaciones medioambientales. Este mapa proporciona información global y agregada de las variables medidas, lo cual se combina con información de pronósticos meteorológicos para definir las alertas medioambientales.

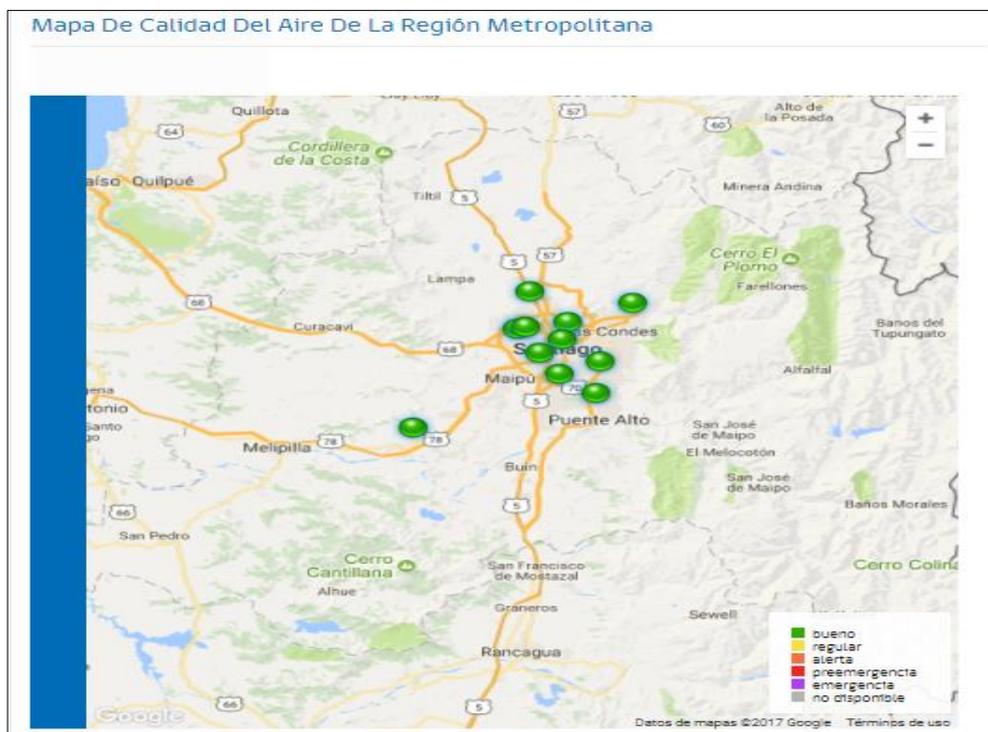


FIGURA 1: Mapa de calidad del aire región metropolitana
(Fuente: SINCA Sistema Información Nacional de Calidad del Aire, Sept. 2017)

Con respecto a la situación de información de contaminación acústica, se puede mencionar que se dispone para la comunidad de un mapa georeferenciado, que distribuye los niveles de ruido medidos en decibeles durante periodos focalizados de

medición, es decir a diferencia con la medición de las condiciones del aire que se realiza en estaciones fijas que miden en forma continua las variables, la medición de contaminación sonora, se mide en momentos y lugares específicos cada periodo y no de forma continua, lo cual representa una fotografía de un momento y no la medición continua de las variables (**Ministerio Medio Ambiente, 2017**).

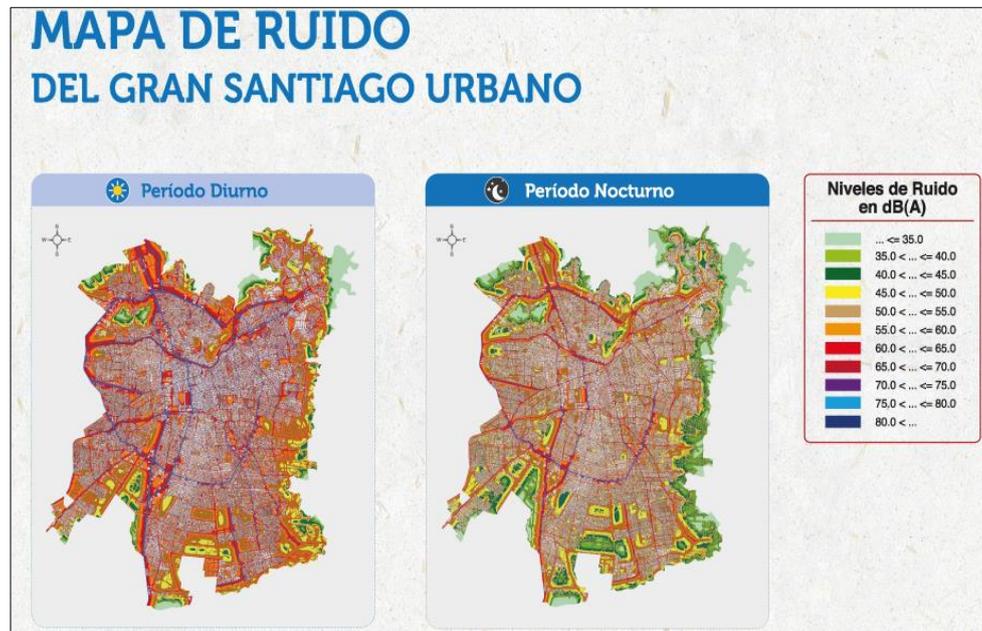


FIGURA 2: Mapa de contaminación acústica región metropolitana (Fuente: Ministerio Medio Ambiente, Gobierno de Chile, Abril 2017)

Como se puede observar, la información que se dispone actualmente (mapa del aire y acústico) se torna insuficiente al momento de caracterizar las condiciones medioambientales en zonas focalizadas como son los jardines infantiles en una comuna en particular. En realidad, no sabemos que pasa a las afueras de los centros de educación y menos disponemos de información actualizada y completa de las condiciones intradomiciliarias.

De acuerdo con esta situación, se abre una oportunidad atractiva de investigación, que permita aportar conocimientos en materia de información medioambiental de zonas de riesgo para niños como son los jardines infantiles, mediante la incorporación un sistema denominado “Observatorio Medioambiental”, que utilizando telemetría remota y una plataforma de despliegue, emita alarmas en tiempo real sobre situaciones de riesgo para la salud y además permita aumentar la certidumbre al momento de seleccionar jardines infantiles en zonas específicas de la ciudad.

1.3 Contexto del Trabajo

El contexto del trabajo es un ámbito de investigación aplicada, proporcionado por un ambiente de experimentación basado en prototipado incremental y luego pruebas controladas de comunicación, para continuar con la captura de datos en el entorno real que posean interés social por ejemplo dependencias de jardines infantiles de una comuna.

La idea fundamental es lograr transitar desde un ambiente de laboratorio científico a un ámbito de ciencia aplicada.

II. Pregunta o Problema a Solucionar

¿Cuánto puede ayudar la información y aviso en tiempo real de las variables medioambientales presentes en jardines infantiles, para reducir los riesgos de enfermedades en niños? ¿Cómo se puede reducir la incertidumbre de los padres al momento de seleccionar un jardín infantil que posea bajos índices de contaminación?, ¿Se puede establecer un mapa medioambiental focalizado de zonas riesgosas para jardines infantiles en cada comuna? Para responder a estas preguntas, se debe primero, comprender las causas que originan la falta de información en tiempo real de variables medioambientales presentes en jardines infantiles.

2.1 Diagrama de Ishikawa

El diagrama Ishikawa, consiste en una representación gráfica de las causas que originan una problemática particular. Este diagrama contempla una línea horizontal, correspondiente al problema a resolver, que se conectan flechas rectas, que representan “causas del problema”, las que a su vez se relacionan con nuevas líneas que representan sub-causas y así sucesivamente, por lo que en esta iteración permite encontrar el conjunto de problemas que representan su causa raíz. Este diagrama de Ishikawa representa el efecto de las causas que se explicitan en las espinas. En este caso, elección informada de jardines infantiles.

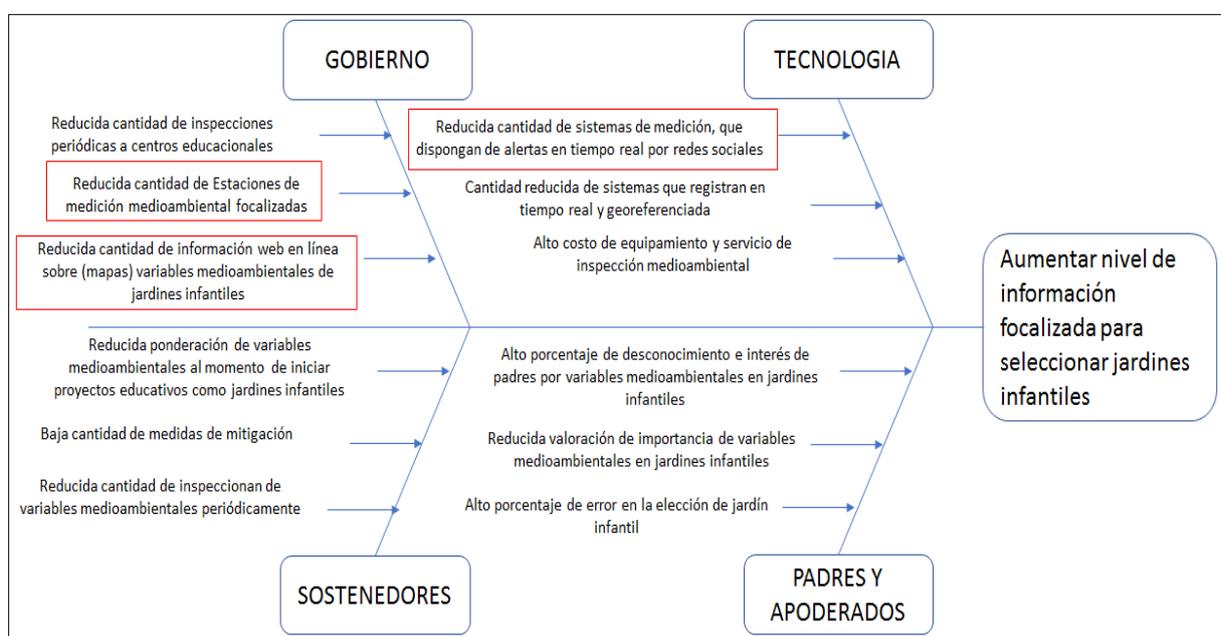


FIGURA 3: Diagrama Ishikawa (Fuente: elaboración propia)

2.2 Selección de causas a trabajar

De acuerdo con el diagrama de Ishikawa, existen diferentes causas que originan la ausencia de información y aviso oportuno de variables medio ambientales presentes en jardines infantiles. Estas causas se relacionan con aspectos de cobertura insuficiente de estaciones de monitoreo que dispone la autoridad, reducida disponibilidad de plataformas de información focalizada, lo cual genera la carencia de un mapa comunal de contaminación de jardines infantiles. En particular, la investigación se enfocará en los aspectos relacionados con la captura, geo posicionamiento, y aviso en tiempo real de dichos parámetros medioambientales a los interesados de la comunidad educativa, agregando un módulo de comunicaciones y registro que permita informar a través de redes sociales la situación periódica que enfrentan los niños en los jardines infantiles.

Causa	Causa	Métricas
C1	Reducida cobertura de estaciones meteorológicas dispuesta por Comuna, lo cual no permite disponer de información focalizada	$Covertura = \frac{\text{Cantidad de nodos de medicion}}{\text{Area que se desea cubrir (Km}^2\text{)}}$
C2	Reducida cantidad de plataformas web de información que muestren variables ambientales en tiempo real	$\%Inform = \frac{\text{Plataformas monitoreo ambiental TR}}{\text{Total plataformas información ambiental}}$
C3	Reducida cantidad de sistemas de medición, que dispongan de alertas temprana en tiempo real por redes sociales	$TAT = \frac{\Delta t_{evento} - t_{respuesta}}{t_{respuesta\ medio/ideal}}$

TABLA 2: Tabla de causas basales (Fuente: elaboración propia)

III. Objetivos

3.1 Definición de objetivos específicos

Codificación	Objetivo
OEM1	Aumentar nodos de medición de variables medioambientales en una comuna, para reducir la incertidumbre en la elección de jardines infantiles
OEM2	Aumentar porcentaje información medioambiental desplegada vía web en tiempo real.
OEM3	Reducir el tiempo de aviso y alerta de eventos de riesgos de salud de los niños

TABLA 3: Matriz de definición de Objetivos medibles del proyecto
(Fuente: elaboración propia)

Matriz de definición de métricas de objetivos específicos

Objetivo	Métrica	VAM	CEM
OEM1	CNM: Cantidad de nodos de medición por km ² (Para efectos del experimento, se elige la comuna Puente Alto)	CNM (estimado) Se disponen de una estación de medición en para cubrir 88 km ²	El objetivo es aumentar a tres nodos de medición: CNM \geq 3
OEM2	PIF: Porcentaje información ambiental en línea (portales web focalizados)	PIF (estimado) Se disponen de reducidas plataformas web que permitan monitoreo ambiental en línea (públicas y privadas, alrededor de 4 sobre un total de 20).	PIF \geq 25%
OEM3	TAT: Índice de alerta temprana de condiciones riesgosas de contaminación para niños en jardín infantil	IAT (estimado), basado en las alertas que entrega el ministerio del medio ambiente que informa un día antes de las emergencias (8 Horas)	El objetivo es disminuir el tiempo de alerta: IAT \leq 15 min.

TABLA 4: Matriz de definición de métricas de los objetivos medibles del proyecto
(Fuente: elaboración propia)

Matriz de Trazabilidad

Objetivo - Causa	C1	C2	C3
OEM1	X		
OEM2		X	
OEM3			X

TABLA 5: Matriz de trazabilidad de objetivos específicos medibles versus causas
(Fuente: elaboración propia)

3.2 Objetivo General

OG: Diseñar e implementar una plataforma de medición en tiempo real que permita desplegar información medioambiental y alertar a la comunidad educativa sobre las condiciones medioambientales de un jardín infantil.

3.3 Hipótesis

H1: Mediante un sistema de medición remota georreferenciado basada en internet de las cosas y tecnología de WebSockets, se puede integrar, procesar y desplegar información medioambiental en tiempo real vía web, que permita alertar sobre niveles de contaminación riesgosos para los niños y servir como fuente de información para decidir la elección de un jardín infantil en una ubicación en particular.

3.4 Alcance

3.4.1 Alcance

De acuerdo con la problemática descrita mediante el diagrama de Ishikawa, se establecieron causas y sub-causas que dieron origen a la construcción de objetivos específicos basados en criterios S.M.A.R.T, que permite determinar un resultado que acompañe al objetivo general planteado. El alcance de la tesis queda definido en la siguiente tabla:

Objetivo	Alcance
OEM1: Aumentar nodos de medición de variables medioambientales en una comuna, para reducir la incertidumbre en la elección de jardines infantiles	<ul style="list-style-type: none"> • Realizar revisión de la bibliografía referente a métodos de medición, sensores y plataformas de procesamiento y despliegue de información en tiempo real. • Seleccionar plataformas, diseñar arquitectura del sistema, realizar la integración de componentes (hardware y software), realizar pruebas de funcionamiento • Realizar experimentos de captura de datos en terreno para validar la solución propuesta
OEM2: Aumentar porcentaje información en línea que muestren variables ambientales	<ul style="list-style-type: none"> • Realizar búsqueda de plataformas de información digitales focalizadas a las mediciones de variables medioambientales
OEM3: Reducir el tiempo de aviso y alerta de eventos de riesgos de salud de los niños	<ul style="list-style-type: none"> • Realizar revisión bibliográfica de plataformas de comunicación IoT que permiten conectar sistemas de hardware a redes sociales, para el envío de alertas de Twitter • Diseñar e implementar modulo alerta tiempo real vía Twitter • Realizar pruebas unitarias, de integración y sistemas para validar la solución propuesta.

TABLA 6: Descripción de alcance del proyecto según los objetivos específicos medibles (Fuente: elaboración propia)

3.4.2 Limitación al Alcance

Como se describe en el diagrama Ishikawa, existen diferentes causas que originan la falta de información de variables medioambientales para la elección y alerta de jardines ambientales. De acuerdo con esto, se presentan las razones de excluir dichas causas en este trabajo de investigación.

Causas	Limitación al Alcance
GOBIERNO	<ul style="list-style-type: none"> • Las sub-causas basales que no se incluirán en este trabajo, se refieren a la deficiente capacidad de inspección periódica de organismos de medio ambiente, que ejecutan un plan de revisión periódica de establecimientos educacionales y solo se basan en la información de las 11 estaciones medioambientales, la cual posee información general de ciertas zonas de la región metropolitana y no el detalle de cada zona. • Otra causa relacionada con la anterior es la carencia de información desagregada y desplegada de parámetros medioambientales para toda la región metropolitana. El trabajo propuesto se encargará más bien de desplegar información medio ambiental acotada, desagregada, pero focalizada de un número reducido de estaciones de monitoreo de jardines ambientales en una zona específica de Santiago y no una red de monitoreo global.
PADRES Y APODERADOS	<ul style="list-style-type: none"> • Las causas basales relacionadas con la valoración e importancia que le dan los padres y apoderados a la información medioambiental presente en los jardines infantiles, quedan excluidas de este trabajo, ya que se estima que son situaciones particulares y con gran diversidad, por lo tanto, la solución se enfocará más bien en proporcionar un elemento adicional de información que pueda ser utilizada ya sea como elemento de alerta para tomar medidas o como elemento decisión, más que como una herramienta o metodología de cambio comportamiento cultural.
SOSTENEDORES	<ul style="list-style-type: none"> • Las causas de falta de planes de mitigación y acciones correctivas, junto con la adopción de protocolos de inspección particulares que encomienden los sostenedores, quedan excluidos por tratarse de decisiones de uso interno. Mas bien el proyecto proporciona información preventiva y de alerta a los interesados, los cuales deberán valorar en forma personal su relevancia.
TECNOLOGICAS	<ul style="list-style-type: none"> • Queda excluido de este trabajo el análisis y calidad de los datos obtenidos por el sistema de percepción, ya que se utilizará dispositivos de bajo costo que no permiten una comparación verosímil con los sistemas de medición utilizados en las estaciones de monitoreo ambiental.

TABLA 7: Limitaciones del alcance del proyecto (Fuente: elaboración propia)

3.4.3 Supuestos

Se contemplan los siguientes supuestos:

- Los interesados de este trabajo son los padres, apoderados, sostenedores, y autoridad medioambiental y entornos académicos de IoT.
- Los modelos, procesos y herramientas, serán validados por entornos académicos.
- Este informe contempla entregar ejemplares con los modelos, arquitecturas, y algoritmos para uso académico.
- La implementación del proyecto está a nivel de prototipo y no de producto comercial
- El foco del proyecto corresponde a la plataforma de monitoreo y no en la precisión y calidad de las mediciones obtenidas

4 Marco Teórico

Dado que la mirada central de esta investigación es la telemetría como servicio basado en internet de las cosas como medio tecnológico de monitoreo del medioambiente, será necesario plantear algunos parámetros que sirvan de ejes conceptuales sobre los que apoyar la lectura interpretativa del núcleo de la investigación. En este sentido, el marco teórico, se referirá a conceptos fundamentales de internet de las cosas, arquitectura de sistemas para internet de las cosas, plataformas de software y hardware necesarios para implementar iniciativas de este tipo.

4.1 Internet de las Cosas

Para comenzar, se revisará el paradigma de Internet de las cosas (IoT, por sus siglas en inglés), este concepto no es una idea nueva. A principios de los años noventa, Mark Weiser, director científico del Xerox Palo Alto Research Center, introdujo el concepto de computación ubicua, que abogaba por un futuro en el que la computación desaparecería de nuestra vista, es decir, que formaría parte integral de nuestra vida diaria y resultaría transparente para nosotros. La primera aparición de la IoT como hoy se conoce procede del Instituto Tecnológico de Massachusetts (MIT), en concreto del trabajo del Auto-ID Center. Este grupo, fundado en 1999, realizaba investigaciones en el campo de la identificación por radiofrecuencia en red (RFID) y las tecnologías de sensores emergentes. Fue ahí donde se empezó a formar el actual concepto que tenemos de telemetría remota e IoT.

Formalmente la utilización del término “Internet de las cosas”, se debe al británico Kevin Ashton, cofundador del Auto-ID Center nombrado anteriormente, quien acuñó por primera vez el nombre de Internet de las Cosas gracias a un artículo publicado en el RFID Journal en Julio de 2009, **(Ashton, 2009)**. En dicho artículo Kevin introdujo el concepto de conectar todas las cosas que nos rodean con la finalidad de poder contarlas, posicionarlas, conocer su estado en cualquier momento, así como aportarnos información sobre el entorno que les rodea.

El paradigma de Internet de las cosas (IoT) considera la presencia generalizada en el entorno de una variedad de cosas y objetos que a través de conexiones inalámbricas y por cable y esquemas de direccionamiento únicos pueden interactuar entre sí y cooperar con otras cosas y objetos para crear nuevas aplicaciones y servicios y

alcanzar objetivos comunes. En este contexto, los desafíos para crear aplicaciones y servicios inteligentes son más grandes. Un mundo donde lo real, lo digital y lo virtual convergen para crear entornos inteligentes que hacen que la energía, el transporte, las ciudades y muchas otras áreas sean más inteligentes. El objetivo de Internet de las Cosas es permitir que las cosas se conecten en cualquier momento, en cualquier lugar, con cualquier persona y cualquier persona que idealmente utilice cualquier ruta y red y cualquier servicio. Internet de las cosas es una nueva revolución de Internet. A continuación, se presenta en la figura nº4, un esquema referente a la diversidad de dispositivos que participan en un sistema basado en internet de las cosas, los cuales tienen elemento en común, la convergencia al protocolo de internet IP.

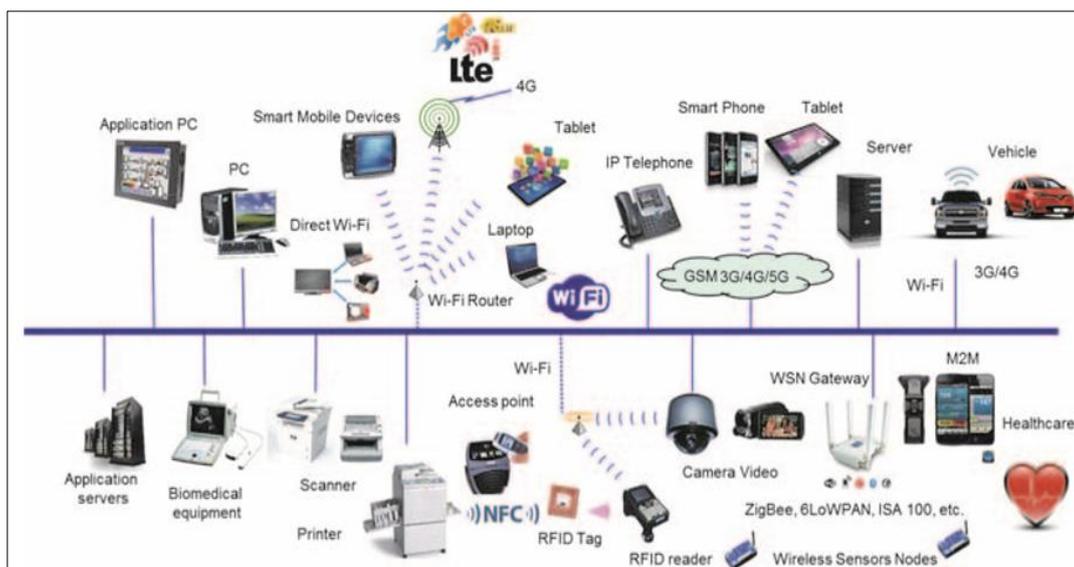


FIGURA 4: Convergencia del Protocolo de Internet IP
(Fuente: Vermesan, O., & Friess, P. 2013)

Por otro lado, los objetos se hacen reconocibles y obtienen inteligencia haciendo o permitiendo decisiones relacionadas con el contexto gracias al hecho de que pueden comunicar información sobre ellos mismos. Pueden acceder a información que ha sido agregada por otras cosas, o pueden ser componentes de servicios complejos. Esta transformación es concordante con la aparición de capacidades de computación en la nube y la transición de Internet hacia IPv6 con una capacidad de direccionamiento casi ilimitada. Los nuevos tipos de aplicaciones pueden involucrar el vehículo eléctrico y la casa inteligente, donde los electrodomésticos y servicios que proporcionan notificaciones, seguridad, ahorro de energía, automatización, telecomunicaciones, computadoras y entretenimiento se integran en un solo

ecosistema con una interfaz de usuario compartida. Obviamente, no todo estará en su lugar de inmediato. En el futuro, los servicios de computación, almacenamiento y comunicación estarán muy extendidos y distribuidos: personas, objetos inteligentes, máquinas, las plataformas y el espacio circundante (por ejemplo, con sensores inalámbricos y cableados, dispositivos máquina a máquina M2M, etiquetas RFID, etc.) crearán un conjunto común de recursos altamente descentralizado (hasta el mismo borde de la "red") interconectados por la red de redes dinámicas. El "lenguaje de comunicación" estará basado en protocolos interoperables, que operan en entornos y plataformas heterogéneos. IoT en este contexto es un término genérico y todos los objetos pueden desempeñar un papel activo gracias a su conexión a Internet mediante la creación de entornos inteligentes. Esta poderosa herramienta de comunicación proporciona acceso a información, medios y servicios a través de conexiones de banda ancha por cable e inalámbricas. **(Vermesan & Friess, 2013).**

4.1.1 Arquitecturas de Internet de las Cosas

Etimológicamente, la palabra arquitectura procede del griego y es la conjunción de dos palabras: arjé, el principal, el que manda, el principio, el primero, y tekton, que significa construir, edificar. El arquitecto es, por tanto, el primero de entre aquellos que realizan la tarea de construir. Por un lado, es el que define las bases, los principios. Por otro, es el que dirige, el que manda en la actividad constructiva. La arquitectura, como actividad, como oficio, es el conocimiento y la práctica que permiten llevar a término estas funciones: determinar aquello que es básico para construir un edificio y también tener la responsabilidad de llevar a término algo determinado **(Morales & De Sola, 2001).**

Este concepto de la Arquitectura tradicional maneja cierta similitud con el concepto de Ingeniería del Software, así como Pressman lo enuncia en su libro: *“Cuando se piensa en la arquitectura de una construcción, llegan a la mente muchos atributos distintos. En el nivel más sencillo, se considera la forma general de la estructura física. Pero, en realidad, la arquitectura es mucho más que eso. Es la manera en la que los distintos componentes del edificio se integran para formar un todo cohesivo. Es la forma en la que la construcción se adapta a su ambiente y se integra a los demás edificios en la vecindad. Es el grado en el que el edificio cumple con su propósito y en el que satisface las necesidades del propietario”* **(Pressman, 2001).**

Pero ¿qué es la arquitectura de software? La arquitectura de software de un sistema o programa de cómputo es la estructura o estructuras del sistema, que comprenden componentes de software, las propiedades externas visibles de esos componentes, y las relaciones entre ellos.

En muchas ocasiones la arquitectura tradicional tiene ciertas fallas ya que no se consideran ciertos escenarios o factores, del mismo modo, en la arquitectura de software no tomar algunas decisiones pueden llevar al fracaso a una solución de software. En cualquier caso, a lo largo de la historia de la ingeniería de software se ha identificado que, a pesar de que la mayoría de los sistemas son distintos entre sí, existen algunas similitudes entre las arquitecturas de los mismos. Estas similitudes suelen seguir lo que se conoce como patrón o estilo arquitectónico que captan la esencia de una arquitectura que se usó en diferentes sistemas de software.

A continuación, en la figura nº5, se presenta un estilo arquitectónico clásico de capas, que puede ser utilizado por diversos sistemas de información.

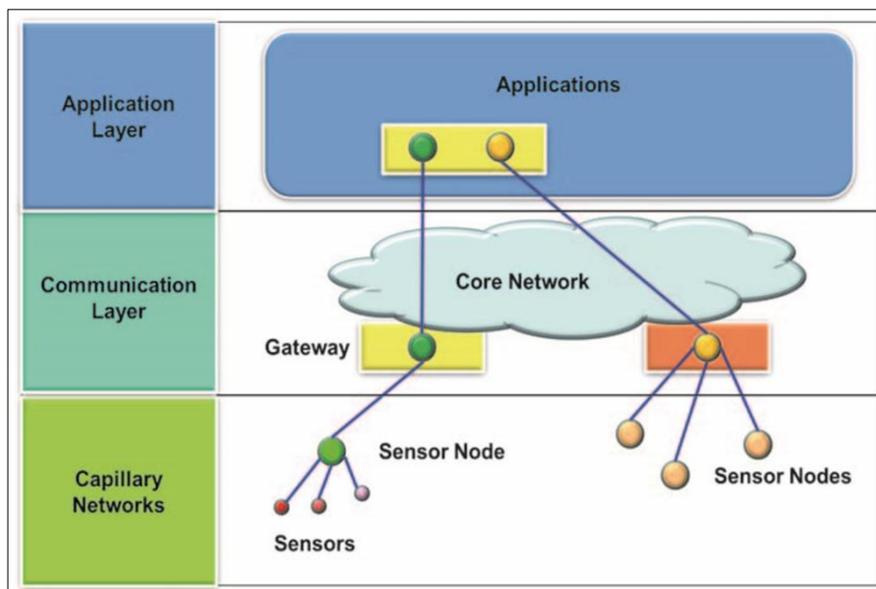


FIGURA 5: Arquitectura de capas o layer
(Fuente: Vermesan, O., & Friess, P. 2013)

El estilo arquitectónico de capas o layer (**Coulouris, 2012**), en términos generales, concibe una organización de componentes basados en niveles con responsabilidades definidas. Específicamente un nivel inferior sirve de recursos o información al nivel superior que actúa como cliente y este a su vez sirve de recursos e información a los niveles superiores. Lo interesante es que cada nivel solo interactúa con el nivel

contiguo, acotando las responsabilidades y simplificando la estructura. Este estilo, es utilizado en la organización de sistemas operativos y la organización de redes de comunicación (modelo OSI).

Como ejemplo particular de arquitectura orientada a internet de las cosas, basado en arquitectura por capas, se puede mencionar el modelo descrito por World Forum referenciado en **(Cazila & Junco, 2015)**, en donde se presenta una visión global de todos los componentes que participan en una arquitectura de capas para IoT. Este modelo de arquitectura para IoT dispone que los niveles 1, 2 y 3 estén totalmente alineados con los primeros niveles modelo de 7 capas de OSI, especialmente los físicos y de conectividad, equivalentes a los niveles de Sensor/Actuador, Gateway y Network. Los niveles 4, 5 y 6 comprenden lo que tradicionalmente se denomina capa de aplicaciones, donde presentamos la aplicación de Gestión o Management, sistemas Analíticos y almacenamiento o sistemas de Big Data. El modelo explicita los niveles de integración de los datos. Esto es, define un nivel de adaptación de información que haga homogénea la información para su tratamiento en los sistemas centralizados, tanto sea para una gestión proactiva (analíticos) o reactiva, gestión tradicional, o tal vez simplemente para ser almacenada en el Data Center. El nivel 7, muestra la necesidad de interacción entre los procesos, personas y negocios, que preferimos dejar fuera de un modelo de capas que se enlazan unas con otras. Pero que si consideramos de suma importancia para que los proyectos de IoT puedan llegar a buen puerto.

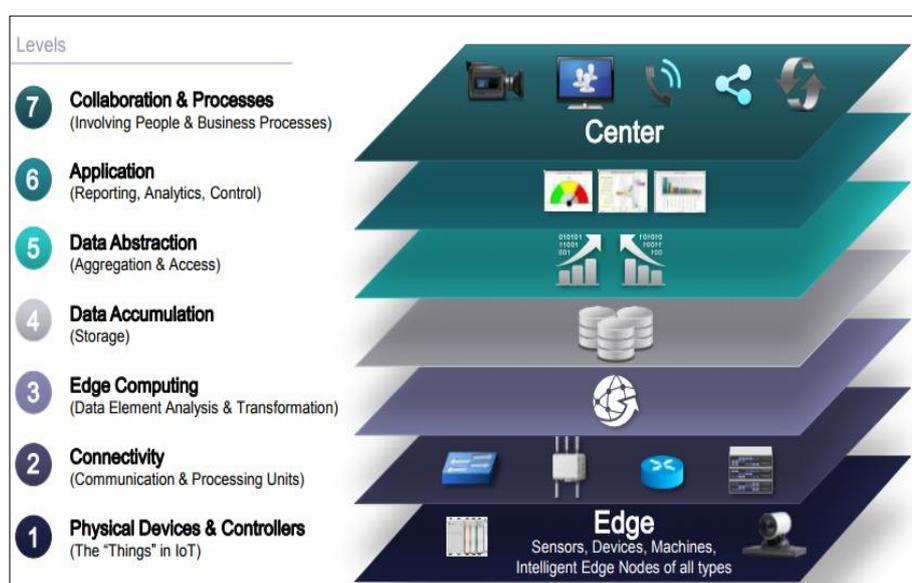


FIGURA 6: Modelo de arquitectura genérica para una aplicación de IoT
(Fuente: Cisco Systems. 2013)

4.1.2 Plataformas de hardware y software

Un requisito importante en el diseño de dispositivos IoT está relacionado con el concepto de "energía mínima". Este requisito también puede extenderse al enfoque de diseño general en la creación de componentes de hardware y software de IoT. Los dispositivos IoT van desde sensores muy livianos accionados por microcontroladores (MCU) de 8 bits hasta dispositivos equipados con procesadores de 32 bits más potentes pero eficientes desde el punto de vista energético. Estos dispositivos se basan en diferentes arquitecturas, como x86, MSP430, ARM7, Atmel AVR, Cortex-M0, Cortex-M3, Cortex-M4. Un sistema operativo IoT debe ejecutarse en diversas plataformas, incluidos los dispositivos integrados y las PC comunes, y para admitir múltiples controladores e interfaces (**Gardasević, 2016**).

4.1.2.1 Dispositivos de procesamiento

Cada nodo que participa en un sistema IoT, debe disponer de algún tipo de unidad de procesamiento que permita transformar señales y procesar flujos de datos. Formalmente este tipo de dispositivos se dispone en una placa reducida (en inglés: Single Board Computer o SBC) que técnicamente es una computadora completa en un sólo circuito. El diseño se centra en un solo microprocesador con la RAM, E/S y todas las demás características de un computador funcional en una sola tarjeta, que suele ser de tamaño reducido, y que tiene todo lo necesario en la placa base. Hay plataformas de hardware libre que ofrecen productos comerciales con un hardware muy potente en este tipo de placas a un precio muy asequible. Gracias a su bajo coste y sus infinitas posibilidades se han convertido en una herramienta perfecta para el mundo del IoT y los sensores.

4.1.2.2 Sensores

Según, (**Duarte, 2016**). Un sensor es un objeto capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en otra magnitud, normalmente una variable eléctrica, para que se pueda cuantificar y manipular. Un sensor es un dispositivo que convierte una forma de energía en otra, aprovechando una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. La variación de la magnitud que se obtiene en el sensor

es un valor directo de la variable a medir, pero los sistemas de control no trabajan con estas señales, sino con tensión o intensidad por lo que, con ayuda de un transductor, se asocia a una variación de voltaje o intensidad. El transductor, por tanto, suele incluir al sensor. Normalmente estos dispositivos se realizan con componentes pasivos que varían su magnitud en función de alguna variable y otros componentes activos. En cuanto a las variables de instrumentación, éstas pueden ser, por ejemplo: intensidad lumínica, temperatura, distancia, aceleración, inclinación, presión, desplazamiento, fuerza, torsión, humedad, movimiento, pH, concentraciones químicas, etc.

Otro elemento importante que hay que considerar en un sistema de sensores y actuadores IoT, corresponde al concepto de sensor virtual que puede considerarse como un producto de información espacial, temporal, el cual proporciona una estructura de datos para la programación. Formalmente los sensores y actuadores virtuales son una abstracción de programación que simplifica el desarrollo de aplicaciones descentralizadas (Vermesan, & Friess, 2013). Los datos adquiridos por una serie de sensores se pueden recopilar, procesar de acuerdo con una función de agregación proporcionada por la aplicación y luego percibirse como la lectura de un único sensor virtual. Dualmente, un actuador virtual proporciona un único punto de entrada para distribuir comandos a un conjunto de nodos reales de actuador.

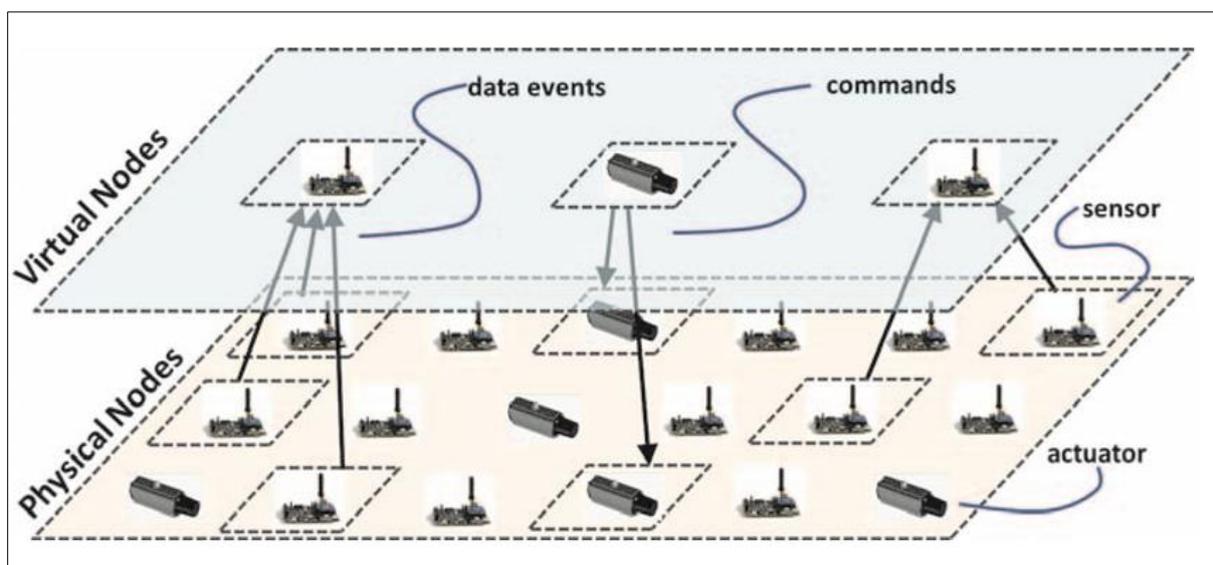


FIGURA 7: Esquema de interacción de sensores y actuadores físicos y virtuales (Fuente: Vermesan, O., & Friess, P. 2013)

La información recopilada del mundo físico y virtualizada en combinación con los recursos y servicios existentes en la red, facilitan métodos mejorados para obtener inteligencia empresarial, lo que permite la construcción de nuevos tipos de aplicaciones y servicios de entrada frontal que pueden ayudar a la evolución a las organizaciones y personas a utilizar servicios de Internet y sus aplicaciones en sus actividades de datos.

4.1.2.3 Protocolos de comunicación

Respecto a los conceptos relacionados con los protocolos de comunicación, también es necesario realizar una distinción entre los protocolos de comunicación de red, llamados de bajo nivel, que se relacionan con las capas inferiores del modelo OSI (capa física, enlace y red) y los protocolos orientados a la comunicación de alto nivel (capa de transporte, sesión, presentación y aplicación), que se encargan de modelar los datos con un nivel de abstracción superior. Según lo anterior, se describe en primer término la comunicación de bajo nivel que considera evidentemente alternativas inalámbricas por la naturaleza del paradigma de internet de las cosas.

En **(Fabregas, 2016)**, se describe la existencia de diferentes tecnologías de comunicación inalámbricas ampliamente conocidas, como Wi-Fi, Bluetooth y redes de datos móviles como GPRS/3G que ya se encuentran integradas en productos y aplicaciones del Internet de las Cosas. Sin embargo, la escalabilidad requerida para desplegar potencialmente miles de equipos, en escenarios tan diversos como ciudades o entornos rurales, y con restricciones como el consumo de energía requerido para operar años con baterías, ha motivado nuevas tecnologías de telecomunicación para el Internet de las Cosas

Nuevas tecnologías emergentes como Sigfox y LoRa plantean una solución a esta necesidad, permitiendo enlaces inalámbricos con kilómetros de rango, con un consumo moderado de energía. La figura nº8 resume las tecnologías inalámbricas más relevantes.

Un inconveniente de las tecnologías mencionadas anteriormente es su carácter privativo, ya que protocolos como LoRa y Sigfox requieren bien el pago de licencias y regalías o restringen su utilización a módulos específicos de hardware. Protocolos como IEEE 802.15.4 (sobre el cual se construyen protocolos como ZigBee, presentan una alternativa viable a estas restricciones, al ser un protocolo abierto y ampliamente

implementado por varios fabricantes de chips y módulos de radio, lo que promete una mayor interoperabilidad entre aplicaciones y productos. El protocolo IEEE 802.15.4 define la capa física y de acceso al medio para radios de baja potencia y baja tasa de datos, para paquetes de datos con una MTU de 128 bytes.

El estándar abierto 6LoWPAN definido en RFC 6282, permite conectar redes IEEE 802.15.4 a redes IP, protocolos de capa de transporte como TCP y UDP son implementados sobre 6LoWPAN, lo que facilita la implementación de productos y servicios que interactúen de manera transparente con la mayoría de los equipos y soluciones existentes.

WIRELESS TECHNOLOGIES AT A GLANCE					
Technology	Frequency	Data rate	Range	Power	Cost
2G/3G	Cellular bands	10 Mb/s	Several km	High	High
802.15.4	2.4 GHz	250 kb/s	100 m	Low	Low
Bluetooth	2.4 GHz	1, 2, 3 Mb/s	100 m	Low	Low
LoRa	< 1 GHz	<50 kb/s	2-5 km	Low	Medium
LTE Cat 0/1	Cellular bands	1-10 Mb/s	Several km	Medium	High
NB-IoT	Cellular bands	0.1-1 Mb/s	Several km	Medium	High
SIGFOX	<1 GHz	Very low	Several km	Low	Medium
Weightless	<1 GHz	0.1-24 Mb/s	Several km	Low	Low
Wi-Fi (11n/h)	2.4, 5, <1 GHz	0.1-1 Mb/s	Several km	Medium	Low
WirelessHART	2.4 GHz	250 kb/s	100 m	Medium	Medium
ZigBee	2.4 GHz	250 kb/s	100 m	Low	Medium
Z-Wave	908.42 MHz	40 kb/s	30 m	Low	Medium

FIGURA 8: Tecnologías de telecomunicación inalámbricas
(Fuente: “Twelve wireless options for IoT/M2M: diversity or dilemma”,
Electronic Design)

En cuanto a los protocolos de comunicación orientados a las capas superiores, se puede mencionar por ejemplo que el popular protocolo HTTP que pertenece a la capa de aplicaciones, no es la opción más viable para un entorno IoT ya que requiere de muchos recursos y ancho de banda. Se debe tener en cuenta que los dispositivos que usen este protocolo tendrán limitaciones en la duración de sus baterías, memoria, ancho de banda, entre otros. Por tanto, la selección adecuada del protocolo de comunicación incidirá tanto en el rendimiento individual de cada dispositivo, y en todo el sistema IoT. El protocolo MQTT está diseñado específicamente para redes de dispositivos Máquina a Máquina (M2M) y aplicaciones móviles con restricciones de

memoria, batería, red, entre otras. Sus mensajes se transmiten en formato binario y trabaja con el modelo publicador/suscriptor (**HiveMQ, 2015**), el cual funciona como una vía de comunicación de uno a varios clientes, lo cual es útil en IoT para el monitoreo de dispositivos remotos. Esto no impide establecer una conexión uno a uno (útil en caso de control de dispositivos en IoT) o de varios a uno. MQTT provee tres niveles de calidad del servicio (QoS) para la entrega de mensajes; nivel 0 (Máximo una vez entregado), nivel 1 (Al menos una vez entregado) y nivel 2 (Exactamente una vez entregado). Los clientes MQTT se comunican mediante tópicos, los cuales funcionan como un canal de comunicación privado. Existe incluso una variante de MQTT destinada a redes de sensores con recursos muy restringidos, la cual es definida en (**Stanford-Clark & Truong, 2015**). Existen varios intermediarios que implementan el protocolo MQTT, siendo el más famoso de ellos el bróker open source Mosca y Mosquitto.

El Protocolo de Mensajería de Aplicaciones Web (**WAMP, 2015**) provee dos modos de comunicación: llamadas de procedimientos remotos (RPC) y publicador/subscriptor (PUB/SUB). Aunque su medio de transmisión es por defecto WebSockets, es posible utilizar medios diferentes, donde los datos son transmitidos en texto plano o en binario. Las llamadas de procedimientos remotos son útiles para comunicaciones de uno a uno. Por otro lado, el modo publicador/suscriptor es utilizado en comunicaciones de uno a varios o de varios a uno. La comunicación entre clientes se establece mediante tópicos representados mediante URIs. WAMP es capaz de comunicarse de forma nativa con un navegador web sin necesidad de intermediarios. Por ende, resulta sencillo servir datos en tiempo real directamente en la interfaz de un usuario conectado desde algún navegador.

El Protocolo de Aplicación Restringida (CoAP) es utilizado también en dispositivos y redes cuyos recursos son limitados. Funciona bajo UDP con el modelo REST, y además utiliza el modo publicador/suscriptor. Difiere de HTTP ya que CoAP no conlleva un alto consumo de recursos y descarta las cabeceras adicionales no necesarias en el entorno IoT. A pesar de sus similitudes con HTTP, CoAP no puede comunicarse nativamente con un navegador web, por lo cual se debe utilizar un gateway como intermediario que traduzca CoAP a HTTP.

4.1.2.4 Gateway y Middleware

La premisa básica de IoT, es que se requiere la interoperabilidad de dispositivos basados en estándares abiertos para cumplir con la visión de IoT. De acuerdo con esto, es necesario incluir una capa intermedia que interconecte los diferentes estándares. En **(Cazila, & Junco, 2015)**, se describen dos capas de interconexión. La primera capa, está el nivel de adaptación o gateway que es crucial en la integración de los sensores o dispositivos de medición a la red IP.

Básicamente hay, al menos, dos tipos de conexión; sensores que se conectan a la red IP de forma directa y otros que se conectan a la red mediante un gateway físico. Aunque muchos de los sensores tendrán una interface propietaria o autónoma de fábrica, no se puede dejar de separar, aunque sea en forma teórica, estos niveles. Esta separación va a reforzar el diseño y desarrollo de nuevas ideas con más libertad y facilidad.

Es fundamental a la hora de hablar de sensores para IoT, definir cómo éstos dispositivos se van a conectar a la red IP de manera segura, con calidad de servicios garantizada y respetando capacidades protocolares adecuadas para la ocasión. Esta capa de interconexión que va a estar dada por la funcionalidad de gateway va a hacer posible vincular la medición de un parámetro determinado producido por un sensor a una consola de gestión inteligente a través de una red de datos determinada, (muy posiblemente IPv6), pudiendo la misma ser una red privada o una pública. En la implementación de IoT sobre redes existentes, el tráfico de red se incrementará sustancialmente, es de esperar que 80% del tráfico provenga de pequeñas piezas de información que viajan desde el Sensor a través de la red IP hacia el Data Center o el Cloud. Este no va a ser voluminoso si lo miramos de manera individual, pero al agruparlo y multiplicarlo por decenas de miles puede representar un desafío crucial a la hora del diseño y la implementación del Datacenter, de allí que se desarrollará más adelante el termino Big Data para tener mayor claridad sobre este punto.

Podemos decir que esta capa es funcional a la capa de red o al sensor. Esto significa que uno podría implementar la funcionalidad de gateway tanto en un sensor como en el componente más próximo de la red, sin necesidad de requerir de un elemento físicamente independiente adicional en el diseño de la solución.

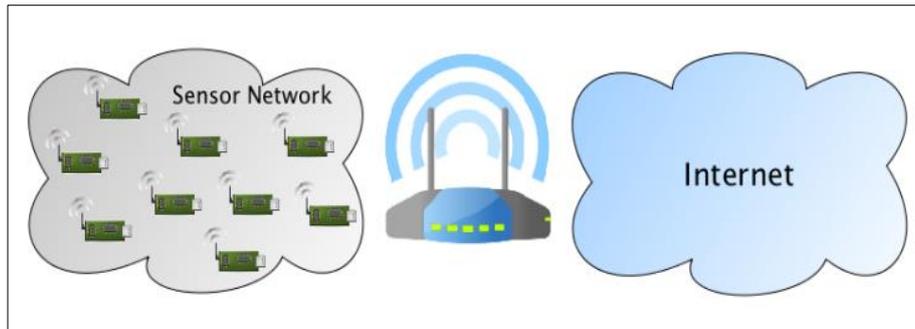


FIGURA 9: Esquema de funcionalidad de Gateway
 (Fuente: Paper: The design of the Gateway for the Cloud of Things, Petrolo R., Morabito R., Loscri V., Mitton N., 2016)

Un segundo nivel de interconexión se relaciona con una capa lógica específicamente la integración entre servicios y aplicaciones que puedan conversar con los niveles inferiores y superiores en el flujo de datos. Aquí cobra relevancia el concepto de middleware.

Middleware es un término acuñado por Lewandowski en **(Lewandowski, 1998)**, que representa una capa de software ubicada entre la capa más alta (usuarios y aplicaciones), y la más baja (sistemas operativos y mecanismos de comunicación básicos). Provee un mismo interfaz para todas las aplicaciones, y por tanto de un mismo medio de comunicación entre diferentes aplicaciones. Al mismo tiempo también oculta las diferencias en hardware, lenguajes de programación, plataformas de comunicación, y sistemas operativos, facilitando el desarrollo de sistemas complejos con diferentes tecnologías y arquitecturas. En particular las aplicaciones de internet de las cosas, posee software distribuido que permiten facilitar la interacción cliente y servidor, que se manifiesta como middleware. Este middleware representa una capa transparente de acceso a servicios y recursos no locales distribuido a través de una red.

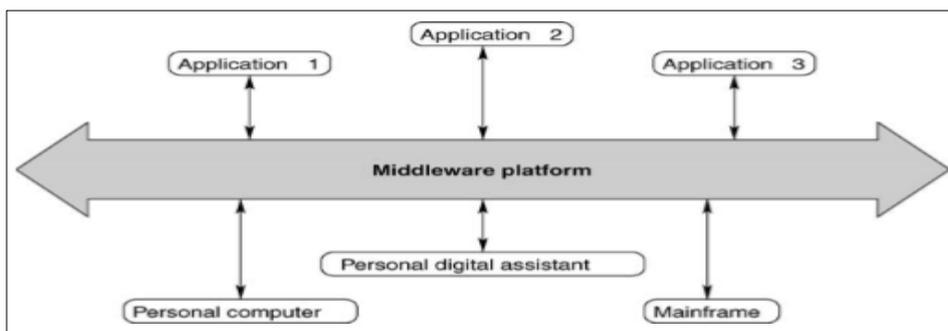


FIGURA 10: Middleware como infraestructura de sistemas distribuidos
 (Fuente: Libro Distributed System Architecture, Arno Puder. 2005)

Ejemplo de Middleware son variados, van desde tecnologías de bajo nivel como pipes, socket tcp-ip, métodos de invocación remota (RPC y RMI), objetos remotos COM y DCOM, tecnología CORBA, hasta tecnologías más de alto nivel como protocolos http, servicios web SOAP, REST, Ajax y WebSockets.

En particular la tecnología web se ha masificado como medio de comunicación de aplicaciones, sin embargo, la web no nació para ser dinámica. Su objetivo era mostrar ficheros de texto de uno en uno. El ciclo web originario partía de un usuario que solicitaba una página web desde el navegador, obtenía una respuesta del servidor, y el mismo navegador mostraba la respuesta que había solicitado. Poco a poco, los desarrolladores fueron creando tecnologías, tales como el JavaScript, para ampliar lo que una página puede hacer, prácticamente desde sus orígenes.

El paso más cercano a los WebSockets lo realizó AJAX. Su gran aportación fue que permitió a los desarrolladores solicitar datos desde un servidor para un elemento concreto de una página web sin tener que actualizar toda la página. Todo esto, además, lo realiza de forma asíncrona, con lo cual la comunicación es mucho más interactiva y fluida. El problema de AJAX es que la comunicación es unidireccional, es decir, solo va del cliente al servidor. Si queremos una comunicación completamente bidireccional, tendremos que hacer uso de los WebSockets, que nos permiten tener una comunicación bidireccional y en tiempo real.

WebSockets es una tecnología que proporciona un canal de comunicación full-dúplex (bidireccional) sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse en cualquier aplicación cliente/servidor. Como las conexiones TCP ordinarias sobre puertos distintos al 80 normalmente son bloqueadas por los administradores de redes, el uso de esta tecnología proporciona una solución a este tipo de limitaciones, ya que provee una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSockets sobre un único puerto TCP.

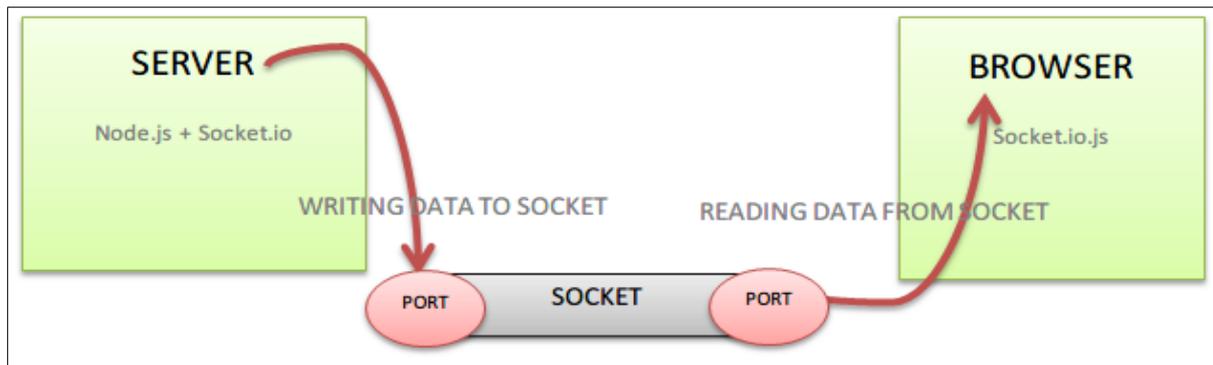


FIGURA 11: Esquema de WebSockets
(Fuente: Libro Distributed System Architecture, Arno Puder. 2005)

4.1.2.3 Plataformas de Analítica

La gestión de datos es un aspecto crucial en Internet de las cosas. Al considerar un mundo de objetos interconectados e intercambiar constantemente todo tipo de información, el volumen de los datos generados y los procesos involucrados en el manejo de esos datos se vuelven críticos

De acuerdo con esto, se requiere como primer elemento para realizar un proceso analítico una etapa de almacenamiento que permita operar con grandes cantidades de información, las alternativas de bases de datos relacionales podrían no satisfacer todas las necesidades requeridas (**McCreary, 2014**). Es ahí donde las bases de datos NoSQL entran en acción debido a la flexibilidad que proporcionan al momento de almacenar y gestionar los datos, siendo esto muy importante en el ámbito de IoT. Este tipo de bases de datos proporcionan la facilidad al momento de realizar consultas, ya que posee un conjunto de funciones con algoritmos lo suficientemente aptos para devolver los datos solicitados en consultas no tan complejas. A pesar de ser un almacén de documentos también permite realizar consultas complejas, distinguiéndose de otras bases de datos y convirtiéndose en una opción muy potente. Proporciona, además una lista cada vez mayor de características importantes (**Chodorow, 2013**) como indexación secundaria, única, compuesta, geoespacial y de texto completo. El teorema CAP (Consistencia-Disponibilidad-Tolerancia de particionado) (**Gilberth, 2012**) define que una base de datos sólo puede cumplir como máximo dos de sus apartados al mismo tiempo. Consistencia se refiere a que todos los clientes de la base de datos son capaces de ver los mismos datos, incluso con actualizaciones concurrentes. La disponibilidad asegura que todos los clientes sean

capaces de acceder a versiones posiblemente distintas de los datos, pero siempre los tendrán a su disposición. Tolerancia de particionado se refiere a la facilidad de la base de datos para desplegarse en distintos servidores. En tal sentido, las bases de datos NoSQL asegura consistencia de sus datos y tolerancia de particionado, importante para IoT, especialmente la consistencia ya que sería un problema si un cliente visualizara en su dashboard (Interfaz de usuario con gráficas representativas de datos históricos y/o en tiempo real.) que un actuador esté apagado, mientras que otro que recibió una versión de ese dato anterior, visualiza que el mismo actuador aún está encendido.

Una segunda etapa necesaria en los procesos de analítica dispuestos en proyectos IoT, se refiere a las técnicas de minería de datos clásicas que permiten extraer información no evidente que se encuentra disponibles en grandes volúmenes de datos, mediante modelos matemáticos derivados de la estadística e inteligencia artificial. A estas técnicas clásicas, se han sumado los nuevos modelos basados en ingeniería social y cognitiva, que buscan patrones de comportamiento basados en la propia información que brindan los usuarios por la interacción con las redes.

4.1.2.4 Plataforma de despliegue

Uno de los componentes esenciales de un sistema basado en Internet de las Cosas (IoT) es disponer de la capacidad de desplegar la información capturada en las capas inferiores con el fin de ponerlas a disposición de las capas superiores manifestadas en aplicaciones o servicios. Las alternativas disponibles son variadas, desde las infraestructuras clásicas compuestas por un servidor o conjunto de servidores interconectados que involucran hardware y software para ofrecer servicios de monitoreo y control (**Campoverde, Hernández, Mazón, 2015**), hasta los modelos actuales de Cloud Computing basados en la flexibilidad y virtualización.

En Cloud Computing existen dos aspectos de vital importancia a considerar, el primero se refiere al modelo de despliegue y el segundo se refiere al modelo del servicio a implementar (**Sosinsky, 2011**). Entre los tres principales modelos de despliegue se encuentran; la nube pública, la cual se caracteriza por su libre acceso desde cualquier parte del mundo con posibles restricciones; la nube privada, la cual comúnmente se implementa dentro de las instalaciones de una empresa y es de su uso exclusivo; y la nube híbrida, la cual es una combinación de una nube pública y privada al mismo

tiempo. Debido a que IoT está obviamente ligado a internet, una nube pública es la opción viable para su implementación, si lo que se desea es una alternativa on-premise, se pierde el concepto IoT y la solución viable en ese caso sería M2M (Machine-to-Machine). Con respecto a los modelos de servicio, las posibilidades son: SaaS (software como servicio), PaaS (plataforma como servicio) e IaaS (infraestructura como servicio). SaaS ofrece de forma predefinida una infraestructura, plataforma y software de aplicación listo para su uso, en donde todo aspecto de la nube es abstracto para el usuario; algunos ejemplos de SaaS son Google Docs, Gmail y Dropbox. PaaS ofrece una infraestructura y plataforma predefinida a partir de las cuales el usuario puede implementar su aplicación mediante herramientas especificadas por el proveedor del servicio (bases de datos, lenguajes de programación, entre otros), en este modelo únicamente la infraestructura de la nube es completamente abstracta al usuario. Microsoft Azure, Google Cloud Platform, Bluemix de IBM, Heroku, Sofia2, Xively, son los servicios PaaS más relevantes del mercado en la actualidad. IaaS provee un entorno de virtualización de recursos físicos para que el usuario sea el encargado de definir una infraestructura que se ajuste a sus necesidades, los servicios IaaS más populares son Amazon AWS y Rackspace. En cloud computing la virtualización es algo muy importante, consiste en simular un solo espacio físico como si fueran varios, aprovecha así de forma óptima los recursos del equipo, brinda mayor facilidad de mantenimiento a la infraestructura y abarata costos de operación. Es común utilizar un software especializado para la virtualización, mejor conocido como Hypervisor o Virtual Machine Monitor (VMM), el cual en el mejor de los casos se instala directamente en el equipo sin necesidad de un Sistema Operativo anfitrión.

5 Estado del Arte

El proceso de búsqueda del estado del conocimiento es una actividad relevante en el desarrollo de una investigación académica aplicada. De acuerdo con esto, es necesario determinar dicho estado para sustentar adecuadamente la investigación propuesta.

Para desarrollar la búsqueda se utilizará la metodología denominada mixta que combina un estudio de mercado referente a las iniciativas y proyectos relacionados con la temática de interés y adema una revisión de los trabajos de investigación académica disponibles en la línea de interés. Tanto la revisión de mercado y revisión científica será implementada mediante una “*revisión sistemática*”, que corresponde a un diseño de investigación observacional y retrospectivo, que sintetiza los resultados de múltiples investigaciones primarias. Formalmente la búsqueda sistemática, se define como una manera de evaluar e interpretar toda la investigación disponible, que sea relevante respecto de una interrogante de investigación particular, en un área temática o fenómeno de interés (**Kitchenham, 2004**). La búsqueda sistemática posee una serie de etapas como: *Formulación de la pregunta u objetivo de la revisión, Definición de las fuentes y del método de búsqueda, Definición y aplicación de criterios de selección de los estudios, Análisis crítico de los estudios, Extracción y síntesis de los resultados de los estudios, Conclusiones e inferencias.*

Particularmente, en esta investigación, se utilizará una versión simplificada de la búsqueda sistemática, estableciendo una estrategia basada en una serie de parámetros de búsqueda, que se utilizaran como filtros en los repositorios automáticos. La búsqueda, se implementará utilizando repositorios científicos validados como las revistas indexadas. De acuerdo con esto, se presentan los siguientes parámetros de búsqueda.

Parámetro Búsqueda	Filtro
Repositorios de datos	<ul style="list-style-type: none">• ScienceDirect
Tipo de fuente	<ul style="list-style-type: none">• Artículos publicados en Ingles
Áreas de búsqueda	<ul style="list-style-type: none">• Informática, Computación, Electrónica, Medicina
Año de búsqueda	<ul style="list-style-type: none">• 2015, 2016, 2017
Palabras claves	<ul style="list-style-type: none">• Aplicaciones IoT de bajo costo de telemetría• Medición de variables medioambientales en la ciudad• Métodos de medición remota de gases y contaminación acústica de bajo costo

TABLA 8: Parámetros de búsqueda sistemática (Fuente: elaboración propia)

5.1 Medición de Variables ambientales

El primer estudio consultado que resulta de interés corresponde al desarrollado en **(Kelly, 2016)**, donde se estudia el desempeño del sensor Plantower PMS3003 de bajo costo y se compara con estaciones profesionales de medición de alto costo para ver si el sensor tiene correlación en sus medidas. Este sensor mide material particulado (MP) y posee un costo comercial de 135 U\$. La electrónica de medición está basada en la dispersión de luz que se convierte en concentraciones de PM1, PM2.5 y PM10. El sensor se acopla a una plataforma de hardware que involucra una red de sensores y se basa en un chip Wifi ESP8266. Cada sensor está integrado con una antena inalámbrica y ejecuta un firmware Arduino personalizado. El sensor PMS 3003 mide 5.5 x 4.5 cm y posee un consumo aproximado de 3 W de potencia, incluida la señal inalámbrica.

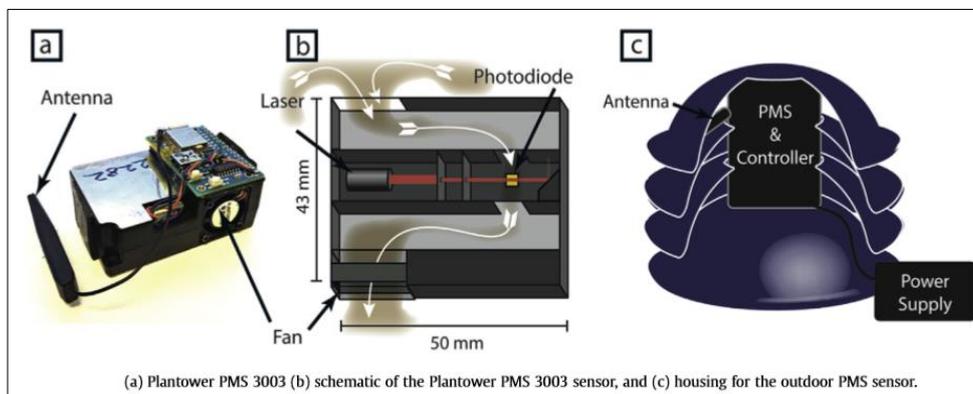


FIGURA12: Sensor Plantower PMS3003

(Fuente: Kelly, 2016)

Otro estudio, relacionado con la medición de condiciones ambientales con sensores de bajo costo, se describe en **(Rogulski, 2017)**. En donde se realiza un estudio de monitoreo ambiental de la ciudad de Varsovia, en la capital de Polonia, que posee un área de 517 km² y solo posee 8 estaciones automáticas. Este estudio presenta el concepto de utilizar dispositivos de medición de PM de bajo costo para aumentar la resolución espacio temporal de las mediciones de calidad del aire que complementan la red de dispositivos de referencia. Se presenta una red prototipo de tales dispositivos y mide las concentraciones de PM10, PM2.5 y Monóxido de Carbono. El sensor está basado en el microcontrolador Arduino, y el sensor de polvo óptico SEN0177, además de sensor de temperatura y humedad y módulo de comunicación.

Varios sensores de bajo costo han sido probados en aplicaciones prácticas y condiciones de laboratorio (**Gao, Cao, Seto, 2016**). Estas pruebas son prometedoras para los sensores de PM, durante el estudio de cuatro días se probó la calidad del sensor Shinyei en la región fuertemente contaminada de China y se estableció una correlación con los instrumentos de investigación profesionales, incluidos los que miden mediante el uso de métodos gravimétricos.

En (**Wang, Li, Jing, Zhang, Jiang, Biswas, 2015**), se presentaron los resultados del análisis de tres sensores de bajo costo: Shinyei PPD42NS, HK-A5 Laser PM2.5/10 y Sharp GP2Y1010AU0F.

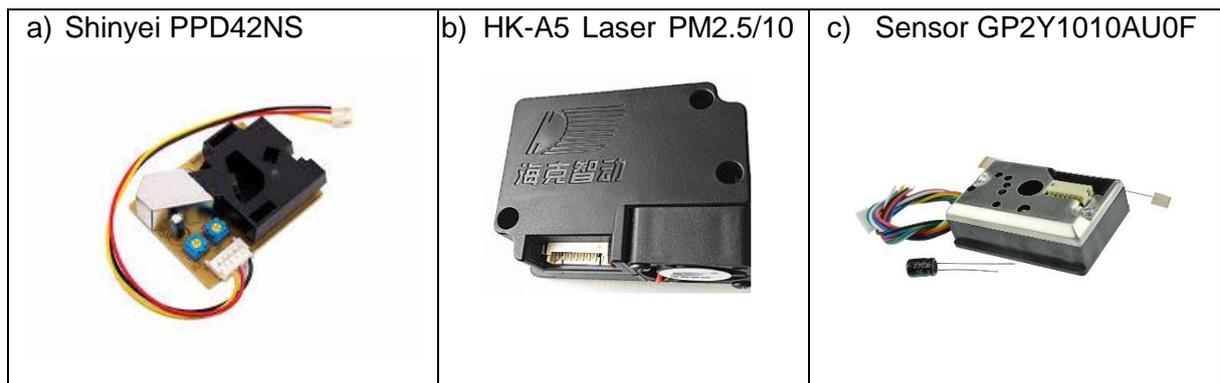


FIGURA13: Sensores comerciales de bajo costo
(Fuente: Gao M, Cao J, Seto E. 2016).

Los resultados obtenidos, se caracterizaron por una alta linealidad con los resultados obtenidos con el uso de equipos profesionales. Los sensores fueron comparados con dos equipos de medición de PM2.5 - DustTrak 8520 TSI y DataRAM personal de Thermo Scientific (costo aproximadamente U\$ 4000). Los autores han demostrado que la correlación de las medidas entre los diferentes tipos de sensores es bastante consistente.

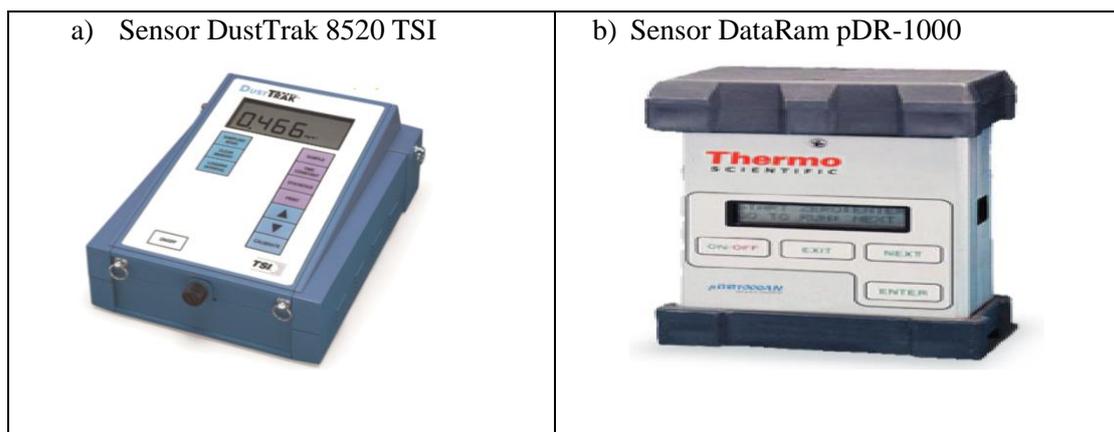


FIGURA14: Sensores Profesionales (Fuente: Gao M, Cao J, Seto E. 2016).

Otro sensor de uso comercial de mediano costo es la unidad AQMesh (**Environmental Instruments Ltd., Reino Unido, www.aqmesh.com**), que es una plataforma estacionaria, descrita en (**Castell, Dauge, Schneider, Vogt, Lerner, Fishbain, Broday, Bartonova, 2017**), donde se utiliza esta tecnología en la monitorización de Oslo, Noruega mediante 24 nodos de AQMesh. Cada nodo mide cuatro componentes gaseosos (CO, NO, NO₂ y O₃) y PM₁₀ y PM_{2.5} mediante refracción. Los nodos AQMesh también miden la temperatura, la humedad relativa y la presión atmosférica. Los nodos AQMesh estándar entregan datos promediados de una hora, pero se pueden configurar para entregar datos promediados de 15 minutos. Un módem GPRS integrado permite la transferencia de datos al servidor de base de datos AQMesh. Los datos pueden luego descargarse de un sitio web dedicado.

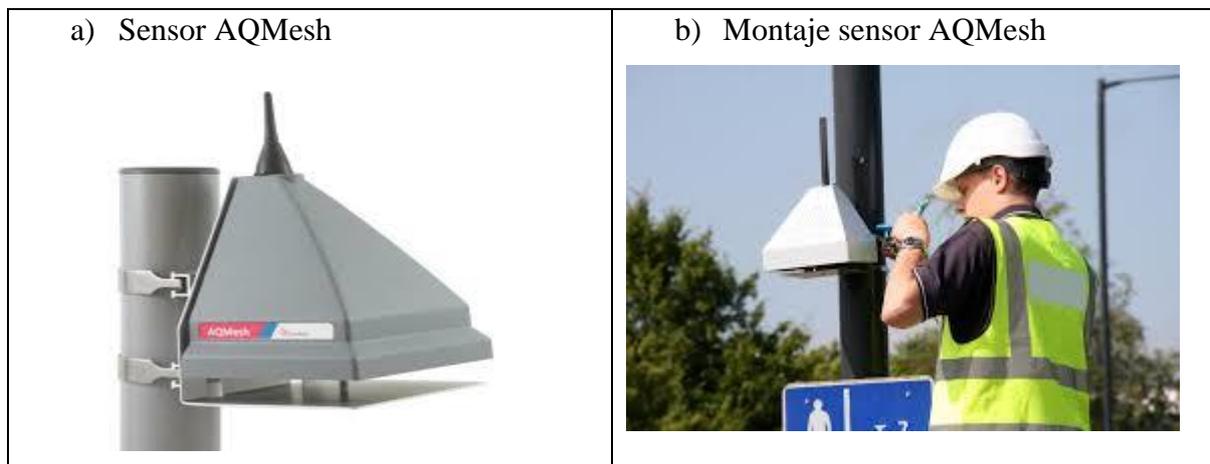


FIGURA15: Sensor comerciales mediano costo AQMesh

(Fuente: Environmental Instruments Ltd., Reino Unido, www.aqmesh.com)

Un cuarto estudio de interés relacionado con la medición con redes de sensores de bajo costo, pero en ambientes interiores, se describe en (**Abraham & Xinrong, 2015**). Aquí, se presenta un sistema de red de sensores inalámbrico de monitoreo de la calidad del aire interior de bajo costo desarrollado con los módulos Arduino, XBee y micro gases. El sistema desarrollado permite recopilar seis parámetros de calidad del aire de diferentes ubicaciones simultáneamente. La arquitectura global del sistema de red de sensores inalámbricos que se emplea se muestra en la figura n^o16. Los principales componentes del sistema incluyen el nodo del sensor, la estación base y la base de datos y el servidor web como se muestra en la figura. La estación base recibe los datos de medición de los nodos del sensor distribuidos periódicamente, luego los datos se envían a un servidor de base de datos para su almacenamiento y

administración. Se puede implementar un servidor web para proporcionar una interfaz web conveniente para que los usuarios accedan a los datos y administren el sistema de red del sensor de forma remota. Este trabajo, está centrado en el desarrollo del nodo del sensor, mientras que los detalles de desarrollo de la estación base, la base de datos y el servidor web, y la interfaz web no se describen en detalle.

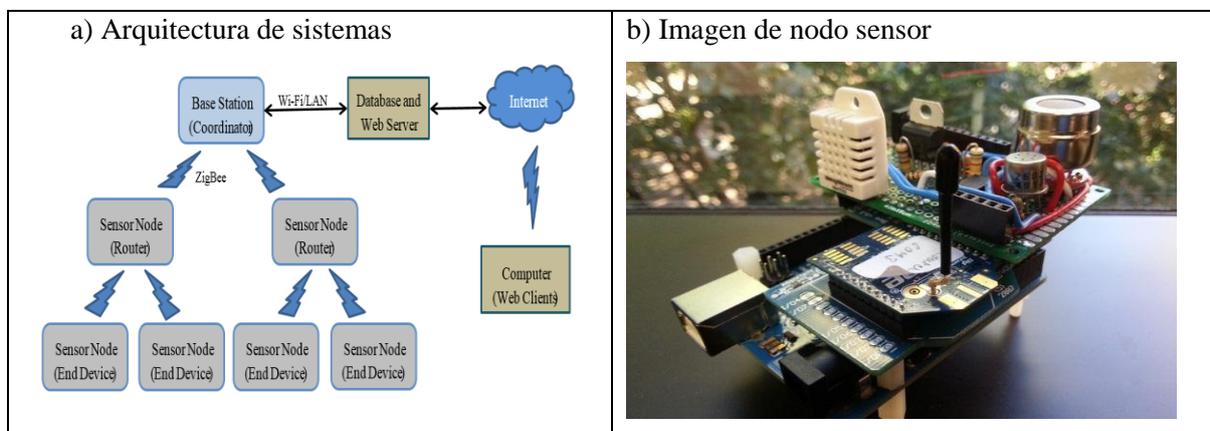


FIGURA16: Esquema de arquitectura y nodo sensor
(Fuente: Abraham & Xinrong, 2015).

El nodo sensor desarrollado, está equipado con múltiples sensores, una unidad de procesamiento y un módulo de red inalámbrica de comunicación y malla. La unidad de procesamiento es Arduino Uno. La comunicación inalámbrica se logra utilizando el módulo XBee. Los sensores utilizados en este proyecto, el sensor COV TGS2602, CO MQ7, El sensor de ozono MQ131 que funcionan con semiconductor basado calentamiento y el sensor de temperatura y humedad RTH03 que es un sensor digital y es compacto con bajo consumo de energía y estabilidad a largo plazo.

Los resultados del trabajo son satisfactorios en función de un proceso de calibración y ajuste previo de los sensores. Evidentemente la red de medición está instalada en un ambiente controlado, en cuanto a la distancia para la comunicación entre nodos y los alcances de cada sensor, situación diferente a la medición en el exterior.

Un quinto proyecto de interés corresponde al desarrollado por la empresa Libelium (www.libelium.com) que lanzo en el año 2016 una nueva plataforma IoT Smart Cities que mejora monitoreo en forma simultánea la calidad del aire y el nivel de ruido.

Libelium ha mejorado el diseño de sus sensores de acuerdo con las regulaciones de ruido y control ambiental. Un nuevo conjunto de sensores de gas calibrados en fábrica permite calcular el AQI (índice de calidad del aire) siguiendo las directivas

internacionales mundiales que miden CO, NO, NO₂, SO₂ y polvo / partículas (PM₁ / PM_{2.5} / PM₁₀). Esta nueva generación de sensores de gas ofrece una precisión adicional detectando variaciones tan pequeñas como 0.1 ppm en muchos casos.

El nuevo sensor de nivel de sonido se puede colocar al aire libre ya que incluye un micrófono a prueba de agua. Ha sido diseñado siguiendo las especificaciones del estándar IEC 61672 para sonómetros con una precisión de +/- 0.5dBA. El valor otorgado por el sensor es Leq (nivel de sonido continuo equivalente), una magnitud utilizada en las regulaciones de control de ruido en lugares de trabajo y exteriores, y perfecta para la creación de mapas de niveles de sonido.

La precisión de estos sensores junto con el hecho de estar comunicados de forma inalámbrica a través de 4G, WiFi, LoRaWAN, Sigfox, 802.15.4 / ZigBee o 868/900 radios con cualquier plataforma de la nube en tiempo real los convierten en el complemento perfecto para extender la capilaridad de estos servicios en Smart Cities.



FIGURA17: Imagen nodo sensor Libelium Wasp mote Plug & Sense
(Fuente: www.libelium.com)

Respecto a investigaciones relacionadas con la medición de la contaminación acústica, se presenta el trabajo realizado en **(Mydlarz & Salamon & Bello. 2017)**. En este trabajo se propone un dispositivo de detección acústica estático, inteligente y de bajo costo, basado en hardware de bajo consumo. Lo interesante de este trabajo es la clasificación que realizan los autores respecto diferentes categorías de sistemas de medición remota disponibles y las características que debe tener un sistema de medición de red. Según los autores, los sensores acústicos desplegados estáticamente se pueden agrupar en tres categorías generales, dependiendo de la funcionalidad y el costo del sensor.

Categoría 1: Estaciones de monitoreo dedicadas

Estos dispositivos comerciales están diseñados y construidos con el propósito de monitorear acústicamente de manera precisa, confiable, silenciosa y exigible, y como tal, pueden costar más de \$ 10,000 USD. Estos sistemas generalmente consisten en medidores de nivel de sonido de registro portátil de alta gama y hardware de red a medida, encerrados en una carcasa climatizada. Un ejemplo de esta categoría es el que se presenta (www.bksv.com/Products/EnvironmentManagementSolutions) que se vende por \$ 15,000 USD e incluye un micrófono aprobado y un sistema de análisis con un rango de medición de 25 a 140 dB, la capacidad de producir Datos de ruido espectral de 1/3 de octava, transferencia inalámbrica de datos en tiempo real, operación autónoma y una carcasa reforzada para operación exterior a largo plazo. Otros ejemplos con especificaciones y precios similares incluyen la 01dB OPER @ Station (www.01db.com) y el Larson Davis 831 Noise Monitoring System (www.larsondavis.com). El hardware y el software utilizados en estos sistemas suelen ser propietarios y, por lo tanto, no permiten personalizar la funcionalidad para fines distintos de la monitorización acústica básica de niveles de ruido, excepto a través de compras de módulos de software tales como detección de eventos basados en umbrales que normalmente cuestan más de \$ 1000 por módulo. Si bien los costos iniciales de los sensores son altos, los costos de mantenimiento generalmente son menores que en las soluciones de menor costo debido a la naturaleza especializada y de alta ingeniería de estos dispositivos.

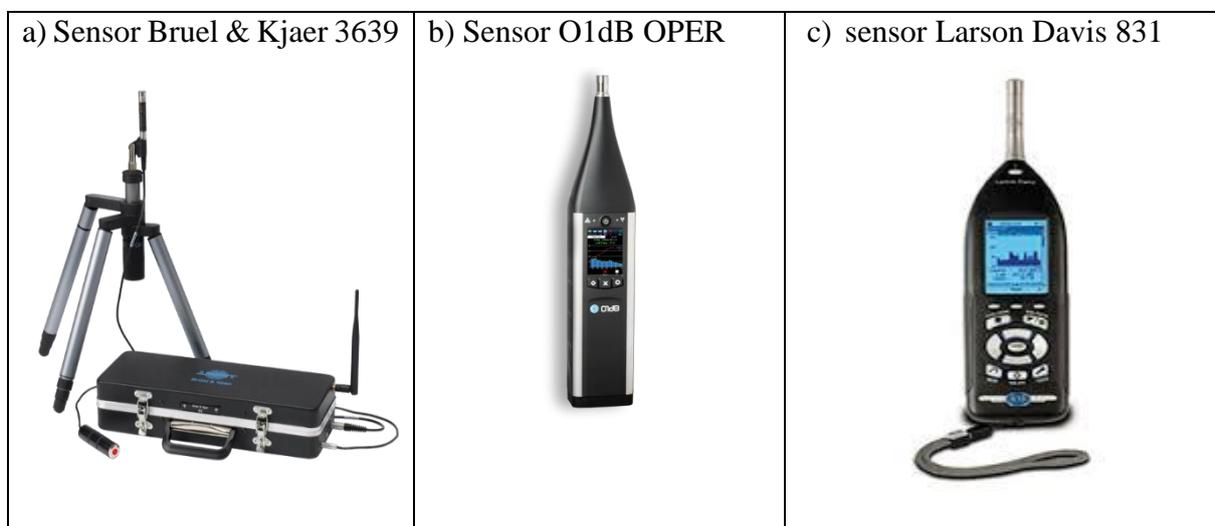


FIGURA18: Imagen sensores de audio de alta gama
(Fuente: Mydlarz & Salamon & Bello. 2017)

Categoría 2: Red de sensores moderadamente escalable.

Este grupo consiste en una combinación de dispositivos desarrollados comercialmente y desarrollados en grupos de investigación que brindan mayores oportunidades para despliegues a mayor escala que los de Categoría 1 con variada precisión de los datos. El punto de precio típico de cada nodo en este grupo es la marca de \$ 600 USD. Entre los ejemplos comerciales se incluyen los \$ 560 USD Libelium Waspote Plug & Sense, dispositivo Smart Cities que, entre otras cosas, mide valores de db simples sin certificación de tipo, para dar una indicación del nivel de presión acústica de una ubicación. El dispositivo Libelium está reforzado para uso exterior, funciona de forma autónoma y puede transferir datos de forma inalámbrica a un servidor central. Este sistema no proporciona medios para procesar los datos de audio entrantes, ya que la conversión a los valores de dB se produce a nivel de hardware en la placa de micrófonos. Un ejemplo diferente en esta categoría es la red RUMEUR desarrollada por el Noise Observatory Group de la organización sin fines de lucro, Bruitparif, con sede en París (www.bruitparif.fr). Su red consta de alrededor de 50 estaciones de monitoreo de \$ 2500 USD que recopilan audio de alta calidad y datos acústicos precisos en el nivel de tipo 1, incluida la detección de eventos acústicos. Esta red también se complementa con 350 dispositivos de bajo costo de \$ 550 USD que registran valores de dB en el nivel de tipo 2. Si bien son más escalables que las redes de Categoría 1, aún están limitadas por costos relativamente altos y, en algunos casos, por imprecisiones de medición.

Categoría 3: Red de sensores de bajo costo.

Esta categoría de red de sensores generalmente consiste en nodos hechos a medida diseñados para ser económicos, de baja potencia y autónomos para implementaciones a gran escala. La mayoría de estos utilizan núcleos de computación de placa única de baja potencia con hardware de audio de bajo costo. El precio de \$ 150 por nodo sensor en esta categoría la convierte en una solución viable para despliegues de red generalizados. Actualmente estas redes están desarrolladas principalmente por grupos de investigación universitarios, incluido el proyecto CamMobSen de Universidad de Cambridge (www.escience.cam.ac.uk/mobiledata) cuyos sensores de bajo costo controlan los niveles de ruido en dB, con un rango efectivo de 55 a 100 dB con una precisión de 3 dB en comparación con un tipo 1 medidor de nivel de sonido. Una iniciativa similar de bajo costo de Finlandia (**Hakala**,

2016) ha producido nodos de sensores con un costo de \$ 150 USD que pueden transmitir valores de dB de forma inalámbrica utilizando un núcleo de computación de bajo consumo y un sistema de audio capaz de un rango efectivo de mediciones de 36 a 90 dB. Esta categoría es claramente la más escalable debido a sus nodos sensores de bajo costo, sin embargo, en los ejemplos dados, la precisión de los datos acústicos es baja y los núcleos informáticos de baja potencia no permiten ningún DSP in situ. El nodo de medición está basado en el sensor MCE-400 (www.monacor.com).

Ahora en el estudio de (Mydlarz & Salamon & Bello. 2017), se presenta las características que debe tener una solución de red de sensores. En este sentido la red debe reunir algunas características como ser escalable, preciso, autónomo y adaptable, se requiere una combinación de atributos de cada una de las categorías mencionadas anteriormente. Con base en estos ejemplos previos de redes de detección acústica, una solución viable para la monitorización del ruido urbano a gran escala y de alta calidad debería proporcionar un mínimo de estas características: La capacidad de monitorear los niveles de presión sonora con un nivel comparable de precisión a los estándares de las agencias de la ciudad, Capacidades informáticas mejoradas para el procesamiento inteligente de señales in situ y transmisión de datos de audio sin procesar, Autónomo en su operación, Un bajo costo por nodo sensor en el punto de precio de \$ 100 USD.

La solución presentada tiene como objetivo cumplir con todos estos requisitos para proporcionar una solución viable para el monitoreo acústico urbano avanzado a gran escala. El nodo sensor propuesto permite capturar datos acústicos con buena precisión y posee un costo aproximadamente de \$ 100 USD.

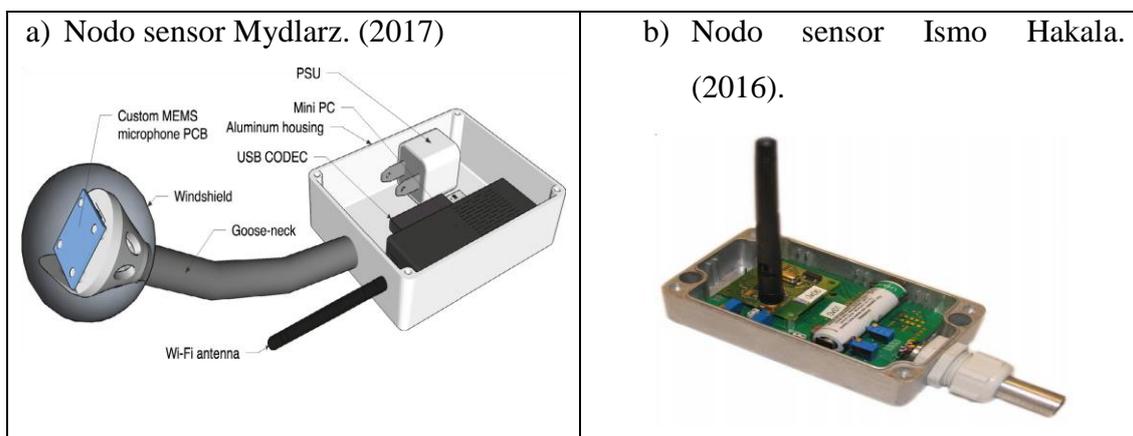


FIGURA19: Imagen sensor de sonido fabricación académica
(Fuente: Mydlarz & Salamon & Bello. (2017) & Ismo Hakala. (2016))

5.2 Arquitecturas de sistemas de medición remota

El primer trabajo consultado, corresponde al descrito en (Gómez, Marcillo, Triana, Gallo, Oviedo, Hernández., 2017), donde se desarrolla una arquitectura basada en internet de las cosas (IoT) que permite el uso de sensores con la capacidad de recopilar información relacionada con variables ambientales y además facilita la integración con cualquier otro tipo de sensores asociados que midan en un entorno de ciudad inteligente. Usando esta información como herramienta, se pretende mejorar el proceso de toma de decisiones, comenzando con las mejoras en el diseño de la ciudad para el aumento y la mejora de los productos básicos para los habitantes de la ciudad.

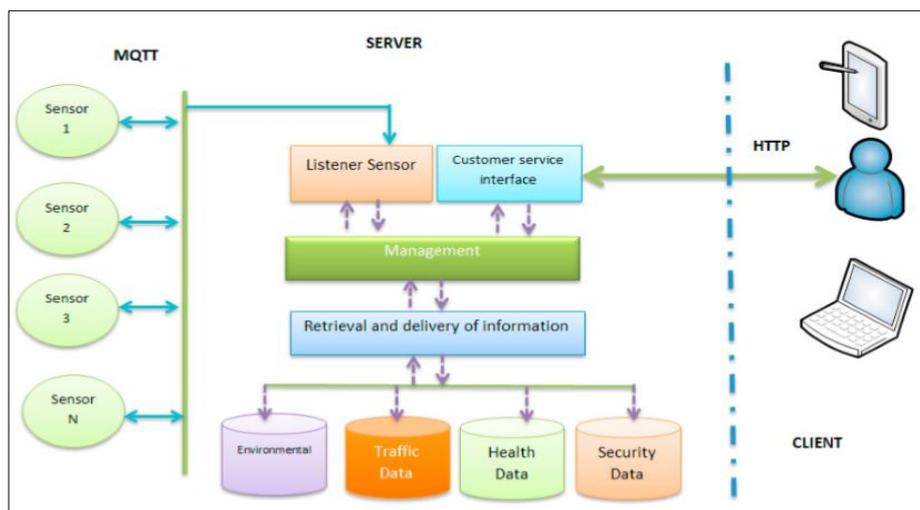


FIGURA 20: Modelo arquitectura genérica para IoT

(Fuente: Gómez, Marcillo, Triana, Gallo, Oviedo, Hernández., 2017).

La arquitectura del sistema propuesta en el trabajo está basada en la filosofía de servidor-cliente, en donde está la presencia de los sensores y es la capa responsable de la escucha permanente a través del protocolo MQTT de las solicitudes derivadas de los sensores que capturan los datos. De la misma manera, los sensores se pueden gestionar, calibrar y conocer el estado real de cada uno de ellos. Esta capa incorpora los datos procedentes del sistema de sensores gestionados a través de un dispositivo Arduino Uno, que funciona como un controlador para ellos, al mismo tiempo, los datos recogidos se envían a un Pi Raspberry, que los dirige al IoT mediador a través de una publicación. La Interfaz de servicio al cliente: esta capa se encarga de obtener las solicitudes del cliente desde la web móvil, que luego se envía a la capa subyacente,

es decir a la capa de gestión, para procesar las solicitudes provenientes de los clientes. Esta capa también es responsable de mostrar las respuestas a los diferentes clientes por medio de HTTP. La siguiente capa se llama de Gestión. Esta capa está a cargo de la suscripción al mediador por un cliente MQTT, recibe los datos leídos a través del tema enviado y se conecta con la base de datos para su almacenamiento, que se redirigen en función de su naturaleza: ambiental, sanitaria, tráfico o seguridad urbana. Al mismo tiempo, gestiona los diferentes aspectos de la aplicación web y móvil para obtener los datos solicitados por los clientes. La última capa es la de persistencia y entrega de información dependiendo de las solicitudes y respuestas de los usuarios, así como de los sensores.

Otra iniciativa interesante corresponde al proyecto “Smart Santander”, (**SmartSantander, 2009**). Este proyecto es una iniciativa para toda la ciudad que incluye una red con más de 12.000 sensores. Esos sensores monitorean los niveles de tránsito, las opciones de transporte público, los niveles de ruido y partículas en el aire, los niveles de iluminación, la calidad del agua y la disponibilidad de lugares para estacionar. El sistema también proporciona acceso abierto a los datos y facilita la interacción remota entre los residentes de la ciudad y los oficiales municipales. El sistema incluye varias aplicaciones para smartphones que colaboran con la función de uso compartido de datos y generación de informes y se ponen a disposición de los ciudadanos a través de una aplicación. con la posibilidad de una conexión en tiempo real de los datos generados por la red de sensores existente en la ciudad de Santander. La red posee 15 receptores gateway posicionados en toda la ciudad que reciben datos de los sensores. Cada uno de los sensores se comunica con los receptores gateway a través de tecnología Wi-Fi o de comunicación de campo cercano RF. Todos esos receptores gateway están conectados por un cable de fibra óptica a los servidores centrales de la ciudad. Gracias a esta infraestructura, la ciudad puede instalar una plataforma para analizar datos. El desarrollo del sistema y el mantenimiento continuo le cuestan a la ciudad de Santander aproximadamente USD 2,75 millones por año.

Tal como el estudio anterior, existen proyectos de investigación que están explorando la posibilidad de recopilar datos sobre la calidad del aire utilizando plataformas de sensores de bajo costo agrupados bajo la denominación de ciudades inteligentes y telemetría social.

Ejemplos de este tipo incluyen al proyecto OpenSense (www.opensense.ethz.ch), que corresponde a una iniciativa de investigación financiada por Nano-Tera y tiene como objetivo estudiar la detección basada en tecnología de red de sensores inalámbricos con el fin de visualizar la contaminación del aire mediante mapas de partículas creados con las mediciones de la red móvil de monitoreo. Esta información es desplegada en una aplicación iOS y Android y proporciona a los habitantes urbanos de Zúrich un servicio de enrutamiento óptimo para la salud. Los ciudadanos pueden reducir su exposición a partículas ultra finas al no tomar el camino más corto entre el origen y el destino, sino una ruta alternativa más sana y ligeramente más larga calculada por la aplicación.

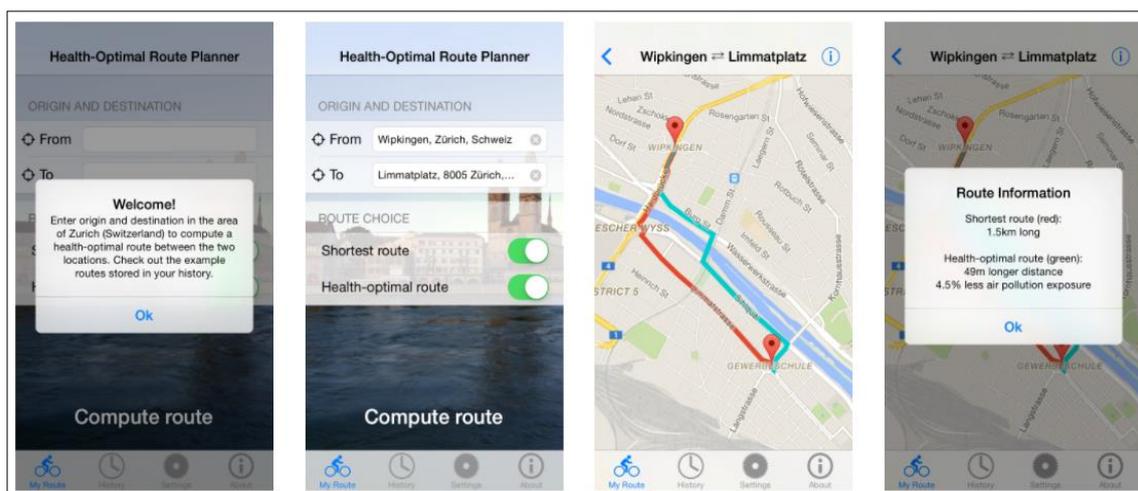


FIGURA 21: Imagen de interfaz aplicación OpenSense (Fuente: <http://www.opensense.ethz.ch/>).

Dentro de esta misma categoría de proyectos, se puede mencionar la iniciativa denominada Citi-Sense-MOB (<http://cwi.unik.no/wiki/CSM:Home>) que usa plataformas móviles para monitorear la variación de la contaminación del aire en las ciudades. Citi-Sense-MOB utiliza una tecnología innovadora para medir, compartir y comunicar continuamente datos ambientales. El objetivo es crear una infraestructura urbana dinámica para la gestión de la ciudad en tiempo real y el progreso sostenible mediante la prestación de servicios relacionados con medición y reducción de los efectos del cambio climático y la calidad del aire.

Citi-Sense-MOB crea conciencia sobre los vínculos entre el cambio climático y la contaminación del aire, el impacto en la salud pública y los beneficios del uso de transporte menos contaminante. El monitoreo móvil de la calidad del aire también

ayuda a una mejor planificación y desarrollo de las áreas urbanas (es decir, transporte público, rutas ciclistas, áreas peatonales). Los usuarios finales del proyecto son personas con asma, infecciones u otras enfermedades respiratorias, trabajadores expuestos a estancias prolongadas cerca de las carreteras, personas mayores y, por supuesto, ciudadanos preocupados por su salud.

La figura siguiente ilustra la arquitectura Citi-Sense-MOB. El sistema de medición está compuesto por numerosos sensores montados en plataformas móviles (autobuses, bicicletas, automóviles, etc.). Los datos reunidos continuamente se transmiten luego a un servidor que los combina con datos ya existentes (modelos, estaciones de monitoreo, etc.). Los datos procesados se devuelven a los ciudadanos presentados en un diseño fácil de usar y visualmente informativo utilizando servicios web y aplicaciones de teléfonos móviles.

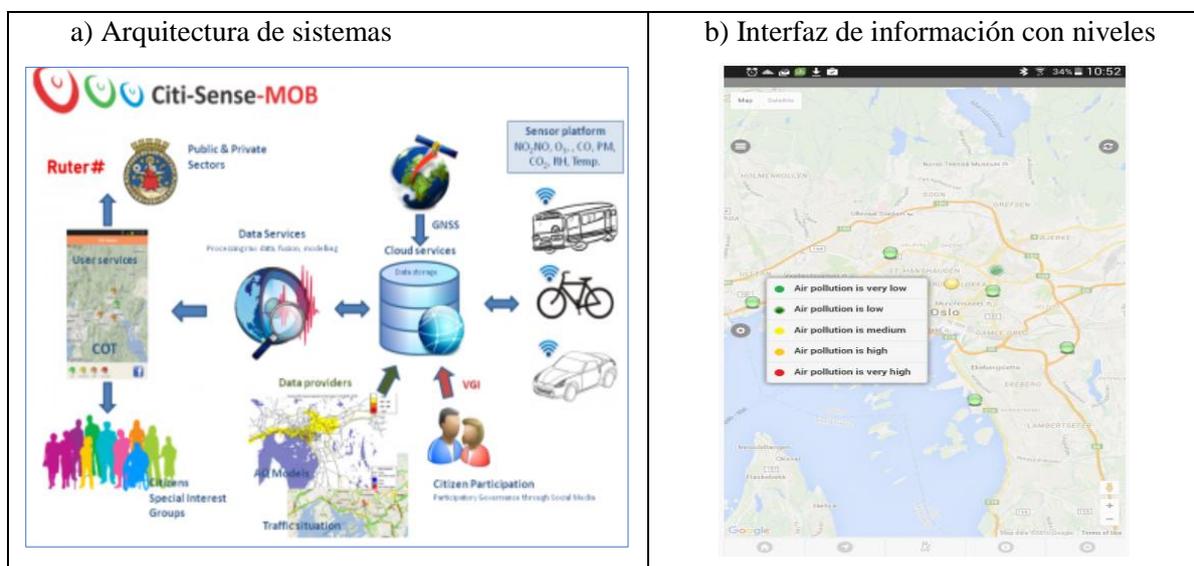


FIGURA 22: Imagen de interfaz aplicación Citi-Sense-MOB

(Fuente: <http://cwi.unik.no/wiki/CSM:Home/>).

Otro proyecto de telemetría como servicio, se encuentra disponible en Everyaware (www.everyaware.eu), que permite ayuda a los ciudadanos a recopilar, compartir y comprender su entorno centrándose en los datos de ruido y contaminación del aire. La plataforma desarrollada permite organizar, recopilar y estimar información ambiental. Esta nueva forma de acceder a la información ambiental puede desencadenar cambios en el comportamiento de los ciudadanos debido a una mayor conciencia de su situación ambiental. Además, los datos recopilados ayudarán a los investigadores a comprender la conexión entre los datos ecológicos objetivos y la

percepción que los ciudadanos tienen de su entorno. De esta forma, los investigadores obtendrán información sobre los procesos subyacentes. Para permitir mediciones de aire por parte de cualquier ciudadano y permitir experimentos científicos, se ha desarrollado una solución de sensor asequible y de bajo costo en combinación con un teléfono inteligente y un servicio web.

También interesa mencionar el proyecto Citi-Sense (www.citi-sense.eu) que permite a las personas utilizar plataformas de calidad del aire de bajo costo en 8 ciudades de Europa. El portal web está diseñado para permitir a los ciudadanos acceder no solo a la información ambiental en tiempo real proporcionada por una gran cantidad de sensores y plataformas de sensores (incluidos sensores portátiles y sensores estáticos, aplicaciones móviles y diferentes tipos de IR contaminación encuestas de percepción), sino también para proporcionar comentarios para la discusión, el debate y el intercambio de sus propias observaciones personales. El proyecto contempla mediciones para ayudar a las escuelas a obtener más información y participar en la mejora de la calidad del aire, equipamos 12 escuelas en Europa con unidades de sensores electrónicos que miden diferentes aspectos del aire interior (<http://schools.citi-sense.eu/>). Las unidades de sensor miden la temperatura, la humedad relativa, el CO₂, el NO₂, las partículas (PM₁₀, PM_{2.5}), el ozono, el CO, el ruido y el radón. Los datos se actualizan con frecuencia y se pueden ver y acceder a través de este portal. Las están ubicadas en cuatro ciudades diferentes de Europa, incluidas Oslo (Noruega), Edimburgo (Reino Unido), Liubliana (Eslovenia). Hay 3 escuelas participantes en cada ciudad.

Un estudio muy interesante y cercano a los objetivos de esta investigación, se describe en **(Castell, Schneider, Grossberndt, Fredriksen, Sousa-Santos, Vogt, Bartonova., 2017)**. Donde se realizan mediciones ambientales a jardines infantiles en Noruega. Como se sabe los niños de los jardines pasan un tiempo significativo al aire libre bajo todas las condiciones climáticas, y existe una preocupación natural por la calidad del aire exterior. Especialmente durante el invierno y la primavera, los jardines de Oslo que están situadas cerca de las calles con tráfico intenso, donde la quema de leña se usa para el calentamiento de la casa, experimentan muchos días con baja calidad. Durante estos períodos, la información actualizada sobre los niveles de calidad del aire puede ayudar a los profesores de kínder a planificar actividades al aire libre apropiadas y así proteger la salud de los niños. El proyecto contempló, la instalación de 17 nodos de medición de calidad del aire con sensor AQMesh

(www.aqmesh.com), descrito anteriormente, en jardines de infantiles en Oslo. Los nodos sensores se ubicaron al aire libre, en el patio de recreo del jardín, a alturas de alrededor de 2.5 a 3 m. Además, se instalaron 6 nodos sensores en las calles de Oslo (figura 22, círculos verdes).

En cuanto al despliegue de la información capturada, el proyecto contempla un portal web para visualizar los datos de los nodos del sensor en tiempo real, como parte de la visualización de toda la información recopilada por el proyecto CITI-SENSE. El portal web muestra la ubicación de cada nodo sensor en Google Maps y proporciona información en tiempo real sobre los niveles de calidad del aire medidos (Figura 23). La información sobre la calidad del aire se mostró con una escala de 5 colores (Indicación de Polución de Aire Indicación), que indica si la contaminación del aire es muy baja, baja, bastante baja, bastante alta o muy alta. Los umbrales se basaron en el Índice de calidad del aire promedio para los valores por hora. El índice promedio se propuso para facilitar la comparación de la calidad del aire en las ciudades europeas en tiempo real. El usuario podría elegir un período específico para ver la calidad del aire a lo largo del tiempo. Si hace clic en un objeto específico, estará disponible información adicional sobre sus datos actuales (por ejemplo, valores del índice de las últimas 24 h). Un gráfico de resumen también indica la calidad del aire medida por todas las unidades de sensores y proporcionaría un índice general para toda la ciudad. El usuario también puede ver el mapa de calidad del aire generado en tiempo real.

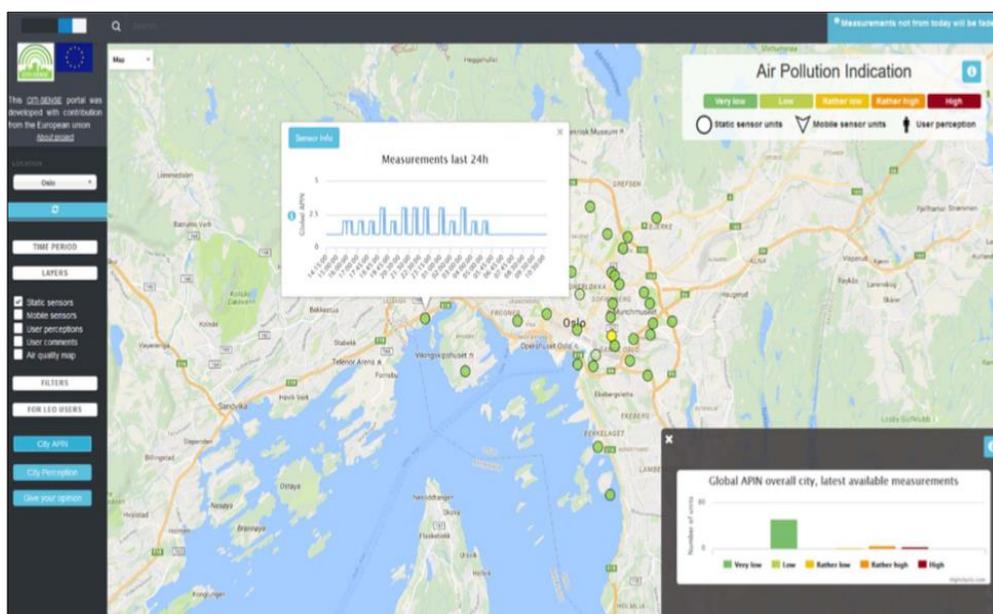


FIGURA 23: Interfaz monitoreo Jardines Infantiles Oslo, (Fuente: Castell, Schneider, Grossberndt, Fredriksen, Sousa-Santos, 2017)

Finalmente es necesario citar la plataforma de medición SINCA (Sistema de información nacional de calidad del aire) perteneciente al Ministerio del Medioambiente del Gobierno de Chile (<http://sinca.mma.gob.cl/>).

El sistema contempla una red nacional de medición con estaciones de alta gama que permiten medir una amplia variedad de parámetros ambientales. La medición se realiza en tiempo real pero el despliegue se actualiza cada una hora. La interfaz gráfica principal del sistema se presenta a continuación.

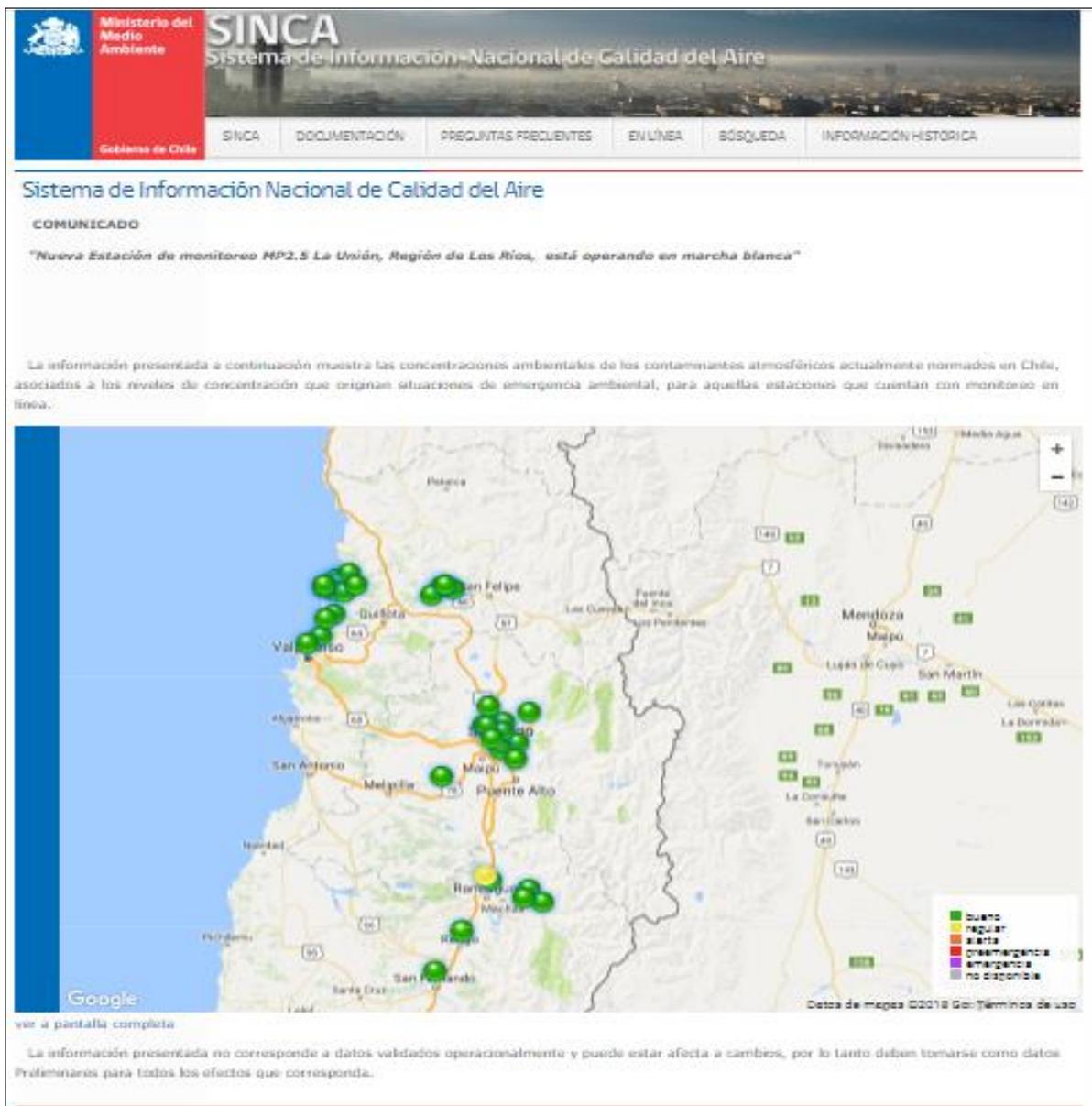


FIGURA 24: Interfaz plataforma SINCA
(Fuente: <http://sinca.mma.gob.cl/>, enero, 2018)

La plataforma dispone información de monitoreo por regiones que incluyen estaciones ubicadas en las grandes ciudades (207 estaciones). En particular los registros disponibles son para cada hora de mediciones, también datos semanales, mensuales y anuales los cuales se despliegan mediante gráficos y un sistema de alerta por colores para cada variable.

Como ejemplo se presenta la interfaz de monitoreo para la estación de la comuna de Puente Alto.

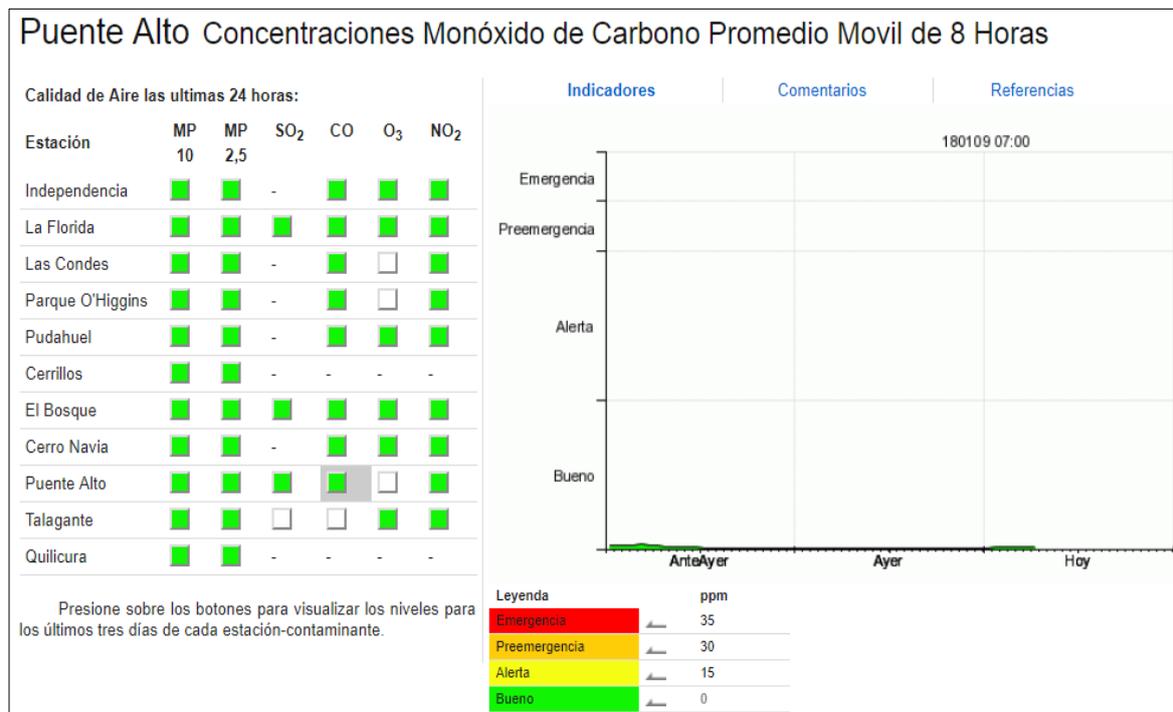


FIGURA 25: Interfaz monitoreo SINCA
(Fuente: <http://sinca.mma.gob.cl>, Enero 2018)

El esquema general de funcionamiento de las estaciones que miden la calidad del aire funciona de forma similar en todo el país. Estas estaciones trabajan mediante una unidad colectora (cabezal), que es capaz de capturar porciones de aire. La muestra se toma del aire ambiente a través del cabezal, con una velocidad de flujo constante del orden de 16.7 litros por minutos, lo que representa el volumen promedio de aire respirado por el ser humano.

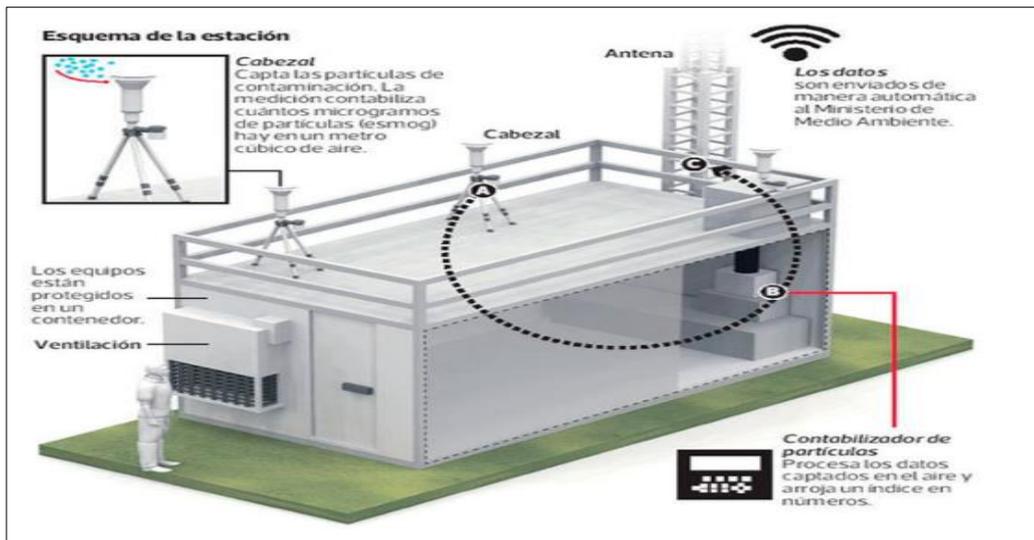


FIGURA 26: Infograma estación monitoreo SINCA

(Fuente: <http://www.latercera.com/noticia/contaminacion-asi-miden-la-calidad-del-aire-las-207-estaciones-de-monitoreo-del-pais>)

Respecto a la medición de monóxido de carbono, las estaciones incorporan el sensor Thermo 48i, que se ilustra a continuación.



FIGURA 27: Esquema de sensor Thermo 48i utilizado por SINCA, Puente Alto

(Fuente: <http://www.thermofisher.com/order/catalog/product/48I>)

El modelo 48i se basa en el principio de que el monóxido de carbono (CO) absorbe la radiación infrarroja con una longitud de onda de 4,6 micrómetros. Como la absorción de infrarrojos es una técnica de medición no lineal, los componentes electrónicos del instrumento tienen que transformar la señal básica del analizador en una salida lineal. El modelo 48i utiliza una curva de calibración exacta para linealizar de manera precisa la salida del instrumento en cualquier intervalo hasta una concentración de 10.000 ppm. El costo del equipo Thermo 48i asciende a U\$ 9500.

Como información adicional está el proyecto World Air Quality Index (<http://aqicn.org>), que es un proyecto de empresa social que comenzó en 2007. El proyecto presenta información transparente de calidad del aire para más de 70 países, abarcando más de 9000 estaciones en 600 ciudades principales. Para el caso de Chile, utiliza la información del SINCA.

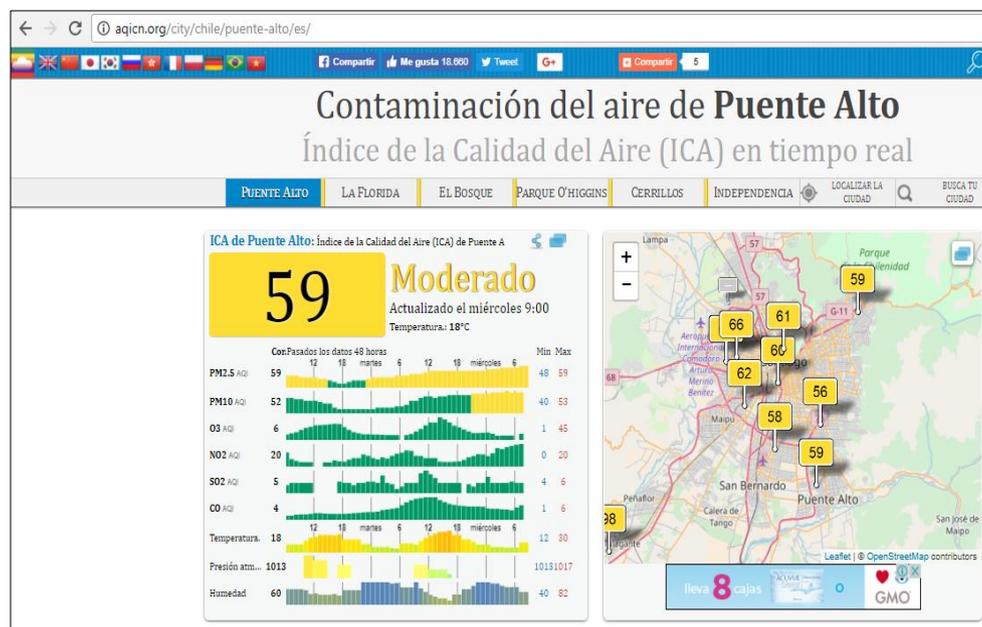


FIGURA 28: Interfaz plataforma monitoreo aqicn.org
(Fuente: <http://aqicn.org/city/chile/puente-alto/es/>, Enero 2018)

5.3 Discusión del Estado del Arte

De acuerdo con los estudios revisados, se puede comentar lo siguiente.

- Se constata la presencia de abundante información actualizada (2015 al 2017), sobre análisis de sensores de bajo costo y la comparación con equipamiento profesional de medición. Los resultados de la calidad de la medida con dispositivos comerciales de bajo costo son confiables cuando se utilizan en ambiente Indoor, sin embargo, para mediciones de alta gama en exteriores no son validados.
- No se observa evidencia abundante de proyectos orientados a mediciones medioambientales que midan variables relacionadas con la calidad del aire en forma simultánea con la medición de variable acústica. Mas bien los proyectos disponibles se enfocan a la medición individual de cada parámetro.

- Existe evidencia de la necesidad que tiene la comunidad urbana de disponer de una mayor cobertura de la red de estaciones de medición, que puedan proporcionar información localizada relevante. Esto se manifiesta en proyectos de ciudades inteligente vigentes en Europa.
- No obstante, de contar con proyectos funcionando, que disponen de plataformas de monitoreo web y dashboard. Se verifica que no se visualizaron iniciativas que dispongan de opciones de aviso y alerta en tiempo real a los usuarios. Mas bien se constata el uso pasivo del usuario al acceder a la información desplegada.
- En cuanto a las arquitecturas utilizadas por cada uno de los trabajos realizados, se evidencia una concordancia respecto a la estructura que deben seguir los sistemas de monitoreo. Específicamente, se verifica una arquitectura de organización basado en capas o layer y una filosofía servidor cliente a nivel superior y una filosofía de publicación-suscripción para las capas inferiores. Además, los mecanismos de integración con otros sistemas, se basa generalmente en arquitecturas Rest, Mqtt, Ajax, Http. Esto permite un campo abierto para la utilización de tecnología de WebSockets que permite enviar un flujo continuo de data desde el servidor con tecnología push y no pull.
- A nivel nacional la información ambiental disponible se cataliza en base al SINCA que dispone de una red certificada de medición, además se verifican alternativas de monitoreo a nivel focalizado por instituciones privadas y académicas con estaciones meteorológica semi profesionales.

VI Enfoque metodológico

6.1 Tipo de proyecto

Para establecer el tipo de metodología a utilizar para esta investigación, es necesario determinar el tipo de proyecto o investigación. En este sentido, la investigación propuesta es de tipo aplicada que tiene como objetivo proponer un servicio de información a los usuarios, que se relaciona con el ámbito de las tecnologías de la información. Este servicio, se traducirá en un prototipo de hardware y software, que será validado mediante pruebas experimentales.

De acuerdo con lo anterior, se propone como metodología de trabajo, la *Investigación Cuantitativa*, descrita en **(Hernández & Fernández & Baptista, 2010)**, el cual se muestra en la figura nº24. La elección de esta metodología se debió a sus características de investigación practica basada en experimentación, con objetivos y requerimientos claramente definidos. Las características de esta metodología se comentan a continuación:

- Exige disciplina en el orden de cumplimiento de cada etapa del proyecto, es decir, antes de iniciar una nueva etapa, debe dar por finalizada la etapa anterior, lo que asegura existan tareas pendientes. Esto permite asegurar que los entregables de cada etapa cumplan plazos comprometidos.
- Está orientado a la creación de una estructura documental.

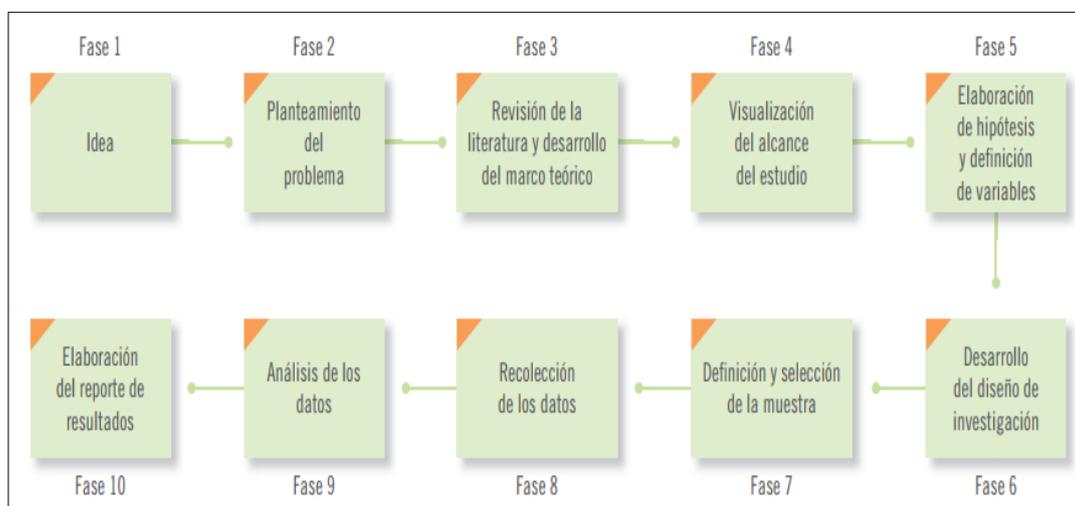


FIGURA 29: Esquema de proceso cuantitativo
(Fuente: Hernández R., & Fernández C. & Baptista P., 2010)

6.2 Metodología de investigación cuantitativa

El enfoque cuantitativo es secuencial y puede ser probado. Cada etapa se debe concluir para comenzar otra y no se pueden saltar pasos del modelo. Es primordial priorizar rigurosamente el orden de la metodología.

Se inicia mediante la definición del alcance, una vez realizada esta tarea, se derivan los objetivos y preguntas de investigación (dependiente el caso), se revisa la literatura asociada al tema y se construye el marco teórico, se procede a establecer el aporte de valor del trabajo y se determinan las variables relevantes que servirán para validar metas propuestas. Luego se desarrolla un plan para probar las variables (diseño), éstas se miden en un escenario controlado analizando resultados obtenidos de la medición de variables y se establecen conclusiones respecto a las metas planteadas. Este proceso se observa en la figura nº29.

- Primera fase es el inicio de la investigación o proyecto, donde se presenta la idea principal, la cual será estudiada, ésta se representará en el primer acercamiento a estudiar, sin importar qué tipo de paradigma sea fundamentado en el estudio ni el enfoque adoptado. En esta fase no se puede descartar la idea por buena o mala, pues hace falta más análisis e información para la decisión.
- Segunda fase plantea el problema de investigación, estableciendo objetivos de investigación, preguntas (si fuese necesario), justificación de la investigación y evaluación de las deficiencias del conocimiento del problema. Es importante señalar, que no sirve contar con un buen método si no se tiene claro qué problema a resolver. El tiempo transcurrido entre las fases 1 y 2, es relativa y depende del dominio del conocimiento en el tema del investigador. Esta etapa necesita que los problemas específicos sean en términos concretos y explícitos, de manera que sea susceptible de investigarse con procedimientos científicos, de acuerdo con los Métodos de investigación en las relaciones sociales. Por ello teniendo un problema bien planteado, estará parcialmente resuelto.
- Tercera fase se procede a la revisión de literatura asociada al tema y se construye el marco teórico. Según los Enfoques de la investigación **(Hernández & Fernández & Baptista, 2010)**.

- Cuarta fase, entrega como resultado, el proceso de inmersión en el conocimiento existente correlacionado con el problema planteado, y por otro lado un producto, el marco teórico utiliza el conocimiento existente.
- Quinta fase define el fin del estudio. Además, de las variables que ayudarán a medir y validar el fin del estudio. La variable es una propiedad de variación que puede medirse u observarse.
- Sexta fase, el investigador debe visualizar la manera práctica y concreta de medir las variables definidas en la fase anterior, para validar los objetivos específicos del proyecto. En esta se diseña el experimento, fijando el contexto basado en los alcances definidos para el proyecto.
- Séptima fase, se centra en los participantes “qué o quiénes”, objetos, sucesos o comunidades de estudio, de las cuales depende del planteamiento del proyecto. Se define la población, se valida el método de selección de la muestra (probabilístico o no probabilístico), se define el tamaño de la muestra, el proceso de selección y la obtención de la muestra.
- Octava fase, se recolectan datos pertinentes sobre los atributos, conceptos o variables del caso de análisis. Recolectar los datos implica la elaboración de un plan que permita conseguir este objetivo. Para esto se debe determinar las fuentes de obtención de datos, la localización de fuentes, proceso de recolección de datos y finalmente analizarlos para responder al planteamiento del problema.
- Novena fase, realiza tarea de análisis de datos obtenidos en la fase anterior. Actualmente, este proceso se realiza por medio de computadores y softwares especializados. Es por esto, que además del análisis de los datos, esta fase se encarga de la interpretación de los resultados del análisis.
- Décima fase, se encarga de presentar los resultados del análisis de los datos realizados en la fase anterior, comunicando los resultados mediante un reporte, el cual puede presentarse en distintos formatos: un libro, un artículo para una revista, diario, presentación, documento técnico, tesis, entre otras posibilidades. En cualquiera de los casos, debemos describir la investigación realizada y los descubrimientos producidos.

6.2.1 Proceso de Obtención De Datos

La información necesaria para el desarrollo y obtención de resultados para el trabajo se realizará mediante la investigación bibliográfica de trabajos similares, evaluación de plataformas de hardware y software disponibles, diseño de arquitectura y generación de vistas de diseño.

A continuación, se deriva a etapas de pruebas de control básico de hardware y software, efectuadas en laboratorio electrónico, en donde se procederá a probar los sensores, alcance, calibración. A continuación, se probará el envío inalámbrico de los datos desde los nodos de medición al nodo maestro. Luego se realizará la captura de datos en dependencias particulares (Comuna de Pirque), para continuar con mediciones en un ambiente real en particular un Jardín Infantil ubicado en una zona urbana de la comuna de Puente Alto. Los datos capturados, serán almacenados y desplegados en la plataforma de monitoreo georeferenciada.

VII Plan de Tesis

El plan de tesis corresponde a la guía que orienta el desarrollo y ejecución del proyecto. Esta planificación considera determinar los paquetes principales de actividades que componen la ejecución del proyecto. Cada paquete está constituido por una serie de actividades concretas que se deben ejecutar y se definen a partir del alcance del proyecto y la metodología cuantitativa utilizada.

7.1 Definición de tareas

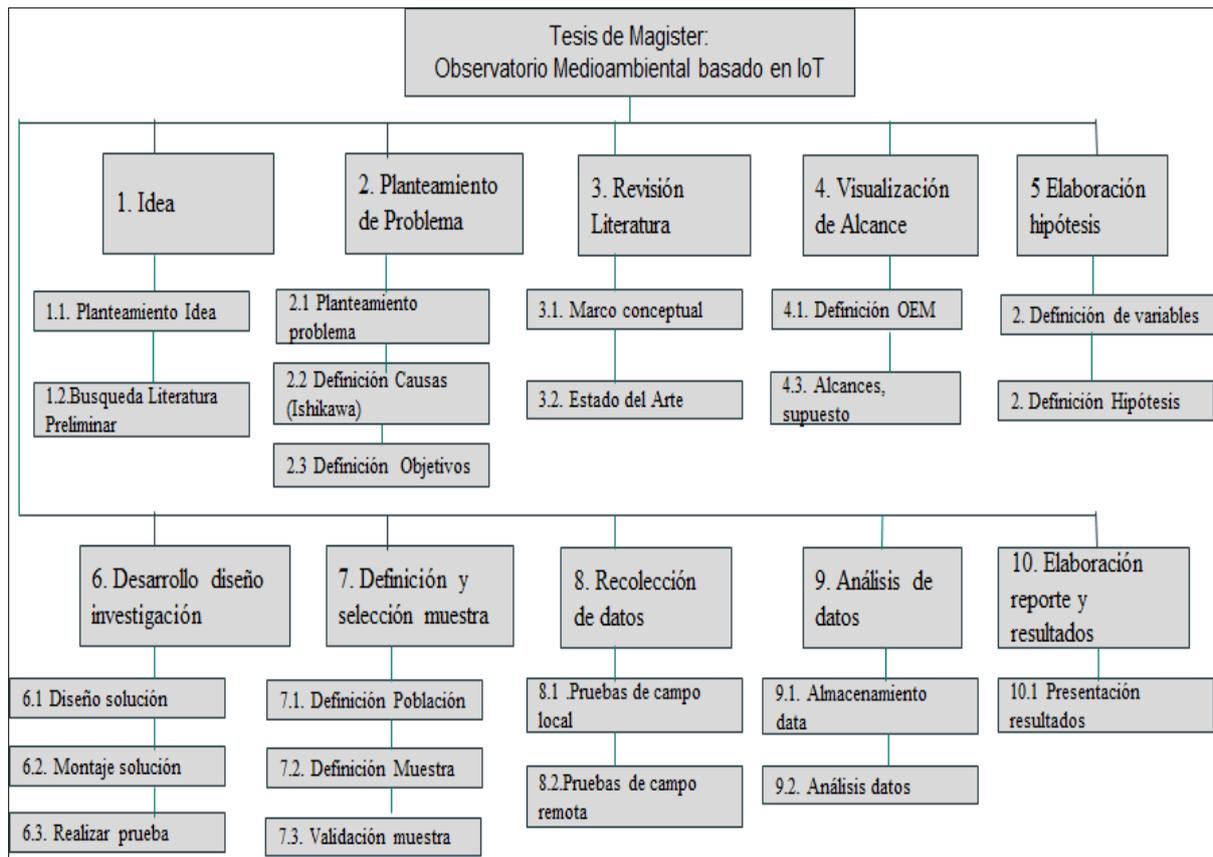


FIGURA 30: Esquema de actividades del proyecto EDT
(Fuente: Elaboración propia)

7.2 Relación de dependencia de tareas

De acuerdo con la naturaleza de la investigación y la metodología adoptada, es necesario disponer de una organización y ejecución de tareas dependientes unas de otras. En este sentido los paquetes definidos anteriormente (figura nº30), con sus actividades correspondiente, poseen una estrecha relación secuencial, es decir se basa en el término de una etapa para comenzar con la otra. En particular, se

manifiesta una dependencia entre las actividades de diseño, codificación, integración, pruebas que requieren desarrollarse en forma secuencial para su éxito. Esto pudiera evidenciar un punto crítico en el desarrollo del proyecto al definirse algún tipo de dependencia que genere retrasos de algunas etapas al generarse algún inconveniente en alguna de ella. De acuerdo con lo anterior, es necesario contemplar un control y seguimiento efectivo de cada actividad, con el fin de detectar algún tipo de desvío y activar la gestión de cambios para resolver la dificultad en el proyecto.

A continuación, se presenta la distribución temporal de la planificación, mediante la carta Gantt, en donde se puede evidenciar mediante flechas rojas la dependencia de las actividades. En particular, se pueden definir algunas tareas simultaneas, pero en general el proyecto tiene una distribución secuencial debido a su naturaleza que dispone de una sola persona que ejecuta todas las actividades.

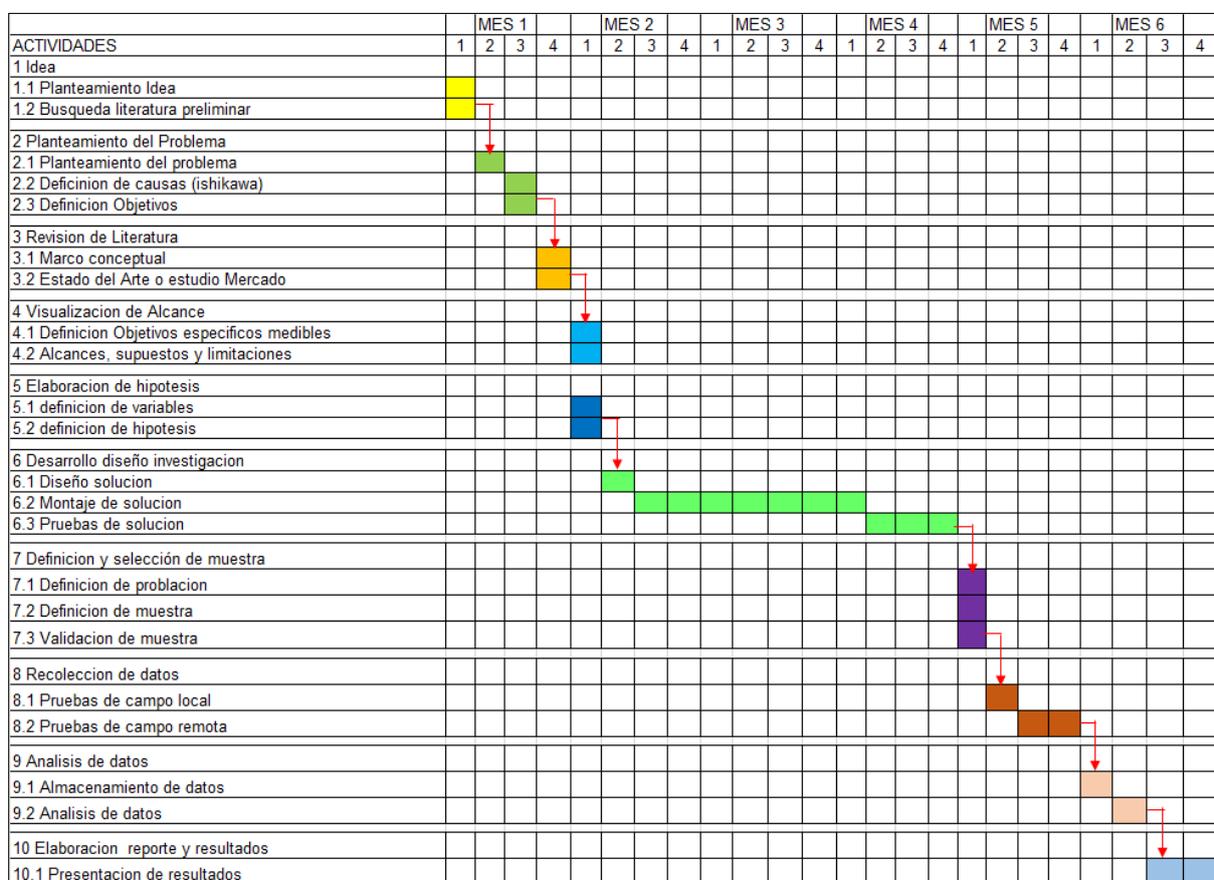


FIGURA 31: Esquema de carta Gantt del proyecto (Fuente: Elaboración propia)

Como se puede observar, la duración estimada del proyecto es de seis meses aproximadamente.

7.3 Asignación de recursos a tareas

Cada actividad del proyecto demanda un esfuerzo necesario para su ejecución. Este esfuerzo, se traduce en el trabajo necesario para implementar cada actividad del proyecto. En particular, el trabajo y esfuerzo de este proyecto, se traducirá en el recurso humano necesario para ejecutar cada actividad (con un rol de integración electrónica y programación de hardware y software) y además la inversión en hardware y software que se debe considerar para implementar la solución. Los criterios de estimación de recursos se detallan a continuación:

- El proyecto será implementado por una persona (Tesisista), que dispondrá de una dedicación de tiempo parcial (4 HH) por día.
- El costo de cada HH se estima de un profesional de nivel medio, valorizado en media UF por HH.
- Por la naturaleza de uno de los objetivos del proyecto, que se refiere a disminuir los costos de las estaciones de monitoreo, se utilizara para la estimación equipamiento de hardware de bajo costo y utilización de software libre cuando sea posible.
- La valoración de todos los recursos se realizará con una distribución beta, utilizando una valoración optimista (**Co**), valoración pesimista (**Cp**) y valoración más probable (**Cm**) de cada recurso asignado.
- De acuerdo con lo anterior, la valoración estimada, se calculará mediante la siguiente expresión matemática:

$$C_e = \frac{C_o + 4C_m + C_p}{6}$$

- Para la estimación de los valores frontera (Co, Cp), se estima un 50% de rango
- Todos los valores de la estimación serán expresados en UF
- Todos los valores de estimación están basados en la distribución temporal presentada en la carta Gantt (Figura nº31).
- La compra de dispositivos electrónicos se realizará, con proveedores locales

A continuación, se presenta una tabla, donde se describen la asignación de recursos de cada paquete expresados en UF y una columna con las HH estimadas para cada actividad.

ACTIVIDADES	RR-HH (valores en UF)					Inversion (UF)					Total (UF)
	Co	Cm	Cp	Ce	HHm	Co	Cm	Cp	Ce	CeT	
1 Idea											
1.1 Planteamiento Idea	2,5	5	7,5	5	10	0	0	0	0	5	
1.2 Búsqueda literatura preliminar	2,5	5	7,5	5	10	0	0	0	0	5	
2 Planteamiento del Problema											
2.1 Planteamiento del problema	5	10	15	10	20	0	0	0	0	10	
2.2 Definición de causas (ishikawa)	5	10	15	10	10	0	0	0	0	10	
2.3 Definición Objetivos	5	10	15	10	10	0	0	0	0	10	
3 Revisión de Literatura											
3.1 Marco conceptual	2,5	5	7,5	5	10	0	0	0	0	5	
3.2 Estado del Arte o estudio Mercado	2,5	5	7,5	5	10	0	0	0	0	5	
4 Visualización de Alcance											
4.1 Definición Objetivos específicos medibles	2,5	5	7,5	5	10	0	0	0	0	5	
4.2 Alcances, supuestos y limitaciones	2,5	5	7,5	5	10	0	0	0	0	5	
5 Elaboración de hipótesis											
5.1 definición de variables	2,5	5	7,5	5	10	0	0	0	0	5	
5.2 definición de hipótesis	2,5	5	7,5	5	10	0	0	0	0	5	
6 Desarrollo diseño investigación											
6.1 Diseño solución	2,5	5	7,5	5	10	0	0	0	0	5	
6.2 Montaje de solución	35	70	105	70	140	10	20	30	20	90	
6.3 Pruebas de solución	15	30	45	30	60	0	0	0	0	30	
7 Definición y selección de muestra											
7.1 Definición de población	2,5	5	7,5	5	10	0	0	0	0	5	
7.2 Definición de muestra	1,13	2,5	3,63	2,458	5	0	0	0	0	2	
7.3 Validación de muestra	1,13	2,5	3,63	2,458	5	0	0	0	0	2	
8 Recolección de datos											
8.1 Pruebas de campo local	5	10	15	10	20	0	0	0	0	10	
8.2 Pruebas de campo remota	15	30	45	30	60	0	0	0	0	30	
9 Análisis de datos											
9.1 Almacenamiento de datos	5	10	15	10	20	0	0	0	0	10	
9.2 Análisis de datos	5	10	15	10	20	0	0	0	0	10	
10 Elaboración reporte y resultados											
10.1 Presentación de resultados	10	20	30	20	40	0	0	0	0	20	
					510					285	
					Total HH				Total Costo UF		

FIGURA 32: Estimación de recursos del proyecto (Fuente: Elaboración propia)

Como se puede observar en la figura n°32, casi la totalidad de los recursos, son asignados a las actividades de recurso humano, expresado en las horas hombre necesarias para la ejecución. En particular se estima que el proyecto requiere de un total de 510 HH aproximadamente, equivalente a 126 días de trabajo (6.5 meses). Respecto a los costos de insumos necesarios para la construcción de las estaciones de medición, el costo total se estima en 20 UF aproximadamente. Finalmente, el costo estimado total de la ejecución del proyecto sería de 285 UF, de acuerdo con esto, se estiman provisiones por desvíos presupuestarios de un 5%, quedando un costo total de **300 UF** aproximadamente.

7.4 Hitos

Los hitos de un proyecto son momentos o eventos relevantes que deben ser alcanzados como elemento vital para el éxito del proyecto. En el caso particular de este proyecto, que involucra un alto nivel de experimentación a nivel de software y hardware, queda en evidencia la importancia de estas actividades para su ejecución. De acuerdo con esto, se definirán como hitos principales del proyecto las entregas de los paquetes principales definidos en el diagrama EDT (Figura n°30).

Hitos	Identificador	Descripción
Hito 1	Diseño de plataformas HW/SW	El cumplimiento de este hito se efectúa con la validación de los documentos de requerimiento y arquitectura.
Hito 2	Construcción de plataforma de medición	Este hito, se cumple al momento de que la plataforma de hardware (nodo de medición) comienza a transmitir en forma inalámbrica el paquete de datos con las mediciones.
Hito 3	Construcción de plataforma de monitoreo	El cumplimiento de este hito se verifica al disponer de la aplicación de monitoreo compuesta por el dashboard en que se visualiza la data de los sensores y el mapa georeferenciado de la ubicación de los nodos de medición y conexión con Twitter.
Hito 4	Pruebas y Experimentos	El ultimo hito relevante, se verifica al disponer de la data obtenida en las pruebas de laboratorio, campo local y campo real. Específicamente, se valida los experimentos al disponer de data obtenida algunos días de la semana.

TABLA 9: Definición de hitos del proyecto (Fuente: elaboración propia)

VIII Desarrollo del Diseño de Investigación y Proyecto

8.1 Especificación de Requerimientos

8.1.1 Introducción

En esta sección se proporcionará una introducción a todo el documento de especificación de Requisitos de Software (ERS). Consta de varias subsecciones: propósito, alcance, personal involucrado, definiciones, referencias.

8.1.1.1 Propósito

El propósito de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que será construido. La especificación de requisitos describe las necesidades del sistema: **Observatorio Medioambiental Georeferenciado para monitoreo de jardines infantiles basado en Internet de las Cosas y WebSockets**, para facilitar el entendimiento de la lectura de los Stakeholders, usuarios claves y programadores, es decir, este documento será el canal de comunicación entre las partes involucradas en el proyecto.

8.1.1.2 Alcance

Este documento contiene la especificación de requerimientos del proyecto, dividido en Requerimientos de negocio, funcionales y no funcionales además las dependencias y fuentes de cada uno de ellos, como también supuestos y restricciones, características de los usuarios y descripción general del prototipo.

8.1.1.3 Personal Involucrado

<i>Stakeholders</i>		
Cargo en la Organización	Rol en el proyecto	Influencia
Director de Carrera	Director gerente	Media
Arquitecto	Diseñador y Desarrollador en jefe	Alta
<i>Usuarios Claves</i>		
	Rol en el proyecto	Influencia
Padres y Apoderados Profesor y Sostenedor	Usuario	Baja

TABLA 10: Matriz de Interesados del Proyecto (Fuente: elaboración propia)

8.1.2 Descripción General

8.1.2.1 Funcionalidad del Prototipo

- Captura de datos medioambientales en tiempo real y georeferenciado mediante nodo de medición basado en plataforma Arduino.
- Despliegue de datos de monitoreo en tiempo real en plataforma web.
- Despliegue de datos históricos (diario y semanal) mediante gráficos.
- Alerta de mediciones riesgosas mediante aviso por Twitter.

8.1.2.2 Características de los Usuarios

- Usuarios del Sistema: Conocimientos básicos de navegación web, en ambiente de computador o dispositivo móvil.
- Administrador del Sistema: Debe contar con conocimientos y experiencias en programación ambiente JavaScript, NodeJS, MongoDB, Arduino.

8.1.2.3 Restricciones

- El proyecto debe estar en pruebas de campo en Marzo 2018.
- No existe posibilidad de sobrepasar el presupuesto ni disponer de financiamiento extra.

8.1.2.4 Suposiciones y dependencias

- Se utilizará el sistema de bases de datos MongoDB.
- Se utilizará plataforma de desarrollo Node.js
- La administración de plataforma web será realizada en forma local
- La infraestructura de Bases de Datos estará ubicada en local
- El proyecto se presenta a nivel de prototipo y no de uso comercial

8.1.3 Requisitos Específicos

Estas secciones de la especificación de requisitos de software contienen todos los requerimientos hasta un nivel de detalle suficiente para permitir a los diseñadores, diseñar un sistema que satisfaga dichos requerimientos, y que permita diseñar las pruebas que ratifiquen que el sistema cumple con los requerimientos.

8.1.3.1 Requisitos comunes de las interfaces

8.1.3.1.1 Interfaces de Usuario

- La interfaz de usuario es de tipo web, orientada a GUI, en especial ventanas y botones, gráficos, y el manejo de la interfaz se realizará mediante mouse, teclado y touch, considerando una interfaz de fácil uso.
- Se puede utilizar un dispositivo móvil o Tablet con pantalla de 8"

8.1.3.1.2 Interfaces de Hardware

- Los nodos de medición serán implementados con plataforma de hardware Arduino Uno, sensor de posición GPS, sensor de Gas MQ, sensor de sonido.
- La conexión entre el ambiente el servidor de monitoreo y los nodos de medición, se realizará en forma inalámbrica (ZigBee).

8.1.3.1.3 Interfaces de Software

- La aplicación se puede operar en los navegadores Internet Explorer 9 en adelante, Google Chrome, Firefox y Safari
- Este software puede operar sobre Android 4 o superior
- El sistema debe interactuar con Base de Datos MongoDB
- El sistema debe interactuar con API de Google Map y API Twitter

8.1.3.1.4 Interfaces de Comunicación

- La aplicación utilizará la filosofía cliente-servidor implementada mediante tecnología push basada en WebSockets.

8.1.4 Requerimientos del Negocio o Dominio

ID	RN_001
Nombre	El sistema deberá considerar interoperabilidad con Bases de datos MongoDB
Características	Interoperabilidad
Descripción	El sistema, debe realizar las consultas a información contenida en sistema de MongoDB
Trazabilidad	Director de Carrera
Prioridad del Requerimiento: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media Alta <input type="checkbox"/> Media <input type="checkbox"/> Media Baja <input type="checkbox"/> Baja	

ID	RN_002
Nombre	El sistema deberá considerar aplicativo web
Características	Funcionalidad
Descripción	La interfaz de comunicación con usuario se realiza mediante aplicativo web, implementado con tecnología NodeJS, HTML, CSS, AngularJS
Trazabilidad	Director de Carrera
Prioridad del Requerimiento: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media Alta <input type="checkbox"/> Media <input type="checkbox"/> Media Baja <input type="checkbox"/> Baja	

ID	RN_003
Nombre	El sistema deberá considerar escalabilidad horizontal y capacidad de agregar más nodos de medición
Características	Eficiencia
Descripción	Se requiere que el sistema proporciones alternativas para incluir nuevos nodos de medición en forma flexible
Trazabilidad	Director de Carrera
Prioridad del Requerimiento: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media Alta <input type="checkbox"/> Media <input type="checkbox"/> Media Baja <input type="checkbox"/> Baja	

ID	RN_004
Nombre	La infraestructura debe ser construida de manera híbrida con una red dedicada (Zigbee) y LAN (monitoreo web)
Características	Fiabilidad
Descripción	Utilizar plataforma híbrida de comunicación para desplegar información fuera de la red de sensores
Trazabilidad	Director de Carrera
Prioridad del Requerimiento: <input type="checkbox"/> Alta <input type="checkbox"/> Media Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Media Baja <input type="checkbox"/> Baja	

8.1.5 Requerimientos Funcionales

ID	RF_001
Nombre	El sistema debe presentar datos de mediciones ambientales en tiempo real en mapa georeferenciado
Características	Presentación información en Dashboard
Descripción	El usuario deberá ingresar a la plataforma y acceder a visualización de mediciones en tiempo real de los nodos desplegados en el mapa, y se indicará la posición del nodo y sus respectivas mediciones
Trazabilidad	Director de Carrera
Dependencias	RN_002, RN_003, RN_004
Prioridad del Requerimiento: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media Alta <input type="checkbox"/> Media <input type="checkbox"/> Media Baja <input type="checkbox"/> Baja	

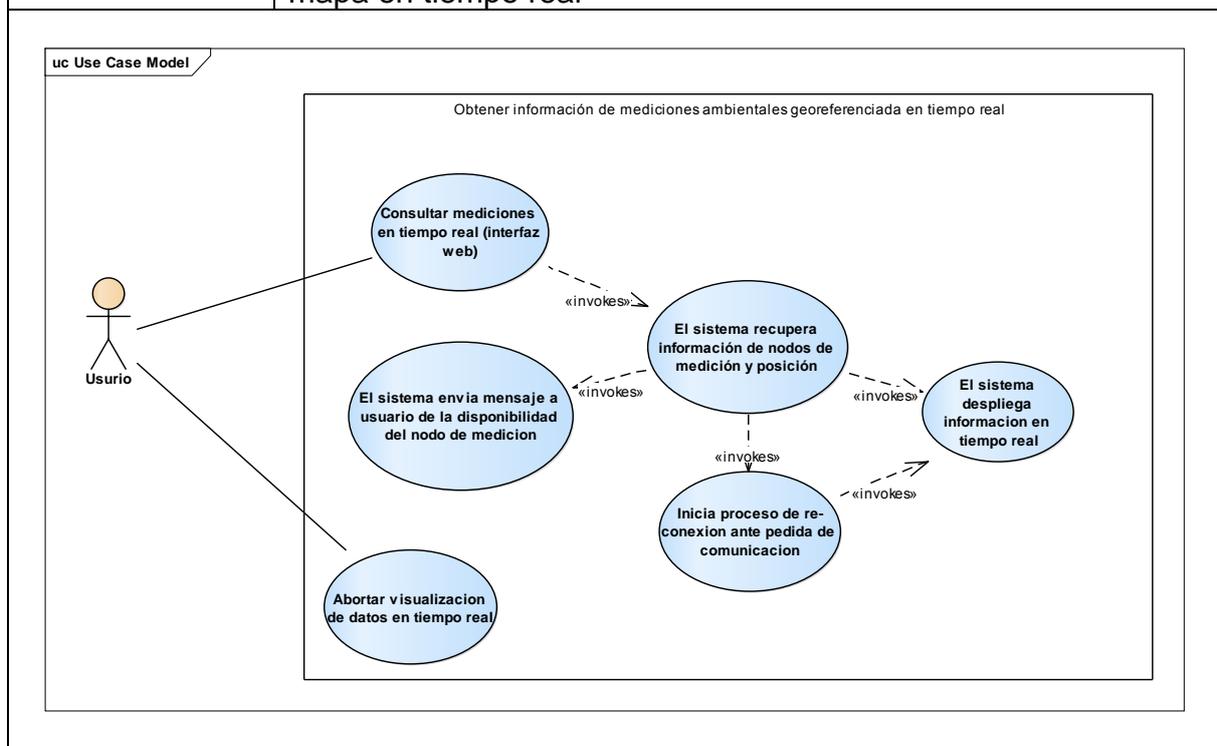
ID	RF_002
Nombre	El sistema debe recuperar datos de mediciones almacenadas en BD y presentar gráficos históricos del día y semana
Características	Lectura datos en BD y presentación información en Dashboard
Descripción	El usuario deberá ingresar al panel de información histórica y podrá visualizar información desplegada en gráficos de día y semana de los comportamientos de las mediciones medioambientales.
Trazabilidad	Director de Carrera
Dependencias	RN_001, RN_002, RN_003, RN_004
Prioridad del Requerimiento: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media Alta <input type="checkbox"/> Media <input type="checkbox"/> Media Baja <input type="checkbox"/> Baja	

ID	RF_003
Nombre	El sistema debe alertar sobre riesgosas de medición mediante un aviso por Twitter a los interesados
Características	Interacción con Redes sociales
Descripción	El sistema avisa mediante Twitter al usuario si algún nodo ha sobrepasado los niveles de riesgo para los niños
Trazabilidad	Director de Carrera
Dependencias	RN_001, RF_001
Prioridad del Requerimiento: <input type="checkbox"/> X Alta <input type="checkbox"/> Media Alta <input type="checkbox"/> Media <input type="checkbox"/> Media Baja <input type="checkbox"/> Baja	

8.1.5.1 Casos de Uso

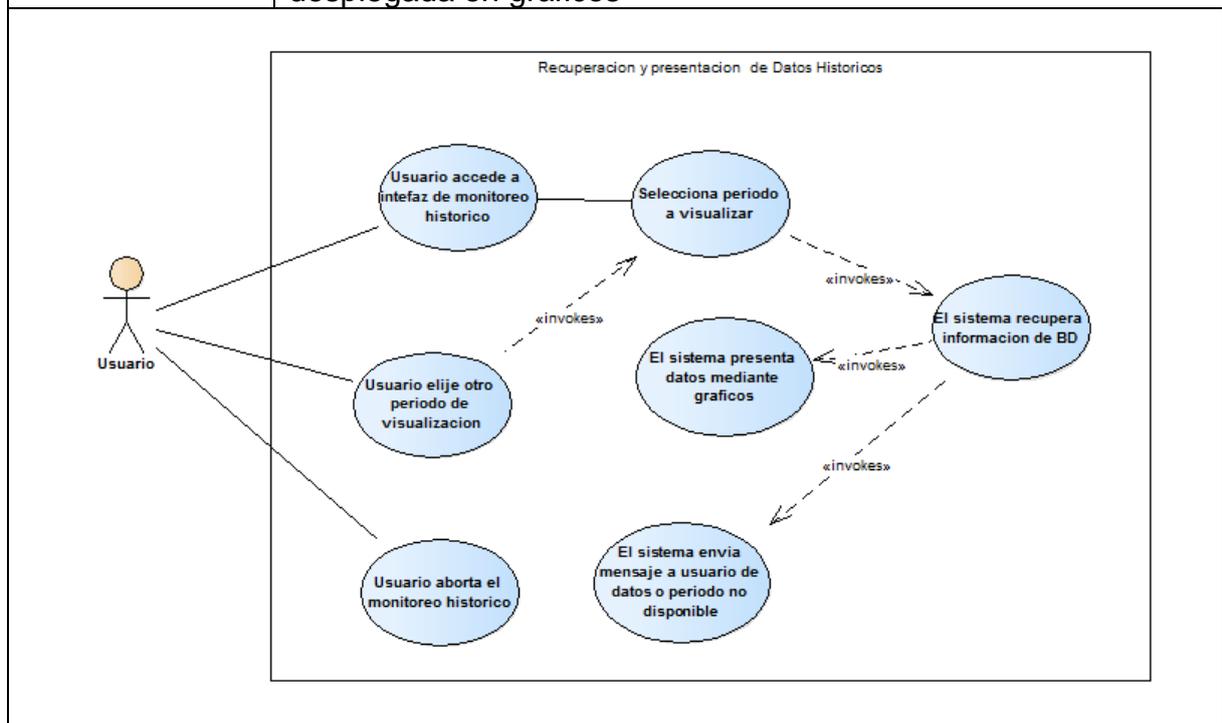
Caso de Uso asociado a requerimiento RF001		
Nombre	Obtener información de mediciones ambientales georeferenciada en tiempo real	
Identificación ID	CU1	
Actor	Usuario y Sistema	
Descripción	El usuario deberá ingresar a la plataforma de medición en tiempo real. El sistema debe rescatar desde los nodos de medición la información de sensores y posición y desplegar en un mapa la posición y el valor de las medidas.	
	Eventos Actor	Eventos de Sistema
Flujo Principal	1. Usuario accede al sitio web 2. Ingresa en interfaz de mediciones en tiempo real	3. El sistema recupera información de nodos de medición y posición 4. El sistema despliega en mapa la posición y mediciones.

Flujo Alternativo	3. Usuario debe actualizar interfaz de plataforma web 4. El usuario aborta el proceso de visualización	1. El sistema envía mensaje de actualizar plataforma para seguir con monitoreo en tiempo real 2. El sistema envía mensaje a usuario para informar que los datos no están disponibles
Precondición	Usuario debe acceder a navegador web y navegador móvil, actualización de WebSockets y módulo de la recepción de data	
Postcondición	El sistema, arroja la información de mediciones y posición en mapa en tiempo real	



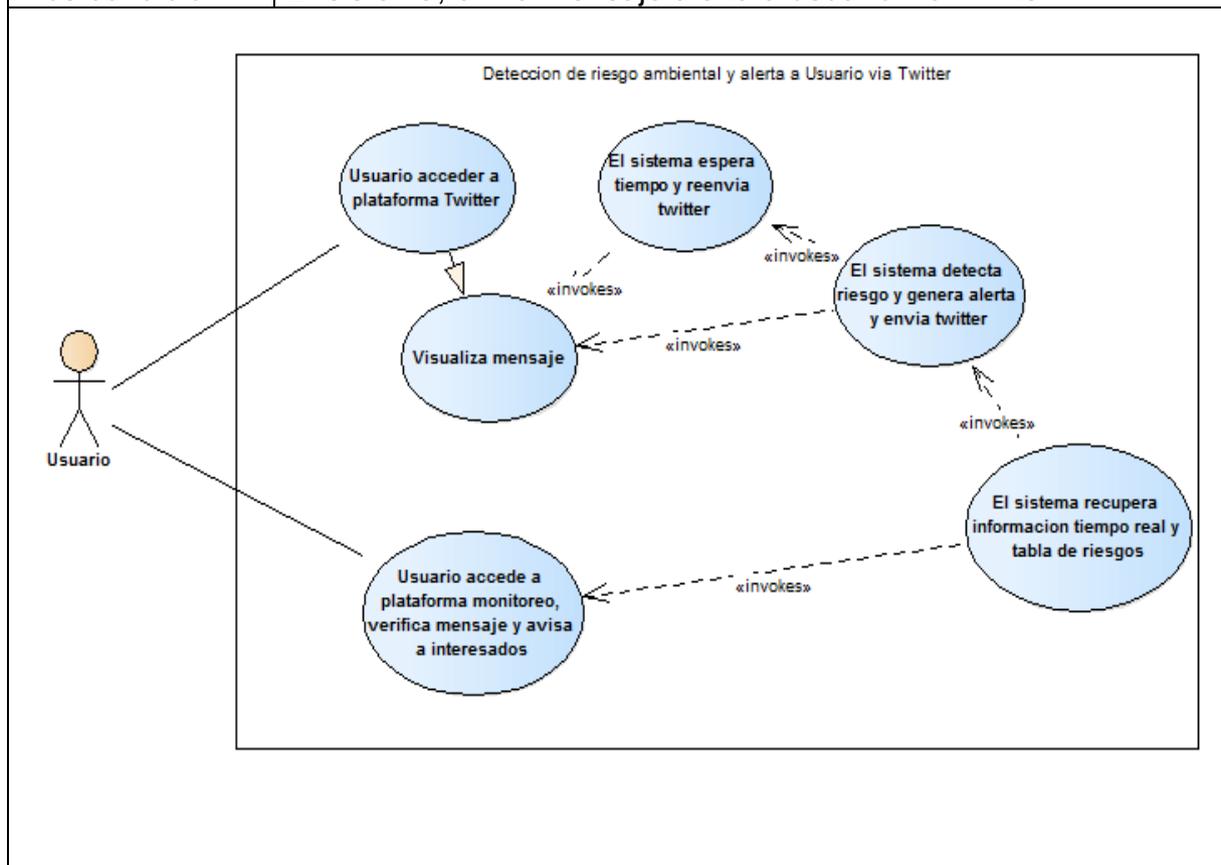
Caso de Uso asociado a requerimiento RF002		
Nombre	El sistema debe recuperar datos de mediciones almacenadas en BD y presenta gráficos históricos del día y semana en dashboard	
Identificación ID	CU2	
Actor	Usuario y Sistema	
Descripción	El usuario deberá ingresar a la plataforma de mediciones históricas y el sistema debe rescatar desde BD los registros diarios o semanal de mediciones y desplegar en gráficos la información.	
	Eventos Actor	Eventos de Sistema
Flujo Principal	1. Usuario accede al sitio web 2. Ingresa a la interfaz de monitoreo histórico 3. Selecciona periodo a visualizar	3. El sistema recupera información de BD de medición y posiciones diaria o semanales

		4. El sistema despliega en gráficos las mediciones diarias o semanal
Flujo Alternativo	4. Usuario debe cambiar de periodo de visualización 5. El usuario aborta el proceso el proceso de visualización	5. El sistema envía mensaje de periodo no disponible para desplegar 6. El sistema envía mensaje a usuario para informar que los datos no están disponibles
Precondición	Usuario debe acceder a navegador web y navegador móvil, actualización de BD y módulo de la recepción de data	
Postcondición	El sistema, arroja la información de mediciones histórica desplegada en gráficos	



Caso de Uso asociado a requerimiento RF003	
Nombre	El sistema debe alertar al interesado mediante él envió de un Twitter cuando el sistema detecte el aumento de los índices de contaminación
Identificación ID	CU3
Actor	Usuario y Sistema
Descripción	El sistema avisa mediante Twitter al usuario si algún nodo ha sobrepasado los niveles de riesgo para los niños
	Eventos Actor
	Eventos de Sistema

Flujo Principal	1. Usuario accede a plataforma Twitter 2. Ingresa a cuenta Twitter 3. Visualizar mensaje	3. El sistema recupera información de modulo tiempo real 4. El sistema acceder a plataforma Twitter y envía mensaje con alerta de riesgo
Flujo Alternativo	4. Usuario visualiza en mapa tiempo real indicadores riesgosos y avisa a jardín infantil del riesgo	1. El sistema reenvía mensaje temporizado si no detecta conexión con Twitter
Precondición	Usuario debe acceder a Twitter, módulo de datos y módulos de WebSockets, api Twitter	
Postcondición	El sistema, envía mensaje alerta a usuario vía Twitter	



8.1.6 Requerimientos no funcionales

ID	RNF_001
Nombre	El sistema deberá ser intuitivo en sus interfaces web
Características	Usabilidad- aprendizaje
Descripción	El usuario de aplicación web realiza la monitorización de mediciones en forma directa, sin necesidad de autenticarse, ya que la información es abierta para el público general. La interfaz debe ser gráfica y simple, sin la necesidad de ingresar datos en forma manual, solo acceder en base a botones o link

Trazabilidad	Director de Carrera, Arquitecto, Usuario
Dependencias	RN_002
Prioridad del Requerimiento: <input checked="" type="checkbox"/> _X__Alta <input type="checkbox"/> __Media Alta <input type="checkbox"/> __Media <input type="checkbox"/> __Media Baja <input type="checkbox"/> __Baja	

ID	RNF_002
Nombre	El sistema deberá interoperar con los sistemas BD locales, API Externas, y plataforma Arduino
Características	Funcionalidad –Interoperabilidad
Descripción	La plataforma de monitoreo realiza la recepción de datos desde plataforma Arduino y almacena los datos en BD, e interactúa con la API de Google Map para visualizar la información y además interactúa con API de Twitter para enviar alerta a usuarios interesados
Trazabilidad	Director de Carrera, Arquitecto
Dependencias	RN_001
Prioridad del Requerimiento: <input checked="" type="checkbox"/> _X__Alta <input type="checkbox"/> __Media Alta <input type="checkbox"/> __Media <input type="checkbox"/> __Media Baja <input type="checkbox"/> __Baja	

ID	RNF_003
Nombre	El sistema deberá proporcionar tiempos óptimos de respuesta en todas las consultas realizadas por web
Características	Eficiencia- comportamiento de tiempo
Descripción	Con el fin facilitar la experiencia del usuario con la interfaz, se debe considerar que el sistema entregue respuestas rápidas a consultas a BD
Trazabilidad	Director de Carrera, Arquitecto, Usuario
Dependencias	RN_001, RN_002, RN_003, RN_004
Prioridad del Requerimiento: <input type="checkbox"/> __Alta <input checked="" type="checkbox"/> _X__Media Alta <input type="checkbox"/> __Media <input type="checkbox"/> __Media Baja <input type="checkbox"/> __Baja	

8.1.7 Matrices de Trazabilidad

	RF	RF1	RF2	RF3
Objetivos Específicos Medibles (OEM)				
OEM1: Aumentar nodos de medición de variables medioambientales en una comuna, para reducir la incertidumbre en la elección de jardines infantiles			X	X
OEM2: Aumentar porcentaje información medioambiental desplegada vía web en tiempo real.	X			
OEM3: Reducir el tiempo de aviso y alerta de eventos de riesgos de salud de los niños				X

	RF	RF1	RF2	RF3
Casos Usos				
CU1: Obtener información de mediciones ambientales georeferenciada en tiempo real	X			
CU2: C El sistema debe recuperar datos de mediciones almacenadas en BBDD y presentar gráficos históricos del día y semana			X	
CU3: El sistema debe alertar al interesado mediante Twitter sobre el aumento de los índices de contaminación				X

	RF	RF1	RF2	RF3
Actividades del sistema				
A1: Capturar data en nodos de medición		X	X	X
A2: Desplegar data en dashboard		X		
A3: Enviar data en tiempo real a los clientes web		X		X
A4: Almacenar data en BD			X	
A5: Desplegar información histórica en gráficos			X	
A6: Detectar niveles riesgos de mediciones		X		
A7: Enviar alerta vía redes sociales		X		X

8.1.8 Atributos de Calidad y Métricas

Atributo de Calidad (Usabilidad)	
Objetivo	Asegurar que el sistema proporcione al usuario un proceso de monitoreo en menos de 5 segundos
Función	Analizar la capacidad de la plataforma de asegurar que la monitorización de las variables medioambientales se realice en forma simple y expedita, no superando los 5 segundos el proceso total de obtener la información de cada estación y los valores históricos, además de las

	alarmas. Específicamente, se debe verificar que la aplicación proporcione facilidad en la visualización de la información mediante la utilización mínima de controles o formularios de llenado
Propósito	Controlar la experiencia de usuario en el proceso de monitorización de datos tiempo real e históricos
Con respecto a	El nivel de usabilidad de la interfaz web y de nodos medición
Contexto	Uso de la aplicación en entorno web
Punto de Vista	Área de monitoreo de aplicativos
Q1	¿Qué tiempo medio de respuesta ofrece el sistema toda vez que se ha detectado un acceso a la plataforma web?
Métricas	
Nombre de la métrica	TMM: Tiempo medio de acceso a información de monitorización y alertas
Entidad	Sistema web, aplicación Arduino, BD, API redes sociales
Atributos	Eficiencia >> Usabilidad
Unidad de Medida	Segundos
Escala de medición	0s-5s: Bueno si es >10 s: Malo
Forma de medir	Se considera la ejecución de monitoreo en tiempo real, acceso a información histórica y sistema de alertas en ese intervalo de tiempo. $TMM = (T1+T2+...+Tn) / TA$, donde: T1, T2, ..., Tn: Tiempo tomado en el proceso de monitorización TA: Cantidad total de accesos

Atributo de Calidad (Interoperabilidad)	
Objetivo	Asegurar que el sistema reciba data de mediciones, acceso a BD y sistema alerta Twitter en el 98% de las operaciones
Función	Analizar la capacidad de la plataforma de asegurar que monitorización se realice en forma transparente con los sistemas de captura, BD, y API de redes sociales. Específicamente, se debe verificar que la aplicación converse fluidamente con las plataformas de medición, BD, Twitter mediante protocolos Zigbee, Wifi, WebSockets, Apis
Propósito	Controlar la comunicación y transferencia de datos que se produce en el proceso de monitorización
Con respecto a	El nivel de interoperabilidad de aplicación, plataforma de medición, BD, Apis
Contexto	Uso de la aplicación en entorno web de monitorización
Punto de Vista	Área de monitoreo de aplicativo
Q1	¿Qué porcentaje de consultas realizadas a plataformas de medición, BD, y Twitter es inconsistente o no se logra concretar es posible de detectar sobre un total de operaciones consultas a las monitorizaciones diarias?

Métricas	
Nombre de la métrica	CMNE: Consulta a nodos de medición, BD, Twitter no exitosa
Entidad	Sistema de medición, WebSockets, BD, api Twitter
Atributos	Funcionalidad >>Interoperabilidad
Unidad de Medida	Porcentaje
Escala de medición	0%-100%: Inaceptable
Forma de medir	Se considerará aquellos casos en los que se detecte alguna consulta de usuario que no sea respondida por la plataforma de medición, BD o Api debido a inconsistencia de formatos, accesos a plataformas web externas, actualización de datos o problemas de comunicación. CDNE=(A/B) *100, donde: A: Total de casos de consultas a plataforma, api no exitosas B: Total de casos de operaciones de consulta a plataforma

Atributo de Calidad (Eficiencia)	
Objetivo	Asegurar que el sistema proporcione respuestas al usuario en promedio de 5 segundos
Función	Analizar la capacidad de la plataforma de asegurar que el proceso de consultas de monitorización y alerta se realice en forma eficiente, no superando los 5 seg. Específicamente, se debe verificar que la aplicación proporcione tiempos de operación mínimos en el proceso de consultas de información de estaciones de medición, visualización de data y gráficos históricos y alerta
Propósito	Controlar la eficiencia de la plataforma en la generación de proceso de monitorización y alerta
Con respecto a	El nivel de eficiencia de la interfaz web, estaciones de medición, integración de servicios y BD
Contexto	Uso de la aplicación en entorno web
Punto de Vista	Área de monitoreo de aplicativos
Q1	¿Qué tiempo medio de respuesta ofrece el sistema toda vez que se ha generado un evento en la plataforma web generado por el usuario?
Métricas	
Nombre de la métrica 1	TME: Tiempo medio de resultado a evento de operación
Entidad	Sistema web, aplicaciones, integración de servicio y BD
Atributos	Eficiencia
Unidad de Medida	Segundos
Escala de medición	0s-5s: Bueno >10 s: Malo
Forma de medir	Se considera la ejecución de eventos de consulta y accesos de clientes a la interfaz web. Todo esto se hará sobre una base de tiempo total de estudio $TME = (T1+T2+...+Tn) / TA$, donde: T1, T2, ..., Tn: Tiempo tomado en la generación de eventos TA: Cantidad total de accesos

8.2 Análisis y Diseño de Arquitectura

8.2.1 Introducción

En este apartado, se describe la etapa de diseño de la solución, en particular se definen las ideas generales de la estructura que debe poseer el sistema en cuanto a la organización e interacción de componentes físicos y lógicos que lo constituyen.

Para tal efecto, se organiza la descripción en tres ámbitos de diseño. El primero corresponde a la *arquitectura de sistemas* que representa una organización global de los componentes, poniendo énfasis en la descripción de la estructura general de la solución. A continuación, se describe un ámbito denominado *arquitectura de software*, que representa una mirada más específica de los componentes lógicos de la solución y su interacción. Y finalmente un ámbito denominado *arquitectura de hardware*, que se refiere a una descripción física de la organización e interacción de los componentes de hardware involucrados.

8.2.2 Arquitectura de Sistemas

Como se menciona en el capítulo cuarto de revisión de marco conceptual, la arquitectura se encarga de definir una organización global de componentes y la interacción entre ellos. En este sentido, cobra importancia el concepto de estilos arquitectónicos que permite disponer de plantillas o modos de organización estándar y útiles para resolver cierto tipo de necesidades comunes a los sistemas.

En particular, un estilo de organización de componentes muy utilizado en los sistemas de información, debido a su simplicidad, es la organización por capas. Este tipo de estilo de arquitectura define funcionalidades y responsabilidades perfectamente acotadas en diferentes niveles de abstracción, que interactúan organizadamente en forma jerárquica del nivel inferior a superior y viceversa, logrando un flujo de información controlado, niveles de modularización adecuados que ayudan a la mantenibilidad y seguridad en toda la estructura.

En este sentido, la arquitectura por capas representa una excelente alternativa para este proyecto, ya que dispone de componentes de hardware y software en distintos niveles que deben interactuar en conjunto, en base a responsabilidades y funciones bien definidas y es consistente con soluciones IoT. El modelo descrito por World Forum citado en **(Cazila & Junco, 2015)**, representa una excelente alternativa de modelo a seguir para este proyecto de IoT.

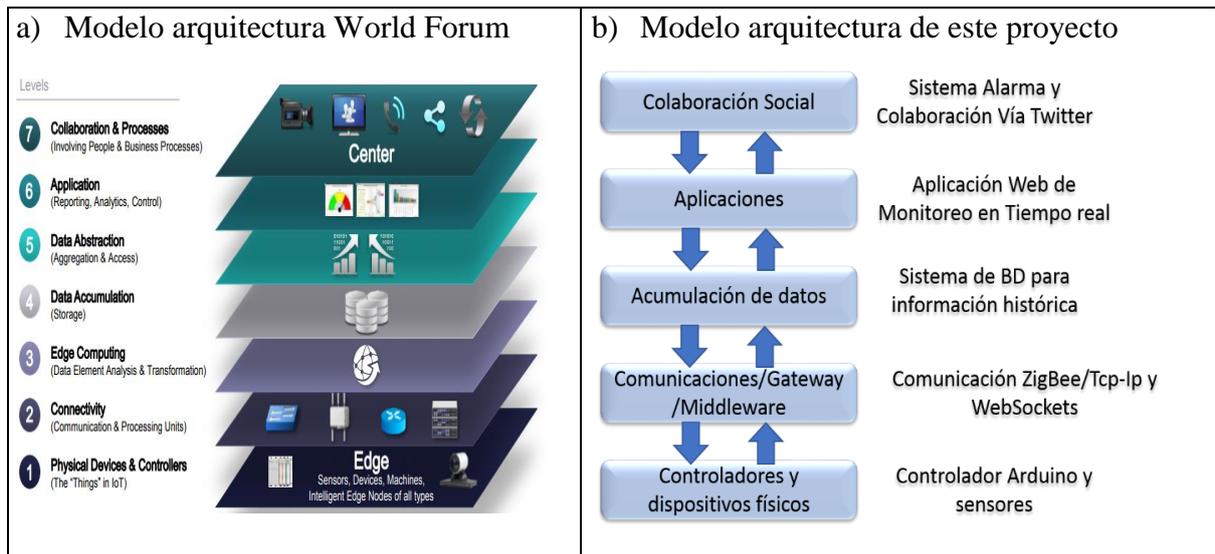


FIGURA 33: Esquema Arquitectura del Proyecto (Fuente: Elaboración propia)

8.2.2.1 Descripción de capas del sistema

El sistema propuesto, se organizará en cinco capas, las cuales se detallan a continuación:

Capa 1: Se encarga de cobijar los sistemas de bajo nivel que permiten capturar los datos de interés, es decir contiene a los nodos de medición que se instalarán en los jardines infantiles. Cada nodo de medición contendrá una unidad de control (Arduino) y tres sensores que permitirán medir la calidad del aire, el nivel de ruido y la posición del nodo. Además, su respectiva unidad comunicaciones y energía (baterías, estación solar o conexión a la red eléctrica).

Capa 2: Esta capa contiene a los sistemas de interfaz física y lógica que permiten la comunicación entre los módulos de hardware de bajo nivel y los módulos de software de mayor nivel. Aquí se distinguen dos tipos de interfaz.

- La interfaz de bajo nivel encargada de comunicar los datos generados en el microcontrolador con el servidor de aplicaciones. Esta interfaz, se realiza mediante el protocolo comunicación (Zigbee-Serial) mediante una comunicación inalámbrica de largo alcance.
- La segunda interfaz se encarga de la interconexión de más alto nivel, en particular la comunicación entre el servidor de aplicaciones con los clientes web. Por las características especificadas del sistema, esta comunicación debe ser

entimiento real, debido a que el cliente debe tener acceso a las lecturas de los sensores en cada instante de tiempo además del registro histórico que puede ser fuera de línea. Esta comunicación será implementada mediante el protocolo de WebSockets que trabaja sobre tcp-ip.

Capa 3: La tercera capa, se encarga del sistema de persistencia de los datos. Esta capa se requiere debido a la necesidad de registrar y desplegar información histórica del comportamiento de las mediciones. En este sentido, la capa se implementará con tecnología NoSQL, debido a la gran cantidad de datos involucrado, pero a su vez la simpleza del modelo de datos requerido.

Capa 4: La cuarta capa, se refiere a la plataforma web que desplegará la información a los usuarios mediante una interfaz web. Esta capa se encarga de recibir los datos desde la capa inferior y presentar los datos mediante un dashboard para que el usuario acceda vía web a la monitorización en tiempo real de los nodos de medición y además acceder a información historia de las variables ambientales presentadas mediante gráficos.

Capa 5: Finalmente la última capa es la encargada de interactuar con la comunidad vía redes sociales, en particular mediante el envío de alertas de riesgo de salud para los niños mediante el aviso a través de mensajes de Twitter

8.2.2.2 Filosofía de interacción del sistema

La interacción entre los componentes de cada capa se implementará mediante la filosofía conocida como cliente-servidor, la cual se basa en los protocolos de solicitud respuesta entre cada componente.



FIGURA 34: Filosofía Cliente-Servidor (Fuente: Elaboración propia)

Como se puede observar en el esquema anterior, el sistema dispone de tres componentes principales. El primer sub-sistema corresponde al Servidor de Datos, que contiene a los nodos de medición (tres nodos) y su sistema de comunicación (capa 1 y 2 de la arquitectura), el segundo corresponde al sub-sistema denominado Cliente de datos y servidor de aplicaciones (capa 3 y 4) que cumple la doble función, es decir ser cliente del servidor de datos y a su vez ser servidor de la aplicación para los clientes web. El tercer sub-sistema se denomina cliente de aplicación (capa 5), que corresponde a los programas navegadores que permiten a los clientes visualizar la información desplegada y la comunicación vía Twitter.

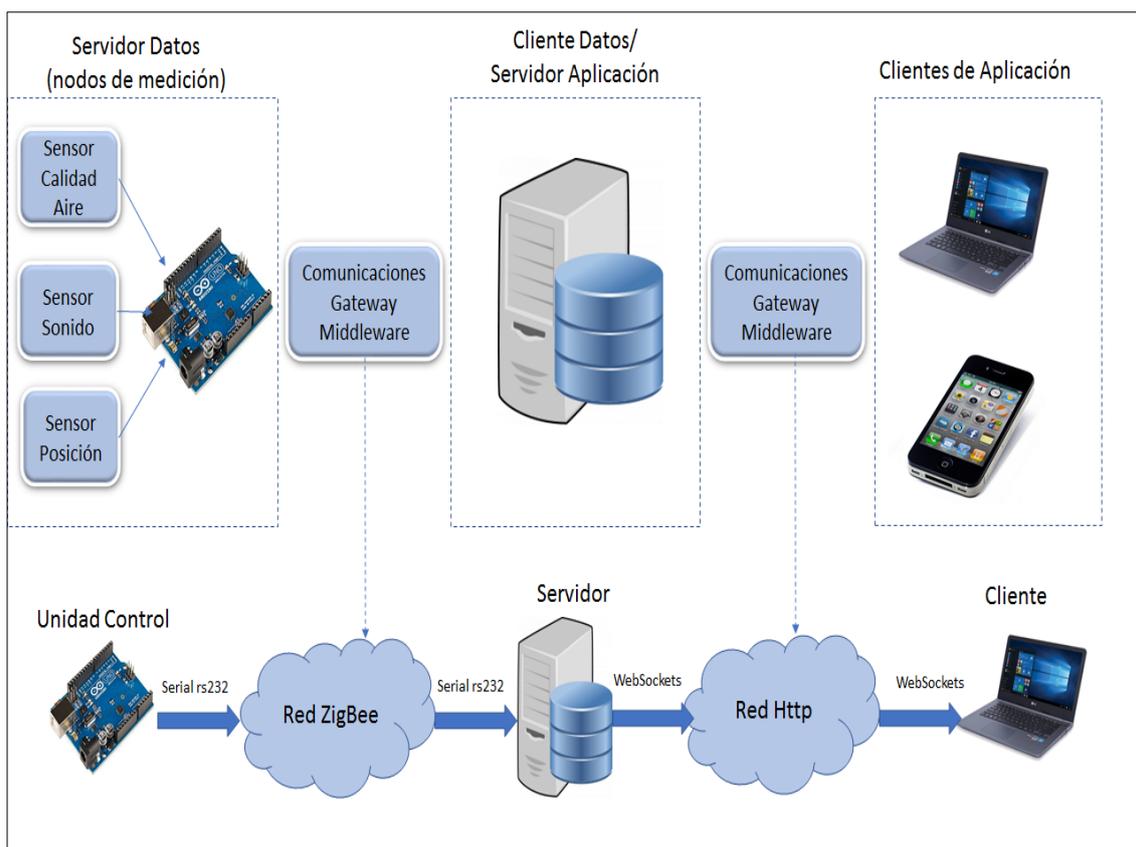


FIGURA 35: Esquema arquitectura de componentes de sistema
(Fuente: Elaboración propia)

8.2.3 Arquitectura de Software

La arquitectura de software corresponde a la presentación de un conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución.

El diseño se implementará mediante modelos gráficos según las recomendaciones de la arquitectura de software del modelo de vistas 4+1 (Kruchten, 1995). Este modelo describe la arquitectura del software usando cinco vistas concurrentes. Tal como se muestra en la Figura nº 36, cada vista se refiere a un conjunto de intereses de diferentes Stakeholders del sistema.

- La vista lógica describe el modelo de objetos del diseño cuando se usa un método de diseño orientado a objetos.
- La vista de procesos describe los aspectos de concurrencia y sincronización del diseño.
- La vista física describe el mapeo del software en el hardware y refleja los aspectos de distribución.
- La vista de desarrollo describe la organización estática del software en su ambiente de desarrollo.
- La vista de escenarios se refiere a la interacción del sistema con los usuarios

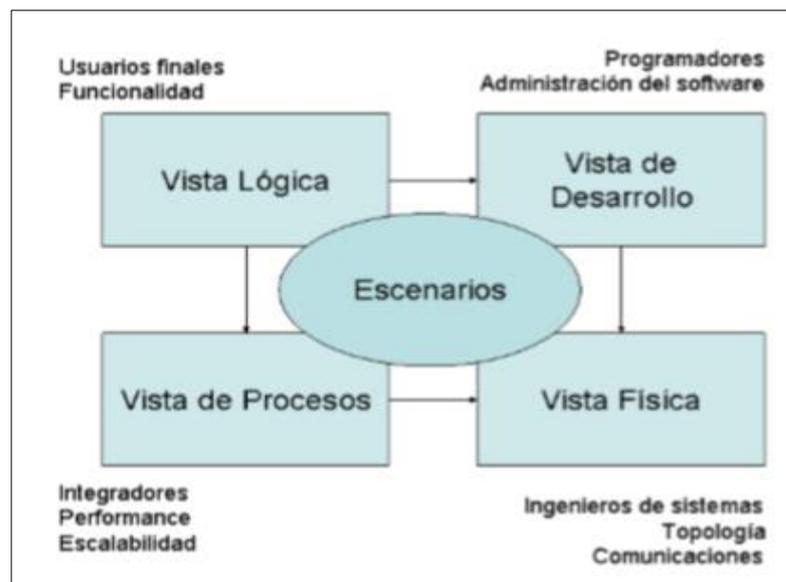


FIGURA 36: Esquema Modelos de “4+1” vistas
(Fuente: Kruchten, 1995)

Los modelos son presentados por vistas o diagramas, con notaciones gráficas como el Lenguaje de modelado unificado (UML).

8.2.3.1 Vistas de arquitectura de software

Vista de Escenarios: Diagrama Casos de uso del Sistema (Simplificado)

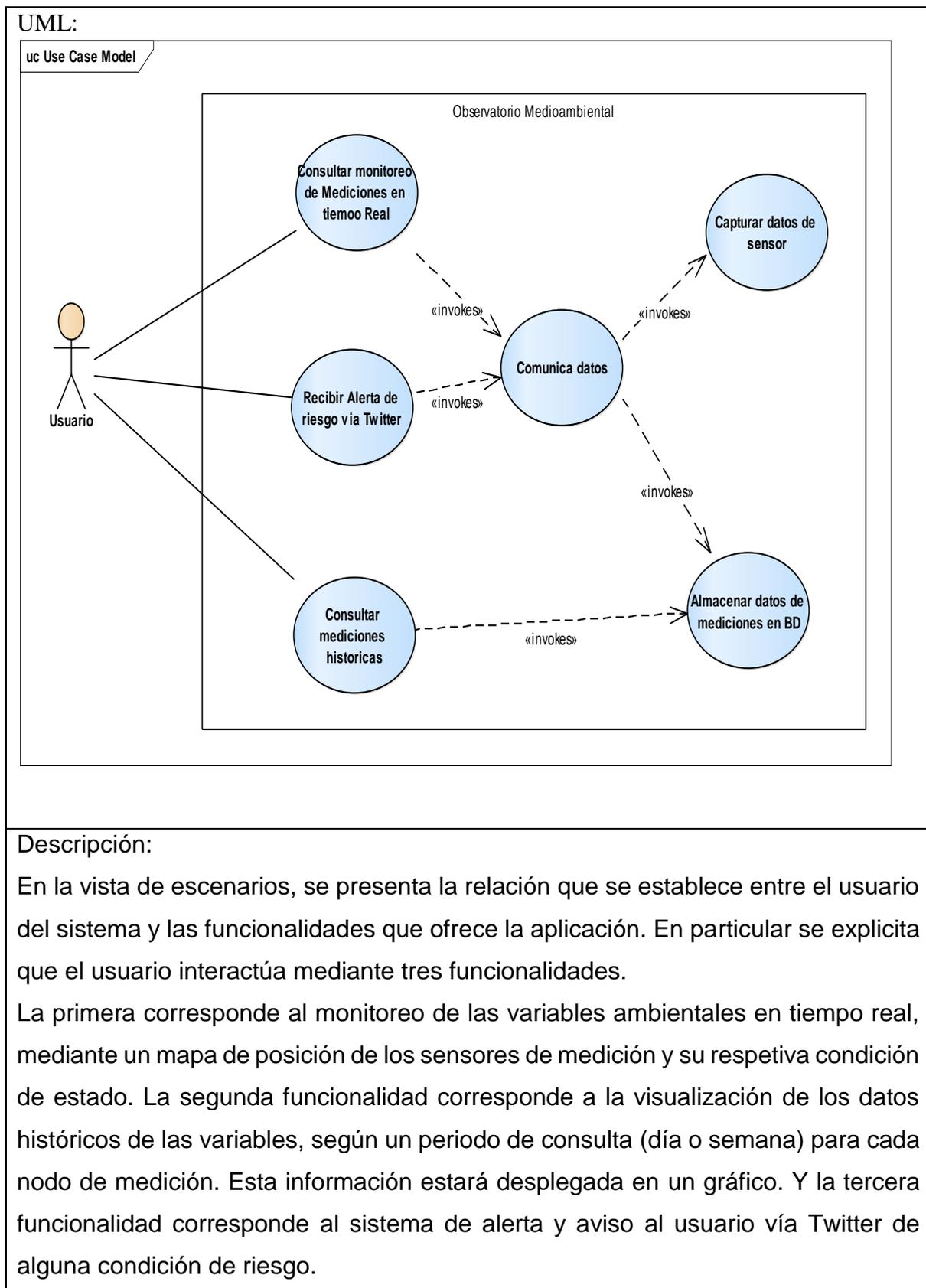
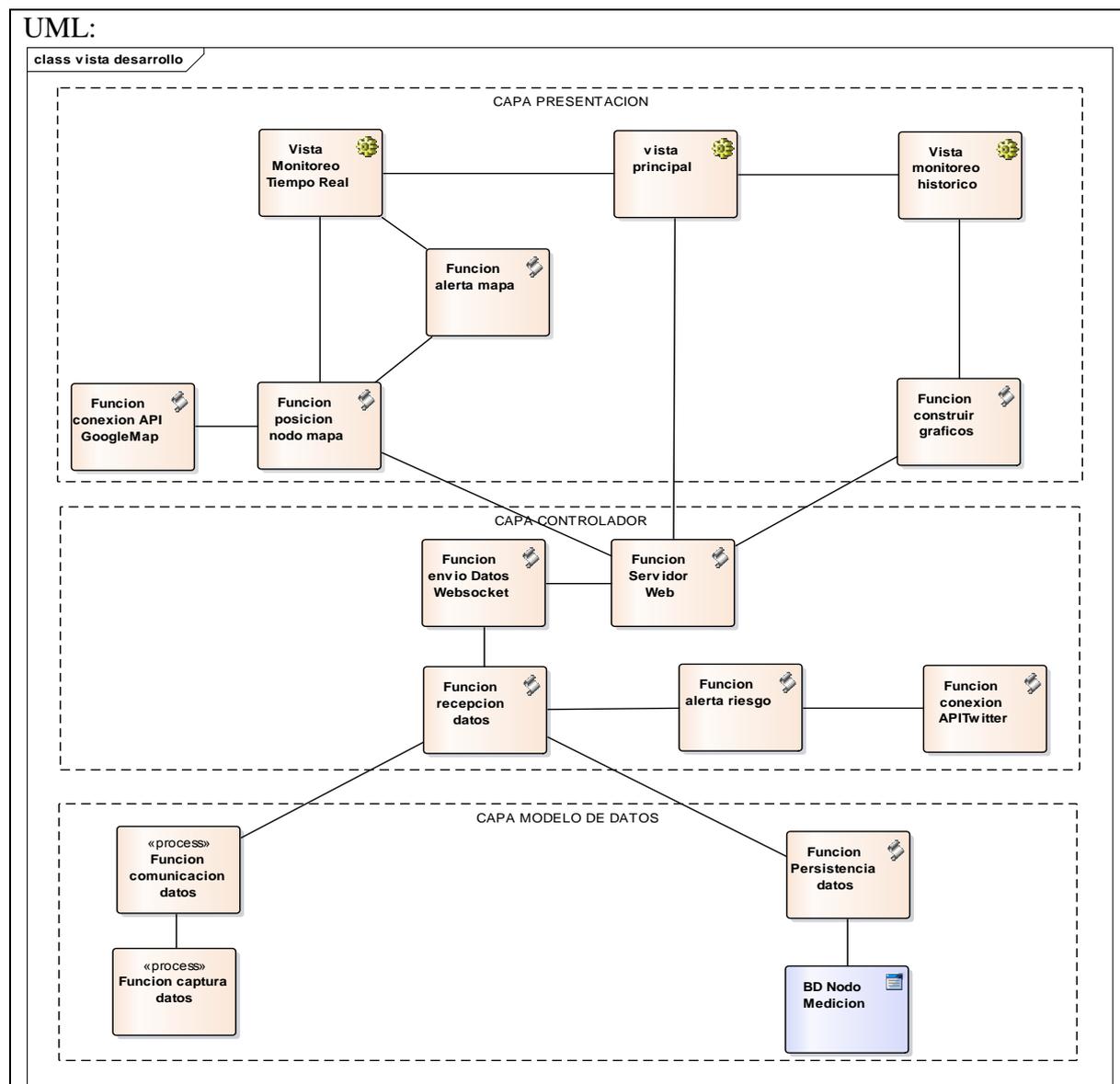


FIGURA 37: Esquema vista Escenarios (Fuente: Elaboración propia)

Vista Lógica: Diagrama de Clase/Funciones

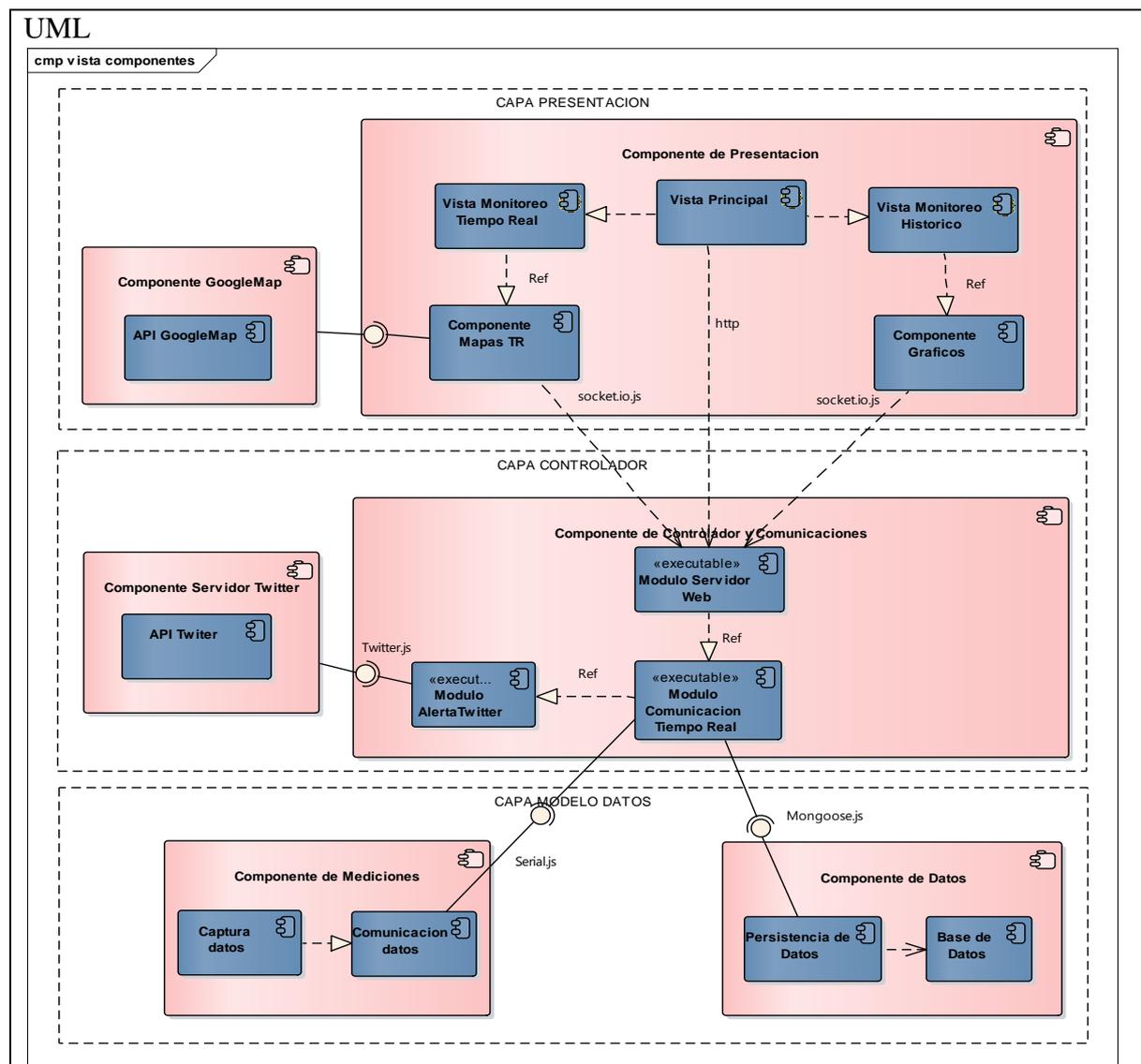


Descripción:

Esta vista de desarrollo presenta a las funciones que participan en cada uno de los módulos y componentes del sistema. Específicamente la idea es visualizar a un mayor nivel de detalle, la operaciones o funciones (JavaScript) que permiten cumplir con los requerimientos definidos previamente. Estas funciones se relacionan con actividades de captura, procesamiento y envío de datos (componente medición), recepción de datos, persistencia de datos, sistema de alerta, conexión con API y servidor web (componente controlador) y funciones dedicadas a presentar la información ya sea en el mapa de posición o mediante gráficos históricos vía interfaces web.

FIGURA 38: Esquema vista Lógica (Fuente: Elaboración propia)

Vista de Desarrollo: Diagrama de componentes general



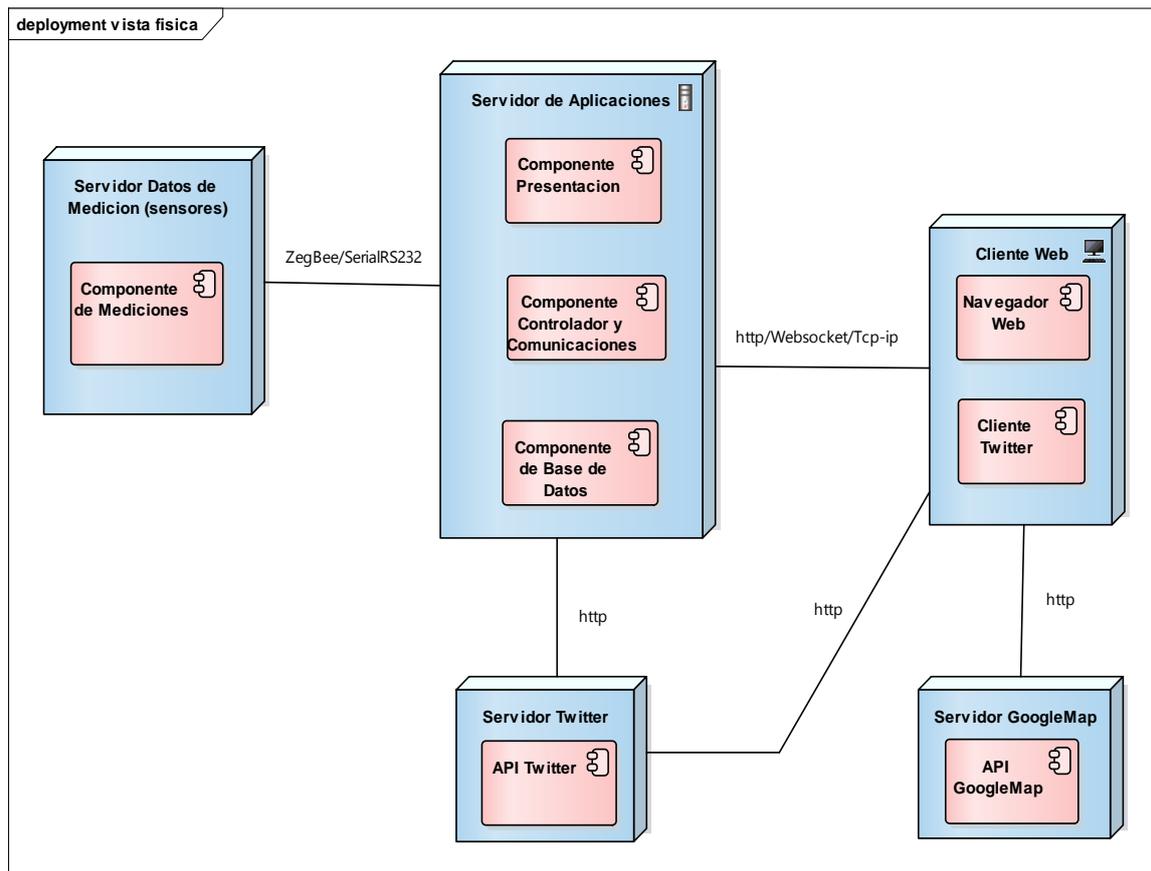
Descripción.

En esta vista, se visualizan los cuatro componentes de software que componen al sistema y su interacción con los componentes externos (API GoogleMap y Twitter). Además, se puede apreciar la arquitectura denominada cliente pesado, en donde el cliente web asume responsabilidades de procesamiento relevante en el sistema. Cada componente contiene una serie de módulos que cumplen diferentes tareas, por ejemplo, la de captura y transmisión de datos (componente de medición), la comunicación en tiempo real, servidor web, persistencia de datos, alerta Twitter (componente controlador y comunicación) y los módulos del componente presentación, por ejemplo, el de preparación de datos para mapas, gráficos y sus respectivas vistas o template.

FIGURA 39: Esquema vista Desarrollo (Fuente: Elaboración propia)

Vista Física: Diagrama de Despliegue

UML:



Descripción:

El diagrama de despliegue identifica la presencia de los tres nodos principales que participan en el sistema.

El primero corresponde a un nodo servidor de mediciones que captura los datos ambientales proveniente de los sensores y que son enviados al nodo central.

El nodo central se encarga de recibir la información del nodo de mediciones (actúa como cliente) para luego almacenar los datos en una Base de Datos local y posteriormente enviar los datos (actúa como servidor) al nodo cliente vía WebSockets. Además, el nodo central interactúa con el nodo servidor de Twitter para enviar una alerta de riesgo vía su API.

También se presenta el nodo cliente web, que despliega las vistas de monitoreo de datos en tiempo real e histórico y además se conecta con la API de GoogleMap para presentar la información mediante un mapa de posición utilizando las coordenadas de las estaciones de medición.

FIGURA 40: Esquema vista Física (Fuente: Elaboración propia)

Vista de Procesos Diagrama de Secuencia

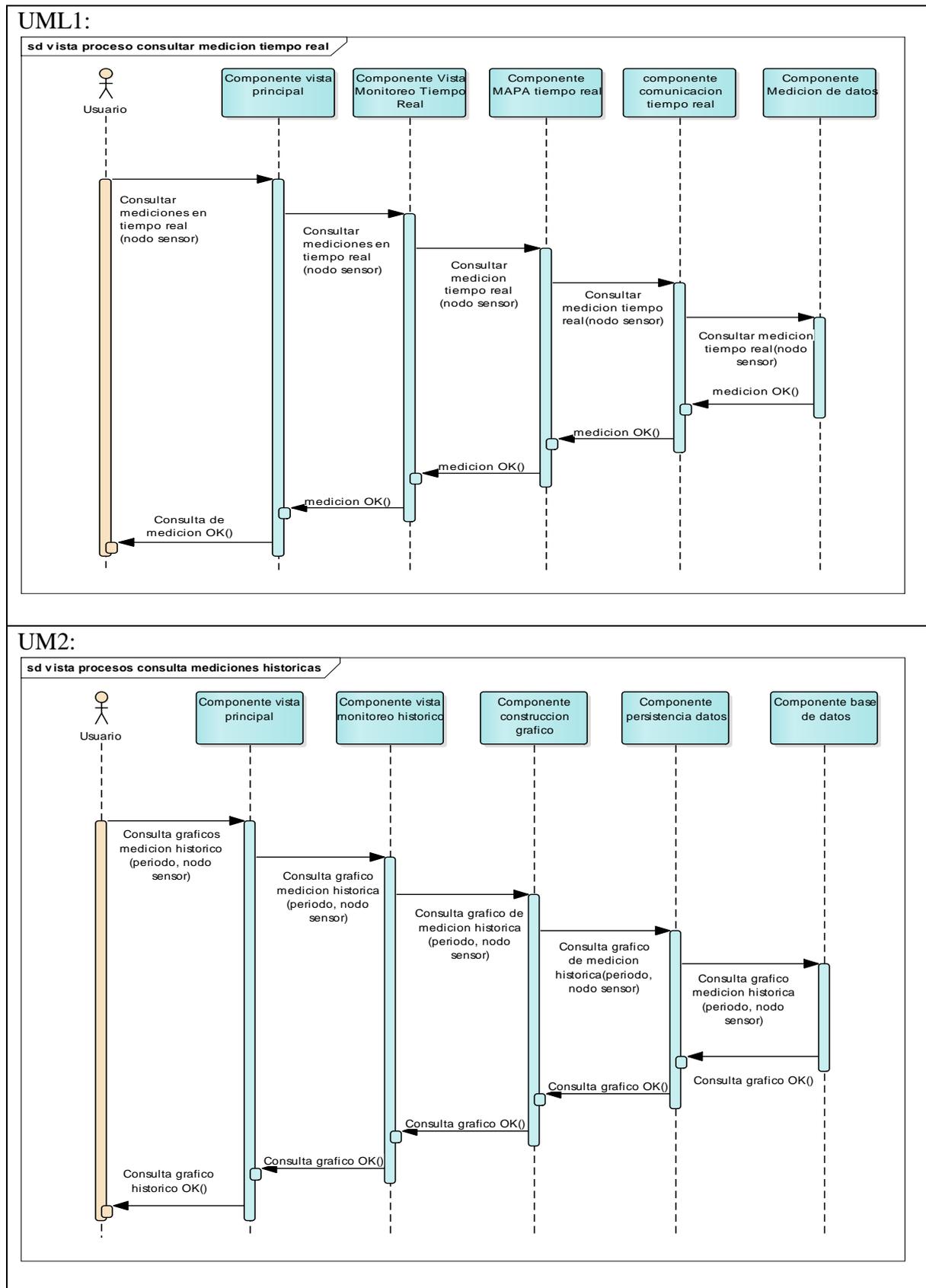
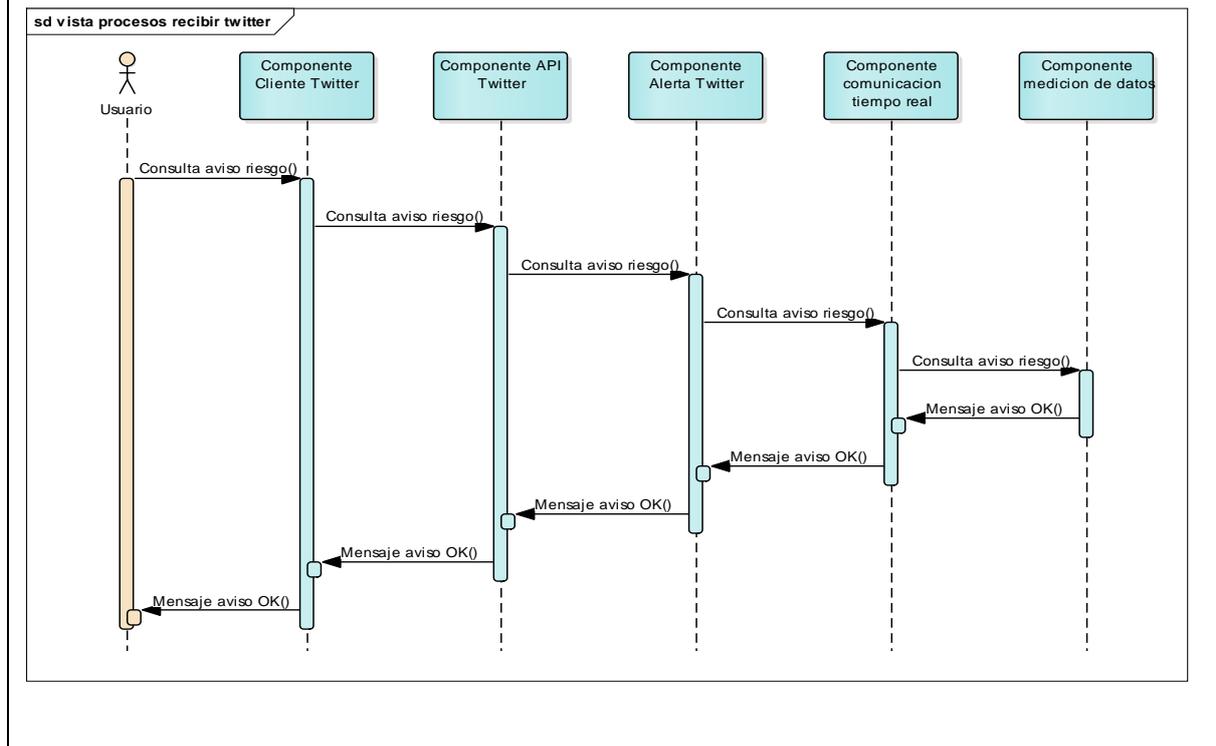


FIGURA 41: Esquema vista de Procesos (Fuente: Elaboración propia)

UML3:



8.2.3.2 Descripción de Componentes de software

Componente SW	Modulo	Descripción	Componente relacionado
Componente de mediciones	Captura de datos	Contiene la lógica de captura de los datos en el microcontrolador Arduino (puerto analógico) y su respectivo empaquetado	Comunicación de datos
	Comunicación de datos	Contiene la logia de envió mediante el puerto de comunicaciones	Componente Controlador y comunicaciones
Componente Controlador y comunicaciones	Comunicación tiempo real	Contiene la lógica para recibir los datos desde sensores y además de	Componente de mediciones

		enviar los datos mediante WebSockets al cliente	
	Persistencia de datos	Contiene la lógica de preparación de datos y conexión con base de datos	Comunicación tiempo real
	Alerta Twitter	Contiene la lógica de decisión de para la generación de alerta de riesgos de mediciones	Comunicación tiempo real
	Servidor web	Contiene la lógica para el despliegue de la información mediante un servidor web, con su respectivo routing y seguridad.	Comunicación tiempo real Persistencia de datos
Componente de datos	Base de datos	Contiene los registros de las mediciones para su consulta	Persistencia de datos
Componente presentación	Construcción Mapa	Contiene la lógica de preparación y conexión con librerías, imágenes y API para el despliegue de la información	Comunicación tiempo real API GoogleMap
	Construcción Gráficos	Contiene la lógica de preparación y conexión con librerías imágenes	Persistencia de datos
	Vista principal	Contiene la lógica para mostrar la vista	Servidor web Vista Tiempo

		principal.	real e Históricos
	Vista Tiempo real	Contiene la lógica para mostrar la vista de los datos en desplegada en mapa en T.R	Construcción Mapa
	Vista Históricos	Contiene la lógica para mostrar la vista de datos históricos mediante gráfico.	Construcción Gráficos
Componente servidor Twitter	API Twitter	Contiene la lógica de autenticación y conexión con API	Comunicación tiempo real
Componente servidor GoogleMap	API GoogleMap	Contiene la lógica de autenticación y conexión con API	Construcción Mapa

TABLA 11: Componentes de software (Fuente: elaboración propia)

8.2.3.3 Descripción de conectores de software

Componentes	Conector	Descripción
Componente de mediciones conectado con componente Controlador	Librería serialport NodeJS Librería Net NodeJS	Estos conectores permiten comunicar a los componentes de medición y controlador mediante librerías externas a la plataforma NodeJS
Componente Controlador conectado a base de datos	Librería MongoDB de NodeJS	Librería que permite conectar una aplicación NodeJS con una base de Datos MongoDB
Componente Controlador conectado a Componente servidor Twitter	Librería Twitter de NodeJS	Librería de NodeJS que permite conectarse a la API de Twitter
Componente Controlador conectado a Componente presentación	Librería socket.io de NodeJS Librería http y Express de NodeJS	Librería de NodeJS que permite conectarse a mediante protocolo de WebSockets a un cliente web

Componente presentación conectado a Componente servidor GoogleMap	Librería JavaScript que permite conectarse a API GoogleMap	Librería de JavaScript que permite conectarse a la API
---	--	--

TABLA 12: Conectores de software (Fuente: elaboración propia)

8.2.3.4 Matrices de Trazabilidad

RN, RF, RNF	RN1	RN2	RN3	RN4	RF1	RF2	RF3	RNF1	RNF2	RNF3
Módulos de SW										
M1: Captura datos			X		X	X	X			
M2: Comunicación dato			X	X	X	X	X			
M3: Persistencia dato	X					X			X	
M4: Base de datos	X					X			X	X
M5: S. Comunicación TR			X		X		X			X
M6: Alerta Twitter							X	X		
M7: Servidor web		X			X	X				X
M8: C. Comunicación TR					X					X
M9: Gráficos						X		X		
M10: Mapas					X			X		
M11: Vista principal		X						X		
M12: Vista Tiempo real		X			X			X		
M13: Vista históricos		X				X		X		

TABLA 13: Matriz de Trazabilidad Módulos SW vs Requisitos RN, RF, RNF (Fuente: elaboración propia)

Módulos SW	M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8	M 9	M 10	M 11	M 12	M 13
Nodos HW													
N1: Servidor datos de medición	X	X											
N2: Servidor/Clientes de aplicación			X	X	X	X	X						
N3: Cliente Web								X	X	X	X	X	X
N4: Servidor GoogleMap												X	
N5: Servidor Twitter						X							

TABLA 14: Trazabilidad Nodos HW vs Módulos SW (Fuente: elaboración propia)

8.2.3.5 Detalles de Implementación de software y tecnología utilizada

Para el desarrollo de esta aplicación se han considerado diferentes lenguajes de programación, sistemas de bases de datos y librerías de comunicación y representación gráfica. Así como diferentes entornos de desarrollo.

En cuanto a lenguajes de programación se utilizarán tres lenguajes de programación, uno para los nodos de servidor de aplicaciones y cliente web mediante JavaScript implementada a través de Node.js. El segundo lenguaje se utilizará para la capa de presentación basado en HTML y el tercer lenguaje utilizado para el servidor de mediciones, en este caso el lenguaje para Arduino, específicamente lenguaje C.

A continuación, se describen cada una de las tecnologías de software que se utilizará.

Node.js: Herramienta lanzado en 2009, utiliza la máquina virtual V8 de Google, la misma que se utiliza en el navegador Google Chrome, para ejecutar código JavaScript del lado del servidor. Se basa en la programación orientada a eventos, cada vez que se realice una conexión, se reciban datos o se cierre una conexión se produce un evento.

Express js: Es una framework de desarrollo de aplicaciones web minimalista y flexible para Node.js que ofrece un router de URLs, facilidades para integrar motores de plantilla y middleware. Con estas características, entre otras, este framework facilita el desarrollo de aplicaciones y/o API's (Application Programming Interface) en Node.js.

MongoDB: Es un sistema de bases de datos no relacional lanzado en 2009 de código abierto que opera bajo la licencia GNU AGPL v3.0. En un sistema de base de datos no relacional la información no se almacena en tablas sino en documentos en BSON (Binary JavaScript Object Notation) que es una representación binaria de JSON (JavaScript Object Notation) con un esquema dinámico, la estructura es mutable en cualquier momento, haciendo que la integración de datos en algunos contextos sea realmente rápida y fácil. Una de las ventajas de MongoDB reside en que las consultas se hacen utilizando una sintaxis similar a la utilización de objetos en JavaScript.

Chart.js: La librería Chart.js es una de las grandes bibliotecas de JavaScript en línea, la cual ayuda a plasmar datos usando el elemento canvas de html5 para dibujar gráficos y tablas. Funciona con html5 por lo que soporta la mayoría de los browsers modernos, incluso funciona en dispositivos móviles. No depende de librerías externas y pesa tan solo 4.5kb. Chart.js es una librería open source bajo licencia MIT.

8.2.4 Arquitectura de Hardware

8.2.4.1 Descripción de la arquitectura de hardware

La arquitectura propuesta para la red de nodos de medición está basada en dos unidades que recolectan los datos de medición, para luego ser acondicionadas para su transmisión. Cada nodo dispone de tres sensores (sensor de audio, CO2 y de posición GPS), que son conectados a una unidad de control de tipo Arduino, junto con su respectiva interfaz de comunicación inalámbrica basada en tecnología zigbee o en su defecto wifi.

Los nodos de medición son coordinados por la estación de aplicaciones que dispone de un dispositivo receptor de las señales y además proporciona la identificación para cada emisor.

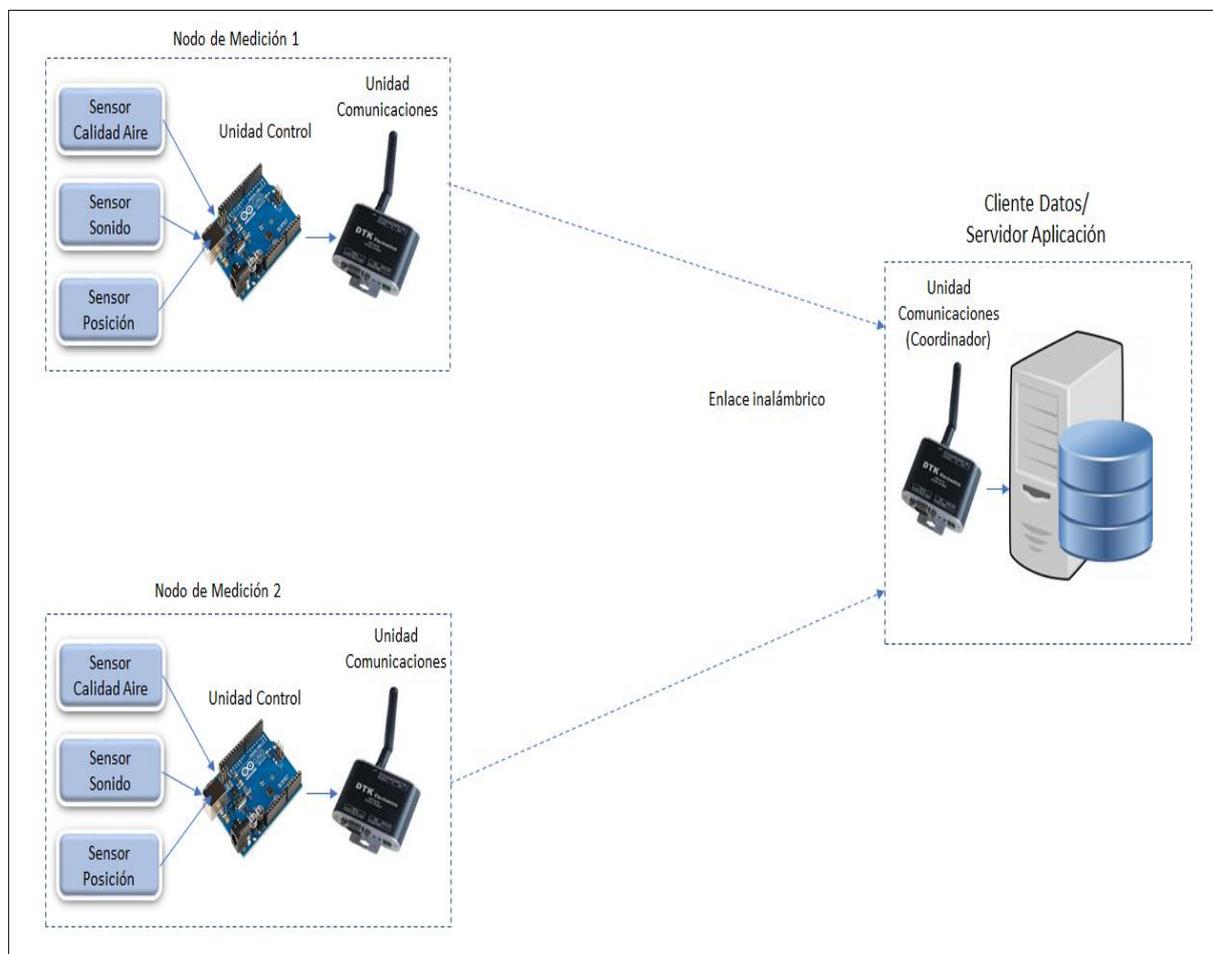
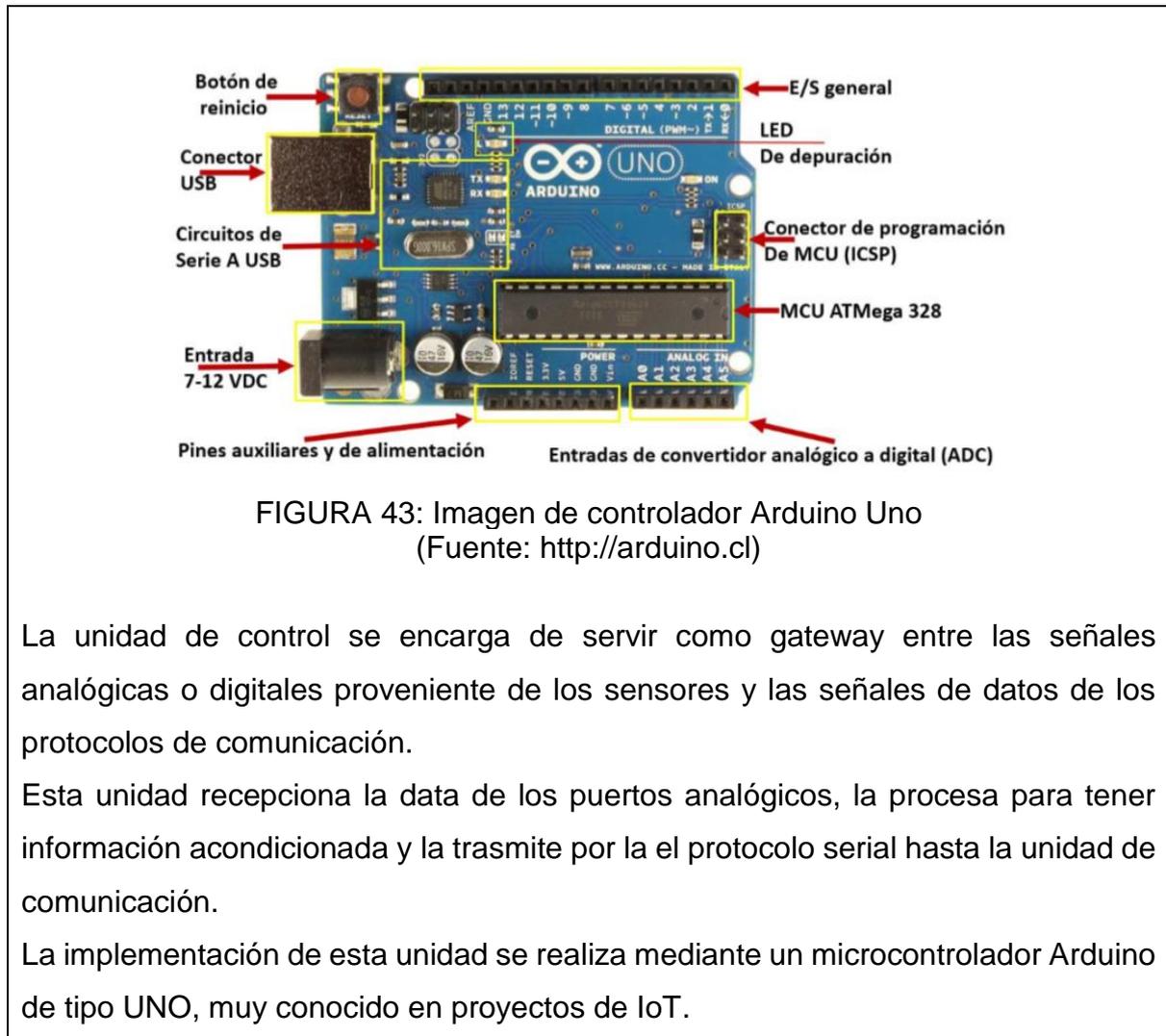


FIGURA 42: Esquema arquitectura de hardware
(Fuente: Elaboración propia)

8.2.4.2 Descripción de Componentes de hardware

Componente: Unidad de Control



Componente: Unidad de Comunicación



La unidad de comunicación que se utilizará corresponde a sistema inalámbrico basado en protocolo Zigbee. Este dispositivo cuenta con la posibilidad de transformar la señal zigbee a señal de protocolo serial RS232. Su alcance, según el fabricante es de 1600 metros y tiene la posibilidad de organizar una red donde se pueden configurar nodos coordinador y router.

Topología de la Red ZigBee:

La topología utilizada es estrella en donde se dispone de un nodo coordinador que está rodeado de dispositivos como se muestra en la Figura 40. Se puede configurar fácilmente. Puede soportar hasta 6000 dispositivos. Pero también hay algunas limitaciones: si coordinador no puede trabajar debido a algún problema técnico, entonces toda la red falla, ya que todo el tráfico pasa por el coordinador de la estrella. En conclusión, esta topología no ofrece una transmisión fiable.

Parámetros electrónicos:

- Voltaje de entrada Estándar: DC 5-12V
- Rango de temperatura: -40C ~ 85C
- Velocidad del puerto serial: 9600 bps, 19200 bps, 57600 bps, 115200 bps
- Radio frecuencia: 2.4G (2460MHz)
- Protocolo inalámbrico: Zigbee2007
- Distancia visual: 1600 Metros
- Corriente de trabajo Envío: 150mA (Máx.) 100mA Recepción: 55mA (Máx.)
- Sensibilidad del receptor: -110 dBm
- Chip: CC2530F256,256KFLASH
- Punto configurable: Se puede configurar como coordinador, enrutador
- Predeterminado: enrutador, ID PAN = 0x199B, Canal de radio = 22 (2460MHz)
- Interfaz: RS232 (D- / B, D- / A, GND)

Componente: Sensor de Gas



FIGURA 45: Imagen Sensor MQ-7
(Fuente: http://wiki.seeed.cc/Grove-Gas_Sensor-MQ7/)

Los sensores de gas de la serie MQ-x son sensores analógicos por lo que son fáciles de implementar con cualquier microcontrolador. Estos sensores son electroquímicos y varían su resistencia cuando se exponen a determinados gases, internamente posee un calentador encargado de aumentar la temperatura interna y con esto el sensor pueda reaccionar con los gases provocando un cambio en el valor de la resistencia. El calentador dependiendo del modelo puede necesitar un voltaje entre 5 y 2 voltios y una corriente de 150mA, el sensor se comporta como una resistencia y necesita una resistencia de carga (RL) para cerrar el circuito y con este hacer un divisor de tensión y poder leerlo desde un microcontrolador.

Debido al calentador es necesario esperar un tiempo de calentamiento para que la salida sea estable y tenga las características que el fabricante muestra en sus datasheet, dicho tiempo dependiendo del modelo puede ser entre 2 a 3 minutos.

En el mercado, generalmente los sensores MQ se encuentran en módulos, lo que nos simplifica la parte de conexiones y nos facilitan su uso, solo basta con alimentar el módulo y empezar a leer el sensor, estos módulos también tienen una salida digital la cual internamente trabaja con un comparador y con la ayuda de un potenciómetro podemos calibrar el umbral y así poder interpretar la salida digital como presencia o ausencia del gas.

La diferencia entre los distintos tipos de sensores MQ es la sensibilidad a cierta gama de gases, más sensibles a algunos gases que a otros, pero siempre detectan a más de un gas, por lo que es necesario revisar los datasheet para escoger el sensor adecuado para nuestra aplicación.

Componente: Sensor de Posición (GPS)



FIGURA 46: Imagen Sensor GPS Neo-6M
(Fuente: <http://www.naylampmechatronics.com>)

El módulo GPS en su modelo GY-GPS6MV2 viene con un módulo de serie U-Blox NEO 6M equipado en el PCB, una EEPROM con configuración de fábrica, una pila de botón para mantener los datos de configuración en la memoria EEPROM, un indicador LED y una antena cerámica. También posee los pines o conectores Vcc, Rx, Tx y Gnd por el que se puede conectar a algún microcontrolador mediante una interfaz serial. Para que nuestro módulo GPS funcione adecuadamente se recomienda hacer las pruebas en un ambiente abierto o cercano a la ventana para una correcta recepción de la señal. Según el fabricante el módulo tiene una precisión de 2,5 metros.

Cabe indicar que nuestro módulo GPS también se puede alimentar con una tensión de 3.3V ya que posee un regulador integrado dentro de sí con un consumo de 40mA.

El módulo GPS posee una velocidad de 9600 baudios que es la velocidad por la que viene configurado por defecto.

Los datos que emitidos por el módulo GPS siguen el protocolo NMEA (siglas de National Marine Electronics Association), las cuales deben ser convertido a formato de latitud y longitud mediante una librería que proporciona Arduino.

Componente: Sensor de Sonido



FIGURA 47: Imagen Sensor Sonido LM393
(Fuente: <http://www.naylampmechatronics.com>)

Este módulo está conformado por un micrófono electro modelo CZN-15E, el cual es sensible a la intensidad sonora ambiental.

La salida del módulo es dual, en donde por un lado se tiene una señal analógica corresponde a una señal de tensión que va desde los 0 a 5 volts, y por otro lado una salida digital de 2 estados (0 o 1 lógico) según un umbral establecido mediante potenciómetro. Ambas salidas pueden ser leídas mediante cualquier entrada analógica de microcontroladores.

Características:

- Alimentación: 5 V DC
- Permite detectar la intensidad de sonido del entorno
- Posee un orificio de instalación para facilitar su uso y montaje
- Incorpora LED Indicador de alimentación y LED Indicador de salida digital
- Conexión de 4 hilos
- Dimensiones: 35 x 15 x14 mm

8.3 Desarrollo Plataformas

La etapa de desarrollo de la plataforma considera dos actividades principales, estas son la construcción de la plataforma de medición compuesta por los nodos de sensores, unidad de control y comunicación. Y también la plataforma de monitoreo con sus funciones de visualización en tiempo real, registro histórico y sistema de alarma.

8.3.1 Plataforma de Medición

8.3.1.1 Montaje de nodos de medición

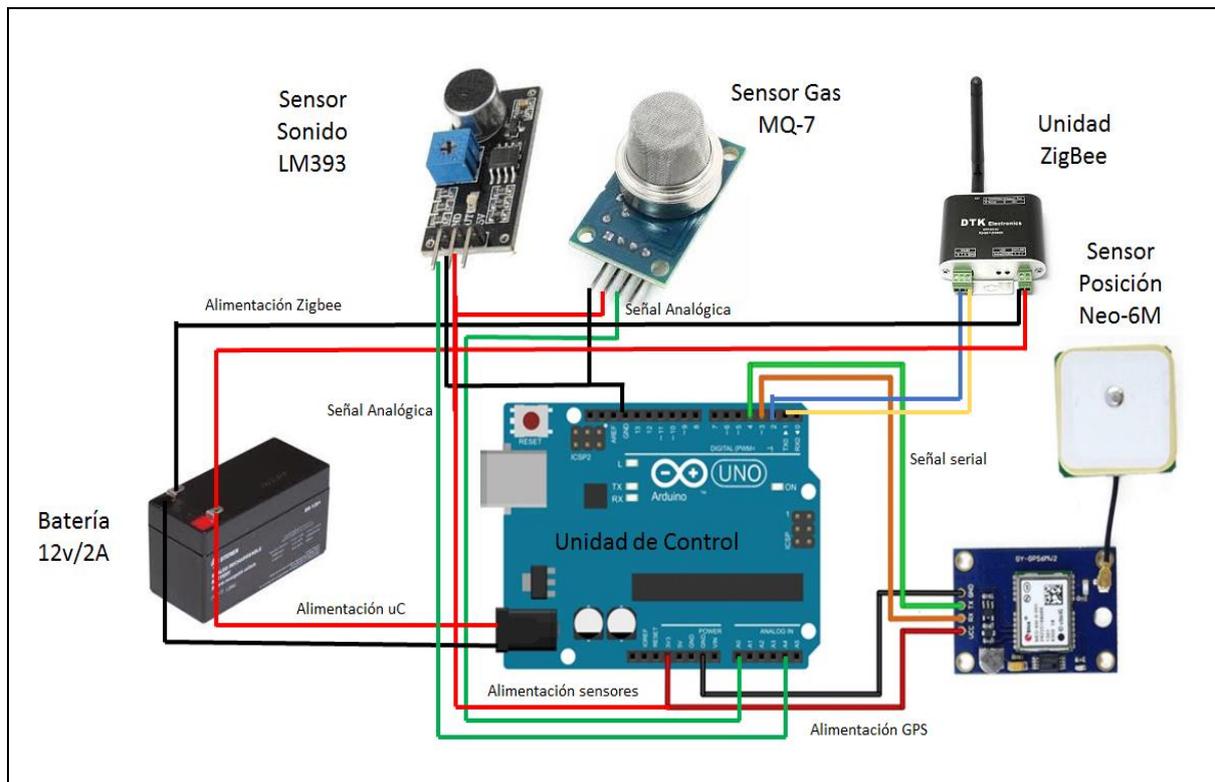


FIGURA 48: Esquema conexiones nodo medición (Fuente: Elaboración propia)

Como se puede apreciar en la figura nº48, se disponen todos los componentes que constituyen el nodo de medición, es decir sensores, unidad de control y comunicación. La alimentación del nodo se realiza mediante una batería de 12 volt/ 2 ampere, además el nodo puede contar con energía de red mediante un transformador y 220 AC/12 CC.

En cuanto a la conexión de los sensores (gas y sonido) se puede comentar que utilizan el puerto analógico (pines A0 y A4) respectivamente. Respecto a la conexión del GPS, se realiza mediante los pines digitales 3 y 4 que deben ser reconfigurados virtualmente mediante la librería serial para transformarlos en pines de comunicación serial. La unidad de comunicación utiliza los pines 1 y 2 que son los pines Tx y Rx que dispone el microcontrolador para a comunicación serial nativamente. Los voltajes de alimentación de los dispositivos son de 5 volts (sensor gas y sonido), 3.3 volts (sensor gps) y 12 volts antena zigbee.

A continuación, se muestran imágenes del montaje del nodo de medición en su respectiva caja de protección.

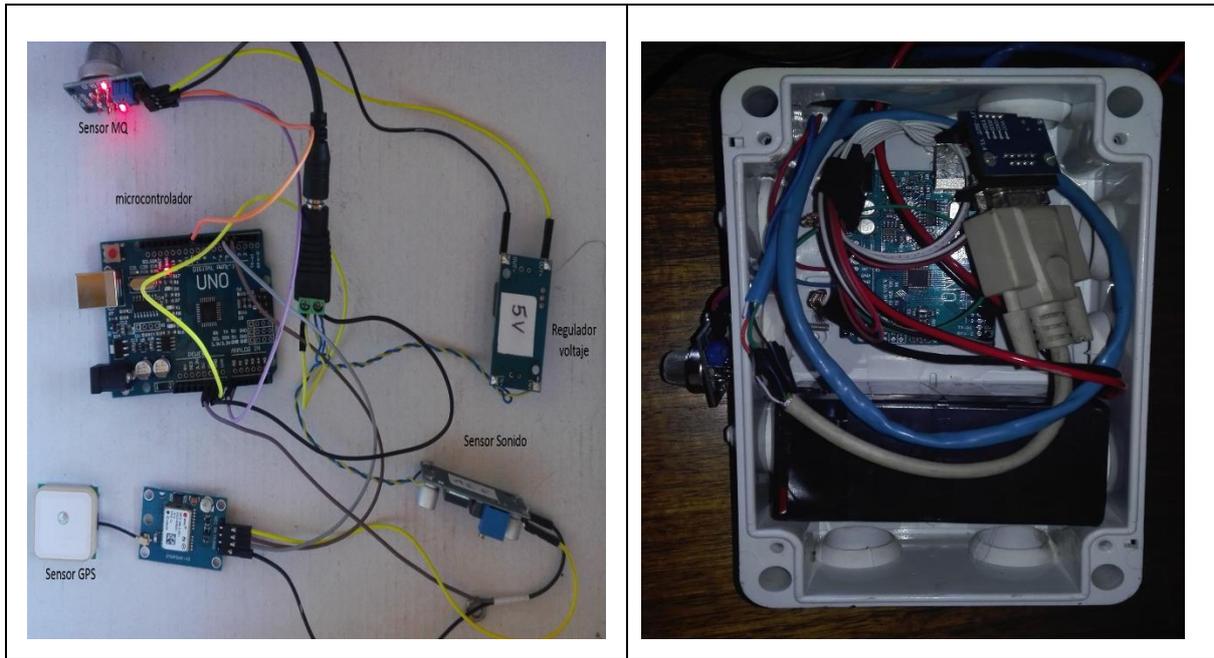


FIGURA 49: Fotografías de montaje y prueba módulo de medición
(Fuente: Elaboración propia)

En la imagen superior izquierda y derecha se puede apreciar los componentes utilizados, los cuales están montados en una caja de protección. En la imagen inferior se puede apreciar el módulo montado con la antena y el trípode para mediciones en terreno.

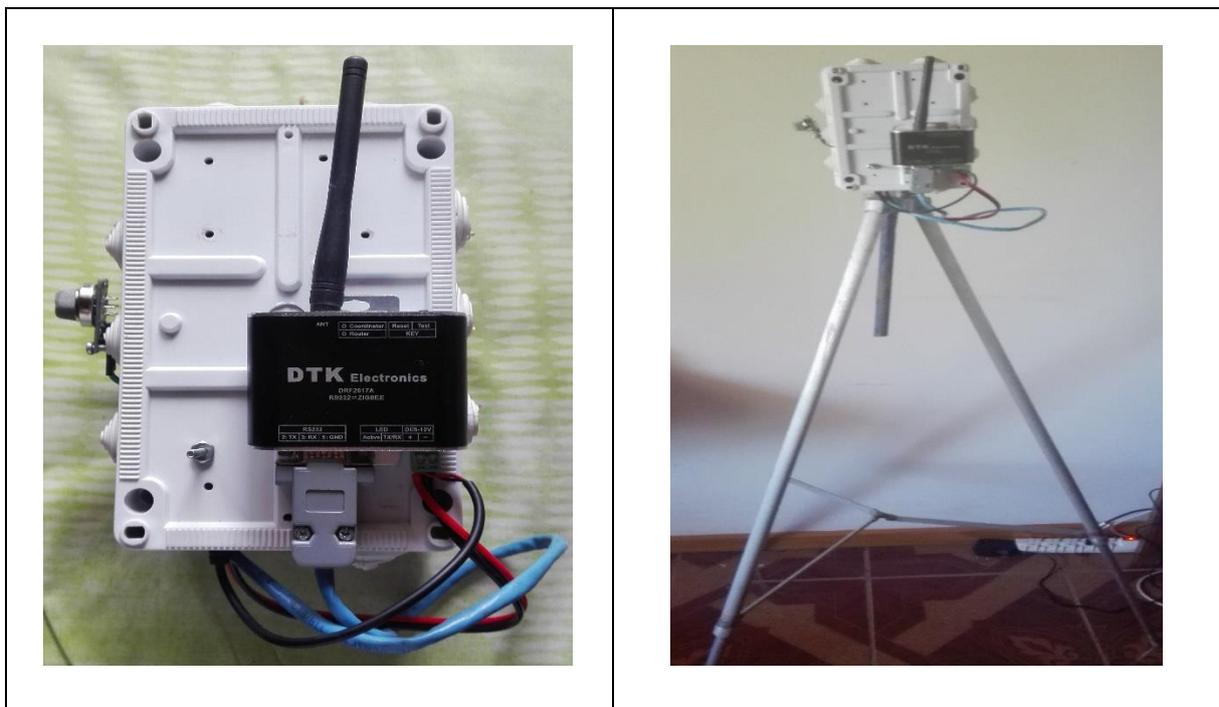


FIGURA 50: Fotografías de módulo de medición (Fuente: Elaboración propia)

8.3.2.2 Implementación de módulo de captura y envío de datos

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>

TinyGPS gps;
SoftwareSerial ss (4, 3);

static void smartdelay (unsigned long ms);
static void print_float (float val, float invalid, int len, int prec);
static void print_int (unsigned long val, unsigned long invalid, int len);
static void print_date (TinyGPS &gps);
static void print_str (const char *str, int len);
const int MQ_PIN = A0;
const int MQ_DELAY = 2000;
const int SO_PIN = A1;

void setup ()
{
  Serial.begin(57600);
  ss. begin (9600);
}

void loop ()
{
  float flat, flon;
  unsigned long age, date, time, chars = 0;
  unsigned short sentences = 0, failed = 0;
  static const double LONDON_LAT = 51.508131, LONDON_LON = -0.128002;
  gps. f_get_position (&flat, &flon, &age);
  print_float (flat, TinyGPS::GPS_INVALID_F_ANGLE, 10, 6);
  Serial.print(",");
  print_float (flon, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6);

  gps. stats (&chars, &sentences, &failed);
  print_int (chars, 0xFFFFFFFF, 6);
  print_int (sentences, 0xFFFFFFFF, 10);
  print_int (failed, 0xFFFFFFFF, 9);
  Serial.println();
  smartdelay (1000);
}

static void smartdelay (unsigned long ms)
{
  unsigned long start = millis ();
  do
  {
    while (ss. available ())
      gps. encode (ss. read ());
  } while (millis () - start < ms);
}
```

```

static void print_float (float val, float invalid, int len, int prec)
{
  if (val == invalid)
  {
    while (len-- > 1)
      Serial.print('*');
    Serial.print(' ');
  }
  else
  {
    Serial.print(val, prec);
    int vi = abs((int)val);
    int flen = prec + (val < 0.0? 2: 1);
    flen += vi >= 1000? 4: vi >= 100? 3: vi >= 10? 2: 1;
    for (int i=flen; i<len; ++i)
      Serial.print(' ');
    int raw_adc = analogRead(MQ_PIN);
    int son_adc = analogRead(SO_PIN);

    // Convertimos los datos a decibeles
    long x=10*val;
    long y=100*x;
    double z=log10(y);
    double decibeles=10*z;
    Serial.print(son_adc);
    Serial.print(raw_adc);
    Serial.println(decibeles);
  }
  smartdelay (0);
}

static void print_int (unsigned long val, unsigned long invalid, int len)
{
  char sz [32];
  if (val == invalid)
    strcpy (sz, "*****");
  else
    sprintf (sz, "%ld", val);
  sz[len] = 0;
  for (int i=strlen(sz); i<len; ++i)
    sz[i] = ' ';
  if (len > 0)
    sz[len-1] = ' ';
  Serial.print(" ");
  smartdelay (0);
}

```

FIGURA 51: Esquema implementación módulo de captura y envío de datos
(Fuente: Elaboración propia & <https://www.luisllamas.es/localizacion-gps-con-arduino-y-los-modulos-gps-neo-6/>)

8.3.2 Plataforma de Monitoreo

La plataforma de monitoreo es el núcleo principal del sistema, ya que se encarga de recibir la información proveniente de la plataforma de medición y preparar la data para ser almacenada en la base de datos, además de enviar vía WebSockets a los clientes web la información disponible, crear un servidor web y conectarse a la API de Twitter. Esta plataforma dispone de variados módulos de software, que se presentan a continuación.

8.3.2.1 Implementación módulo servidor web

El módulo servidor web permite proporcionar un sistema de comunicación basado en el protocolo http de la capa de aplicaciones del modelo OSI.

Las funciones que posee el módulo consideran la apertura de un socket de conexión para el puerto 3000, donde puede atender a múltiples clientes (computadores y móviles), mediante peticiones http.

El módulo servidor web se encarga además de servir los archivos estáticos HTML que componen el módulo de vistas. Para eso, utiliza la librería express, el middleware que permite hacer ruteo de direcciones al estilo API Rest.

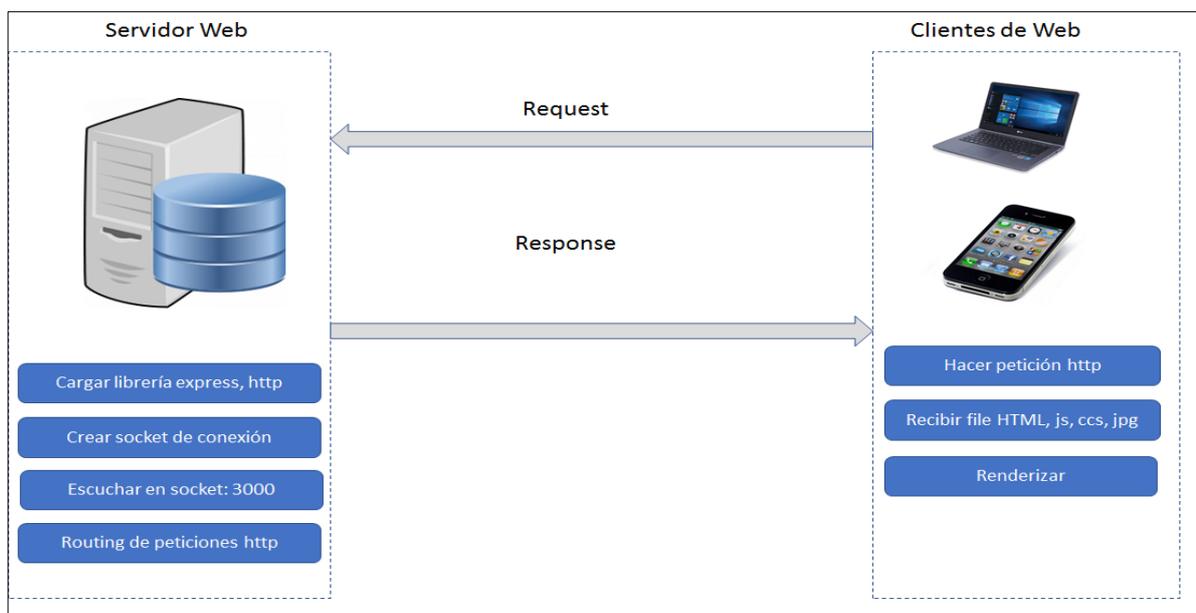


FIGURA 52: Esquema módulo servidor web (Fuente: Elaboración propia)

Como se puede observar en la imagen, el servidor incorpora la funcionalidad de cargar

las librerías de comunicación, luego crear el socket y abrirlo además de establecer un sistema de routing que permita acceder a las rutas de las vistas disponibles. Vista principal (index.html), vista monitoreo tiempo real (real.html) y monitoreo histórico (histórico.html).

Respecto a la implementación del servidor básico, que incorpora las funcionalidades descrita, se presenta un extracto del código para el módulo servidor web.

```
//cargar librerías de comunicación y creo servidor con express

var express=require('express');
var app = express ();
var server = require('http'). server(app);

//establecer middleware para acceder a directorio público de vista

app.use(express.static('public'));

//Routing de peticiones del navegador mediante método GET

app.get('/',function(req,res){
    res.sendFile(__dirname+'public/index.html'); //ruta página principal
});
app.get('/real.html',function(req,res){
    res.sendFile(__dirname+'public/real.html'); //ruta página monitor TR
});
app.get('/historico.html',function(req,res){
    res.sendFile(__dirname+'public/historico.html'); //ruta página histórico
});

//Poner a escuchar al socket

server.listen(3000,function(){
    console.log("el servidor http arranco");
});
```

FIGURA 53: Esquema código módulo servidor web (Fuente: Elaboración propia)

Otro elemento relevante del módulo servidor web, es la seguridad, en este sentido se considera como primera capa de seguridad una funcionalidad básica de control de *exploit* en especial los relacionados con el “*Directory Traversal*” que se refiere al acceso no autorizado de cualquier directorio padres sin control. Esta capa de seguridad es ofrecida por el propio framework express.js

8.3.2.2 Implementación módulo comunicación tiempo real

El módulo de comunicación en tiempo real es el módulo matriz del sistema de monitoreo ambiental, ya que se encarga de recibir la información del módulo de mediciones y enviarla mediante protocolo WebSockets hasta el cliente web, para lograr la comunicación en tiempo real que corresponde a uno de los requerimientos de este proyecto.

Para cumplir los objetivos planteados, el módulo debe incorporar las librerías de comunicación serial que corresponde al middleware con el bajo nivel. Además, la librería socket.io que permite establecer un canal virtual bidireccional permanente entre el servidor y cliente como middleware de más alto nivel.

Para lograr la comunicación, ambos actores servidor y cliente deben disponer de la librería *socket.io.js*.

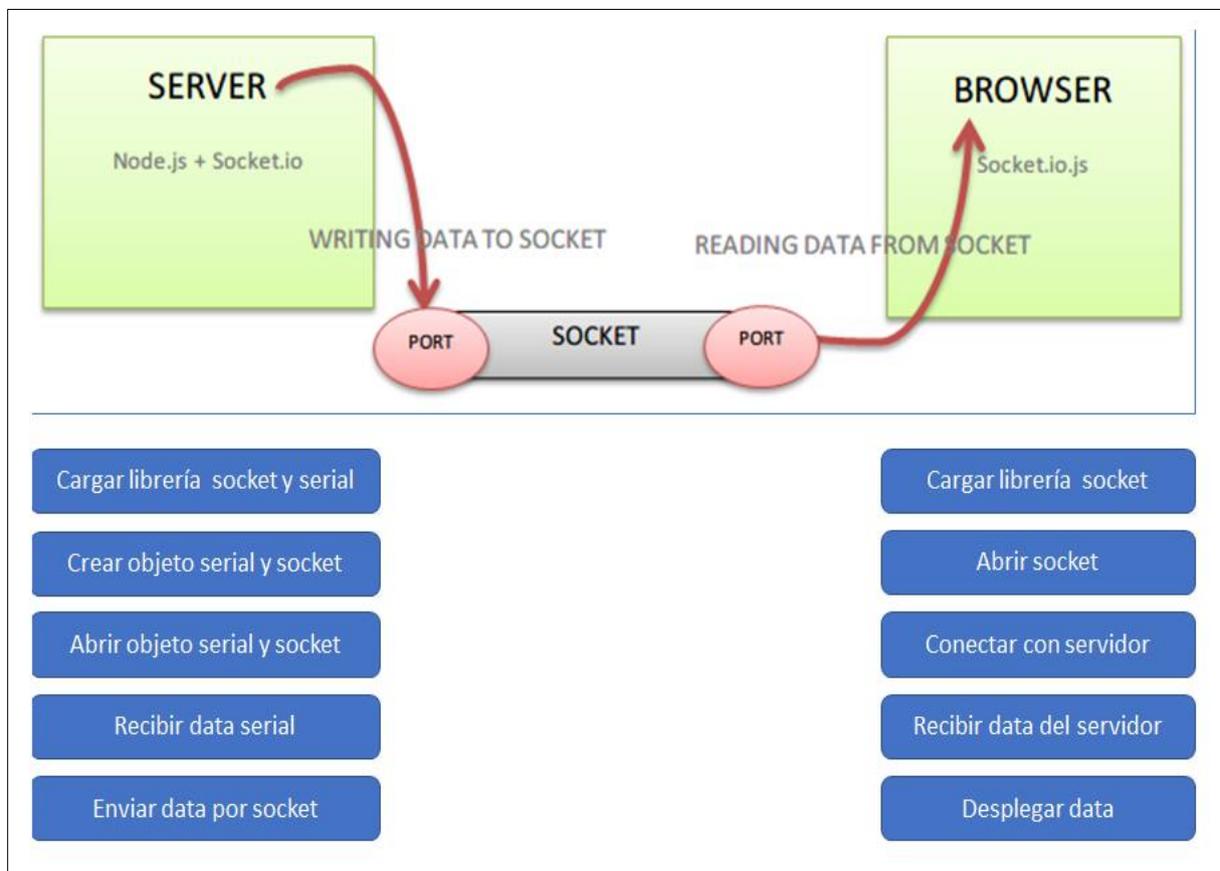


FIGURA 54: Esquema módulo comunicación tiempo real
(Fuente: Elaboración propia)

La implementación del módulo de comunicación en tiempo real se presenta a continuación.

```

//Cargar módulo de comunicación WebSockets y serial

var io = require ('socket.io'). listen(server);
var serialport = require('serialport');
var Serialport = serialport.SerialPort;

//conexión de WebSockets con cliente web

io.on('connection', function (socket){
  console.log("Alguien se conectó");
})

//crear objeto serial para recepción nodo de medición

var myPort = new Serialport("COM5",{
  baudrate:57600,
  parser: serialport.parsers.readline("\n")
});

//llamar al método myPort con la función onOpen de parámetro

myPort.on('open',onOpen);

//llama al método myPort con la función onData de parámetro

myPort.on('data',onData);

// enviar data de sensores a cliente web vía WebSockets

function onData(dato){
  io.sockets.emit('lectura', dato);
};

//funcion para abrir el puerto serial para lectura de data de sensores

function onOpen(){
  console.log("Arduino conectado");
};

```

FIGURA 55: Esquema código módulo comunicación tiempo real
(Fuente: Elaboración propia)

8.3.2.3 Implementación modulo persistencia de datos

El módulo de persistencia está encargado de almacenar en forma permanente la data capturada por los sensores. Esta data almacenada permitirá confeccionar los gráficos de mediciones históricas (día y semana) que provee el sistema.

Para la confección del módulo de persistencia, se recurre al entorno MongoDB, que

corresponde a un sistema de bases de datos NoSQL, el cual se basa en el almacenamiento de colecciones y documentos, en vez de tablas y registros como los sistemas relacionales.

En particular la organización del módulo es la siguiente.

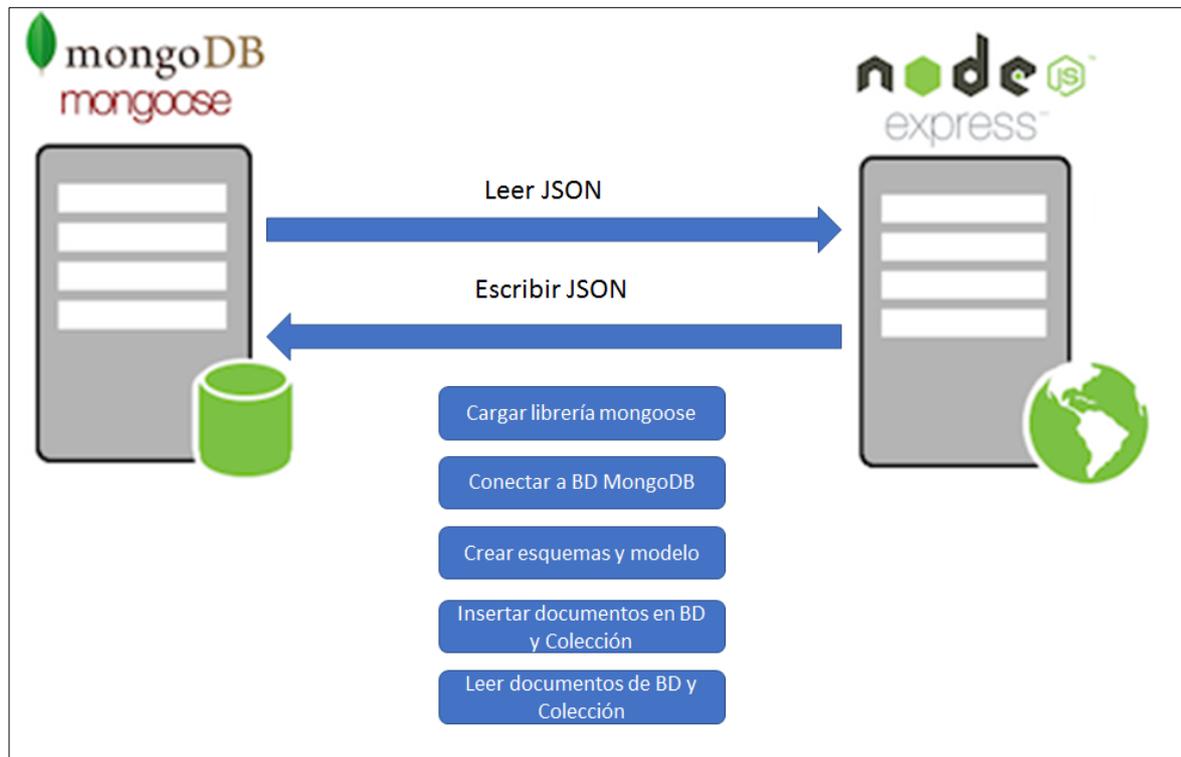


FIGURA 56: Esquema módulo persistencia de datos (Fuente: Elaboración propia)

Como se puede apreciar en el esquema, el módulo realiza varias funciones. Por ejemplo, se debe cargar la librería mongoose (ORM) que permite establecer un middleware entre la aplicación y la base de datos para transformar colecciones de datos en objetos.

Además, se debe establecer la conexión con la base de datos, crear los esquemas y modelos que permiten operar con los objetos JSON. Finalmente se incluyen las funciones que permiten hacer lectura y escritura de documentos desde la base de datos.

El objeto JSON, que se envía es del tipo clave: valor, en particular se indica la medida del sensor y su respectivo valor numérico.

El modelo de datos se presenta a continuación.

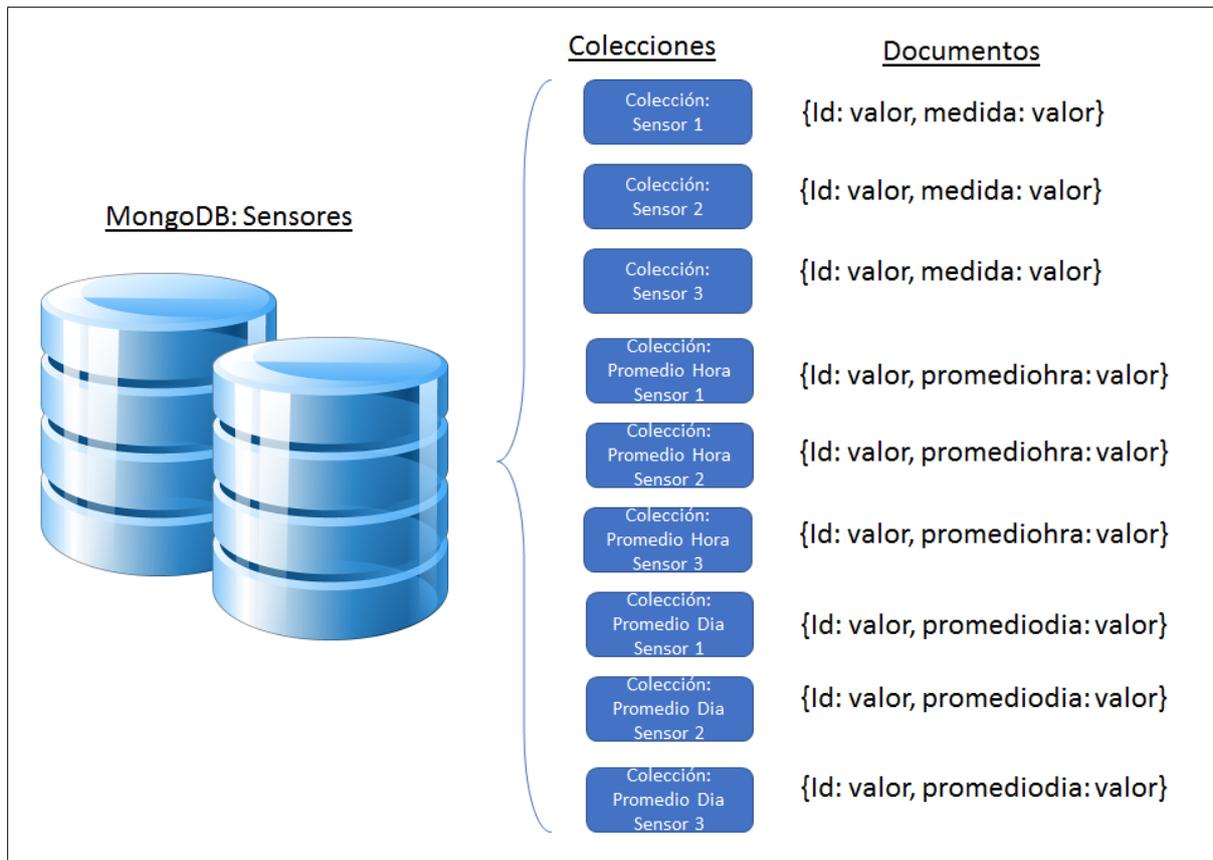


FIGURA 57: Esquema modelo de datos (Fuente: Elaboración propia)

El código necesario para implementar el módulo de persistencia se detalla a continuación:

```
// agregar módulo mongoose para la conexión entre NodeJS y MongoDB
var mongoose = require('mongoose');

//conexión a BD Mongo
mongoose.connect('mongodb://localhost/users', function(err){
  if (!err){
    console.log('conectado a mongo');
  }else {
    throw err;
  }
});

//crear schema1 para el primer nodo de medición
Nodo1 = new mongoose.Schema({
  medida: Number,
},{collection: 'nodo1'});

//crear modelo del esquema 1
Nodo1 = mongoose.model('users', Nodo1);
```

```

//crear schema2 para el promedio de datos de nodo de medición 1
PromNodo1 = new mongoose.Schema({
  promedio: Number,
},{collection: 'PromNodo1'});

//crear modelo del esquema 2
PromNodo1 = mongoose.model('users1', PromNodo1);

//insertar datos en la BD users, colección Nodo1
Nodo1.collection.insert({medida: dato}, function(err, response){
  if (err) throw err;
});

//leer datos de BD users, colección Nodo1
Nodo1.find(function(err, data){
  if(err) {return console.error(err);}
  if(data===null){
    console.log('no hay datos');
  }
});

//recorrer la colección, eliminar el id y deja solo promedio para forma un paquete
de envió
var paquete=new Array();
data.forEach(function(element) {
  paquete.push(element.medida);
});
});

```

FIGURA 58: Esquema código modulo persistencia de datos
(Fuente: Elaboración propia)

8.3.2.4 Implementación de módulo de Mapas

El módulo de mapa se encuentra alojado en el componente de presentación. El objetivo fundamental del módulo es desplegar en forma gráfica la información de ubicación del nodo de medición georeferenciado y además la capacidad de monitoreo de la condición ambiental mediante una funcionalidad de alerta dinámica de estado que permite cambiar el icono de alerta del nodo, según la condición normal o peligro. Para la implementación del módulo se debe interactuar con la API de GoogleMap, que provee de diversas funcionalidades de presentación adicional en el mapa, como incluir iconos personalizados y mensajes dinámicos.

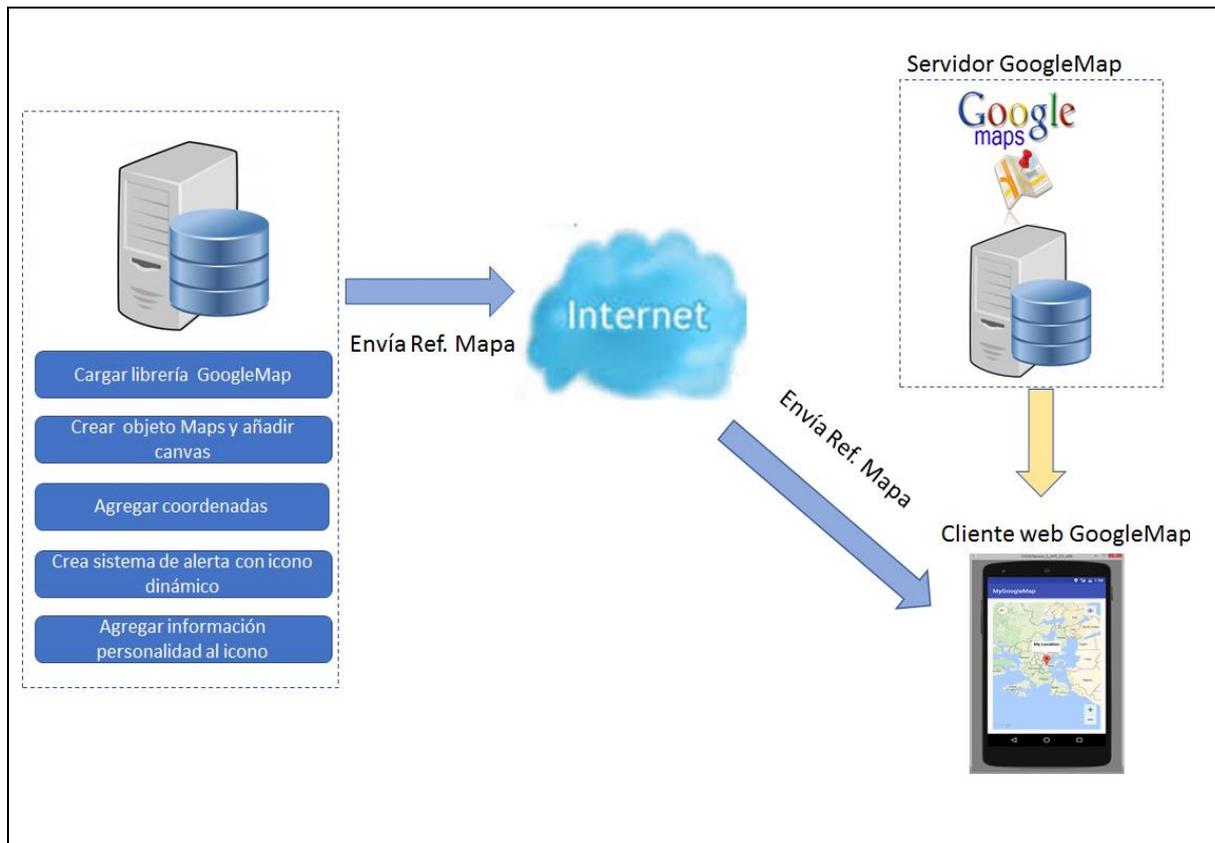


FIGURA 59: Esquema módulo mapas (Fuente: Elaboración propia)

La implementación del código base, se detalla a continuación:

```
function crear_mapa () {

    //Insertar mapa centrado en coordenadas
    var map;
    var marker;
    var mapOptions= {
        center: new google. maps. LatLng (-33.4349083,-70.6170132),
        zoom: 15,
        mapTypeId: google. maps. MapTypeId.ROADMAP
    }
    map=new google. maps. Map (document. getElementById("map"),
    mapOptions);

    //crear socket
    var socket=io ();

    //escuchar en el socket y recibir el evento 'lectura' desde el servidor
    socket. on ('lectura', function(data) {

        document. getElementById("pepe").value=data;
        var a=-33.4352352;
        var b=-70.6150000;
    });
}
```

```

document. getElementById("x").value=a;
document. getElementById("y").value=b;

//Cambiar icono de mapa si hay alerta (sobrepasa un umbral definido)
if (data<umbral) {
    var normal='NORMAL';
    document. getElementById("juanito").value=normal;
    //cargar icono1
    var verde = new google. maps. MarkerImage (
'https://www.shareicon.net/data/128x128/2016/05/28/572079_valid_30x30.png',
    new google. maps. Size (30,30)
    );
    var a=-33.4352352;
    var b=-70.6150000;
    var place = new google. maps. LatLng (a, b);
    marker = new google. maps. Marker ({
    position: place,
    title:"Nodo Medicion 1",
    map: map,
    icon: verde
    });
    //Agrego mensaje alerta personalizado
    google. maps. event. addListener (marker,"click", showInfo);
    function showInfo () {
        map. setZoom (16);
        map. setCenter (marker. getPosition ());
        var infowindow= new google. maps. Infowindow ({
        content:"Medicion Sensor 1: "+data + " Condicion:
PELIGRO"
        });
        infowindow. open (map, marker);
    }
}
if (data>=umbral) {
    var peligro='PELIGRO';
    document. getElementById("juanito").value=peligro;
    //cargar icono2
    var rojo = new google. maps. MarkerImage (
'http://drhoffman.com/wp-content/uploads/2015/03/rss.png',
    new google. maps. Size (30,30)
    );
    var a=-33.4352352;
    var b=-70.6150000;
    var place = new google. maps. LatLng (a, b);
    marker = new google. maps. Marker ({
    position: place,
    title:"Nodo Medicion 1",
    map: map,
    icon: rojo
}

```

```

    });
    //Agrego mensaje alerta personalizado
    google.maps.event.addListener (marker,"click", showInfo);
    function showInfo () {
        map.setZoom (16);
        map.setCenter (marker.getPosition ());
        var infowindow= new google.maps.Infowindow ({
            content:" Sensor 1: "+data + " Condicion: PELIGRO"
        });
        infowindow.open (map, marker);
    }
});
}

```

FIGURA 60: Esquema código modulo mapas (Fuente: Elaboración propia)

8.3.2.5 Implementación módulo de Gráficos

El módulo de gráficos pertenece al componente de capa de presentación que está basado en un patrón de cliente pesado.

Este módulo se encarga de recibir la información histórica enviada por el servidor (base de datos) mediante WebSockets para desplegarla en gráficos a través de librería “char.js”.

Para la implementación del módulo, se requiere establecer una conexión con la base de datos mongodb mediante WebSockets y recibir la data almacenada, en particular los promedios obtenidos cada cuatro horas de medición y los promedios de cada día para construir el grafico de monitoreo diario y semanal.

La librería “char.js” posee variados tipos de gráficos que permiten presentar la información de las mediciones en forma adecuada para su lectura. En particular se ha elegido gráficos de barras para la presentación.

El detalle de la implementación del módulo base, se presenta a continuación:

```

function histórico () {
    //creo variable para socket
    var socket=io ();
    //escucho en el socket y recibo data para graficar
    socket.on ('envio_mensaje', function(data) {
        //agregar grafico con data (promedios colección lunes)
        var ctx = document.getElementById("myChart").getContext('2d');
        var myChart = new Chart (ctx, {
            type: 'bar',
            data: {
                labels: ["00:00", "4:00", "8:00", "12:00", "16:00", "20:00"],
            }
        });
    });
}

```

```

datasets: [{
  label: '# Mediciones Promedio últimas 4 horas',
  data:data,-
  backgroundColor: [
    'rgba (255, 99, 132, 0.2)',
    'rgba (54, 162, 235, 0.2)',
    'rgba (255, 206, 86, 0.2)',
    'rgba (75, 192, 192, 0.2)',
    'rgba (153, 102, 255, 0.2)',
    'rgba (255, 159, 64, 0.2)'
  ],
  borderColor: [
    'rgba (255,99,132,1)',
    'rgba (54, 162, 235, 1)',
    'rgba (255, 206, 86, 1)',
    'rgba (75, 192, 192, 1)',
    'rgba (153, 102, 255, 1)',
    'rgba (255, 159, 64, 1)'
  ],
  borderWidth: 1
}]
},
options: {
  responsive: false,
  scales: {
    yAxes: [{
      ticks: {
        beginAtZero: true
      }
    }]
  }
}
});
});
}

```

FIGURA 61: Esquema código modulo Gráficos (Fuente: Elaboración propia)

8.3.2.6 Implementación módulo de alerta Twitter

El módulo de alerta Twitter corresponde a la funcionalidad de la quinta capa revisada en la arquitectura general del proyecto, que corresponde a la interacción del sistema con actores sociales.

La idea es que el sistema de monitoreo ambiental tenga la capacidad de interactuar en forma dinámica con los interesados mediante el envío de mensaje de twitter cuando los sensores detecten condiciones ambientales desfavorables.

Para la implementación del módulo, se desarrolla un cliente de la API de twitter que permite mediante un proceso de autenticación previa, poder interactuar con el servidor Twitter leyendo twitter disponibles en la red o enviando twitter a una cuenta especifica. En este caso una cuenta desarrollada para pruebas del sistema.

El sistema se activa si se sobrepasa un valor umbral de medidas de los sensores definido previamente.

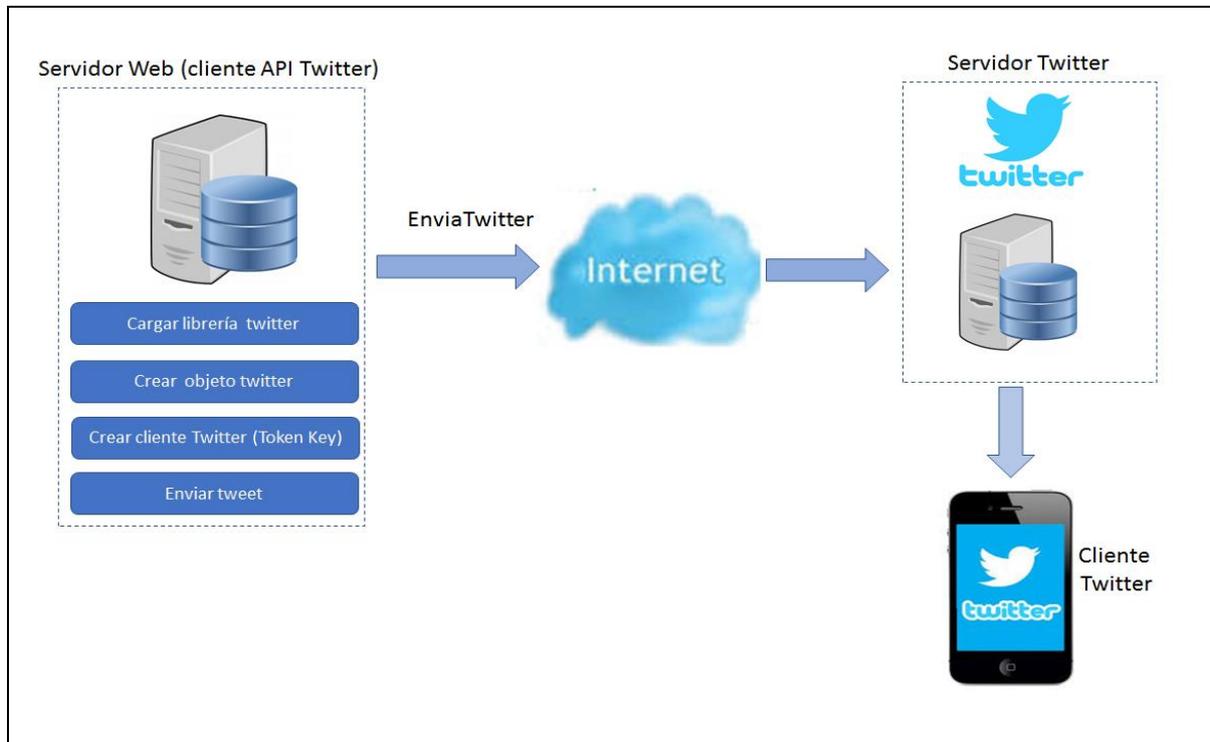


FIGURA 62: Esquema módulo Twitter (Fuente: Elaboración propia)

A continuación se detalla la implementación base del módulo:

```
//agregar librería Twitter para NodeJS

var Twitter = require('twitter');

//Crear objeto Twitter con información de API y conexión cuenta

var cliente = new Twitter ({
  consumer_key: 'xxxxxxxxxxxxxxxxxxxxxxx',
  consumer_secret:'xxxxxxxxxxxxxxxxxxxxxxx',
  access_token_key: 'xxxxxxxxxxxxxxxxxxxxxxx',
  access_token_secret: 'xxxxxxxxxxxxxxxxxxxxxxx'
})
```

```

//Si valor sensores sobrepasan umbral definido de riesgo envía alarma con data a
Twitter
if (dato>umbral) {
    cliente.post ('statuses/update', {status: 'PELIGRO: Valores riesgosos:' +
dato})
    . then((tweet)=> {
        console.log(tweet);
    })
    . catch((error)=> {
        console.log(error);
    })
}
}

```

FIGURA 63: Esquema código modulo Twitter (Fuente: Elaboración propia)

9.4.1.1 Implementación de Interfaces gráficas y vistas

Las interfaces graficas son el medio de comunicación principal del sistema con el usuario. Estas interfaces, se componen de tres vistas web, compuesta por archivos html5. A continuación se muestra un esquema de interacción entre el servidor las vistas.

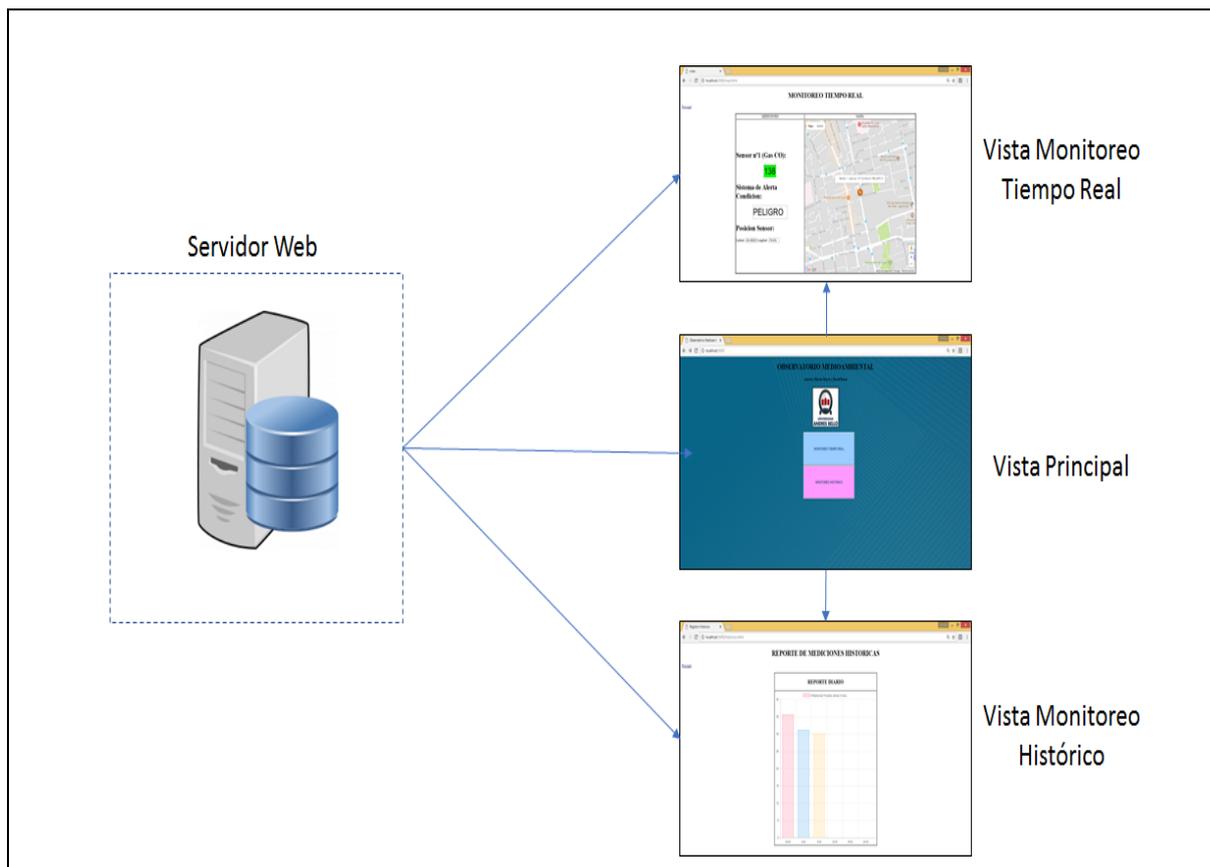


FIGURA 64: Esquema de vistas (Fuente: Elaboración propia)

La primera vista corresponde a una página de presentación del sistema, en donde se destaca el proyecto “Observatorio Medioambiental” perteneciente a la Universidad Andrés Bello. Además, posee dos botones principales que permiten acceder a el monitoreo ambiental en tiempo real y monitoreo histórico.

La segunda vista consiste justamente en la página de monitoreo en tiempo real de las condiciones ambientales. Esta vista presenta dos bloques principales, el primero corresponde al área de mediciones en donde se presenta la información de los nodos de medición, en particular el valor de la medida, posición del nodo y la condición de alerta. El segundo corresponde al área donde se incluye el mapa georeferenciado de la posición de los nodos con su respectivo icono de alerta con la información dinámica. La tercera vista corresponde a la presentación del monitoreo histórico. Esta página presenta la información historia del promedio obtenidos por cada sensor con una frecuencia de una hora. Los datos son presentados mediante gráficos de barra. Además, se incluye los gráficos de los promedios obtenidos para cada día de la semana. Cada página dispone de un botón que permite retornar a la página principal, si lo desea.

8.4 Pruebas

Las pruebas son una etapa fundamental en cualquier proyecto de hardware y software, ya que permiten comprobar cada una de las funcionalidades particulares y generales que permiten satisfacer los requerimientos de los interesados del proyecto. En este sentido se establece un plan de pruebas orientadas a probar las funcionalidades de los nodos de medición, plataforma de monitoreo y sistema global. De acuerdo con esto, el plan de prueba contempla test básicos que permiten hacer la trazabilidad de la funcionalidad desde que se captura el dato de interés hasta que se despliega al usuario. Además, se presentan algunos test adicionales que permiten determinar el rendimiento de la aplicación en cuanto a la capacidad de recibir conexiones simultaneas y el tiempo de carga de la plataforma para ser concordantes con el cumplimiento de los requisitos no funcionales de usabilidad e interoperabilidad. Los test se detallan a continuación.

8.4.1 Set de pruebas

Test n°1: Prueba captura y envío de datos desde Arduino

El objetivo de esta prueba es verificar la unidad más básica de funcionalidad del proyecto que considera la obtención de data por parte de los sensores conectados en el puerto analógico (Pines A0, A2) y digital (Pin D3 y 4) del microcontrolador Arduino Uno.

La prueba considera la captura de datos para diversas velocidades de muestreo (100 ms, 500 ms, 1 s, 2 s) y envió por el puerto serial (Pines Tx: 1 y Rx: 2). En definitiva, se configuro él envió de data cada 1 segundo con una velocidad de comunicación de 57600 baudios en el puerto COM1 del microcontrolador. A continuación, se muestra los resultados del envió de datos presentados en la consola del monitor serial que trae incorporado el IDE de Arduino.

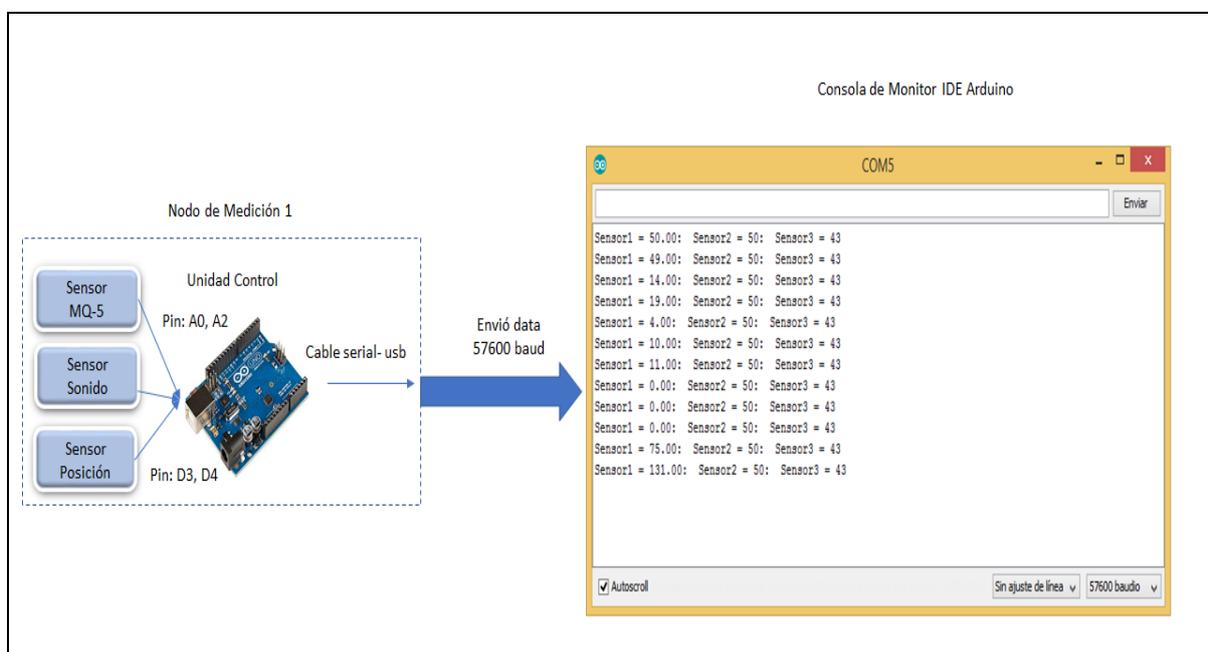


FIGURA 65: Esquema de Consola monitor IDE Arduino (Fuente: Elaboración propia)

Como se puede apreciar en la figura n°65, la unidad de medición envía los datos por el puerto serial y pueden ser visualizados en la consola de Arduino.

La razón que la tasa de envió de datos por el puerto serial (Un dato por segundo y 57600 baud) no sea más alta, se debe a que no se requiere una actualización tan rápida en el servidor debido a las características de las variables medidas ya que no

se producen grandes cambios en las condiciones ambientales en periodos de tiempo pequeños y también una mayor tasa de transferencia genera un aumento de consumo de energía en el sistema de comunicación.

Test nº2: Prueba de la recepción de data en plataforma de Monitoreo NodeJS

Esta prueba tiene como objetivo probar el middleware que proporciona la plataforma NodeJS para interactuar con la comunicación serial de bajo nivel. Para prueba se utiliza el módulo *serialport.js* que permite leer datos desde el puerto serial virtual que tiene incorporado el servidor para este caso el COM5.

Para la implementación en NodeJS, se debe crear un objeto serial de comunicación que disponga de la configuración de puerto y velocidad de la recepción, además crear un evento de la recepción de datos y apertura y cierre del puerto.

Respecto a la recepción de datos, se probaron tasas de la recepción de 100 ms, 500 ms y 1 segundo, comportándose bastante bien la plataforma NodeJS.

Los datos recibidos de la plataforma de medición son almacenados en el buffer que dispone la aplicación.

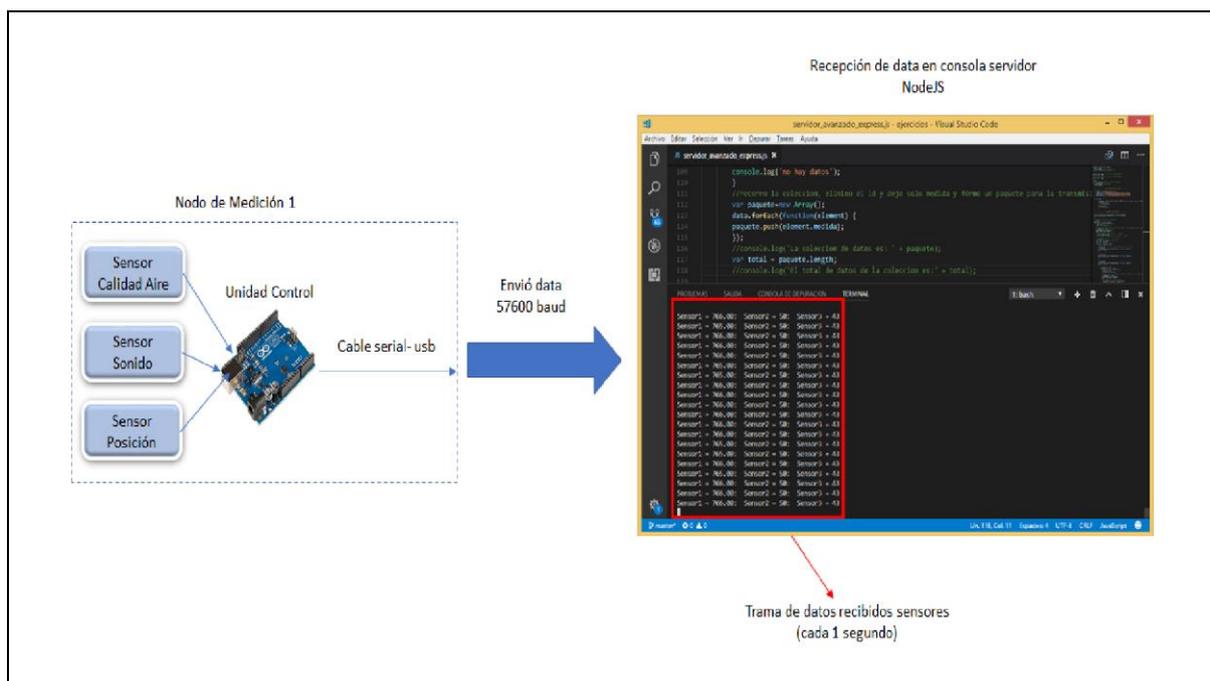


FIGURA 66: Esquema de Consola Servidor NodeJS (Fuente: Elaboración propia)

Como test adicional, se probaron la capacidad de recibir datos desde tres puertos (COM 3, 4 y 5) en forma simultánea, para simular la recepción de diferentes fuentes

de datos y observar el comportamiento del buffer al recibir data desde tres COM diferentes, observando que la aplicación es transparente y no presenta alteraciones de ningún tipo al recibir mayor carga de flujo de datos por diversos puertos simultáneos.

Test nº3: Prueba de persistencia de datos en mongodb

El test de persistencia permite verificar el proceso de almacenamiento en un medio permanente la data capturada. Para esto se implementa una prueba básica de escritura de datos en la base de datos mongo, verificando la interoperabilidad con la aplicación NodeJS a través del módulo *mongoose.js*. Este módulo sirve como middleware entre la base de datos y la aplicación.

En el esquema, se puede apreciar el servidor de base de datos que abre la conexión en el puerto 27017 y recibe datos desde la aplicación NodeJS. Para la visualización de la data almacenada se crea un cliente de mongodb y mediante la consola DOS, se visualizan la data de los tres sensores capturados.

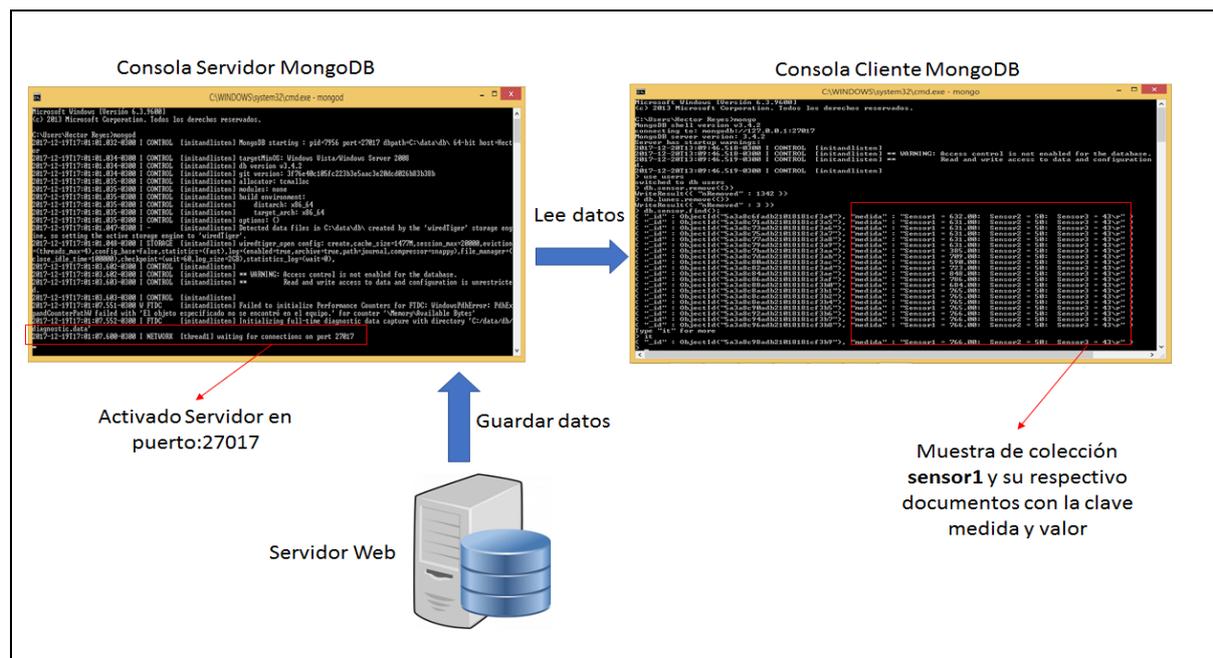


FIGURA 67: Esquema de Consola Servidor Mongodb (Fuente: Elaboración propia)

Como resultado de esta prueba, se puede comentar que el middleware de comunicación *mongoose*, es bastante transparente en la comunicación entre Node y mongo. Los datos son almacenados adecuadamente en colecciones previamente

creadas. Además, como prueba adicional, se pudo ver la capacidad de carga de datos respecto a volumen, para esto se procedió a guardar un día completo de data, es decir casi 260.000 muestras, que serán la data básica que genera un nodo de medición por día. Evidentemente solo es una prueba porque el sistema en funcionamiento solo guardara los promedios cada cuatro horas y el promedio del día.

Test nº4: Prueba de envío de datos en tiempo real a cliente web

Esta prueba permite verificar la parte más importante del proyecto, que es el envío en tiempo real de los datos desde un servidor web a un cliente web, específicamente un browser.

Para la implementación de la prueba, se utiliza el middleware de comunicación vía WebSockets dispuesto por el módulo de Node *socket.io.js*. Este módulo permite abstraer toda la lógica del protocolo push que requiere el saludo inicial a nivel de http y la conexión posterior vía tcp-ip.

Además, el módulo se integra fácilmente en el documento html a través de un archivo JavaScript.

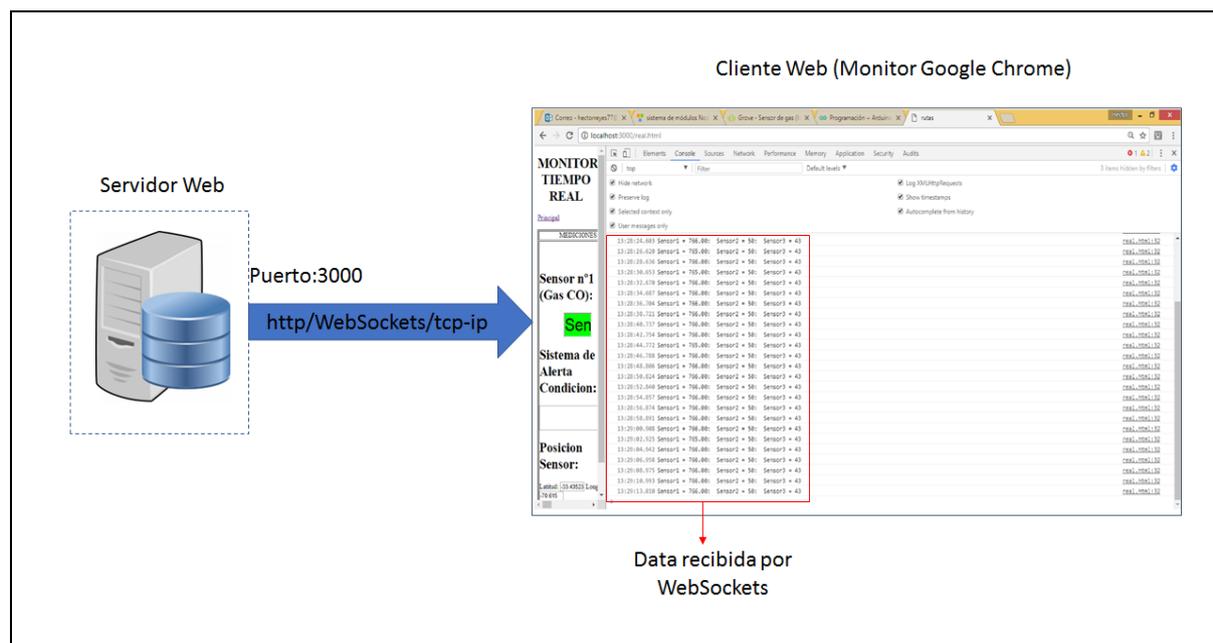


FIGURA 68: Esquema de Consola Cliente Web (Fuente: Elaboración propia)

En el esquema se aprecia que el servidor web envía la data obtenida por los sensores mediante un canal virtual permanente proporcionado por el protocolo WebSockets. Esta data llega al navegador y se visualiza mediante la consola de Google Chrome.

Se observa que los datos llegan con total integridad y velocidad. Además, se probaron instancia de estrés para verificar si el aumento del flujo de datos generaba algún tipo de desbordamiento o si se requería actualizar la página para continuar la recepción, lo cual fue descartado debido al excelente comportamiento del sistema. Como prueba adicional, se verificó la recepción de datos en un browser de dispositivo móvil, obteniéndose resultado similar.

Test n°5: Prueba de visualización de datos en dashboard cliente web

Esta prueba tiene como objetivo verificar el despliegue adecuado en términos de claridad de lectura, continuidad de envío y recepción vía web socket y consistencia de datos. A través del objeto DOM se puede manipular los objetos html lo cual permite desplegar adecuadamente la información renderizada. También se puede probar la conexión de la interfaz con el api de GoogleMap. De acuerdo con esto, se observa que es necesario disponer de conexión de internet para la carga del programa lo que influye en el despliegue del mapa y el icono de alerta. Respecto del flujo de información, no se observa dependencia con la conexión de internet ya que el socket continúa funcionando adecuadamente.

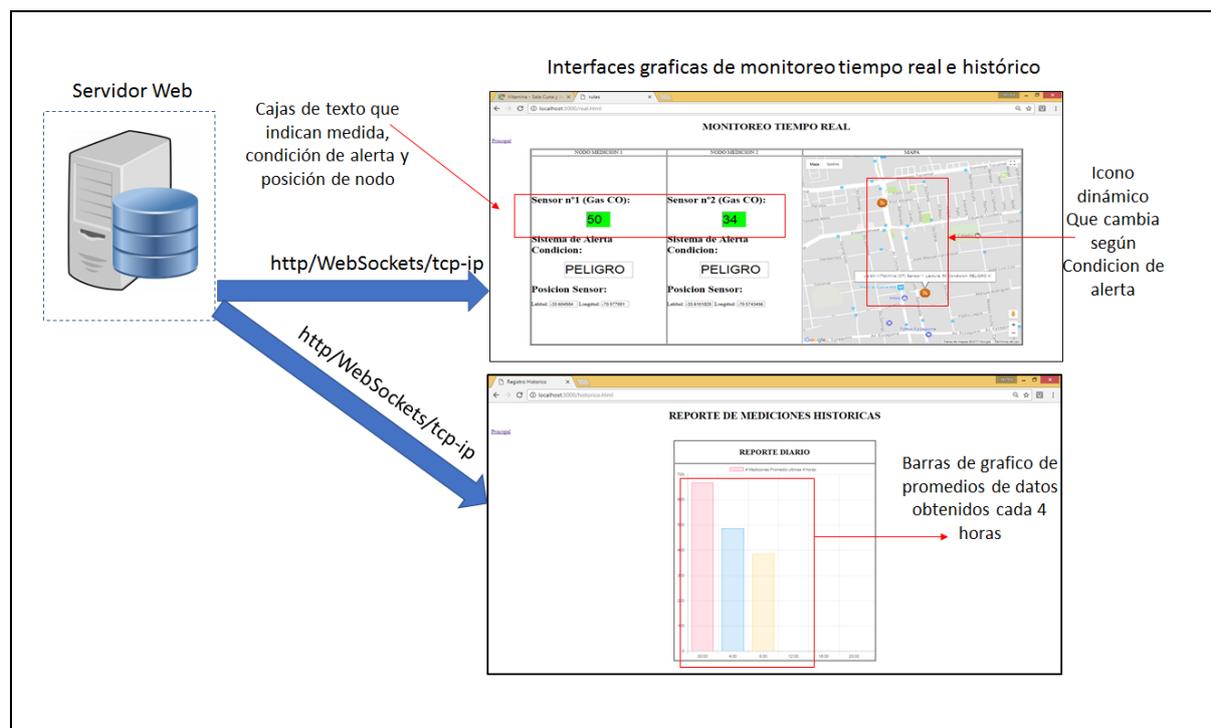


FIGURA 69: Esquema de Dashboard (Cliente Web)

(Fuente: Elaboración propia)

Respecto al despliegue de la información en el gráfico, se puede observar que la librería *char.js*, permite desplegar y actualizar convenientemente el grafico histórico. La carga del grafico es rápida debido a que también recibe los datos vía WebSockets (cada segundo). Finalmente se verifica en todas las pruebas realizadas que la trazabilidad del tiempo que fluyen los datos desde su captura hasta su despliegue se mantiene en torno al segundo.

Test nº6: Prueba de envío de alerta a Twitter

La prueba de Twitter representa el fin del ciclo de funcionamiento de la aplicación, desde la captura hasta el aviso al usuario.

En este sentido, la prueba contempla verificar la sensibilidad que posee el sistema al generar producirse una alerta y a su vez la velocidad con la que se recibe el mensaje en la plataforma twitter. De acuerdo con las pruebas realizadas se puede mencionar que desde el momento que el servidor detecta la alerta, el periodo de respuesta (recepción del mensaje) es en promedio de 2 segundos. Además, se puede observar que el módulo *Twitter.js* proporcionado por node.js permite filtrar las alertas que posee el mismo mensaje en cuanto al valor de medida. Esto es una medida de seguridad que se dispone contra el uso de Bot que generan mensaje indefinidamente.

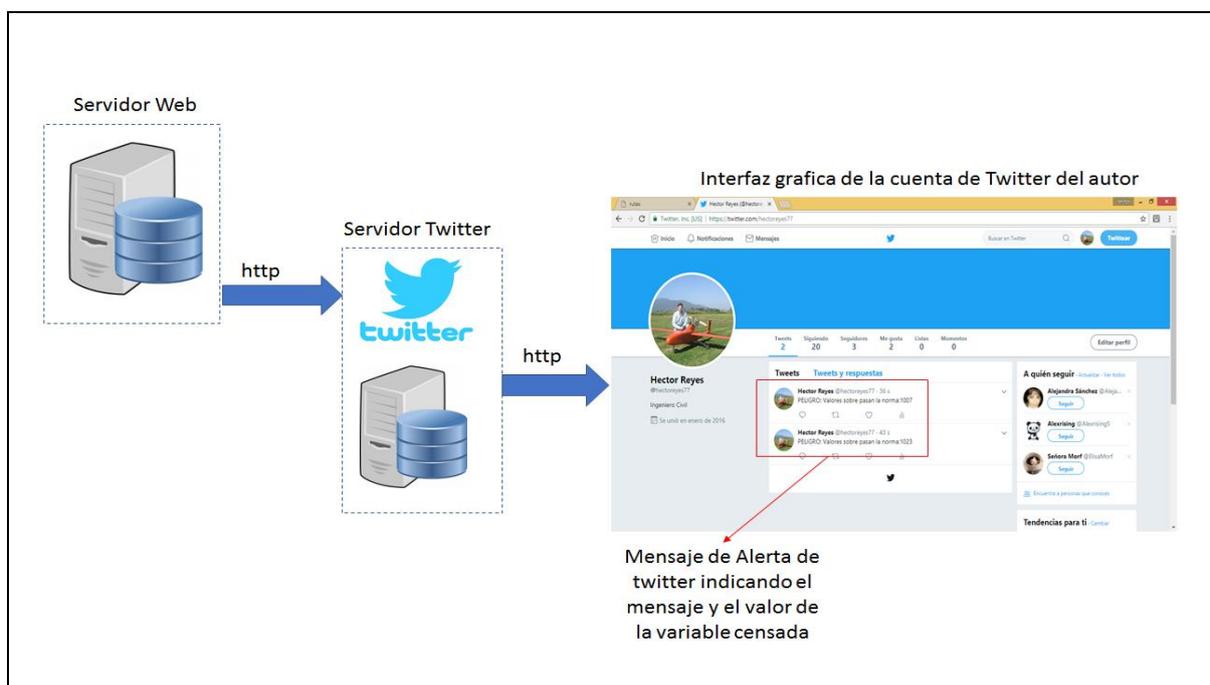


FIGURA 70: Esquema de Interfaz de Twitter (Cliente Web)

(Fuente: Elaboración propia)

La prueba contempló la utilización de la cuenta de Twitter del autor de este trabajo. En caso de que más usuarios quieran acceder al sistema de alerta, solo deben suscribirse a la cuenta y recibirán los mensajes. Evidentemente, esta funcionalidad requiere de la conexión con internet.

Test nº7: Prueba de carga de plataforma

El objetivo de esta prueba es verificar la capacidad de respuesta que posee el servidor web y servidor de bases de datos frente a solicitudes de clientes. En particular interesa saber los tiempos de respuesta de los servidores, frente a diversos escenarios de usuarios virtuales.

Para realizar las pruebas se utilizó la herramienta Jmeter cuya versión es la 3.2. Esta herramienta es open source muy completa implementada en Java que permite realizar test de comportamiento funcional y medir rendimiento.

Las variables medidas por la herramienta son las siguientes:

- Muestras: cantidad de Thread utilizados para la URL.
- Media: tiempo promedio en milisegundos para un conjunto de resultados.
- Min: tiempo mínimo que demora un Thread en acceder a una página.
- Max: tiempo máximo que demora un Thread en acceder a una página
- Rendimiento: rendimiento medido en los requerimientos por segundo / minuto / hora.
- Kb/sec: rendimiento medido en Kbyte por segundo.
- Media en bytes: tamaño medio de respuesta del servidor (en bytes).

Nota: Es necesario mencionar que las pruebas fueron realizadas considerando que los servidores de base de datos y web se encuentran alojados en la misma maquina en donde se realiza la prueba y por lo tanto no se considera la latencia de la red para las velocidades de respuesta.

Test de Rendimiento: Las pruebas de rendimiento permitirán saber los tiempos de respuesta y el porcentaje de éxito de las peticiones, además de indicadores de la tasa de transferencia para a cada petición. Los escenarios (E) de simulación para el servidor web se han definido para 1, 10, 100, 1000 y 10000 conexiones virtuales y

para el servidor de bases de datos mongodb, son de 10 y 1000 conexiones. Los resultados se presentan a continuación.

E1: 1 usuario:

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo										Navegar...	
Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos <input type="checkbox"/> Configurar											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
Petición HTTP	1	13	13	13	0,00	0,00%	76,9/sec	88,49	13,75	1178,0	
Total	1	13	13	13	0,00	0,00%	76,9/sec	88,49	13,75	1178,0	

E2: 10 usuarios

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo										Navegar...	
Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos <input type="checkbox"/> Configurar											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
Petición HTTP	10	5	4	11	1,87	0,00%	10,9/sec	12,59	1,96	1178,0	
Total	10	5	4	11	1,87	0,00%	10,9/sec	12,59	1,96	1178,0	

E3: 100 usuarios

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo										Navegar...	
Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos <input type="checkbox"/> Configurar											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
Petición HTTP	100	17	4	105	22,87	0,00%	93,9/sec	108,02	16,78	1178,0	
Total	100	17	4	105	22,87	0,00%	93,9/sec	108,02	16,78	1178,0	

E4: 1000 usuarios

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo										Navegar...	
Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos <input type="checkbox"/> Configurar											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
Petición HTTP	1000	2324	38	3559	598,48	36,80%	190,3/sec	284,14	21,50	1528,7	
Total	1000	2324	38	3559	598,48	36,80%	190,3/sec	284,14	21,50	1528,7	

E5: 10000 usuarios

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo										Navegar...	
Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log Sólo Errores <input type="checkbox"/> Éxitos <input type="checkbox"/> Configurar											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
Petición HTTP	10000	3059	19	17369	2587,10	89,52%	436,2/sec	865,15	8,17	2031,1	
Total	10000	3059	19	17369	2587,10	89,52%	436,2/sec	865,15	8,17	2031,1	

Además de las pruebas de carga al servidor web, se aplicaron pruebas al servidor de base de datos mongodb. Esta prueba se realizó utilizando el protocolo tcp-ip puerto 27017. Los resultados para los escenarios de 10 y 1000 conexiones de clientes son los siguientes:

E1: 10 usuarios

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo <input type="text"/> <input type="button" value="Navegar..."/> Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log <input type="checkbox"/> Sólo Errores <input type="checkbox"/> Éxitos <input type="button" value="Configurar"/>											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
Muestreador T...	10	3	2	8	1,56	0,00%	11,0/sec	0,00	0,00	,0	
Total	10	3	2	8	1,56	0,00%	11,0/sec	0,00	0,00	,0	

E2: 1000 usuarios

Reporte resumen											
Nombre: Reporte resumen											
Comentarios											
Escribir todos los datos a Archivo											
Nombre de archivo <input type="text"/> <input type="button" value="Navegar..."/> Log/Mostrar sólo: <input type="checkbox"/> Escribir en Log <input type="checkbox"/> Sólo Errores <input type="checkbox"/> Éxitos <input type="button" value="Configurar"/>											
Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes	
Muestreador T...	1000	1128	24	1840	299,01	39,40%	329,6/sec	0,00	0,00	,0	
Total	1000	1128	24	1840	299,01	39,40%	329,6/sec	0,00	0,00	,0	

A partir de los resultados de cada escenario, se puede comentar lo siguiente:

- En los escenarios de 1, 10 y 100 conexiones al servidor web se logran tiempos de respuesta inferiores a los 20 ms con 0% porcentaje de error en la respuesta. Además, el resultado del rendimiento indica que se pueden atender casi 90 usuarios por segundo, sin dejar de responder ninguna conexión.
- A partir de las 1000 conexiones, los tiempos de respuesta suben a 2 y 3 segundo en promedio, aumentando el porcentaje de error en las respuestas del servidor a 36% aproximadamente, con un rendimiento de 190 usuarios por segundo.
- En el caso de 10000 conexiones, los tiempos de respuesta sobre pasan los 3.5 segundo, teniendo respuesta con tiempos máximo de 17 segundo, que supera cualquier nivel de experiencia de usuario aceptable. Además, el porcentaje de error en la respuesta aumenta a casi el 90%.
- Para el caso del servidor de bases de datos se obtuvieron resultados similares.
- Lo interesante de los resultados es que el sistema, se comporta bastante bien al aumento de conexiones, sabiendo que el sistema tendría una cantidad de usuario focalizados inferior a las pruebas simuladas.

Test nº8: Prueba de enlace inalámbrico

Las pruebas de enlace inalámbrico tienen como objetivo, determinar el nivel de alcance que tiene la red de sensores en cuanto a distancia, integridad de envío de datos y rendimiento de las baterías.

Estas pruebas serán realizadas en un escenario controlado en el sentido que no corresponde a la ubicación final donde se realiza los experimentos (jardines infantiles en zona urbana), por el contrario, se efectúa en un terreno abierto con la disposición de los nodos de medición en línea vista a la estación de recepción, esto solo con el objetivo de comprobar el alcance, integridad de los datos y rendimiento de la batería. Estas pruebas, serán efectuadas en la localidad de Pirque contigua a Puente Alto donde se proyecta efectuar las mediciones finales.

A continuación, se presenta una imagen de GoogleMap de la zona de pruebas.



FIGURA 71: Esquema de zona de pruebas de alcance de red sensores

(Fuente: Elaboración propia)

Una vez seleccionada la zona de pruebas, se procede a realizar el montaje de los nodos de medición en cada lugar, como se puede apreciar en la figura nº72, los nodos de medición están ubicados a una distancia de 800 metros (nodo nº1) y 650 metros (nodo nº2), medidos de la estación de monitoreo. Esta distancia es la máxima que se puede lograr sin salir de la propiedad.



FIGURA 72: Fotografías de instalación de nodos de medición en zona de pruebas
(Fuente: Elaboración propia)



FIGURA 73: Fotografías estación monitoreo (izquierda) y nodo de medición
(derecha) (Fuente: Elaboración propia)

Para realizar las pruebas de comunicación, se debe configurar las unidades ZigBee, específicamente definir la velocidad de transmisión, los identificadores de los nodos, la función que cumplen (coordinador o router) dentro de la red, el canal de comunicación que sincronizara la red de sensores. Para realizar esta labor, la empresa DTK Electronics dispone de una aplicación simple de configuración de cada parámetro. A continuación, se presenta los parámetros para cada nodo de la red (<http://www.dtk-electronics.com/>).

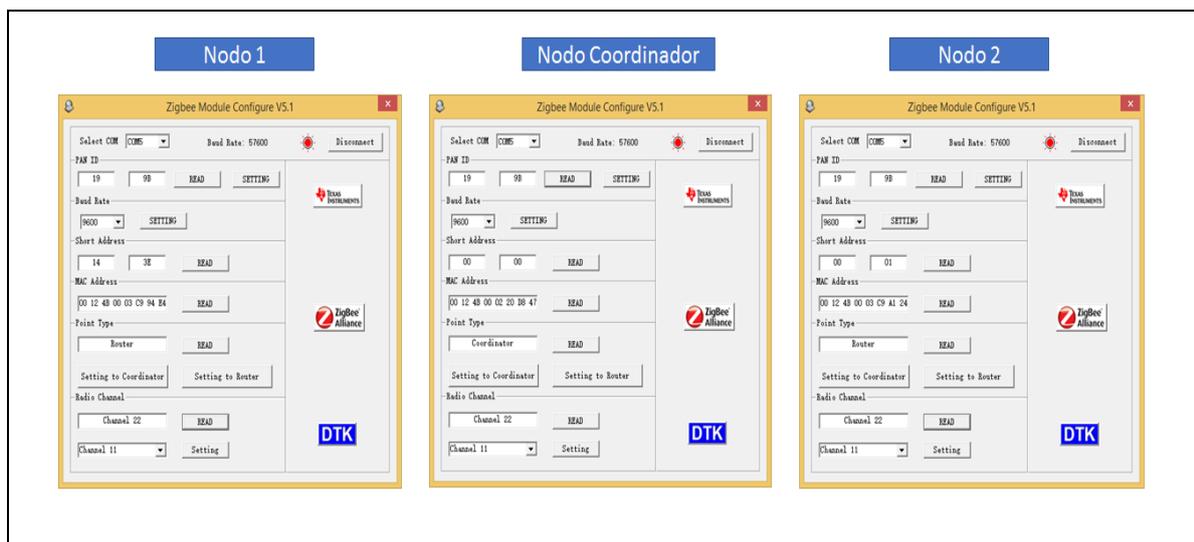


FIGURA 74: Configuración red de sensores zigbee (Fuente: Elaboración propia)

Test n°1: Integridad de datos versus distancia:

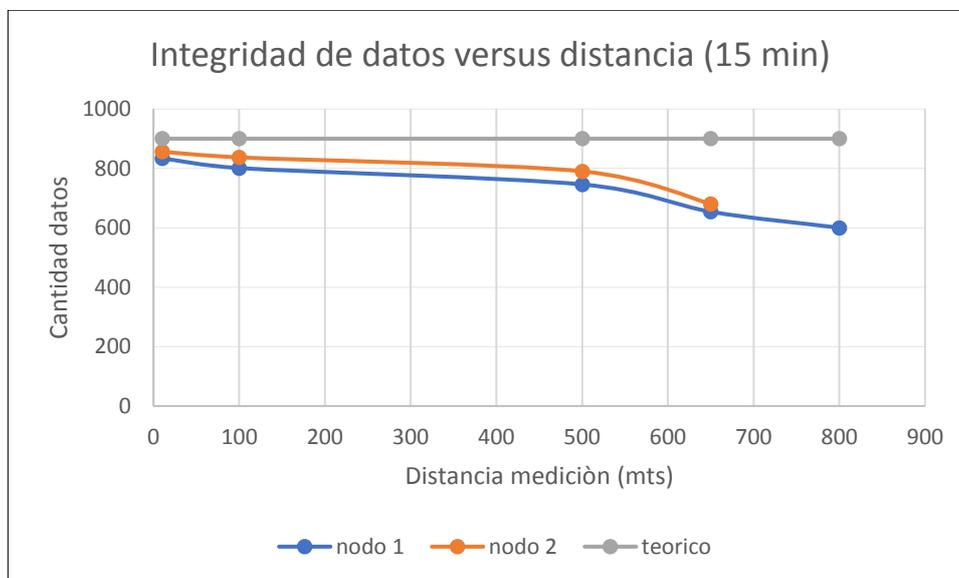


GRAFICO 1: Integridad de datos versus distancia de envío (Fuente: Elaboración propia)

Test nº2: Rendimiento Batería:

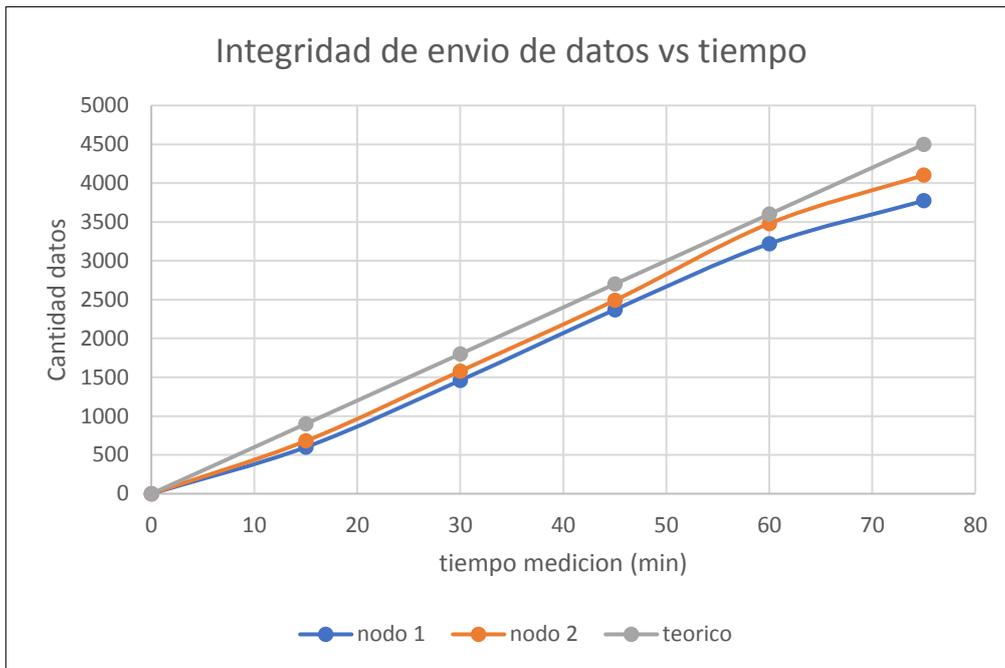


GRAFICO 2: Integridad de datos versus tiempo de medición
(Fuente: Elaboración propia)

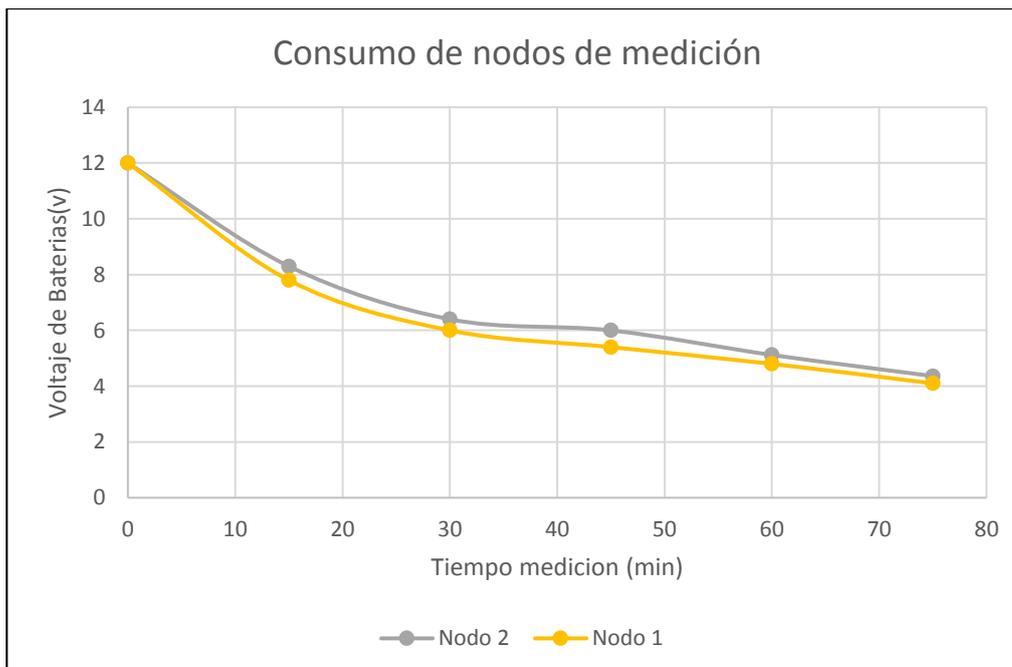


GRAFICO 3: Consumo de batería de nodos de medición datos versus tiempo
(Fuente: Elaboración propia)

A partir de los resultados de prueba, se puede comentar lo siguiente:

- Para medir el alcance del sistema de medición, se utiliza el concepto de integridad de datos versus la distancia de medición, es decir se verifica si el sistema ubicado a cierta distancia es efectivo comparándolo con la cantidad de datos teóricos que se deben recibir. De acuerdo con esto, se puede comentar que existe una pérdida de datos de un 5% aproximadamente a una distancia de 10 metros y 34% para el caso de 800 metros de distancia. Esta situación se muestra en el grafico nº1 para una prueba de 15 minutos de medición a diferentes distancias y se compara datos recibidos (línea naranja y azul) con la captura teórica de datos que se deberían recibir para cada nodo (línea ploma).
- Otra prueba realizada es comprobar el comportamiento de la recepción de datos en función de la batería a la distancia máxima disponible (800 y 650 metros). Esta medida se obtiene en base al tiempo de medición. Como se puede apreciar en el grafico nº2, también se mantiene una tasa de pérdida de datos constantes entre los datos teóricos que se esperan versus los datos recibidos.
- Finalmente se efectúa la prueba de rendimiento de la batería a distancia máxima en función del tiempo (grafico nº3), en donde se puede apreciar que la duración máxima de la batería de 12 volt y 2 Amper es aproximadamente 75 minutos, en donde ya se deja de recibir datos (4 volts). Además, la tasa de pérdida de voltaje en función del tiempo es de 100 mv/minuto, ósea cada dato requiere 1,6 mv para ser enviado aproximadamente.

Test nº9: Prueba de sensores

Los sensores utilizados en este proyecto son de baja gama, es decir de reducida precisión que se recomienda su aplicación en ambientes de interiores para proyectos de domótica simple. Sin embargo, debido a las características del proyecto que busca demostrar la posibilidad de captura de data en tiempo real y no la calidad de la data, se utilizará de todas formas estos dispositivos a pesar de que los experimentos serán en ambientes exteriores.

Los sensores de gas de la serie MQ, utilizan métodos electroquímicos derivados de la variación de su resistencia cuando se exponen a determinados gases, internamente

posee un calentador encargado de aumentar la temperatura interna y con esto el sensor pueda reaccionar con los gases provocando un cambio en el valor de la resistencia.

En el mercado, generalmente los sensores MQ se encuentran en módulos, lo que nos simplifica la parte de conexiones y nos facilitan su uso, solo basta con alimentar el módulo y empezar a leer el sensor.

La diferencia entre los distintos tipos de sensores MQ es la sensibilidad a cierta gama de gases, más sensibles a algunos gases que a otros, pero siempre detectan a más de un gas, por lo que es necesario revisar los datasheet para escoger el sensor adecuado para nuestra aplicación.

Tan pronto se alimente el módulo este empezará a calentarse, hay que esperar unos minutos para que el módulo caliente y tener un censado que permita obtener la lectura analógica directa del sensor.

Para efectos de las pruebas y experimentos, el proyecto contempla utilizar la captura analógica del sensor que se transformada a lecturas de concentración de monóxido de carbono medidas en ppm (partículas por millón) mediante el proceso descrito en el anexo A1.

Las lecturas obtenidas por los sensores son referenciales ya que es inverosímil realizar un proceso de comparación efectivo con los sistemas utilizados en las estaciones de SINCA (Sistema de información nacional de calidad de aire, dependiente del Ministerios del Medio Ambiente).

La estación SINCA que captura datos de la comuna de Puente Alto, utiliza el sistema de fotometría infrarroja de filtro de correlación de gas "Thermo 48i", cuya precisión en la concentración de monóxido de carbono va desde un rango de 0 a 10.000 ppm versus los sensores MQ que tienen un rango de medición de 20 a 10000 ppm. Esta variable es relevante considerando que los datos históricos disponibles de mediciones en la estación de la comuna de Puente Alto fluctúan en el rango de 0 a 30 ppm.

Para el caso de las lecturas del sensor GPS, la validación de las mediciones, fue realizada en base a la aplicación de GoogleMap, en donde se ingresaron las coordenadas que entregaba el sensor y se obtuvo efectivamente la ubicación de los nodos de medición en la comuna de Pirque.

IX. Experimentación

El presente capítulo tienen como objetivo, verificar la efectividad de la plataforma de monitoreo en un ambiente de mediciones real.

La idea es comprobar las funcionalidades de captura y despliegue de la data obtenida en ambientes de interés social. Como ya se ha mencionado anteriormente, la aplicación particular de la plataforma está asociada al monitoreo de jardines infantiles con el fin de disponer de información focalizada complementaria a la información global disponible con medios tradicionales.

Específicamente con los experimentos se pretende verificar la capacidad que posee la plataforma de capturar y enviar datos desde los nodos de medición hasta el servidor. Es característica se cuantifica mediante el *“Porcentaje de efectividad del sistema para la captura de datos”*.

Además, se considera evaluar en forma cualitativa, la capacidad de la plataforma de detectar mediante las mediciones información sobre algún patrón de comportamiento primitivo que pudiera ayudar a categorizar jardines infantiles y otras dependencias en función de las mediciones ambientales y ubicación. Lo anterior mediante alguna categoría de detección de tendencia en las mediciones capturadas con el fin de verificar si existen diferencias en las lecturas entre las dos zonas de ubicación de los nodos de medición.

9.1 Selección de población y Muestra

El experimento fue realizado en una condición de terreno real (jardines infantiles en zonas urbanas), pero en condiciones de temporalidad diferentes a lo real debido al periodo en que se realiza este trabajo de investigación (Enero y Febrero) que no corresponde a los meses de funcionamiento regular del jardín infantil (Marzo a Diciembre), lo que significa que la data obtenida no es representativa de las condiciones ambientales de un periodo académico normal, situación que no afecta a la veracidad del experimento ya que el objetivo es verificar la capacidad de capturar data y despliegue en tiempo real mediante un sistema web.

La zona de experimentación seleccionada corresponde a una zona urbana céntrica de la comuna de Puente Alto en donde se localizan dos jardines infantiles pertenecientes a la red Vitamina. Estos lugares fueron seleccionados por encontrarse

en avenidas con gran flujo vehicular y además están a una distancia entre ellas de 823 metros medidos linealmente, que corresponde a una distancia menor al alcance que describe el fabricante de las antenas (1,6 kilómetros). Lo anterior se aprecia en la figura nº 75.

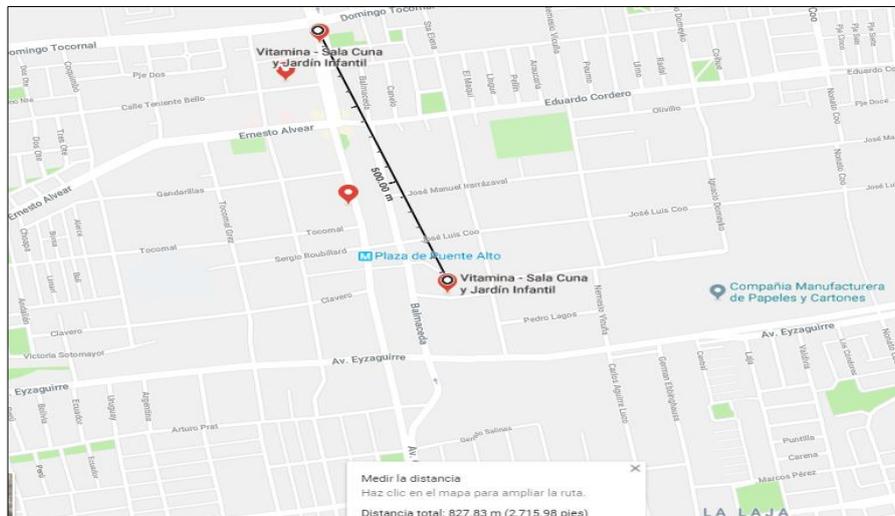


FIGURA 75: Esquema de ubicación zona de experimentación, Comuna Puente Alto
(Fuente: GoogleMap)

Otra variable de interés que se debe considerar para el experimento corresponde al tipo y cantidad de datos que se requieren obtener. En este sentido como el objetivo del experimento es la verificación de la captura de data por la plataforma y no la calidad de la data propiamente tal, se procuró realizar el experimento tratando de igualar las condiciones de medición, es decir disponer de varias jornadas de medición con el fin de almacenar un volumen de datos considerable que emule las condiciones de operación continua que debería tener un sistema de monitoreo.

De acuerdo con lo anterior, se determina que el periodo de muestreo será de 2 semanas en horario diurno. Es necesario mencionar que las mediciones, se podrán realizar solo en horario diurno debido a la solitud de la administradora del Jardín Vitamina que requiere dejar desconectado el sistema durante la noche por motivos de seguridad, definiendo que las mediciones se desarrollarán de lunes a viernes de 10:00 a 20:00 horas.

La dinámica de captura de la data consistió en instalar y preparar la plataforma (servidor, sistema de comunicaciones, activación de base de datos y principalmente precalentar el sensor MQ todos los días (9 días). Esta situación se debió repetir en las dos estaciones de medición.

El servidor de datos para las pruebas iniciales se ubicó en una zona equidistante a las dos estaciones de medición para verificar la captura de data, pero la ubicación final fue variando en los jardines infantiles, es decir la semana uno se ubicó el servidor en el Jardín Vitamina nº1 y la semana dos en Vitamina nº2.

A continuación, se presentan imágenes demostrativas de las instalación y pruebas iniciales de captura de datos en las afueras de los jardines infantiles Vitamina. Estas pruebas fueron realizadas para verificar si las mediciones en el interior y exterior variaban significativamente.



FIGURA 76: Fotografías estaciones de medición en exteriores de Jardines Vitaminas

9.2 Recolección de datos

La recolección de la data fue realizada cada segundo, pero el registro histórico se confeccionó con los promedios obtenidos cada una hora de medición en el respectivo nodo. A continuación, se presenta una captura del panel de monitoreo en tiempo real obtenidas en las afueras del Jardín Vitamina nº1.

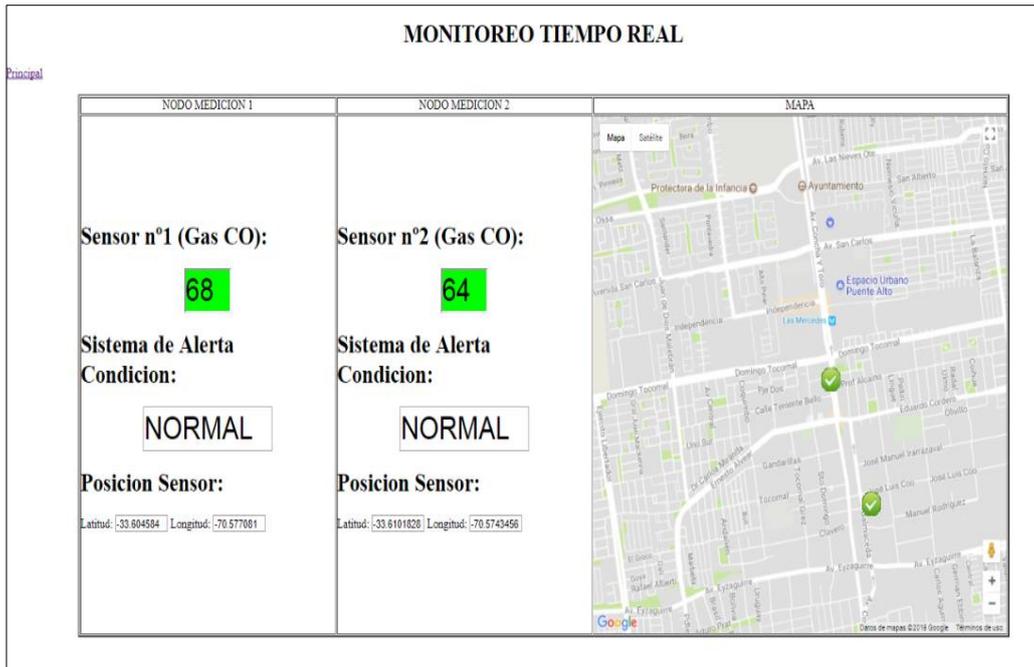


FIGURA 77: Captura de plataforma de monitoreo de data en Jardines Vitaminas
(Fuente: Elaboración Propia)

La identificación de los nodos de medición corresponde al Jardín Vitamina nº1 (V1) ubicado en la calle Concha y Toro y Vitamina nº2 (V2) ubicado en la calle Manuel Rodríguez respectivamente. A continuación, se presentan los resultados del sensor MQ del primer día de mediciones.

		Sensor MQ Vitamina 1	Sensor MQ Vitamina 2
		Valor Concentración Co(ppm)	Valor Concentración Co(ppm)
Martes 02-01-2018	hora		
	10:00	15,18	11,03
	11:00	14,34	9,69
	12:00	15,04	10,13
	13:00	15,12	6,09
	14:00	13,67	4,9
	15:00	12,88	3,05
	16:00	13,56	3,9
	17:00	18,056	7,68
	18:00	18,9	7,56
	19:00	18,38	7,46
20:00	19,01	7,9	
promedio		15,8	7,2

TABLA 15: Promedios de mediciones día 1

(Fuente: Elaboración propia)

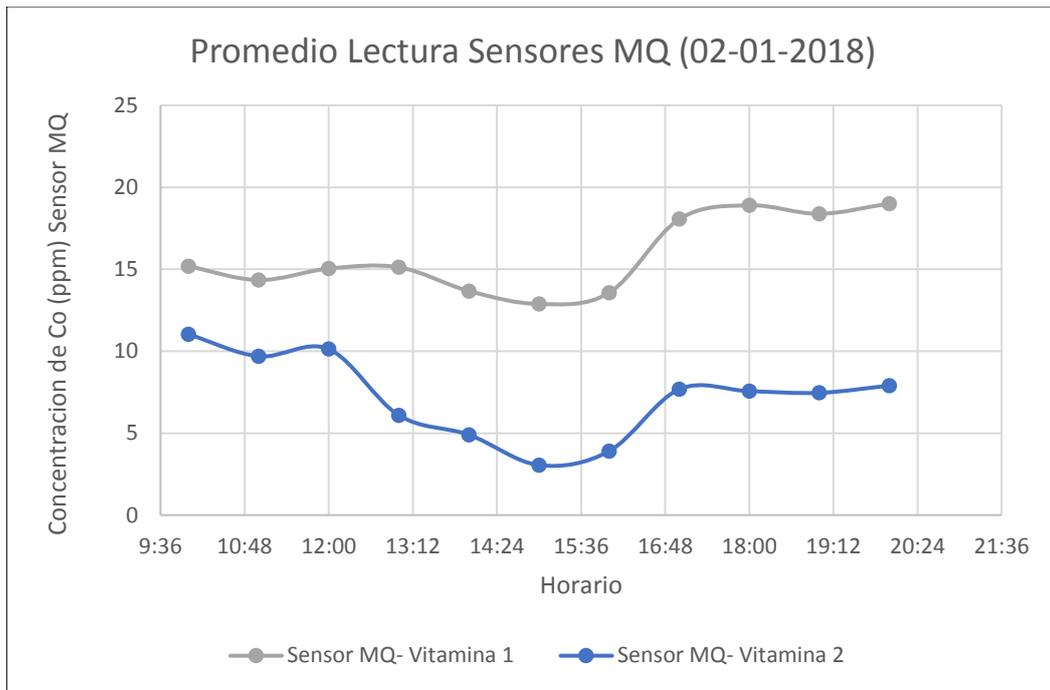


GRAFICO 4: Promedios de mediciones día 1
(Fuente: Elaboración propia)

Además del valor de las mediciones, se registró la cantidad de datos capturados por cada estación de medición. Es necesario mencionar que previo al experimento, se debe instalar la plataforma y preparar los sensores (MQ, GPS, Sonido). Este procedimiento dura una hora aproximadamente. A continuación, se realizan las mediciones de prueba instalando el nodo coordinador en una posición intermedia a las dos estaciones de medición, lo cual tuvo una duración de dos horas aproximadamente para luego ubicarlo en el Jardín Vitamina nº1, en donde permaneció el resto de las horas.

Para determinar el total de datos teórico por hora que se esperan recibir, se multiplica la cantidad de datos enviados teóricos que son cinco por nodo (2 datos de gps, 2 de sensores y 1 de id nodo) por la tasa de transferencia que es de 1 dato por segundo por 3600 (60 segundos x 60 minutos).

Martes 02-01- 2018	hora	Estación Vitamina 1	Estación Vitamina 2	Datos teóricos
		Numero de Datos recibidos	Numero de Datos recibidos	
	10:00	15.678	16.450	18.000
	11:00	15.789	16.790	18.000
	12:00	17.980	15.690	18.000
	13:00	17.920	15.676	18.000
	14:00	17.988	16.023	18.000
	15:00	17.760	15.689	18.000
	16:00	17.996	15.667	18.000
	17:00	18.013	15.634	18.000
	18:00	17.983	15.688	18.000
	19:00	17.845	15.720	18.000
	20:00	17.984	15.601	18.000
	Total	192.936	174.628	198.000

TABLA 16: Volumen de datos capturados primer día de mediciones

(Fuente: Elaboración propia)

A modo de ejemplo, se presentan los resultados de las mediciones del segundo día

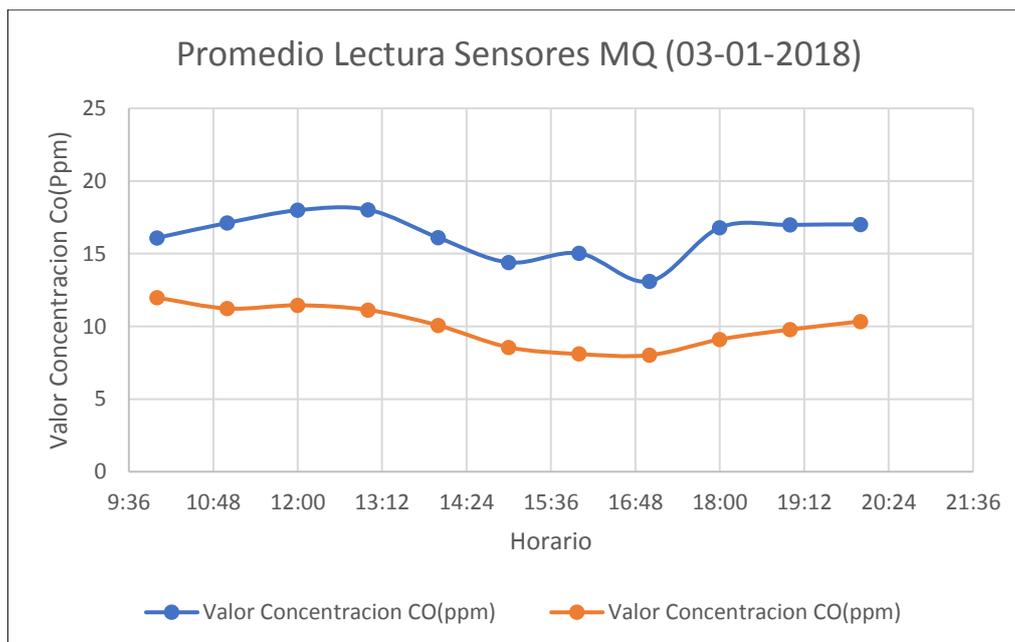


GRAFICO 5: Promedios de mediciones segundo día

(Fuente: Elaboración propia)

A continuación, se presentan una gráfica resumen de los promedios de las mediciones diarias de la primera semana.

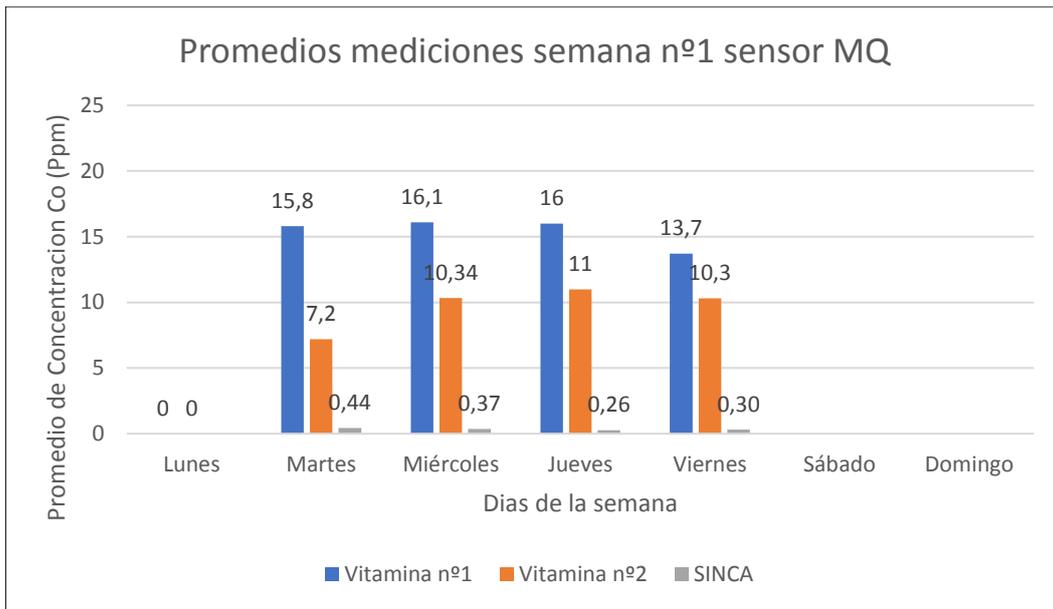


GRAFICO 6: Promedios de mediciones primera semana
(Fuente: Elaboración propia)

Los resultados de las mediciones de los sensores MQ, para las dos semanas se presenta a continuación.

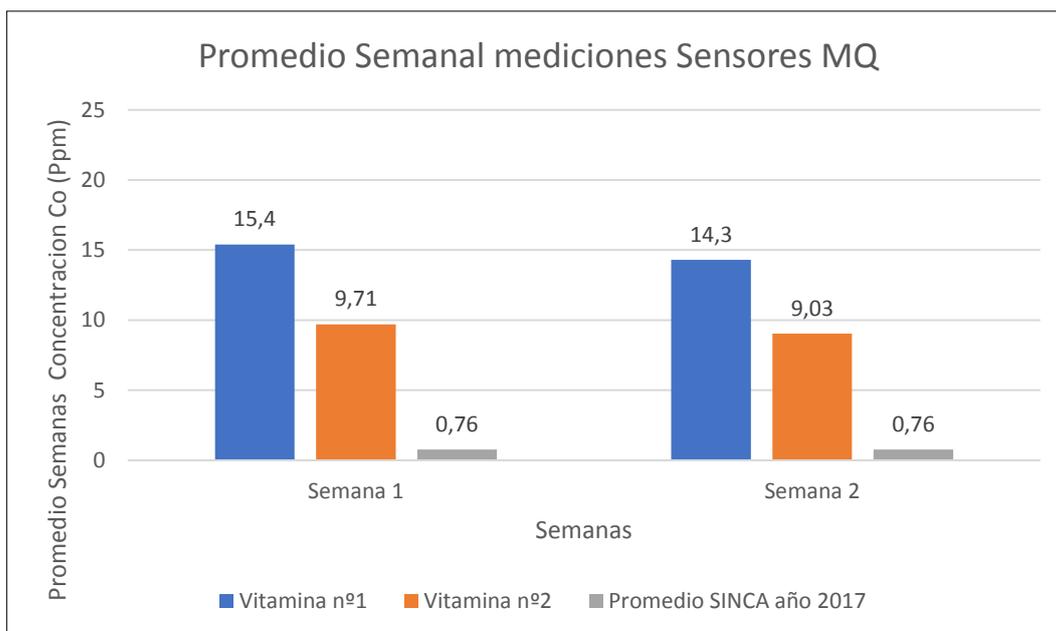


GRAFICO 7: Resumen de Promedios semanales de lectura sensor MQ
(Fuente: Elaboración propia)

En cuanto a las mediciones de ruido, se presentan los gráficos que son solo

referenciales ya que el sensor de sonido utilizado presentaba una sobre valoración de las magnitudes de decibeles, por lo cual se optó por registrar la lectura analógica. La idea fundamental es observar los datos, la factibilidad de detección de patrones de comportamientos cualitativos.

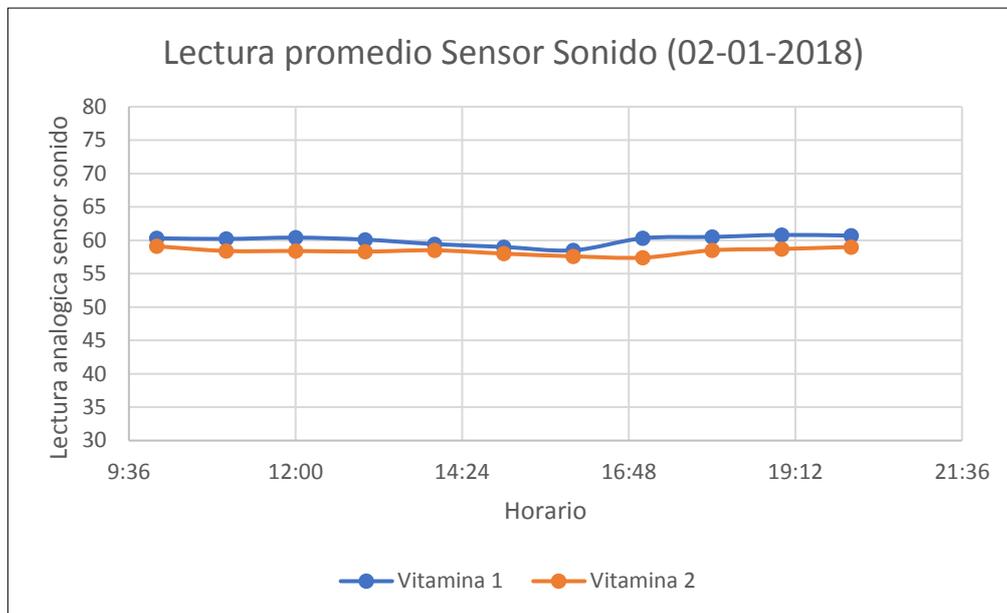


GRAFICO 8: Promedios de lectura analógica sensor sonido primera día
(Fuente: Elaboración propia)

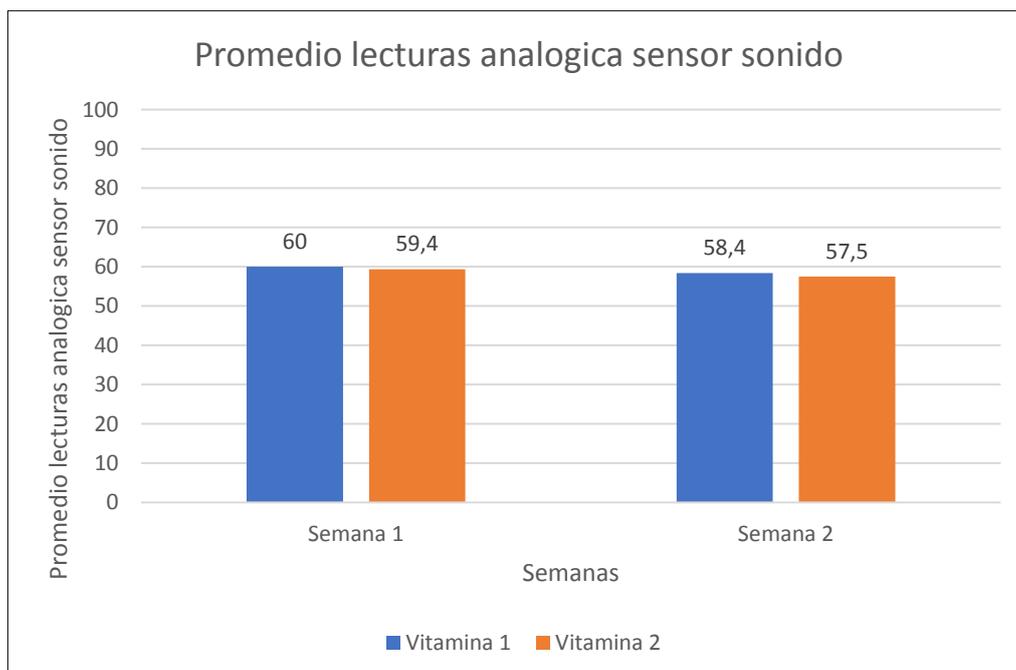


GRAFICO 9: Resumen de Promedios semanales de lectura sensor sonido
(Fuente: Elaboración propia)

Respecto a los datos de captura y almacenamiento de data, se presenta una gráfica resumen de la primera semana, con sus respectivos porcentajes de captura respecto al total teórico estimado.

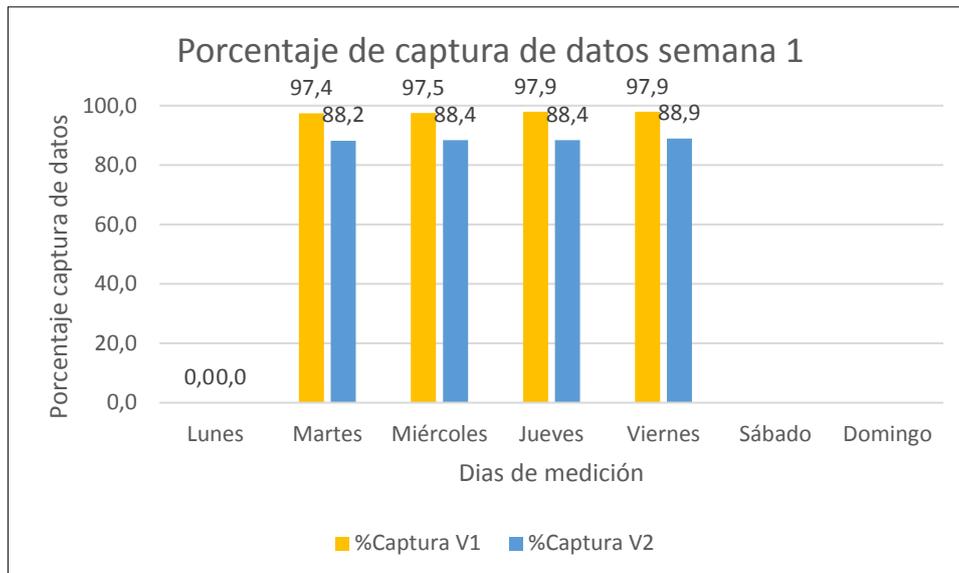


GRAFICO 10: Promedios de porcentaje de datos capturados primera semana
(Fuente: Elaboración propia)

Respecto a los porcentajes de captura de datos obtenidos durante las dos semanas, se presenta el siguiente grafico

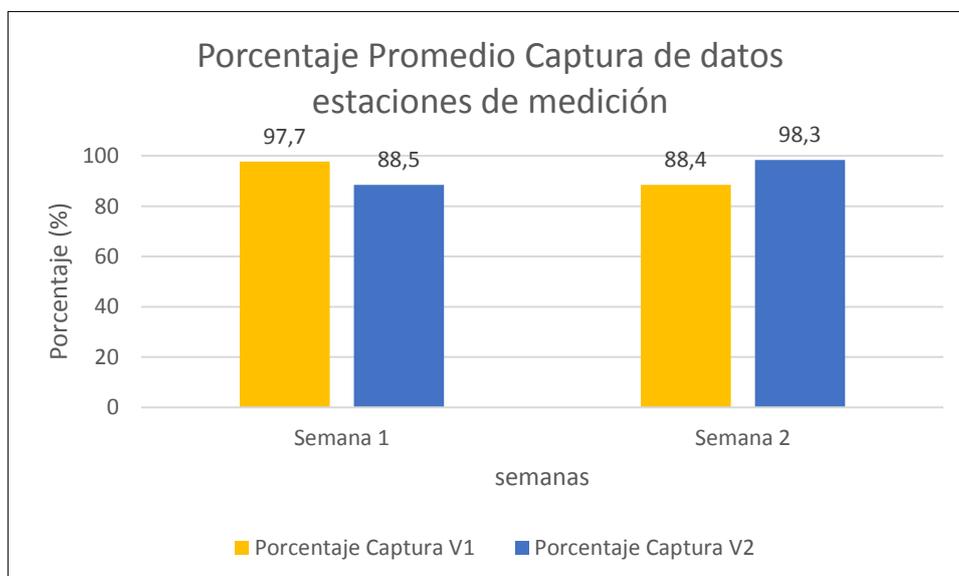


GRAFICO 11: Resumen de Porcentaje Promedio semanales de datos capturados
(Fuente: Elaboración propia)

9.3 Análisis de datos

- Se observa que las mediciones obtenidas por el sensor MQ-7 ubicado en la estación Vitamina nº1 son levemente superiores a Vitamina nº2 durante todo el periodo del experimento. Para validar los datos, se realizó un intercambio de sensores entre los nodos y se obtuvo la misma tendencia. Esta situación puede deberse a que el Jardín Vitamina nº1 está ubicado en una avenida principal de Puente Alto con más flujo vehicular que el Jardín Vitamina nº2
- En los resultados de las mediciones diarias, se logra visualizar comportamientos diferentes en los promedios obtenidos en la mañana y tardes versus los promedios del medio día (12:00 a 16:00). Una interpretación inicial de este comportamiento puede ser la dependencia de la medición con el flujo vehicular a diferentes horas, es decir aumentan en las mañanas y tarde y disminuye al medio día.
- En el grafico nº11 sobre porcentaje de captura de datos, se puede verificar que existe una relación directa entre la distancia del nodo de medición con respecto a la cantidad de datos recibidos. Esto se manifiesta en el mayor porcentaje de datos recibidos por la estación ubicada en el jardín vitamina nº1 donde se encontraba el servidor de monitoreo, (97%) versus el (88%) de la estación de medición de Jardín Vitamina nº2, esta situación se es concordante con las pruebas realizadas en el Test nº8.
- Queda en evidencia que el rango de concentraciones medido por el sensor MQ-7 está entre 5 y 20 ppm. Este rango se ha mantenido durante todo el periodo de mediciones. Estos valores debieran ser comparado con mediciones en los meses de mayor actividad laboral (marzo y abril). A modo referencial, se puede realizar comparaciones primitivas de los rangos de medición obtenidos por los sensores MQ-7 versus los datos disponibles para igual fecha en el SINCA. De acuerdo con esto, se observa que las concentraciones medidas corresponden a 10 o 20 veces superiores a los obtenido por el sensor Thermo 48i, lo que permite verificar el nivel de sensibilidad de cada sensor y los rangos en que trabajan cada uno de ellos.
- Respecto a las mediciones de sonido, se pudo comprobar que los sensores utilizados no proporcionaban información verosímil cuando se realizaba la transformación a decibeles, por lo tanto, se decidió registrar solo la información de la lectura analógica.

X. Conclusiones

Con relación a la hipótesis planteada y objetivos propuesto, se puede mencionar que:

- La plataforma permite desplegar en forma sencilla, rápida y escalable la captura de datos georeferenciado en zonas de interés, lo cual proporciona la capacidad de hacer monitoreo focalizado y apoyar a los sistemas de monitoreo tradicionales y a usuarios interesados. Esta cualidad permite verificar la hipótesis matriz de este trabajo que mediante los experimentos de monitoreo realizados en terreno, se valida la capacidad que poseen los componentes y protocolos de IoT (ZigBee y WebSockets) como tecnología robusta para la transmisión y despliegue en tiempo real de información georeferenciada. Esto se comprueba mediante el grafico nº11 que evidencia un nivel de captura promedio superior al 80%.
- La plataforma contribuye al aumento del nivel de información ambiental disponible por kilómetro cuadrado. En particular en la comuna de Puente Alto se agrega a la estación actual de medición (SINCA), dos prototipos de medición de monóxido de carbono que permiten mediante sensores de bajo costo identificar patrones primitivos de comportamiento de variables ambientales en función de la ubicación geográfica.
- El despliegue de la data capturada por la plataforma permite aumentar el porcentaje de información web de monitoreo en línea disponible actualmente. Esta cualidad se verifica mediante un catastro de sitios web que disponen de información meteorología o ambiental (Anexo 2). En este catastro se identifican 20 sitios nacionales con información relacionada, de los cuales un 20% despliegan información medioambiental en línea. De acuerdo con esto, la plataforma contribuye en aumentar a un 25% la cantidad de sitios relacionados con el monitoreo ambiental en línea.
- La plataforma permite interactuar en forma instantánea con los interesados mediante la utilización con redes sociales, en particular enviar mensajes de alerta a twitter, esto reduce los tiempos de aviso considerablemente al verificarse el aumento de algún índice ambiental en la zona de interés. Esta cualidad permite diferenciar la plataforma con las similares de su tipo hasta el año 2017.

El detalle del cumplimiento de los objetivos de la tesis se presenta en la tabla n°17.

Objetivo	Descripción	Métrica	Resultado Esperado	Resultado Final
OEM1	Aumentar nodos de medición de variables medioambientales en una comuna	Cantidad de nodos de medición por km ² en comuna Puente Alto	Aumentar de una estación de medición a tres, para cubrir 88 km ² en la comuna de Puente Alto.	La plataforma agrega dos nodos de medición para la comuna de Puente Alto, proporcionando información focalizada de concentración de Monóxido de Carbono.
OEM2	Aumentar porcentaje información focalizada de variables ambientales en tiempo real	Porcentaje información medioambiental focalizada	Aumentar a un 25% las plataformas de información medioambiental en línea	De acuerdo con el catastro Anexo 2, se dispone de un total de 20 sitios con información relacionada, de los cuales 4 de ellos disponen de información medioambientales en línea, por lo tanto, se aumenta a 5 aumentando en 25% del total
OEM3	Reducir el tiempo de aviso y alerta de eventos de riesgos de salud de los niños	Índice de alerta temprana de condiciones riesgosas de contaminación para niños en jardín infantil	Sistema permita enviar alertas a interesados en menos de 15 minutos	La plataforma permite vía Twitter, alertar en menos de 1 minuto a los padres y profesores, si algún índice medioambiental aumenta.

TABLA 17: Tabla de Validación de Objetivos Específico Medibles

(Fuente: Elaboración propia)

Respecto a conclusiones de carácter específico y técnico, se puede mencionar que:

- La utilización de tecnología de WebSockets ha proporcionado una excelente alternativa para el envío en tiempo real de data entre un servidor y un cliente web, sin necesidad de la utilización de tecnología basadas en servidor mqtt o tecnología pull como Ajax.

- La tecnología Zigbee de largo alcance basada en el módulo (ZDK Electronics) ha permitido proporcionar una excelente alternativa de transmisión inalámbrica de bajo costo con niveles de autonomía de comunicaciones eficiente al trabajar con una red de sensores dedicada con protocolo estándar.
- La plataforma es poliglota ya que permite utilizar variedad de sensores y unidades de control (Arduino, PIC y PLC Siemens) sin realizar modificaciones al núcleo del sistema debido a la arquitectura de capas y la utilización de protocolos de comunicación estandarizados.
- Con los datos obtenidos durante dos semanas se ha podido establecer un patrón primitivo de comportamiento focalizado. Es decir, se evidencia una diferencia de 2 a 5 puntos en el nivel de concentración de monóxido de carbono entre cada Jardín Infantil, lo cual permitiría categorizar a un nodo de medición en función de sus mediciones ambientales y ubicación. Esta información debe ser ratificada en el futuro con mayor precisión con algún sistema de percepción de mayor calidad y además debe ser contrastado con mediciones en diferentes meses del año. Sin embargo, esta detección de un patrón de comportamiento permite aumentar el nivel de información ambiental que se dispone de una determinada ubicación proporcionando información geográfica desagregada útil que se puede complementar con la información agregada geográfica que entrega el SINCA.

Como reflexiones finales, se puede comentar lo siguiente:

- Se propone continuar con la investigación y mejoramiento de la plataforma incluyendo un sistema de percepción de mejor calidad, que incorpore sensores de mediana gama los cuales proporcionen datos comparables con sistemas utilizados en las estaciones de medición certificadas.
- La siguiente etapa del trabajo debe incorporar el despliegue de los datos en una plataforma cloud tipo PAAS que proporcione interconectividad, persistencia y despliegue de cada componente del sistema.

XI. Referencias Bibliográficas

José Olavo Da Paz Teixeira, (2015), Una Red de Sensores para las Smart Cities, Universidad Complutense de Madrid, España.

Marc Fabregas, (2016), Diseño e implementación de una estación meteorológica de bajo coste con conectividad a Internet, Universidad Politécnica de Valencia, España

Gauderman WJ, (2006). Air pollution and children-an unhealthy mix. The New England journal of medicine. 6;355(1):78-9. PubMed PMID: 16823000.

Eder W, Ege MJ, von Mutius E. (2006). The Asthma Epidemic. New England Journal of Medicine;355(21):2226-35. PubMed PMID: 17124020

Sébastien Sonko, (2013), “Privacy Risk Area Assessment Tool for Audio Monitoring – from legal complexity to practical applications”, Proceedings on 8th International Conference on Legal, Security and Privacy Issues in IT (LSPI).

Ostro B, Sánchez JM, Aranda C, Eskeland GS. (2006). Air pollution and mortality: results from a study of Santiago, Chile. J Expo Anal Environ Epidemiol

Román O, Mancilla P, Prieto MJ, (2004), Contaminación atmosférica y daño cardiovascular. Revista Médica de Chile; 132: 761-7.

Rosales JA, Torres VM, Olaiz G, Borla VH, (2001), Los efectos agudos de la contaminación del aire en la salud de la población: Evidencias de estudios epidemiológicos. Salud Pública; 43: 544-55.

Página Oficial de European Commission – Environment:

http://ec.europa.eu/environment/index_en.htm [Accedido en 09-1-2017].

Ministerio de Medio Ambiente. (2017). Mapa De Calidad Del Aire De La Región Metropolitana. 2017, de Ministerio de Medio Ambiente Gobierno de Chile Sitio web: Mapa De Calidad Del Aire De La Región Metropolitana

Ministerio Medio Ambiente. (2017). Mapa del ruido diurno y nocturno identifica lugares con mayor y menor contaminación acústica en Gran Santiago. 2017, de Ministerio Medio Ambiente, Gobierno de Chile Sitio web: <http://portal.mma.gob.cl/mapa-del-ruido-diurno-y-nocturno-identifica-lugares-con-mayor-y-menor-contaminacion-acustica-en-gran-santiago/>

Kevin Ashton. (2009). That 'Internet of Things' Thing In the real world, things matter more than ideas. RFID journal, 10, 12,13,14.

Estrin, D., Culler, D., Pister, K., & Sukhatme, G. (2002). Connecting the physical world with pervasive networks. IEEE Pervasive Comput. IEEE Pervasive Computing, 1, 59-69. doi:10.1109/mprv.2002.993145

Gordana Gardašević. (2016). The IoT Architectural Framework, Design Issues and Application Domains. Sciencedirect, 92, 127–148.

Vermesan, O., & Friess, P. (2013). Internet of things: Converging technologies for smart environments and integrated ecosystems. The River Publishers Series in Communications.

García Duarte (2016). Diseño, desarrollo e implementación de un protocolo de comunicaciones entre sensores en red y computadores, (Tesis de Magister), Universidad Autónoma de Madrid, España, Recuperado de <https://repositorio.uam.es/handle/10486/673231>, Noviembre 2017.

Morales, Ignasi de Sola (2001). Introducción a la arquitectura: Conceptos fundamentales. Segunda Edición, España: Ediciones UPC.

Roger S. Pressman. (2001). Ingeniería del Software: Un enfoque práctico. Séptima Edición. USA: Mc Graw Hill.

Cazila, J., & Junco, R. (2015). Descubriendo Internet of Things: Una nueva manera de relacionarnos con nuestro entorno, Segunda Edición, Canadá: Cisco Systems.

Scott M. Lewandowski, (1998). Framework for Component-Based Client/Server Computing. ACM Computing Surveys, 30, 1.

Petrolo R., Morabito R., Loscri V., Mitton N. (2016). The design of the Gateway for the Cloud of Things. Annals of Telecommunications, Springer, 10, 2.

Campoverde A., Hernández D, Mazón B. (2015), Cloud computing con herramientas open-source para Internet de las cosas, Universidad Técnica de Machala, Ecuador, MASKANA, CEDIA 2015

Gilerth, S, N. Lynch, (2012). Perspectives on the CAP Theorem. Computer: IEEE Computer Society.

McCreary, D., A. Kelly, (2014). Making sense of NoSQL, Shelter Island: Manning, 19-20 pp.

Anderson, C., Lehnardt, N. Slater, (2015). CAP en CouchDB. Disponible en <http://guide.couchdb.org/draft/consistency.html>.

Sosinsky, B. (2011). Cloud Computing Bible. Indiana: Wiley Publishing Inc., 3 pp.

MongoDB para IoT, (2015). MongoDB en entornos IoT. Disponible en <http://www.mongodb.com/usecases/internet-of-thing>

HiveMQ, (2015). Paradigma de mensajería PUB/SUB. Disponible en <http://www.hivemq.com/mqttessentials-part2-publish-subscribe/>.

Stanford-Clark, A., H. Truong, (2015). MQTT-SN Specification. Disponible en http://mqtt.org/new/wpcontent/uploads/2009/06/MQTT-SN_spec_v1.2.pdf.

WAMP, (2015). Transportes alternativos para WAMP, 2015. Disponible en <https://tools.ietf.org/html/draft-oberstet-hybi-tavendo-wamp-02#section-5.3>.

LoRa Technology (Online) Disponible en: <https://www.lora-alliance.org/What-IsLoRa/Technology>

SIGFOX (Online) Disponible en: <https://www.sigfox.com/>

Narrowband IoT – Enabling New business opportunities, Huawei (Online) Disponible en: <http://www.huawei.com/minisite/4-5g/img/NB-IOT.pdf>

“Twelve wireless options for IoT/M2M: diversity or dilemma”, Electronic Design (Online) Disponible en: <http://electronicdesign.com/iot/12-wireless-options-iotm2m-diversity-or-dilemma>

George Coulouris, (2012). Distributed Systems: Concepts and Design. Quinta Edición, Estados Unidos: Addison-Wesley.

Carmen Medina López. (2010). Una propuesta metodológica para la realización de búsquedas sistemáticas de bibliografía. Working Papers on Operations Management, 2, 13-30.

Hernández R., & Fernández C. & Baptista P. (2010). Enfoques de la investigación: Cuantitativo y Cualitativo. Metodología de la Investigación (5ta. Edición). México: Ed. Mc Graw Hill.

Project Management Institute. (2013). Guía de los fundamentos para la dirección de proyectos (guía del PMBOK), Quinta Edición, Estados Unidos: Project Management Institute.

Kitchenham, B., (2004)., Procedures for Performing Systematic Reviews, Keele University Technical Report TR/SE-0401, 33 p

K.E. Kelly, J. Whitaker, A. Petty, C. Widmer, A. Dybwad, D. Sleeth, R. Martin, A. Butterfield. (2016). Ambient and laboratory evaluation of a low-cost particulate matter sensor. ScienceDirect-Elsevier-Environmental Pollution, 221, 491-500.

Mariusz Rogulski. (2017). Low-cost PM monitors as an opportunity to increase the spatiotemporal resolution of measurements of air quality. ScienceDirect - Energy Procedia, 128, 437-444.

Gao M, Cao J, Seto E. (2016) A distributed network of low-cost continuous reading sensors to measure spatiotemporal variations of PM_{2.5} in Xi'an, China. *Environ. Pollut*; 199:56–65.

Wang Y, Li J, Jing H, Zhang Q, Jiang J, Biswas P. (2015). Laboratory evaluation and calibration of three low-cost particle sensors for particulate matter measurement. *Aerosol Sci. Technol*; 49:1063–1077.

Sherin Abraham, Xinrong Li. (2015). A Cost-Effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications. *ScienceDirect-Procedia Computer Science*, 34, 165-171.

Jorge E. Gómez, Fabricio R. Marcillo, Freddy L. Triana, Víctor T. Gallo, Byron W. Oviedo, Velssy L. Hernández. (2017). IoT FOR ENVIRONMENTAL VARIABLES IN URBAN AREAS. *ScienceDirect- Procedia Computer Science*, 109, 67-74.

Castell, N., Kobernus, M., Liu, H.-Y., Schneider, P., Lahoz, W., Berre, A., Noll, J., 2015. Mobile technologies and services for environmental monitoring: the Citi-Sense-MOB approach. *Urban Clim.* 14, 370–382.

Castell N, Philipp Schneider, Sonja Grossberndt, Mirjam.F. Fredriksen, Gabriela Sousa-Santos, Mathias Vogt, Alena Bartonova. (2017). Localized real-time information on outdoor air quality at kindergartens in Oslo, Norway using low-cost sensor nodes. *ScienceDirect-Environmental Research*, 90, 40-46.

Ismo Hakala, (2016). Design of Low-Cost Noise Measurement Sensor Network: Sensor Function Design. *Science-Direct*, 98, 35-41.

Ian Sommerville. (2005). Ingeniería del software. Séptima Edición. Estados Unidos: Pearson.

Kruchten, Philippe (1995). Planos Arquitectónicos: El Modelo de “4+1” Vistas de la Arquitectura del Software

XII. Anexos

A1: Proceso de Ajuste Sensor MQ-7

El proceso de calibración y medición del sensor MQ-7 se realiza mediante los siguientes pasos:

1.- Dada la hoja de característica del sensor, se recogen los datos de la gráfica logarítmica correspondiente al gas a detectar (C_0 en este caso).

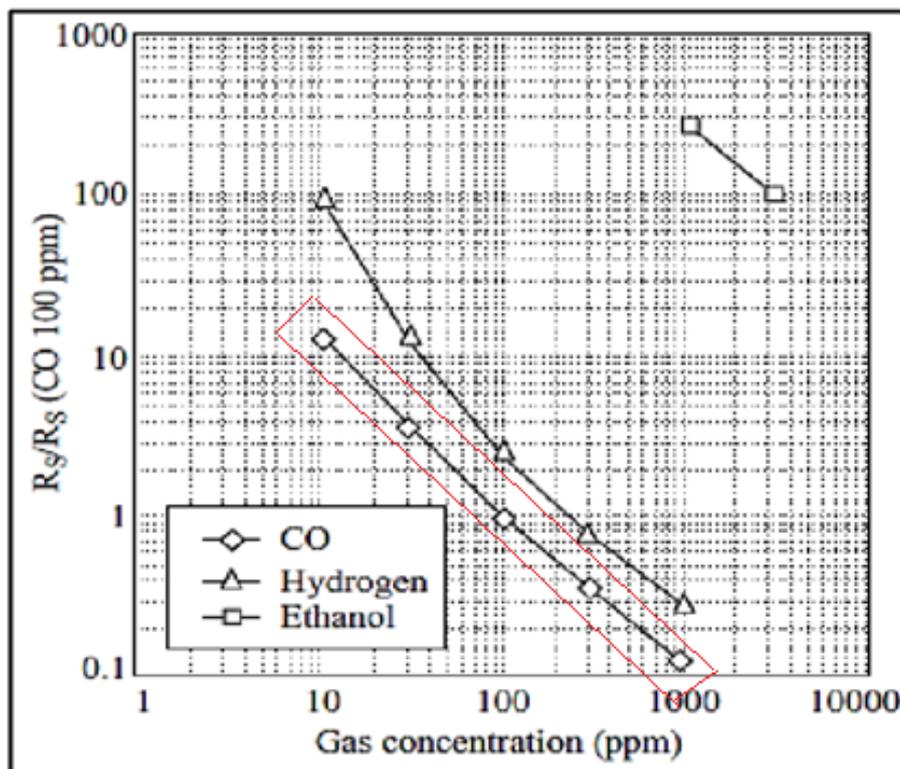


GRAFICO 12: Curva comportamiento logarítmica Sensor de Gas MQ
(Fuente: <http://www.hwsensor.com>)

La grafica del sensor de gas es del tipo logarítmica para un rango de mediciones de 10 a 1000 ppm. Específicamente para el monóxido de carbono se indica con un rectángulo rojo.

Esta grafica es construida en base la razón que existe entre los parámetros R_s , que corresponde a la resistencia variable obtenida por la lectura analógica del sensor. Y

el parámetro R_o que corresponde al valor del sensor obtenido para un valor de concentración conocida de 100 ppm.

2. Con los datos de la curva del sensor, se construye una tabla que se plasma en una hoja de cálculo y se confrontan en una gráfica lineal.

Ppm (C_o)	R_s/R_o
1000	0,1
700	0,2
200	0,5
100	1
60	2
30	4
10	10,5

TABLA 18: Extracción datos desde curva logarítmica
(Fuente: Elaboración propia)

La tabla se grafica con el fin de obtener una representación lineal, además se incluye una línea de tendencia y se calcula el índice de dispersión para encontrar la ecuación que satisfaga los datos. En este caso el coeficiente de correlación de los datos cercano a 1 da la seguridad de que la ecuación abarca la mayoría de los datos.



GRAFICO 13: Curva tendencia comportamiento lineal Sensor de Gas MQ
(Fuente: Elaboración Propia)

3.- Dada una concentración conocida del gas se mide el valor correspondiente de la relación R_s/R_o , correspondiente al eje de abscisas.

4.- Midiendo el valor obtenido en ese entorno de la lectura proporcionada en el microcontrolador de la resistencia en el pin (R_s), se puede conocer el valor R_o despejando la relación.

$$R_o = R_s / \text{relación (valor de x)}$$

5.- Conocido los valores de R_o y de la ecuación de la línea de tendencia se puede obtener la concentración del gas a partir de la ecuación.

$$Y \text{ (ppm)} = 111.3 \times (R_o/R_s) \text{ elevado a } -1.002$$

Empleando esta ecuación se puede obtener con facilidad el valor de concentración, dada R_o (conocida tras la calibración y constante) y la R_s (variables en función de la lectura)

Para el caso dado y calibrando el sensor para 100 ppm de CO_2 , el valor de R_o que se empleará será 6,5 (valor obtenido en http://wiki.seeed.cc/Grove-Gas_Sensor-MQ7/).

A2: Catastro de sitios web con información ambiental o meteorológica

Sitio	Descripción
Monitoreo Ambiental en Línea: http://sinca.mma.gob.cl/	Plataforma oficial del sistema de información nacional de calidad del aire dependiente del ministerio del medio ambiente del gobierno de Chile.
Monitoreo Ambiental en Línea: http://agicn.org/city/chile/puente-alto/es/	Plataforma web que permite integrar información global de estaciones de medición ambiental distribuidas en las grandes ciudades del mundo. En particular la plataforma integra la información del SINCA.
Monitoreo Ambiental en Línea: http://www.redmonitoreoventanas.cl	Portal Medioambiental Codelco División Ventanas y AES Gener
http://www.meteochile.gob.cl/PortalDMC-web/index.xhtml	Portal de la Dirección Meteorológica de Chile, que despliega información meteorológica en tiempo real de 12 estaciones en la región metropolitana.
http://ambiente.usach.cl/meteo/	Portal de la Universidad de Santiago de Chile, que despliega información meteorológica de la comuna de Estación Central.
http://agromet.inia.cl/estaciones.php	Plataforma web de la red agrometeorología de Chile que posee 100 estaciones a nivel nacional midiendo variables meteorológicas administrada por el Instituto de Investigación Agropecuaria.
http://www.agroclima.cl/	Plataforma de medición constituido una red de estaciones meteorológicas en un sistema integrado que administra Fundación desarrollo frutícola.
http://explorador.cr2.cl/	Portal de la Universidad de Chile, que despliega información meteorológica histórica nacional
http://sinia.mma.gob.cl/que-es-el-sinia/	Portal web de acceso al Sistema Nacional de Información Ambiental (SINIA), administrado por el Ministerio del Medio Ambiente y conformado por un conjunto de bases de datos
Monitoreo Ambiental en Línea: http://estaciones.monitoreosserpram.cl/	Red de monitoreo en línea de plantas de generación térmica de empresa ENEL.

Sitio	Descripción
http://www.ceazamet.cl/	Sitio web del Centro de Estudios Avanzados en Zonas Áridas, que posee estaciones meteorológicas en línea.
http://geografiapucv.cl/estacion/estacion/historico_vina.htm	Sitio de la Facultad de ciencias del Mar y Geografía de la PUCV, que permite ver en línea parámetros meteorológicos de una estación ubicada en Viña del Mar.
https://www.centroccb.cl/clima/	Sitio de la Universidad Austral con información en línea de estación meteorológica de una estación ubicada en Valdivia.
http://www.shoa.cl/	Plataforma del Servicio Hidrográfico y Oceanográfico de la Armada de Chile que despliega información meteorológica Nacional.
http://meteo.uv.cl/estaciones/meteo-uv/	Sitio de la Universidad de Valparaíso despliega información meteorológica en línea de 4 estaciones ubicada en la Quinta Región.
http://meteoarmada.directemar.cl	Sitio de la Armada de Chile despliega información meteorológica.
http://www.meteorologia.ucn.cl/	Sitio de la Universidad Católica del Norte despliega información meteorológica en línea de la ciudad de Antofagasta.
http://164.77.222.61/RedEmaNacional/php/VisorMMA.php?codigoNacional=520012	Sitio de la Universidad Magallanes despliega información meteorológica en línea de la ciudad de Punta Arenas
http://www.citrautalca.cl/new/?page_id=233	Sitio de la Universidad Magallanes despliega información meteorológica en línea de la ciudad de Talca.
http://daw.dgeo.udec.cl/	Sitio de la Universidad de Concepción despliega información meteorológica en línea de la Octava Región.
http://sistema.meteorologiaupla.cl/Clima/	Sitio de la Universidad de Playa Ancha despliega información meteorológica en línea de la Quinta Región.

Sitio	Descripción
http://190.151.71.168/melon/publico/	Sitio web de empresa cementera Melón, que posee estaciones de calidad del aire en línea en la V región.
http://nuevaerainforma.enap.cl/seccion/variables-meteorologicas/	Sitio de la empresa ENAP que posee estaciones de calidad del aire en línea en la VIII región.

TABLA 19: Tabla resumen de plataformas web de monitoreo ambiental
(Fuente: Elaboración propia)

A3: Mediciones adicionales en zonas focalizadas

Lugar	Ubicación	Mediciones
		Promedio Concentración Monóxido de carbono (Ppm), Una hora de medición.
Pirque Rural	Sitio 28 San Vicente de Pirque Latitud: -33.4347398 Longitud: -70.5646013	Co = 4,6 ppm (10:05)
Pirque Urbano	Avenida Concha y Toro 2548 Latitud: -33.6347446 Longitud: -70.5736018	Co = 4,87 ppm (11:013)
Estación SINCA Puente Alto	Avenida Ejercito Libertador 2433 Latitud: -33.59313 Longitud: -70.5936109	Co = 4,03 ppm (12:43)
Mall Plaza Independencia	Avenida Concha y Toro 1034 Latitud: -33.6014619 Longitud: -70.5808228	Co = 17,6 ppm (14:05)

TABLA 20: Tabla mediciones adicionales en zonas focalizadas
(Fuente: Elaboración propia)