



ESCUELA DE INGENIERÍA EN COMPUTACIÓN
PROGRAMA DE MAESTRÍA EN COMPUTACIÓN

Diseño de redes de sensores colaborativas para monitorización de salud estructural de puentes

Tesis para optar al grado de Magister Scientiae en Ciencias de la
Computación

Cartago, Costa Rica
Junio de 2019

AUTOR

Aníbal I. Ruiz Barquero

PROFESSOR ASESOR

Dr. César Garita.

APROBACIÓN DE TESIS

“Diseño de redes de sensores colaborativas para monitoreo de salud estructural de puentes”

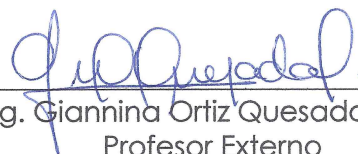
TRIBUNAL EXAMINADOR



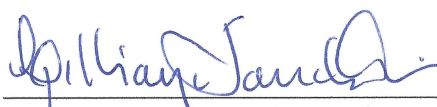
Dr. César Garita Rodríguez
Profesor Asesor



Dr. Carlos Meza Benavides
Profesor Lector



Ing. Giannina Ortiz Quesada, MSc.
Profesor Externo



Ing. Lilliana Sancho Chavarría, MSc.
Coordinadora
Unidad de Posgrado en Computación

Junio, 2019



Resumen

La monitorización de la salud estructural de los puentes implica el almacenamiento y análisis de grandes cantidades de datos provenientes de sensores asociados a variables específicas tomadas en los mismos, las cuales dan un buen indicio para describir la condición general de cada estructura. Normalmente la labor de monitorizar estructuras no es simple y requiere de la integración de varias herramientas que permitan un análisis integral de los datos.

En general la condición de los puentes de Costa Rica es regular, por lo que son estructuras que requieren de la atención tanto de la población como de las autoridades. Con la finalidad de apoyar la prevención de desastres naturales o estructurales en los puentes de Costa Rica, durante la realización de esta tesis, se ha desarrollado el prototipo de red de sensores colaborativa, que utiliza la altura libre inferior como variable de demostración de funcionamiento.

En este contexto, este trabajo tiene como objetivo principal: el desarrollo de una red colaborativa de sensores para monitorizar puentes, buscando poner a disposición tecnología que permita dar algún nivel de autonomía a las estructuras puente en caso de emergencia, mediante la capacidad de los nodos sensor de colaborar entre sí para realizar un razonamiento del estado actual de la estructura a la que pertenecen e incluso de estructuras relacionadas entre sí, apoyando así la prevención de desastres en los puentes. Además se brinda un fácil acceso a los analistas de estructuras, encargados de mantenimiento, encargados de puentes e incluso investigadores; a la información en plazos de tiempo relativamente cortos entre la toma de los datos y la visualización en la interfaz web.

Palabras clave: monitorización, puente, red colaborativa, nodo sensor, RESTful.

Agradecimientos

Especialmente, quiero dar las gracias a mi asesor Prof. Dr. -Ing. César Garita, quien me impulsó a realizar este proyecto y quien me dió la oportunidad de incursionar en investigación y publicación de artículos científicos. Además a todos los profesores que tuve durante la maestría.

Mis más sincero agradecimiento a todo el equipo del CIVCO y especialmente a la M. Sc.-Ing. Giannina Ortiz Quesada por todo su soporte tanto técnico como financiero para la ejecución de la tesis.

Gracias a los asistentes del proyecto de investigación e-Bridge David Gómez Vargas, Steven Sánchez González y Marcelo Sánchez Solano, por toda la ayuda con diversas tareas y que con gran esfuerzo hicieron sus labores.

Un gran agradecimiento a toda mi familia y especialmente a Luisita Montoya mi especial compañera de todo, que con gran cariño me toleraron y soportaron en momentos difíciles, los quiero mucho.

Muchas gracias a todos.

Índice general

1. Introducción	1
1.1. Descripción General	1
1.2. Antecedentes	1
1.2.1. Monitorización de salud estructural (<i>SHM</i>)	2
1.2.2. e-Bridge	3
1.3. Definición del problema	4
1.4. Justificación del proyecto	5
1.4.1. Innovación	5
1.4.2. Impacto	6
1.4.3. Profundidad	6
1.4.4. Hipótesis	7
1.5. Objetivos	7
1.5.1. Objetivo General	7
1.5.2. Objetivos Específicos	7
1.6. Organización del documento	8
2. Trabajos Relacionados	9
2.1. Antecedentes y trabajo relacionado	10
2.1.1. Trabajos relacionados en monitoreo	10
2.1.2. Trabajos relacionados en redes colaborativas y arquitecturas REST-ful	11
2.2. Monitoreo de salud estructural SHM	12
2.2.1. Variables de medición en SHM para puentes	13
2.2.2. Tecnologías utilizadas en sistemas de SHM para puentes	13
2.3. Redes colaborativas(CN), conceptos y aplicaciones	14
2.3.1. Aplicaciones de redes colaborativas	14
2.3.2. Importancia de la colaboración	15
2.4. Internet de las cosas (IoT)	15
2.4.1. Modelos de comunicación del IoT	16
2.4.1.1. Patrón de comunicaciones Device-To-Device	17
2.4.1.2. Patrón de comunicación Device-To-Cloud	17
2.4.1.3. Modelo Device-to-Gateway	18
2.4.1.4. Modelo Back-End Data-Sharing	19

2.4.2.	Arquitectura orientada a los recursos(ROA) para la Web de las cosas(WoT)	19
2.4.3.	Definición de requerimientos y criterios para la implementación de servicios web en IoT	20
2.4.4.	Mejores prácticas de en servicios web en IoT	21
2.5.	Redes de sensores	22
2.5.1.	Redes de sensores inalámbricos(WSN)	22
2.5.1.1.	Comunicaciones en una WSN	22
2.5.2.	Redes de sensores inalámbricas(WSN) para SHM	24
2.5.2.1.	Consideraciones de un SHM con WSN	25
2.5.2.2.	Requerimientos y criterios para <i>SHM</i> en implementación como <i>WSN</i>	26
2.5.2.3.	Mejores prácticas para <i>SHM</i> en implementación como <i>WSN</i>	26
2.5.2.4.	Implementaciones reales de tecnologías de sensores / redes inalámbricos	27
2.5.3.	Redes colaborativas de sensores	27
2.5.3.1.	Modelo de servicio Web colaborativo	27
2.5.3.2.	Requerimientos y criterios para el diseño del protocolo de comunicación entre nodos sensor en una red colaborativa	28
2.5.3.3.	Mejores prácticas en una red colaborativa de sensores	29
2.5.3.4.	Redes Grid y P2P	29
2.5.4.	Servicios Web en redes colaborativas de sensores	29
2.5.4.1.	Generalidades de la interacción Web	30
2.5.4.2.	Algoritmos y generalidades de implementaciones	31
2.6.	Proyecto e-Bridge	32
2.6.1.	Enfoque e impacto de e-Bridge	33
2.6.2.	Etapas y actualidad del proyecto	33
2.6.3.	Logros del proyecto	34
3.	Análisis de Requerimientos	35
3.1.	Diagrama de contexto	35
3.2.	Requerimientos funcionales del prototipo desarrollado	37
3.2.1.	Creación y operación de la red colaborativa de alerta	38
3.2.1.1.	Servidor de Aplicaciones	39
3.2.2.	Monitorización de la estructura puente	41
3.2.3.	Gestión de alarmas	43
3.2.4.	Administración de la configuración	44
3.2.4.1.	Determinación de nodo maestro en una estructura puente	45
3.2.5.	Análisis de los datos del sistema	47
3.3.	Modelo de datos	48
3.3.1.	Mediciones	49

3.4. Requerimientos técnicos de componentes del prototipo	50
3.4.1. Sensor	50
3.4.2. Empotrado	50
3.4.3. Módulo de conexión local	52
3.4.4. Módulo de conexión a la red global	52
3.4.5. Servidor	52
3.4.6. Red inalámbrica	53
4. Diseño	54
4.1. Diseño de la arquitectura de la red	54
4.1.1. Configuraciones de puentes	56
4.1.2. Arquitectura del servidor en la nube	56
4.1.3. Nodo sensor de medición y conexión a la red	57
4.2. Diseño del protocolo	58
4.2.1. Estructura de datos	58
4.2.1.1. Modelo Conceptual de la base de datos	58
4.2.1.2. Modelo Entidad-Relación de la base de datos	59
4.2.2. API e-Bridge	60
4.2.2.1. <i>Login</i>	61
4.2.2.2. Registro	62
4.2.2.3. Configuración	63
4.2.2.4. Puentes	64
4.2.2.5. Dispositivo	65
4.2.2.6. Reportes	66
4.2.3. Protocolo y mensajería del Hardware	67
4.2.3.1. Comunicación entre Raspberry Pi 3 y sensor SRF-08	67
4.2.3.2. Comunicación entre Raspberry Pi 3 y módulo xBee S2C	68
4.2.4. Sistema de alertas	68
4.2.4.1. Alertas Locales	69
4.2.4.2. Alertas Globales	69
4.3. Principales funciones y algoritmos	71
4.3.1. Filtro de mediana móvil	71
4.3.2. Falsos positivos	71
4.3.3. Determinación de mejor señal	72
4.4. Configuración del sistema	72
4.4.1. Configuración de dispositivos	73
4.4.2. Configuración de redes	74
4.4.3. Creación de puentes	75
4.5. Mapa o plano general del sitio	75
4.6. Wireframe de la página principal	76
5. Implementación	78
5.1. Descripción de los ambientes de programación y software	78

5.1.1. Sistemas operativos	78
5.1.2. Lenguajes de programación	79
5.1.3. Scripts de arranque de empotrados	81
5.2. Servidor en la nube, módulos y clases	82
5.2.1. Servidor de aplicaciones	83
5.2.2. Servidor Web	84
5.2.3. Base de datos	84
5.3. Nodo sensor	85
5.3.1. Flujo, clases y paquetes	86
5.3.2. Protocolo de alertas	88
5.3.3. Protocolo de comunicación	88
5.3.3.1. Servicios Web	88
5.3.3.2. Comunicación entre nodos sensor por medio de SMSs	89
5.3.3.3. Xbee	90
5.3.4. Filtro y toma de mediciones	90
5.4. Repositorio en GitHub	91
6. Validación y Pruebas	94
6.1. Escenarios de prueba	94
6.1.1. Escenario de demostración para redes en un puente	95
6.1.2. Escenario de demostración para redes de varios puentes	96
6.2. Pruebas funcionales	97
6.2.1. Pruebas de altura libre para escenario de un puente	97
6.2.2. Pruebas de altura libre para escenario de varios puentes	102
6.2.3. Protocolo de comunicación para estructuras - local	108
6.2.4. Protocolo de comunicación para redes de estructuras- global	110
6.2.5. Generación de alertas para estructuras	113
6.2.6. Protocolo de alertas para redes de estructuras	114
6.3. Pruebas de rendimiento	116
6.3.1. Tiempos de propagación de alertas	116
6.3.2. Costos operativos entre SMS vs REST por unidad de tiempo	119
6.3.3. Tiempos de autonomía con baterías ICR18650 4400mAh 3.7V	120
6.4. Consulta y validación con expertos	120
6.5. Análisis de resultados	122
7. Conclusiones	124
7.1. Conclusiones	124
7.1.1. Conclusiones sobre trabajos relacionados	124
7.1.2. Conclusiones sobre el análisis de los requerimientos	125
7.1.3. Conclusiones sobre el diseño	125
7.1.4. Conclusiones sobre la implementación	126
7.1.5. Conclusiones sobre las pruebas	126
7.2. Trabajo Futuro	127

ÍNDICE GENERAL

IX

8. Referencias

129

Índice de cuadros

2.1. Operaciones HTTP/REST(CRUD)	20
2.3. Requerimientos de tasa de transferencia en nodos sensor, para toma de datos de valores físicos comunes en SHM. Tomado de [68]	25
3.1. Definición de alarmas de emergencia para nivel de agua bajo el puente	42
4.1. Método y URL en el login	61
4.2. Parámetros por método en el Login	61
4.3. Respuesta por código en el Login	62
4.4. Método y URL en el Registro	62
4.5. Parámetros por método en el Registro	62
4.6. Respuesta por código en el Registro	63
4.7. Método y URL en la Configuración	63
4.8. Parámetros por método en la Configuración	63
4.9. Respuesta por código en la Configuración	64
4.10.Método y URL para Puentes	64
4.11.Parámetros por método para Puentes	64
4.12.Respuesta por código para Puentes	65
4.13.Método y URL para Dispositivos	65
4.14.Parámetros por método para Dispositivos	65
4.15.Respuesta por código para dispositivos	66
4.16.Método y URL para Reportes	66
4.17.Parámetros por método para Reportes	66
4.18.Respuesta por código para reportes	67
4.19.Parámetros para comunicación de hardware	68
4.20.Algunos escenarios del Protocolo para Alertas Locales	69
4.21.Algunos escenarios del Protocolo para Alertas Globales	71
6.1. Mediciones de red local - Un solo puente	98
6.2. Variables estadísticas de los datos tomados para el escenario de red colaborativa de un sólo puente, cuadro 6.1	101
6.3. Mediciones de red global - redes de varios puentes	102
6.4. Variables estadísticas de los datos tomados para el escenario de varios puentes, cuadro 6.3	107

6.5. Mensajes transmitidos entre nodos sensor maestro y esclavo para el escenario de un solo puente y su hora de propagación	109
6.6. Mensajes transmitidos entre nodos sensor maestro para el escenario de varios puentes y su hora de propagación	111
6.7. Mediciones de red local con alarma - un solo puente	113
6.8. Mediciones de red global con alarma - varios puentes	114
6.9. Mensajes transmitidos entre puentes por medio de tecnología SMS . .	117
6.10. Mensajes transmitidos entre puentes por medio de tecnología REST . .	118
6.11. Costos de operación para las redes colaborativas con diferentes tecnologías	120
6.12. Comparación entre tecnologías para las redes colaborativas con diferentes tecnologías	122

Índice de figuras

1.1. Topología general de la red de sensores colaborativa.	4
2.1. Diagrama de la taxonomía de la investigación.	9
2.2. Patrón de comunicación Dispositivo-Dispositivo.	17
2.3. Patrón de comunicación Dispositivo-Nube.	18
2.4. Patrón de comunicación Dispositivo-Gateway.	18
2.5. Patrón de comunicación Back-End Data-Sharing.	19
2.6. Modelos de representación ROA para WoT.	20
2.7. Topología Single-Hop[66].	23
2.8. Topología Multi-Hop[66].	24
2.9. Puertos de usuario y de recursos de un servicio web[29].	28
2.10. Condición General de los puentes en Costa Rica - proyecto e-Bridge.	33
2.11. Diagrama de desarrollo general del proyecto e-Bridge.	34
3.1. Diagrama de contexto.	36
3.2. Diagrama de casos de uso para la red colaborativa de prevención de desastres.	37
3.3. Diagrama de secuencia para la creación y operación de la red colabo- rativa de sensores.	38
3.4. Diagrama de secuencia del Servidor de Aplicaciones.	40
3.5. Casos de uso del Servidor de Aplicaciones.	41
3.6. Diagrama de secuencia para monitorizar la estructura puente.	42
3.7. Casos de uso para la obtención de los datos en los nodos sensor.	43
3.8. Diagrama de secuencia para la gestión de alarmas.	44
3.9. Casos de uso de la gestión de alarmas.	45
3.10. Casos de uso para la administración de la configuración.	46
3.11. Casos de uso para selección del nodo de enlace a la red colaborativa.	47
3.12. Diagrama de secuencia de la selección del nodo de enlace a la red co- laborativa.	48
3.13. Casos de uso para el análisis de los datos.	49
3.14. Diagrama de secuencia para el análisis de los datos del sistema.	50
3.15. Estructura del esquema de la base de datos del sistema.	51
3.16. Tabla de Mediciones.	52

4.1. Diseño general de la red.	55
4.2. Servidor en la nube.	57
4.3. Nodo sensor Maestro-Esclavo.	58
4.4. Diagrama conceptual de la base de datos.	59
4.5. Diagrama relacional de la base de datos - Notación Crowsfoot.	60
4.6. Diagrama de módulos del API.	61
4.7. Ejemplo de alerta global.	70
4.8. Interfaz web.	72
4.9. Interfaz para crear dispositivos	73
4.10. Configuración de dispositivos.	73
4.11. Interfaz para crear redes	74
4.12. Configuración de redes.	74
4.13. Interfaz para crear puentes	75
4.14. Plano general del Sitio.	76
4.15. Boceto de la interfaz web, Wireframe. Notación tomada de [54].	77
5.1. Respuesta a comandos <i>lsb_release - a</i> y <i>hostnamectl</i>	79
5.2. Versión de Raspbian para dispositivos empotrados.	79
5.3. Arquitectura general del sistema a base de componentes de software.	80
5.4. Versión de servidor de Node.js, NPM y Express.	80
5.5. Versión de servidor de Angular y Typescript.	81
5.6. Sitio web de e-Bridge.	81
5.7. Script de arranque de los dispositivos empotrados(Nodo Sensor).	82
5.8. Diagrama de clases del servidor de aplicaciones.	83
5.9. Diagrama de módulos de la página web.	84
5.10. Script de deployment de la base de datos “ebridge”.	85
5.11. Diagrama de flujo de empotrados.	86
5.12. Diagrama de clases del empotrados.	87
5.13. Diagrama de paquetes del empotrados.	87
5.14. Implementación del protocolo de alertas.	88
5.15. Implementación de servicios web en estructura RESTful.	89
5.16. Implementación de funciones para mensajería SMS entre dispositivos de diferentes estructuras puente.	89
5.17. Implementación de Xbee.	90
5.18. Filtrado de los datos mediante media y mediana móvil.	91
5.19. Repositorio principal de la red colaborativa de sensores de e-Bridge.	91
5.20. Repositorio del servidor de aplicación de la red de sensores e-Bridge.	92
5.21. Repositorio del servidor web de la red de sensores e-Bridge.	93
6.1. Escenario de demostración del protocolo para una estructura puente.	95
6.2. Configuración de nodos para escenario de un solo puente.	96
6.3. Configuración de nodos para escenario de varios puentes.	96
6.4. Escenario de demostración del protocolo global para varios puentes.	97

6.5. Gráfico de la distribución normal de los datos del cuadro 6.1.	102
6.6. Gráfico de la distribución normal de los datos del cuadro 6.3 para el puente llamado “PuenteEntrada”.	108
6.7. Gráfico de la distribución normal de los datos del cuadro 6.3 para el puente llamado “PuenteResidencias”.	109
6.8. Medición manual de la altura de cada puente.	111
6.9. Costo de kB vs SMS para la operadora Kolbi.	119
6.10. Datos de consumo de datos del dispositivo 22.	120
6.11. Medición de duración de batería 4400mHh.	121
6.12. Resumen de las respuestas obtenidas en la consulta a expertos.	122

Capítulo 1

Introducción

En este documento se presenta la realización de tesis de posgrado para optar por el grado de Magister Scientiae en Ciencias de la Computación del Instituto Tecnológico de Costa Rica. El tema propuesto se relaciona con el paradigma de ciencia de diseño[2] y corresponde a la elaboración de una red de sensores colaborativos inteligentes para la monitorización de puentes en Costa Rica.

La monitorización para salud de estructuras viales se ha convertido en prioridad para muchos países alrededor del mundo, cuyo propósito es el de reducir vulnerabilidades y mejorar la protección y mantenimiento a estructuras críticas contra desastres naturales, e incluso, ante otros acontecimientos dados en la actualidad como lo son los ataques terroristas.

El prototipo desarrollado pretende a largo plazo concentrar y facilitar la información generada por varios tipos de sensores que forman parte del proyecto e-Bridge en un sólo sistema, poniendo a disposición de las diferentes entidades involucradas (MOPT, CONAVI, CIVCO, etc), una estructura computacional colaborativa en función de la determinación del estado de salud de las estructuras. Los recursos computacionales utilizados, permiten habilitar la comunicación entre dispositivos que normalmente no son capaces de hacerlo.

En este primer capítulo de introducción, se presentan las secciones de:

- Antecedentes.
- Definición del problema.
- Justificación del proyecto.
- Objetivos.

1.1. Descripción General

1.2. Antecedentes

En esta sección se presenta una introducción al marco general de la investigación propuesta, enfocándose en los conceptos básicos de monitorización de salud

estructural y del proyecto e-Bridge, dentro del cual se desarrolla este trabajo. En el capítulo 2, se profundizará en otros aspectos de antecedentes y trabajos relacionados.

1.2.1. Monitorización de salud estructural (*SHM*)

La monitorización de la salud estructural es un campo emergente en ingeniería civil que tiene el potencial para la evaluación continua y periódica de la seguridad e integridad de la infraestructura civil. El objetivo de SHM es conocer la condición de la estructura y así llevar a cabo ciertas medidas preventivas para prolongar la vida útil de una estructura, además de prevenir fallas catastróficas. Los parámetros están asociados a cualquier variable de la que se obtienen valores a través de un sensor, como lo sería la medición del nivel del agua del río que pasa por debajo de un puente[12][14]

Mediante el conocimiento de la condición de las estructuras puente, se pueden llevar a cabo ciertas medidas preventivas para prolongar la vida útil de las mismas y prevenir fallas catastróficas. Las estrategias de detección de daños pueden, en última instancia, reducir el costo del ciclo de vida. En cuanto a estructuras de ingeniería civil se define como daño a toda acción que deteriora el rendimiento de la estructura; por ejemplo, los cambios en los materiales, las conexiones, las condiciones de contorno, etc[14][31].

Todo sistema diseñado para monitorizar salud estructural, debe contemplar la integración de tres subsistemas principales para poder cumplir con los requisitos de un sistema SHM y poder organizar las actividades de monitoreo. Los tres subsistemas son: el subsistema de detección y adquisición de datos, el subsistema de gestión, y el subsistema de acceso y recuperación de datos[1].

Al desarrollar soluciones para SHM se debe tomar en cuenta las limitaciones que tienen ciertas tecnologías; algunos desarrollos recientes incluyen tecnología de sensores y mejoras de las capacidades informáticas y de redes; han permitido la instalación de redes de sensores in-situ, que son responsables de la supervisión de estructuras. Sin embargo; muchas de estas técnicas son costosas y en muchos casos no son viables para SHM, ya que tienen varios requisitos para su instalación; lo que limita su aplicación a un pequeño número de puentes[42][6][12][14].

El daño estructural en puentes puede ser causado de varias maneras; situación que es inherente a todo tipo de estructura. Desde las actividades normales que introducen desgaste, hasta corrosión y el envejecimiento. El tráfico y las cargas de viento causan los principales daños en los puentes. La naturaleza y funcionalidad de cada tipo de estructura establece el tipo de variables que se deben monitorizar.

Existen datos, mediciones e implementaciones sobre varias tecnologías utilizadas en *SHM*, entre las cuales se encuentran: fotogrametría, radares en general, georradars, radares interferométricos, tecnologías de monitorización de corrosión, termografía infrarroja, efecto Doppler acústicos (ADCP), velocímetros gráficos de partículas a gran escala (LSPIV), medición de vibraciones mediante filtros y transformada rápida de Fourier, entre otras[42][6][12][14][16][15][26].

Diversos diseños, prototipos y experimentos con *WSN* para *SHM* se han propuesto, implementado y llevado a cabo tanto en el ámbito académico como en el comercial, algunos proyectos de los que se tiene evidencia son: Wisden, en Los Ángeles, California, EE.UU. [52][62]; sistema *WSN* que se implementó en el puente Golden

Gate en San Francisco, EE.UU; WISAN (Sistema Inteligente Inalámbrico Sensor y Actuador) para detectar daños y localizar estructuras [25]; BriMon en Malasia; entre otros.

Los puentes representan uno de los elementos más críticos de la infraestructura vial. Son de suma importancia para la calidad de vida general de las personas y para actividades económicas como el turismo, el transporte de mercancías y las empresas. En el caso particular de Costa Rica, la mayoría de las estructuras de puentes se construyeron hace más de 30 años y la inversión en el mantenimiento adecuado ha sido mínima. Actualmente, el programa e-Bridge del Instituto Tecnológico de Costa Rica (TEC) está realizando un inventario nacional de puentes en conjunto con el Consejo Nacional de Carreteras[16]. El proyecto e-Bridge ha adquirido con el paso del tiempo, un nivel de importancia alto en esferas incluso políticas de Costa Rica, debido a que sus propios datos han hecho de dominio público el estado real de la infraestructura de puentes en el país.

1.2.2. e-Bridge

El estado actual de la infraestructura vial en Costa Rica es una gran debilidad de la economía y los puentes presentan un porcentaje significativo en condición promedio o crítica. E-Bridge nace como un proyecto en respuesta a la problemática, que es parte del Centro de Investigación en Vivienda y Construcción (CIVCO) del Instituto Tecnológico de Costa Rica[16][15]. Esta tesis se desarrolla dentro del contexto del proyecto e-Brige.

Se destaca que al pasar del tiempo, el proyecto ha adquirido una gran relevancia en el ámbito gubernamental de Costa Rica, parte de las razones de su éxito es que los datos recopilados y mediciones realizadas han sobre-justificado la razón de su existencia. En el ámbito académico su éxito es indiscutible, donde se puede apreciar a nivel de referencias en este documento, que la cantidad de publicaciones al respecto y salidas del proyecto, es basta.

La predicción remota de fallas en puentes surge como el enfoque de la investigación en desarrollo, bajo ciertas condiciones de requerimiento como adaptabilidad, bajo costo de implementación, dispositivos , fácil instalación, entre otras. E-Bridge cuenta con el soporte del TEC, el CONICIT, el MOPT y el CONAVI. El principal objetivo de este proyecto es el de diseñar y desarrollar soluciones tecnológicas para la monitorización de salud estructural de puentes en Costa Rica.

El proyecto de investigación del CIVCO, e-Bridge está definido por las siguientes etapas: e-Bridge 1.0 (2011), como proyecto inicial; E-Bridge 2.0 (2013), etapa del proyecto de integración de información e E-Bridge 3.0 (2016) de diseño de prototipo para análisis de riesgos ambientales, características de diagnóstico y técnicas de inteligencia.

La reducción en el costo de implementación de un sistema *SHM* con el uso de tecnología *WSNs*, fomenta su utilización en la infraestructura pública y privada; situación que ha generado un incrementando en su uso para aplicaciones de monitorización estructural debido a que tienen un menor plazo de instalación[8].

La monitorización de la salud estructural debe tener consideraciones tales como sensores comunes, parámetros comúnmente medidos y detección de daños y algoritmos de localización. Los principales desafíos son la escalabilidad, la sincronización de tiempo, la optimización de ubicación de sensores y el procesamiento de

datos[8].

Es importante mencionar que los resultados preliminares de esta investigación han sido publicados y presentados en el congreso 2018 IEEE Central America and Panama Convention [41], cuyo artículo ha sido indexado en SCOPUS.

1.3. Definición del problema

En esta sección se introduce la definición del problema en investigación, su nivel de innovación e importancia. Se explica de manera resumida, algunos factores que resaltan la complejidad del mismo o bien justifican el carácter específico del problema.

La mala condición de los puentes en Costa Rica, aunado a la cantidad de desastres naturales relacionados con lluvias, inundaciones y otros eventos naturales; hacen que las estructuras de puentes sean objeto sensible para las autoridades en caso de emergencia. En nuestro país todavía existen comunidades que están comunicadas entre sí por un único puente, por lo que quedar incomunicados es un riesgo inminente.

El entorno que envuelve a la estructura vial de nuestro país genera la necesidad de monitorizar la salud de las estructuras, como lo son los puentes (1395 puentes en 2016[23]). Una gran cantidad de los puentes en Costa Rica ha alcanzado edades considerables y un nivel de mantenimiento mínimo o nulo durante su uso, por lo que su monitorización constante podría evitar o anticipar un colapso estructural, que puede ser provocado ya sea por una emergencia ambiental o por envejecimiento per se.

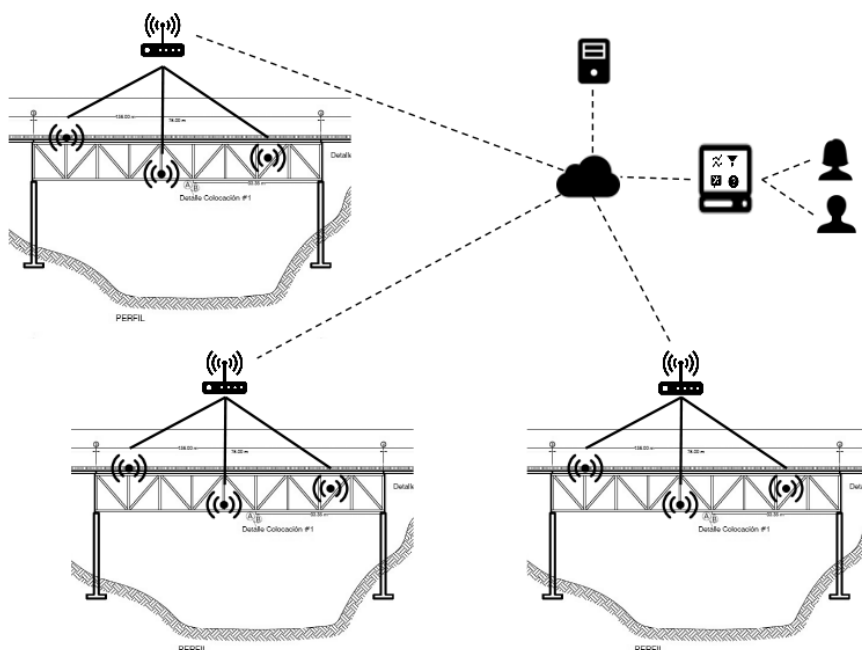


Figura 1.1: Topología general de la red de sensores colaborativa.

Surge también la necesidad de establecer un medio que permita dar soporte en escenarios de colaboración para alertas como las establecidas por la Comisión Nacional de Emergencias (CNE), que involucre a diferentes actores tales como sensores, equipo instalado en general y a organizaciones encargadas de velar por la población; para las cuales es de gran apoyo el contar con soporte que facilite su labor.

En la figura 1.1 se muestra la arquitectura general del sistema requerido para confrontar el problema planteado. Se observa como cada estructura puente es capaz de enlazarse con la red; permitiendo la transmisión de datos y colaboración en caso de emergencia. Cada estructura puente puede tener más de un nodo o sensor, donde la transmisión de datos a la red estará en función de la mejor disposición tenga para hacerlo. Es posible pensar que en caso de emergencia, un puente aguas arriba es capaz de transmitir su estado de alerta a la red, para que así se tomen medidas preventivas e incluso de evacuación o clausura de las estructuras aguas abajo; es importante recalcar que el escenario no es limitado, ya que podría establecerse alguna estrategia de sectorizado (áreas cercanas), donde no necesariamente debe ser para estructuras aguas abajo (mismo río) sino más bien dentro del mismo sector o área.

1.4. Justificación del proyecto

Con el fin de justificar la propuesta expuesta en este documento de investigación se presenta la misma en tres dimensiones. Primeramente, la innovación, luego el impacto y finalmente la profundidad de la misma, en la que se especifica la manera en que será tratado el tema o problema.

1.4.1. Innovación

Las tareas de análisis de SHM generalmente implican la integración de diferentes herramientas que pueden ser altamente heterogéneas, autónomas y estar físicamente distribuidas entre las diferentes organizaciones; como parte de estos componentes están los computacionales, que en general existen o han sido desarrollados previamente por diferentes organizaciones y dicho sea de paso exhiben un alto grado de heterogeneidad en cuanto a los modelos de datos y el software que utilizan [14][12].

Por definición una red colaborativa es donde los módulos en un nodo y los nodos de sensor en las redes se comunican y cooperan para una tarea específica [67]. En el presente trabajo de investigación se busca establecer un protocolo de red de sensores de bajo costo, donde se plasme una filosofía colaborativa; cuyo ámbito le permita funcionar tanto a nivel de una sola estructura, como entre estructuras diferentes; la filosofía colaborativa debe establecerse para fines de prevención de emergencias; donde además de un mejor manejo de los recursos, se tenga puntos de alarma para desastres naturales y emergencias bien definidos en un protocolo de red WSN para SHM. El enfoque basado en redes colaborativas busca promover la colaboración entre diferentes entes incluyendo sistemas, organizaciones y personas para proveer una solución al monitoreo de salud estructural de puentes.

La revisión de trabajos relacionados tanto a nivel académico como industrial permite visualizar el estado del arte desde varios puntos de vista. En la actualidad prevalece un enfoque del problema espléndido económicamente y simplificado a grandes estructuras; con el objetivo de satisfacer necesidades a nivel de salud estructural

con cierta inclinación hacia el ámbito de seguridad. En este sentido, la innovación de esta investigación radica en los siguientes elementos:

- Focalización en la búsqueda de opciones de bajo costo que cumplan los requerimientos necesarios a nivel de hardware en un sistema SHM. Situación que adquiere una gran importancia en el contexto nacional e investigativo del proyecto.
- Inclusión de una capa de interoperabilidad entre dispositivos por medio de un protocolo computacional inteligente. De esta manera, se puede generalizar el tipo de nodo incorporado a la red, la variable a medir y además la estructura en la que se conecte dicho nodo (Ver figura 1.1).
- Inclusión de un sistema que permitiría a la comisión Nacional de Emergencias obtener datos con alguna antelación, ante situaciones de emergencia debido ya sea a desastres naturales ó a eventos humanos premeditados ó inesperados.

1.4.2. Impacto

En el mercado existe una gran variedad de equipos y sensores comerciales para realizar monitorización de salud de estructuras que funcionan de forma adecuada en países industrializados (ver [69][12][45]). Los sistemas comerciales implican un alto costo de instalación y mantenimiento; por otro lado las inspecciones constantes durante un largo periodo de tiempo normalmente se ven restringidas, en muchos casos los datos obtenidos son tomados de forma manual de los equipos para luego realizar un análisis detallado.

El impacto que esta investigación puede generar, está directamente relacionado con las instituciones que podrían hacer uso de un sistema de este tipo. Entre las instituciones que podrían verse beneficiadas están la Comisión Nacional de Emergencias (CNE), el Gobierno central de Costa Rica, municipalidades, Ministerio de Obras Públicas y Transportes, CONAVI, empresas contratistas de obra civil, Laboratorios y Centros de Investigación como el CIVCO, e incluso ciudadanos en general que podrían obtener información de la red de monitoreo.

Actualmente no se cuenta con ningún sistema similar que cubra siquiera parcialmente con las funciones que podría cubrir un *SHM* adaptado a las necesidades nacionales; donde se considere tanto el tipo de estructuras como su estado actual, esto último en función de la imposibilidad de instalar sistemas comerciales que requieren de una configuración previa, que incluye el trazado de cableado durante la construcción de la estructura per se.

1.4.3. Profundidad

Desde el punto de vista de la profundidad de la investigación, se destaca que el enfoque principal se centrará en el desarrollo del prototipo de red inalámbrica para monitorizar salud estructural de puentes, donde la definición del protocolo computacional de comunicación entre nodos de la red, formará parte esencial del prototipo. Las variables implicadas en el diseño se comprenderán en la naturaleza del entorno.

Mediante una revisión de trabajos relacionados, se definirá el estado del arte; para luego llevar a cabo un análisis de requerimientos que describa el escenario y señale aspectos diferenciadores obtenidos con esta investigación. Posteriormente se planteará el diseño de prototipo de acuerdo al escenario y requerimientos identificados, para luego desarrollar el prototipo funcional y ejecutar su validación de funcionamiento.

En la sección de metodología se detallará la inclusión de pruebas repetitivas para determinar el comportamiento tanto del protocolo propuesto como del prototipo realizado.

1.4.4. Hipótesis

El prototipo de una red inalámbrica colaborativa para monitorización de salud estructural en puentes de Costa Rica basada en Servicios Web, apoya la prevención de desastres naturales relacionados con el nivel de caudal de agua bajo un puente.

1.5. Objetivos

Esencialmente un proyecto de investigación tiene como núcleo a los objetivos. En torno a los mismos se establecen los entregables, las actividades a realizar y el cronograma a seguir.

A continuación se describen tanto el objetivo general y como los objetivos específicos de la propuesta de tesis.

1.5.1. Objetivo General

Desarrollar un prototipo de una red inalámbrica colaborativa para realizar monitorización de salud estructural de puentes en Costa Rica.

1.5.2. Objetivos Específicos

- Realizar una revisión del estado del arte de trabajos relacionados con el tema propuesto.
- Llevar a cabo un análisis de requerimientos que permita describir un escenario particular de monitorización de puentes e identificar las funcionalidades requeridas para el diseño del prototipo.
- Diseñar una red inalámbrica colaborativa que permita realizar monitorización de puentes según el escenario y los requerimientos identificados.
- Desarrollar un prototipo funcional del diseño de la red inalámbrica colaborativa.
- Validar el funcionamiento del prototipo desarrollado.

1.6. Organización del documento

La información contenida en este documento está estructurada en los siguientes capítulos:

- Capítulo 1 Introducción: que contiene una descripción general del documento, los antecedentes; la definición del problema, justificación del proyecto en función de innovación, impacto y profundidad; objetivos general y específicos; el alcance y por último los entregables.
- Capítulo 2 Trabajos Relacionados: también conocido como marco conceptual, en éste capítulo se describen los conceptos, definiciones, métodos y procedimientos que permiten abordar y comprender de una mejor manera el ámbito del proyecto, además de determinar el estado del arte.
- Capítulo 3 Análisis de Requerimientos: en éste se describe el proceso y los resultados de la investigación, es donde se compila la información necesaria para comprender el contexto de la red diseñada. Se aborda inicialmente un diagrama de contexto general, posteriormente se profundiza en los requerimientos funcionales y la estructura de datos, finalizando con los requerimientos técnicos de componentes.
- Capítulo 4 Diseño de la Red: en éste se describe el proceso de diseño de la arquitectura de la red colaborativa, donde se incorpora las estrategias propuestas en el capítulo anterior.
- Capítulo 5 Implementación de Prototipo: en éste se describe el proceso de implementación del diseño de la red, la configuración necesaria y el desarrollo que se plantea para lograr la implementación del diseño del prototipo planteado en el capítulo anterior.
- Capítulo 6 Validación y Pruebas: donde se describe la validación de la propuesta de la arquitectura de información así como la implementación del diseño, con el fin de valorar el nivel de aceptación y el cumplimiento de los objetivos que se plantean. En el análisis de resultados se describe el producto generado; se estudian, identifican y contrastan los resultados obtenidos.
- Capítulo 7: Conclusiones: presenta un resumen con los descubrimientos más relevantes del capítulo anterior y se discute si se lograron los objetivos propuestos, así como si se cumplieron los entregables definidos y además se incluyen las secciones de conclusiones generales, limitaciones del proyecto y trabajo futuro.

Capítulo 2

Trabajos Relacionados

La infraestructura vial forma una parte importante de la economía y es de suma importancia para la calidad de vida en general de las personas y para las diversas actividades económicas del país[18]. Los puentes representan uno de los elementos más críticos, ya que por la naturaleza del país son en muchas ocasiones el único punto de unión entre dos comunidades e incluso provincias.

En Costa Rica, la mayoría de las estructuras de puentes se construyeron hace más de 30 años y la inversión en el mantenimiento adecuado ha sido mínima[16]. E-Bridge del Instituto Tecnológico de Costa Rica (TEC) está realizando un inventario nacional de puentes junto con el Consejo Nacional de Carreteras. De entre lo que se ha podido ingresar al inventario hasta el momento, se ha detectado que un porcentaje significativo de puentes muestra una condición regular o crítica.

En un país en desarrollo como Costa Rica, la priorización de los proyectos de infraestructura vial es esencial para las instituciones de gobierno correspondientes. Sin embargo, para poder priorizar las inversiones de recursos, antes es necesario conocer de forma precisa la condición de vías y puentes[18].

En e-Bridge 3.0 se plantea generar un prototipo que permita tomar datos en los puentes, considerando que son estructuras que no contemplan la instalación de sistemas tradicionales de monitoreo.

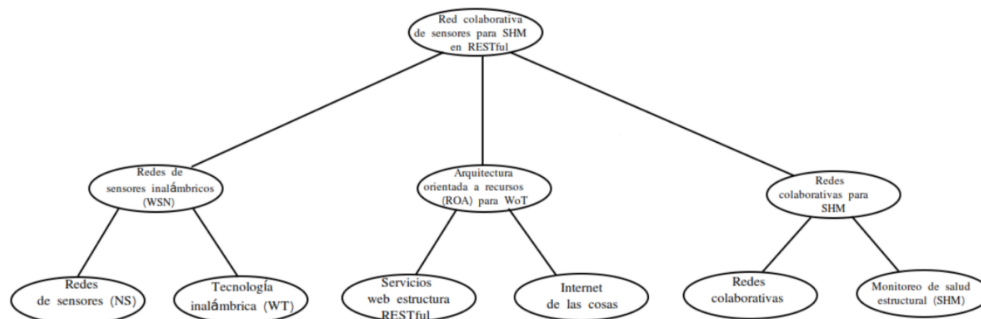


Figura 2.1: Diagrama de la taxonomía de la investigación.

Se plantea parametrizar un prototipo colaborativo de monitoreo de salud estructural con tecnología inalámbrica y uso de servicios web (ver figura 2.1). El prototipo propone solucionar el hecho de que las estructuras puente de Costa Rica no están diseñadas para albergar un cableado para los sensores, además de hacer uso de tecnologías de código abierto y en la medida de lo posible hardware de bajo costo. En este capítulo se presenta el marco del contexto de la investigación propuesta. Se considera en la implementación del prototipo el uso de redes de sensores inalámbricos en estructuras *RESTful* que permitan la escalabilidad del sistema, y algo de procesamiento a nivel de nodos sensor tal y como se especifica en la literatura, mencionado tanto para estructuras en *REST* (*Representational State Transfer*, *REST* por sus siglas en inglés), como en Internet de las Cosas, dicha característica incrementa la capacidad de procesamiento global del sistema[68].

2.1. Antecedentes y trabajo relacionado

E-Bridge 3.0 es ya la tercera etapa del mapa de ruta establecido por un grupo de investigación del TEC sobre monitoreo de la salud estructural en puentes. Este grupo de investigación nace en 2011 con la colaboración de varias escuelas, entre ellas: Ingeniería de la Construcción, Informática, Electrónica, Ingeniería de Producción Industrial e Ingeniería Forestal. Forma parte también de un programa de e-Ciencia y mantiene fuertes colaboraciones con organizaciones relacionadas de usuarios finales, tales como el Ministerio de Transporte de Costa Rica, el Consejo Nacional de Carreteras y las unidades del gobierno local. El objetivo general del grupo de investigación es generar herramientas para analizar la condición y el comportamiento de las estructuras de puentes y así apoyar mejor las actividades de planificación y mantenimiento[17].

2.1.1. Trabajos relacionados en monitoreo

Proyectos e iniciativas existentes relacionados con sistemas de monitoreo de puentes, se pueden encontrar varios informes que analizan varias herramientas y plataformas disponibles[17][41]. Los sensores de nivel de ríos instalados en puentes (*BMRSS*, por sus siglas en inglés para “*Bridge-Mounted River Stage Sensors*”), son uno de los principales referentes para el proyecto en cuestión, ya que realizan acciones como la medición y el análisis de variables como nivel del agua, humedad y temperatura, envío de datos a un servidor y respaldo de información en caso de errores en la transmisión[41].

Algunos de los sistemas propietarios más completos incluyen *BRIMOS* y *SHM Live*[17]. *BRIMOS* es un sistema de monitoreo de puentes, que ofrece un método para la detección de daños en estructuras civiles, basado principalmente en el monitoreo de vibraciones; además admite diferentes tipos de aplicaciones de monitoreo, incluyendo monitoreo de punto caliente, permanente y de cable. *SHM Live* por su parte, es un sitio web que administra y muestra datos sobre estructuras monitoreadas en tiempo real en cualquier parte del mundo; permite la configuración en tiempo real de las alarmas de eventos y ofrece diferentes opciones de visualización de datos.

En [17] se prevé que miles de millones de dispositivos y redes inteligentes, como las redes de sensores inalámbricos ó *WSN* (*Wireless Sensor Network*, *WSN* por sus siglas en inglés), no se aislarán, sino que se conectarán e integrarán con redes de

computadoras en el futuro *Internet of Things (IoT)*. Para mantener bien esos dispositivos sensores, a menudo es necesario evolucionar los dispositivos para que funcionen correctamente al permitir que las entidades de administración de dispositivos (DM) monitoreen y controlen dispositivos de forma remota sin consumir recursos significativos. Para manejar la cantidad de sensores proponen un enfoque ligero del servicio web *RESTful(WS)* para permitir la gestión de dispositivos de sensores inalámbricos.

En relación con e-Bridge en [46], se describe el proceso de diseño de prototipos y el desarrollo de un sistema integrado inalámbrico, que utiliza sensores específicos tipo acelerómetro para generar datos relevantes en el monitoreo de la salud de puentes (vibraciones en este caso). Su contenido se basa en el análisis de requisitos, el diseño y la implementación de un prototipo de sistema que utilice principalmente tecnologías gratuitas o de bajo costo. Además se habla de incluir una interfaz web que permita el análisis de los datos, entre otras características.

Para finalizar en [47], se describe el diseño y desarrollo de una red inalámbrica de sensores por medio de un dispositivo embebido que se encarga de la transferencia de datos para realizar tareas de monitoreo y toma de datos según los requerimientos identificados mediante reuniones que se realizaron con ingenieros e investigadores con el fin de obtener un conjunto de requerimientos y especificaciones. Los requerimientos extraídos se clasificaron en aquellos asociados al dispositivo embebido y en aquellos asociados a servicios web.

2.1.2. Trabajos relacionados en redes colaborativas y arquitecturas RESTful

El sistema integrado para la determinación del desempeño de estructuras de Puentes desarrollado en [4], es un prototipo de una red inalámbrica de sensores que apoya tareas relacionadas con el monitoreo de salud de estructuras de puentes en Costa Rica. La red fue diseñada utilizando sistemas tipo PC-Duino con una versión de Linux. El diseño realizado presenta ventajas específicas con respecto a sistemas propietarios existentes pues permite construir redes inalámbricas abiertas, escalables, de bajo costo y consumo de energía, que se adaptan bien a los requerimientos de escenarios de monitoreo remoto de estructuras civiles. Los Gateways en este caso funcionan como un punto de agregación de los sensores a la red.

En [9] se propone un método de diseño e implementación de *gateways* para WSN con estructura *RESTful* utilizando un *framework* basado en Javascript, donde Node.js utiliza un modelo de programación asíncrono, que puede manejar una gran cantidad de conexiones de red simultáneamente. El *gateway* está diseñado como un dispositivo empotrado con Linux y que maneja múltiples accesos tanto de los sensores como de los servidores en la nube. Todos los *APIs* de comunicación que residen en el *gateway*, son diseñados para comunicarse con los servidores en la nube por medio de *APIs RESTful*, *HTTP* y con los sensores por medio de un *CoAP(Constrained Application Protocol, CoAP* por sus siglas en inglés).

Otro trabajo desarrollado en [3], son los servicios de ubicación *RESTful* de *Open Mobile Alliance (OMA)* con servicios web *RESTful* estándar para ubicación de terminales. Son independientes de la tecnología de ubicación y permiten la portabilidad y la interoperabilidad de las aplicaciones. Los sensores inalámbricos son dispositivos electrónicos que pueden detectar el contexto: espacio, medio ambiente y fisiología. Los sensores inalámbricos pueden detectar la ubicación con alta precisión y que

otras tecnologías no pueden proporcionar.

En la sección de redes colaborativas se desarrolla de forma más amplia el tema de redes colaborativas.

2.2. Monitoreo de salud estructural SHM

El monitoreo de la salud estructural es un campo emergente en ingeniería civil que tiene el potencial para la evaluación continua y periódica de la seguridad e integridad de la infraestructura civil. El objetivo de *SHM* (*Structural Health Monitoring*, *SHM* por sus siglas en inglés) es conocer la condición de la estructura, que permita llevar a cabo medidas preventivas para prolongar la vida útil de una estructura, además de prevenir fallas catastróficas. En síntesis, el monitoreo de salud estructural (*SHM*) se refiere al almacenamiento y análisis de grandes cantidades de datos de sensores asociados a variables específicas; el registro permanente, continuo o periódico de parámetros indica la condición y el rendimiento de una estructura civil, como por ejemplo, un puente en una vía de transporte público[12][14][19]. Los parámetros están asociados a cualquier variable de la que se obtienen valores a través de un sensor, como lo es la medición del nivel del agua del río que pasa por debajo de un puente.

Las estrategias de detección de daños pueden, en última instancia, reducir el costo total del ciclo de vida de las estructuras. En términos generales, el daño puede definirse como el cambio introducido en un sistema que afecta negativamente su desempeño (Farrar et al., 1999). En cuanto a estructuras de ingeniería civil se define como daño a toda acción que deteriora el rendimiento de la estructura; por ejemplo, cambios en los materiales, conexiones, condiciones de contorno, etc[14][31].

El daño estructural puede ser causado de varias maneras. Las actividades normales pueden introducir daño a la estructura. Las estructuras pueden dañarse debido a la corrosión, el envejecimiento y las actividades diarias; como por ejemplo, el tráfico y las cargas de viento, que causan daños en los puentes. Por otro lado, las cargas excesivas producidas por tornados, huracanes y terremotos también pueden causar daños en estructuras[31]. Propiamente, las crecidas espontáneas constituyen otro fenómeno natural que afecta a las estructuras de puentes en Costa Rica, incrementando los niveles de caudal, aguas abajo en poco tiempo y causando gran destrucción en las estructuras y zonas aledañas; muchas veces con repercusión en vidas humanas, flora y fauna.

Todo sistema diseñado para evaluar salud de estructuras(*SHM*), debe contemplar la integración de tres subsistemas principales; etapa de detección y adquisición de datos, etapa de gestión, y etapa de acceso y recuperación de datos. El componente de adquisición de datos, incluye sensores para obtener datos bajo diversas condiciones ambientales, así como un sistema de comunicación de datos para la transmisión de los datos medidos a un servidor en tiempo real. El diseño de este componente tiene implicaciones sobre los módulos de detección en la estructura, tales como: su elección, el número de módulos requeridos y su ubicación[1].

El subsistema de gestión de datos consiste en técnicas de recopilación de datos, almacenamiento de datos y su procesamiento; haciendo posible proporcionar una evaluación del estado de la estructura. Las tareas de procesamiento de datos son fundamentales para extraer la característica que permite la identificación del daño, o su ausencia, en una estructura[1].

Por último, el acceso y recuperación de datos se basa en el control y acceso a datos, en línea, o fuera de línea. En esta etapa del sistema, el propósito es que el análisis e interpretación de los datos permita proporcionar información útil para la toma de decisiones. Una implementación satisfactoria de esta etapa, permitirá al usuario ver los resultados del análisis de los datos de forma remota a través de Internet.[1].

A la hora de desarrollar soluciones para SHM, es importante tomar en cuenta las limitantes del uso de ciertas tecnologías; algunos desarrollos recientes que incluyen diversas tecnologías de sensores y mejoras de las capacidades informáticas y de redes, han permitido la instalación de redes de sensores *in-situ*; dichas redes son responsables de la supervisión de parámetros tales como, tensión y deformación de las estructurales, agrietamiento de la cubierta de concreto, entre otros. Sin embargo, las técnicas de toma de datos, son costosas y en muchos casos no son viables para SHM, ya que requieren de una instalación en lugares difíciles de alcanzar o deben ser colocadas durante la construcción de la estructura; lo que limita su aplicabilidad a tan solo el número de puentes que se construye hoy en día, en comparación con la población actual de puentes en servicio[42][6][12][14].

Tomar en cuenta las lecciones aprendidas y la información disponible en materia de SHM, permite evitar la reproducción de errores fatales, que afectan la calidad de los datos y por ende complican el análisis a la hora de enfrentar un problema en este área[42].

2.2.1. Variables de medición en SHM para puentes

El daño estructural en puentes puede ser causado de varias maneras -inherente a todo tipo de estructura-, desde las actividades normales que introducen desgaste, hasta la corrosión y envejecimiento. El tráfico y las cargas de viento, son las principales causas de daño en los puentes. La naturaleza y funcionalidad de cada tipo de estructura establece el tipo de variables que se debe monitorear.

Investigaciones previas han generado tecnologías y datos de medición, relacionados a diversas variables que se consideran importantes de medir en SHM; algunas infieren sobre estructuras en general y algunas sobre puentes, específicamente. Entre las variables consideradas importantes, se encuentran: grietas en el concreto, carga del viento, vibraciones, nivel del agua, precipitaciones, velocidad del viento, tensión estructural, deformación estructural, flujo vehicular, entre otras[42][6][12][14][16][15].

2.2.2. Tecnologías utilizadas en sistemas de SHM para puentes

El estado del arte refleja que existen, datos, mediciones e implementaciones sobre varias tecnologías utilizadas en SHM. Entre las cuales se encuentran: fotogrametría, radares en general, geo-radares, radares interferométricos, tecnologías de monitoreo de corrosión, termografía infrarroja, efecto Doppler acústicos (*Acoustic Doppler current profiler, ADCP*¹ por sus siglas en inglés), velocímetros gráficos de partículas a gran escala (*Large-Scale Particle Image Velocimetry, LSPIV*² por sus siglas en inglés),

¹Sistema y método para medir velocidades de corrientes utilizando señales acústicas de banda ancha de pulso codificado.[33]

²PIV refiere al diagnósticos de flujo óptico que se basan en refracción, absorción o dispersión, de la luz (visible) en medios no homogéneos, por ejemplo Imágenes de la superficie del río muestreadas con una cámara[37][39].

medición de vibraciones mediante filtros y transformada rápida de Fourier, entre otras[42][6][12][14][16][15][26].

Los ingenieros encargados del análisis de salud estructural utilizan diferentes tipos de sensores para monitorear estructuras, entre ellos: sensores de desplazamiento, medidores de tensión, acelerómetros, entre otros. A manera de ejemplo los acelerómetros miden las aceleraciones de la superficie sobre la que están montados. Las aceleraciones se traducen en cambios en la capacitancia o en otras propiedades eléctricas y dichas señales analógicas se muestrean a una frecuencia específica.

Para el desarrollo de métodos de detección y localización de daños, los expertos se basan en amplios conjuntos de datos de respuesta estructural, dichos conjuntos de datos pueden ayudar a validar, comparar y proporcionar intuición para tales métodos. Hoy en día los datos se SistInfoIntegrecopilan mediante costosos sistemas electrónicos cableados (o sistemas inalámbricos one-hop -una sola ruta-) y sistemas de adquisición muy poderosos[42][52].

2.3. Redes colaborativas(CN), conceptos y aplicaciones

Una red colaborativa o *CN*(*Collaborative Network*, *CN* por sus siglas en inglés) es una alianza constituida por varias entidades (e.g. organizaciones, sistemas y personas) de carácter autónomo, geográficamente distribuidas y heterogéneas en términos de su ambiente operativo y metas, pero que interactúan entre sí de forma cooperativa para poder alcanzar metas comunes, y cuyas interacciones son apoyadas por redes de computadores [38][12][10]. Organizaciones virtuales, clústeres de industrias, comunidades virtuales profesionales, laboratorios virtuales, equipos virtuales, cadenas de abastecimiento, gobierno virtual, entre otros, son ejemplos de clases de redes colaborativas. En este contexto, las entidades colaboran estrechamente, por ejemplo para ofrecer productos y servicios por parte de la red colaborativa, que son llevados a cabo por medio de procesos desarrollados en varias organizaciones o por diversas personas. El paradigma de redes colaborativas ha sido aplicado exitosamente a dominios muy variados[12].

La monitorización de salud de estructuras de puentes puede verse como un caso específico de red colaborativa. Para realizar monitorización en una zona particular, se debe contar con una red de entidades autónomas, distribuidas y heterogéneas que consisten de organizaciones (municipalidades, ministerios, laboratorios, universidades), personas (ingenieros, investigadores, ciudadanos), sensores de variables (vibración, precipitación, altura libre inferior,etc), sistemas empotrados y sistemas de información que deben colaborar estrechamente para lograr una meta común[14]. Por ejemplo, la meta puede ser apoyar la prevención de desastres mediante la generación de alertas de incrementos alarmantes en niveles de agua bajo puentes de una zona geográfica determinada. En este caso, se deben definir modelos y protocolos de comunicación entre todas las entidades que permitan manejar adecuadamente las alertas detectadas. Según este concepto, en el país podrían definirse varias redes colaborativas según áreas de interés estratégico.

2.3.1. Aplicaciones de redes colaborativas

En [11] se describe la estructura detallada de un curso de Organizaciones Virtuales o *VO*(*Virtual Organization*, *VO* por sus siglas en inglés), que se imparte en pro-

gramas de licenciatura y maestría de Informática en el TEC. Además del análisis y la integración adecuada de componentes tecnológicos, definen y desarrollan arquitecturas de referencia generales y herramientas genéricas para admitir diferentes tipos de escenarios de VO colaborativos. Se definen escenarios de colaboración entre organizaciones siguiendo el paradigma VO, para el entendimiento de los estudiantes.

En el sector turismo una VE (*Virtual-enterprise*, VE por sus siglas en inglés) puede definirse como un consorcio temporal de organizaciones de proveedores de servicios (agencias de viajes, proveedores de alojamiento, etc)[32], que se unen para fortalecer sus habilidades y recursos para ofrecer servicios integrados. En [13] se describe el diseño e implementación de un componente del Catálogo de Definiciones de Interfaz de Servicio destinado a apoyar la interoperabilidad de los servicios en las VE de turismo. El soporte para servicios de valor agregado o VAS (*Value-Added Services*, VAS por sus siglas en inglés) es un ejemplo de estos escenarios avanzados de colaboración.

2.3.2. Importancia de la colaboración

Un sistema con sentido colaborativo puede ser tratado como una gran empresa y debe de ser lo suficientemente flexible como para usar múltiples estrategias concurrentes. Estas estrategias deben ser eficientes dada la variedad de usuarios y elementos que lo componen. Una colaboración estrecha es, por lo tanto, un imperativo para mejorar la organización y el rendimiento en general de un sistema[21]. El desafío en un sentido colaborativo está en: el grado de colaboración al que se desea llegar, la cantidad de elementos involucrados, su duración y la naturaleza de los procesos; los anteriores no son fáciles de definir. Sin embargo, al crear ambientes de este tipo, la confiabilidad de los elementos es un requisito previo.

2.4. Internet de las cosas (IoT)

Internet de las cosas (*IoT*) se puede definir como un sistema global basado en el conjunto de protocolos de Internet, en el que los objetos equipados con sensores, etiquetas *RFID* (*Radio Frequency Identification*, *RFID* por sus siglas en inglés) que consisten en pequeños transpondedores unidos a objetos físicos[5] o códigos de barras que tienen una identidad única; operan en un entorno inteligente y se integran perfectamente en la red de información mediante el uso de interfaces inteligentes; generalmente se refiere a escenarios donde la conectividad de red y la capacidad de computación se extiende a elementos cotidianos, que normalmente no se consideran computadoras, lo que permite que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana. Sin embargo; no existe una definición única y universal[1][49].

La idea de combinar computadoras, sensores y redes para monitorear y controlar dispositivos ha existido por décadas. La reciente confluencia de varias tendencias del mercado tecnológico, sin embargo, está acercando el Internet de las cosas a su realidad generalizada, incluyendo: conectividad ubicua, adopción generalizada de redes basadas en IP, economía informática, miniaturización, avances en análisis de datos y aumento de la computación en la nube.

El *IoT* está forzando a un cambio en el pensamiento, donde la interacción con Internet proviente del compromiso pasivo con objetos conectados en un entorno,

es cada vez más común, ayudando a tener datos derivados e intercambiados a partir de esa interacción. La realización potencial de este resultado es un mundo hiperconectado, que no impone limitaciones inherentes a las aplicaciones o servicios que pueden hacer uso de la tecnología.

Como resultado del acelerado desarrollo de las tecnologías de detección, tales como la auto-identificación por radiofrecuencia (*RFID*), los sensores y la convergencia de las tecnologías de la información como la comunicación inalámbrica e Internet; *IoT* emerge como una tecnología importante para los sistemas de monitoreo. La cantidad de datos generados por los dispositivos de detección es voluminosa y más rápida que antes de *IoT*; se utilizan soluciones de *Big Data* para tratar la compleja y gran cantidad de datos recopilados de los sensores instalados en las estructuras.

La computación en la nube es un paradigma informático en el que los recursos, como los servidores, el almacenamiento y las aplicaciones, se pueden aprovisionar y acceder en tiempo real a través de redes de comunicación avanzadas. En la era de *IoT* y *Big Data*, la computación en la nube se ha desarrollado ampliamente en muchas aplicaciones industriales que implican un gran volumen de datos. El despliegue a largo plazo de un sistema de monitoreo de salud estructural (*SHM*), incurriría en una cantidad significativa de datos de diferentes tipos[1][49][59].

El desarrollo de sistemas *IoT* depende en gran medida de baterías estándar y de protocolos disponibles para interconectar dispositivos de baja y pequeña potencia, así como de un intermediario de datos o de una aplicación. Muchas tecnologías están involucradas en el paradigma de *IoT* e incluyen protocolos tales como: *NFC*(*Near-Field Communication*, *NFC* por sus siglas en inglés) para áreas muy pequeñas de hasta 4 centímetros[7], *RFID* y *WSN*; que se usan ampliamente para varios objetos de *IoT*. Hoy en día y debido a la necesidad y su constante uso, se están estableciendo diferentes estándares para *IoT* para hacer frente a los problemas de interoperabilidad, sin embargo, los estándares existentes que se usan ampliamente para lograr el intercambio de datos entre dispositivos *IoT*, son IEEE 802.15.4 e IEEE 802.11[1]. Más recientemente uno de los protocolos en *WSN* más populares que está siendo utilizado se conoce como *ZigBee*.

En vista de las facilidades disponibles para el acceso al internet en la actualidad y a la capacidad que tienen las arquitecturas *IoT*, en esta investigación se pretende hacer uso de dicha tecnología para convertir nuestros nodos de sensores en objetos con capacidad de envío de información, con lo cual se pretende realizar una recolección de datos para dar lugar a futuras investigaciones en áreas tales como servicios como el *Big Data*, *Machine Learning* y otras tecnologías relacionadas con el análisis de datos.

2.4.1. Modelos de comunicación del IoT

Desde una perspectiva operativa, se sabe cómo se conectan y comunican los dispositivos de *IoT* en términos de sus modelos de comunicación técnica. En el marco de comunicaciones comunes, utilizados por los dispositivos de *IoT*, se definen cuatro modelos de comunicación, especificados en el Documento Guía de Arquitectura para Redes de Objetos Inteligentes (RFC 7452) 39, lanzado en marzo de 2015, por *Internet Architecture Board (IAB)*[34][49].

2.4.1.1. Patrón de comunicaciones Device-To-Device

Al tener dos o más dispositivos conectados directamente, comunicándose entre sí, sin necesidad de hacerlo a través de un servidor de aplicaciones intermediario, se define un enlace tipo dispositivo a dispositivo. Estos dispositivos se comunican a través de muchos tipos de redes, incluidas las redes IP o Internet.

En imagen de la figura 2.2 se observa un ejemplo de comunicación, entre un monitor de ritmo cardíaco y un *cadence sensor*, donde los dispositivos son manufacturados por fabricantes distintos y establecen una comunicación directa por medio de una red inalámbrica, que representa un patrón de comunicación dispositivo-dispositivo[34][49].

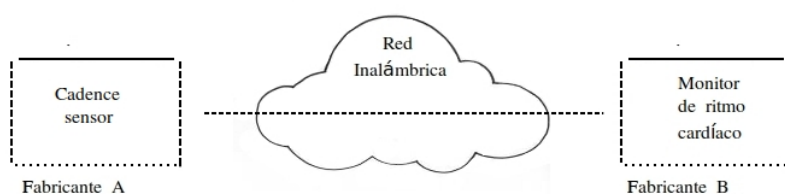


Figura 2.2: Patrón de comunicación Dispositivo-Dispositivo.

Para poder cumplir con el propósito de que los dispositivos de diferentes fabricantes puedan comunicarse de manera inmediata, los proveedores deben establecer un acuerdo sobre un *stack* de protocolos a soportar. Por ejemplo aspectos de diseño de protocolos, tales como: la capa física (Bluetooth Smart, IEEE 802.15.4, etc), capa de transporte (UDP, etc), arquitecturas de comunicación, modelo de los datos, modelo de información, entre otros[34][49].

2.4.1.2. Patrón de comunicación Device-To-Cloud

El dispositivo *IoT* se conecta directamente a un servicio de nube de Internet como un proveedor de servicios de aplicaciones para intercambiar datos y controlar el tráfico de mensajes. Con este enfoque a menudo se aprovechan mecanismos de comunicación existentes, como las conexiones tradicionales por cable o Wi-Fi, para establecer una conexión entre el dispositivo y la red IP, que en última instancia se conecta al servicio en la nube.

En la imagen de la figura 2.3 se ilustra un patrón de comunicación para cargar datos de sensores a un servidor de aplicaciones. A menudo, el proveedor de servicios de aplicaciones (ejemplo.com) también vende objetos inteligentes. En cuyo caso, toda la comunicación ocurre internamente al proveedor y no surge la necesidad de interoperabilidad. Sin embargo, es útil para ejemplo.com reutilizar las especificaciones existentes para reducir el esfuerzo de diseño, implementación, prueba y desarrollo [34][49].

La interfaz de protocolo utilizada para comunicarse con la infraestructura del servidor debe estar disponible, y hay varios estándares que lo permiten, tales como CoAP[9], Seguridad de la capa de transporte de datagramas (DTLS), UDP, IP, etc., como se muestra en la figura 2.3.

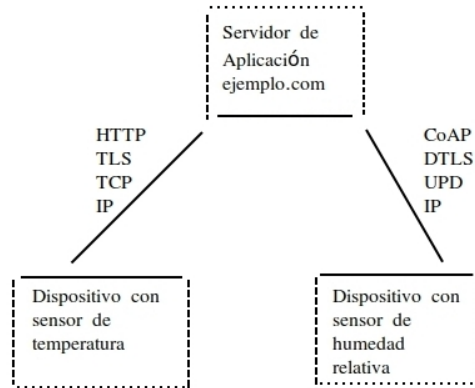


Figura 2.3: Patrón de comunicación Dispositivo-Nube.

2.4.1.3. Modelo Device-to-Gateway

En el típicamente conocido como modelo *device-to-application-layer gateway (ALG)*, el dispositivo *IoT* se conecta a través de un servicio *ALG* como un conducto para llegar a un servicio en la nube. Es decir que existe un software de aplicación que opera en un dispositivo *gateway* local, que actúa como un intermediario entre el dispositivo y el servicio en la nube, proporciona seguridad y otras funciones, como la traducción de datos o protocolos (Ver imagen de la figura 2.4).

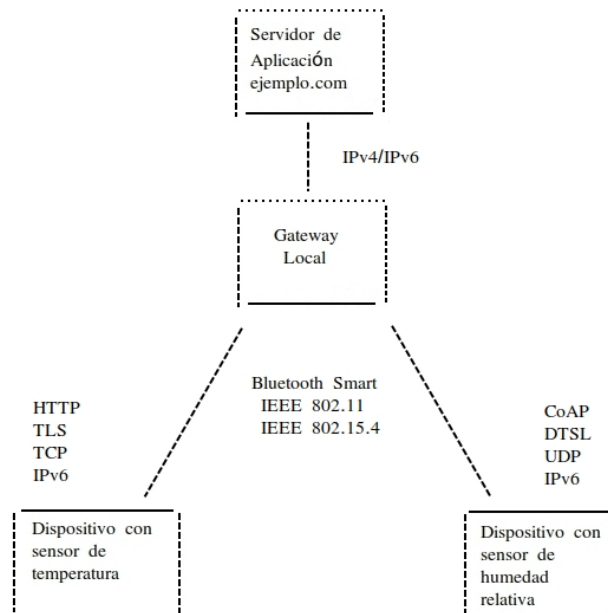


Figura 2.4: Patrón de comunicación Dispositivo-Gateway.

El modelo funciona correctamente si se utiliza tecnología de radio que se implementa ampliamente en el mercado, como el *Wi-Fi* basado en IEEE 802.11, en ocasiones, se requiere de tecnologías de radio menos disponibles (como IEEE 802.15.4) o se debe proporcionar una funcionalidad de capa de aplicación especial (autenticación y autorización local) o se necesita interoperabilidad con dispositivos heredados que no están basados en IP[34][49].

2.4.1.4. Modelo Back-End Data-Sharing

Se refiere a una arquitectura de comunicación que permite a los usuarios exportar y analizar datos de objetos inteligentes de un servicio en la nube, en combinación con datos de otras fuentes. Los dispositivos *IoT* cargan datos a un único proveedor de servidor de aplicaciones. Sin embargo, los usuarios a menudo exigen la capacidad de exportar y analizar datos en combinación con datos de otras fuentes. Por lo tanto, surge la necesidad de otorgar acceso a los datos de sensores desde terceros (Ver imagen de la figura 2.5)[34][49].

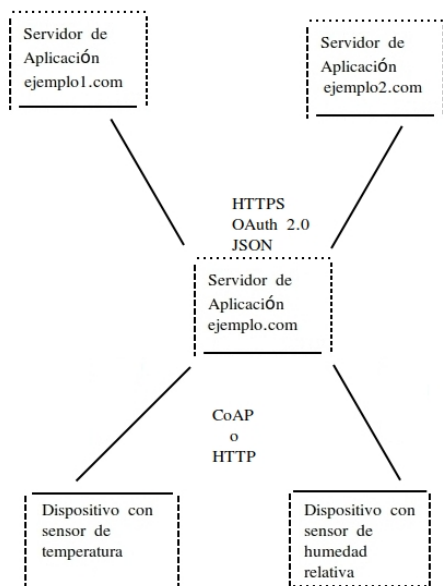


Figura 2.5: Patrón de comunicación Back-End Data-Sharing.

2.4.2. Arquitectura orientada a los recursos (ROA) para la Web de las cosas (WoT)

En el entorno de *WoT* (*Web of Things*, *WoT* por sus siglas en inglés) se espera que una “cosa” se comunique con otra “cosa” a través de la web, eventualmente por sí misma. Lo que indica que cada elemento puede ser indistintamente sujeto u objeto para realizar sus propias tareas. De acuerdo con la arquitectura web de estilo *REST* (estilo arquitectónico para sistemas distribuidos altamente escalables, mencionado por primera vez en la disertación de Fielding[63][48]), cada cosa puede representarse a sí misma como recursos web que se identifican con los URI[63][51].

WoT integra “cosas” inteligentes con tecnologías *well-known* de la web. Los elementos habilitados para la web pueden interoperar libremente a través de la Web utilizando los estándares abiertos.

La arquitectura *ROA* (*Resource-oriented Architecture*, *ROA* por sus siglas en inglés) se aplica en *WoT* mediante el uso de tres modelos de representación de recursos (ver figura 2.6) y el uso de los cuatro métodos básicos *HTTP/REST* para las operaciones *CRUD* (*Create, Read, Update, and Delete*) a los recursos REST mostrados en el cuadro 2.1 [63][51].

Cuadro 2.1: Operaciones HTTP/REST(CRUD)

Operación	Acción
POST	Crea
GET	Lee
PUT	Actualiza
DELETE	Elimina

En la figura 2.6 se muestran los modelos de representación de la arquitectura, donde a) indica que un recurso tiene un *URI* único e información de atributos, en b) y en C) se indica que un grupo de recursos también puede ser representado como un tipo de recurso[63][51].

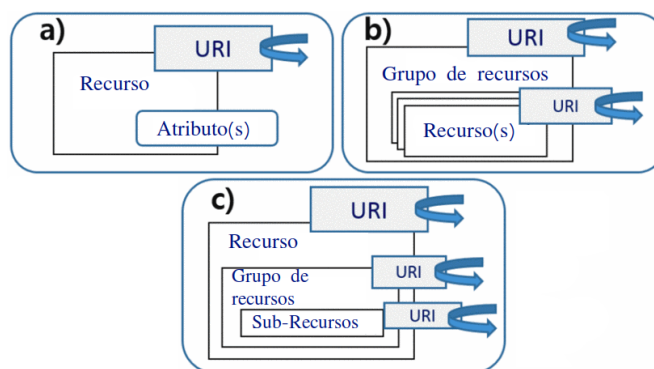


Figura 2.6: Modelos de representación ROA para WoT.

2.4.3. Definición de requerimientos y criterios para la implementación de servicios web en IoT

La falta de un protocolo común dificulta la logística de la comunicación dispositivo-dispositivo y aumenta la complejidad de las redes. El paradigma web de las cosas (*WoT*), donde los dispositivos utilizan *HTTP* como protocolo de comunicación, presenta una solución al problema de la interoperabilidad de las cosas. Es decir donde todas las “cosas inteligentes” tienen una representación y una presencia en internet,

requiriendo una intervención humana mínima para generar, intercambiar y consumir datos; su conectividad permite la toma remota de datos.

La evolución de las redes *IoT* conduce a un mundo conectado, automóviles, casas, máquinas y diferentes dispositivos interrelacionándose con algún nivel de autonomía, desatando una ola de nuevas posibilidades para la recolección de datos, análisis predictivo, y automatización del entorno. En el paradigma cada dispositivo individual tiene acceso a Internet. El mayor desafío o requerimiento se da al interconectar una variedad de dispositivos individuales diversos en una red flexible y efectiva.

La conectividad entre los dispositivos (o nodos) y la aplicación, y entre los dispositivos y el usuario final, se utiliza para que la capa de toma de decisiones funcione. De acuerdo a los principios del paradigma *WoT*, se utiliza un protocolo *RESTful* y un servidor *HTTP* para configurar a los dispositivos y colocar la información proporcionada por ellos en una estructura tipo *path*, donde los recursos están disponibles tanto para la aplicación, como para el usuario. La arquitectura *REST* para sistemas distribuidos se basa en una comunicación sin estado que emplea datos *self-explanatory* y una estructura cliente-servidor. Sus principales ventajas son la escalabilidad y la compatibilidad.[24][49].

2.4.4. Mejores prácticas de en servicios web en IoT

Los sistemas *RESTful* deben ser centrados en los recursos, que en el caso de una red inalámbrica serían los nodos, los recursos se representan mediante un *URI* (útil para su manejo). En una red que utilice el protocolo *RESTful*, se recomienda que los nodos se conviertan en los recursos del sistema y que se ordenen en una jerarquía similar a la de un sitio web. El sistema debe estar basado en un servidor *HTTP*, que actúa como la interfaz principal, de forma tal que se pueda cumplir con la premisa de generalidad de las interfaces. El servidor *HTTP* debe poder acceder y tener permisos de modificación sobre los recursos, además puede proporcionar información de estado a cualquier dispositivo (ó nodo) que lo solicite.

Un sistema construido alrededor del servidor, permite cumplir con la premisa de la independencia en la implementación de dispositivos, porque los dispositivos simplemente están conectados al servidor, pero a su vez tienen la capacidad de hablarse entre sí, ya que se trata de un ambiente de red colaborativo. Mientras que la escalabilidad se logra gracias a que el servidor es *HTTP*, por lo cual puede administrar cualquier número de dispositivos, ya que el sistema *URI* adoptado permite agregar cualquier número de dispositivos adicionales. El acceso a los dispositivos a través del servidor hace que el sistema sea más rápido, pero eso sacrifica algo de seguridad que se puede ganar mediante una cadena de componentes conectados en secuencia, en este punto dependerá de cual de los dos criterios es más importante en el tipo de aplicación.

El hecho de utilizar una arquitectura *RESTful* en una red *IoT* implica trabajar con un protocolo *HTTP*, que proporciona un paralelismo intuitivo y eficiente entre el modo de acceso a un sitio web común y cualquier dispositivo en la red. La forma en que se envían los datos, la estructura del *Path* y los comandos (si es que los hay), son los tres componentes del protocolo particular[24][49][1][24][35].

2.5. Redes de sensores

Sistemas de adquisición de datos estructurales y redes de sensores inalámbricos, son las opciones de redes de sensores entre las que se decide a la hora de realizar un diseño de sistemas para monitoreo de salud estructural. En redes inalámbricas los nodos sensor tienen libertad de cableado y a la vez restricciones de colocación asociadas, que algunas veces dictan una pauta en el diseño. Esto, junto con el enrutamiento de múltiples saltos, permite una infraestructura de instrumentación muy flexible. Por otro lado, varias limitaciones de las redes de sensores hacen que sea difícil diseñar un sistema de adquisición de datos estructurales inalámbricos. El más importante de estos es la energía: no solo la comunicación inalámbrica consume mucha energía, sino que también el procesamiento de los datos consume una parte significativa de la energía en nodos sensor inalámbricos[52].

2.5.1. Redes de sensores inalámbricos(WSN)

Muchos sensores se conectan a controladores y estaciones de procesamiento directamente (por ejemplo, mediante redes de área local), sin embargo en la actualidad un número creciente de sensores realiza su comunicación para envío de datos recopilados de forma inalámbrica a una estación de procesamiento centralizada. Una arquitectura de sensores inalámbrica adquiere importancia para muchas aplicaciones de red que requieren cientos o miles de nodos sensores, a menudo implementados en lugares remotos y áreas inaccesibles.

Cuando muchos sensores monitorean cooperativa-mente grandes entornos físicos, forman una red de sensores inalámbricos ó *WSN*. Los nodos son sensores y se comunican no solo entre sí, sino también con una estación base (*BS*) utilizando su capacidad de comunicación inalámbrica, permitiendo difundir sus datos medidos a sistemas remotos de procesamiento, visualización, análisis y almacenamiento[66].

Las capacidades de los nodos sensores en una *WSN* pueden variar ampliamente, mientras nodos sensores simples pueden monitorear un solo fenómeno físico; los dispositivos más complejos pueden combinar muchas técnicas de detección diferentes (por ejemplo, acústica, óptica, magnética). Las capacidades de los nodos sensores pueden diferir en términos de comunicación también, por ejemplo, mediante el uso de tecnologías de ultrasonido, *Wi-Fi*, infrarrojo o radio frecuencia, que a su vez tienen velocidades y latencias de datos variables. Si bien los sensores simples solo pueden recopilar y comunicar información sobre el entorno observado, los dispositivos más potentes también pueden realizar funciones extensas de procesamiento y agregación[66][1].

2.5.1.1. Comunicaciones en una WSN

La tecnología de red inalámbrica más común para sistemas móviles es la familia de estándares IEEE 802.11, que fue ratificada en 1997 por el Instituto de Ingenieros Eléctricos y Electrónicos (*IEEE*). La familia de estándares utiliza diferentes bandas de frecuencia, por ejemplo, la banda de 2.4 GHz es utilizada por IEEE 802.11b y IEEE 802.11g, mientras que el protocolo IEEE 802.11a usa la banda de frecuencia de 5 GHz. IEEE 802.11 se utilizó en gran medida en las primeras redes de sensores inalámbricos, que aún se puede encontrar en las redes actuales cuando las demandas de ancho de banda son altas.

Sin embargo, el consumo de potencia de las redes basadas en IEEE 802.11 hacen que este estándar no sea adecuado para redes de sensores de baja potencia. Esto ha llevado al desarrollo de una variedad de protocolos que satisfacen mejor la necesidad de las redes de bajo consumo de energía y bajas tasas de datos. Por ejemplo, el protocolo IEEE 802.15.4 ha sido diseñado específicamente para comunicaciones de corto alcance en redes de sensores de baja potencia y es compatible con la mayoría de los nodos de sensores académicos y comerciales.

Cuando los rangos de transmisión de todos los nodos sensores son muy grandes y los sensores transmiten sus datos directamente a la estación base, se establece una topología en estrella como se muestra en la figura 2.7, donde cada nodo sensor se comunica directamente con la estación base mediante con un *Single-Hop* (un sólo paso).

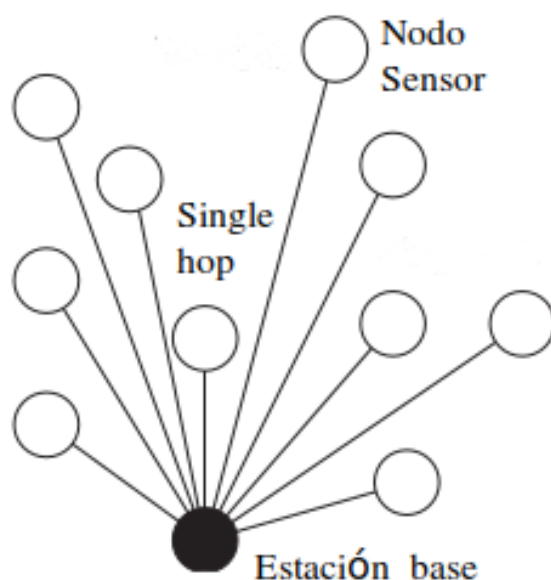


Figura 2.7: Topología Single-Hop[66].

Por otro lado las redes de sensores a menudo cubren grandes áreas geográficas y la potencia de transmisión debe mantenerse al mínimo para conservar energía; en consecuencia, la comunicación *Multi-Hop* (de múltiples saltos) es el caso más común para las redes de sensores (Ver imagen de la figura 2.8). En esta topología tipo malla, los nodos sensores no solo deben capturar y difundir sus propios datos, sino que también deben servir como medio de transmisión para otros nodos sensores (nodos de agregación), es decir, deben colaborar para propagar los datos del sensor hacia la estación base o para propagar información importante para la red[66][1]. Una excelente estrategia para maximizar el uso de energía, es que en nodos de agregación se cambie de tecnología inalámbrica a una de menor consumo energético, por ejemplo *ZigBee*.

ZigBee es un protocolo de baja potencia en redes de área personal inalámbrica, que está basado en el estándar IEEE 802.15.4 y que es adecuado para redes de sensores ubicuas. Si se comparan las tecnologías *Wi-Fi* vs *ZigBee*, *ZigBee* trabaja a baja velocidad de transmisión de datos, pero la mayor ventaja es el bajo consumo de

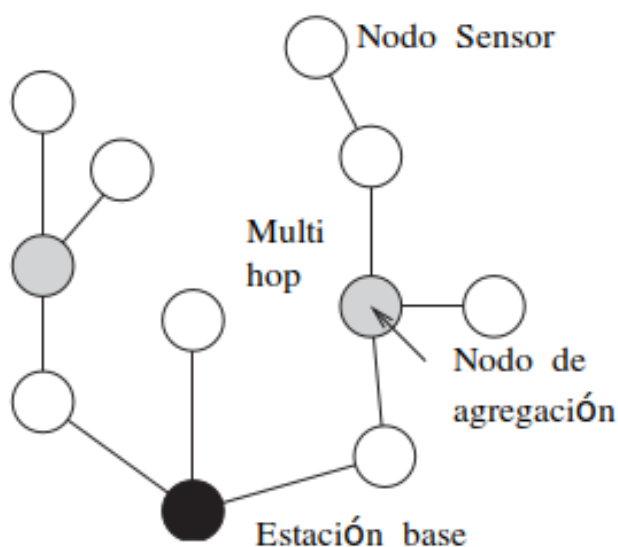


Figura 2.8: Topología Multi-Hop[66].

energía ya que los dispositivos tienen una batería limitada[1].

2.5.2. Redes de sensores inalámbricas(WSN) para SHM

SHM puede considerarse como la integración de los siguientes sistemas: sistema sensorial, sistema de adquisición de datos, sistema de comunicación, sistema de procesamiento de datos y sistema de detección y modelado de daños. Por lo tanto, la investigación en *SHM* abarca detección, comunicación, procesamiento de señales, gestión de datos, tecnología de la información y modelado de sistemas.

La mayoría de los sistemas *SHM* utiliza redes de sensores cableadas (generalmente fibras ópticas) para recopilar información sobre estructuras, sin embargo, los cambios o reconfiguraciones en una red cableada, es en la mayoría de los casos, engorroso, costoso y difícil de mantener. En redes de sensores inalámbricas, una reconfiguración o reubicación de algún(os) nodo(s) es menos complejo de implementar, incluso es más fácil de mantener en el tiempo, independientemente de la magnitud de la estructura en cuestión[68].

El monitoreo de la salud de estructuras usando *WSN* ha ganado interés de investigación debido a la habilidad para reducir costos asociados a la instalación y mantenimiento de sistemas *SHM*. Típicamente, los sistemas *SHM* han utilizado redes de sensores cableadas; pero, la alta confiabilidad y los bajos costos de instalación y mantenimiento de las *WSN*, las han convertido en una plataforma alternativa atractiva. Debido a sus altos costos de instalación, las redes de sensores cableados generalmente solo son factibles para aplicaciones de *SHM* a largo plazo donde la salud de la estructura es de vital importancia.

La reducción en el costo de implementación de un sistema *SHM* con el uso de tecnología *WSN*, fomenta su utilización en la infraestructura pública y privada; situación que ha generado un incremento en su uso para aplicaciones de monitoreo estructural debido a que tienen un menor plazo de instalación. Dichos sistemas po-

drían extender la vida útil de numerosas estructuras al permitir la detección temprana de daños, eliminar el costo de las inspecciones de rutina y lo más importante de todo, mejorar la seguridad pública.

En las *WSN* para *SHM*, los nodos sensor se implementan en varios lugares a lo largo de una estructura para recopilar información sobre su entorno, por ejemplo aceleración, vibración ambiental, carga y estrés a frecuencias de muestreo superiores a 100 Hz. Los nodos de sensor transmiten los datos detectados al receptor directamente o reenviando los paquetes de los demás. La agregación y el procesamiento de datos son necesarios para la detección y localización de daños estructurales y pueden ocurrir en diferentes ubicaciones (por ejemplo, nodos, cabezales de clúster y / o servidor central) dependiendo de la topología de la red[8].

2.5.2.1. Consideraciones de un SHM con WSN

Los sistemas *SHM* han venido siendo utilizados en estructuras para mejorar la seguridad pública y reducir las pérdidas económicas. Sin embargo en algunos sistemas *WSN* diseñados para *SHM*, se ha pasado por alto algunos requerimientos de sistema; superponiendo la necesidad de realizar un análisis de requerimientos necesarios en *WSN* para *SHM*. Los requisitos de sistemas planteados para aplicaciones de *SHM* han de ser considerados para plantear los protocolos en un *WSN* para *SHM*.

El diseño de la red de sensores inalámbricos para *SHM* debe estar estrechamente relacionado con el diseño de la metodología de monitoreo y no necesariamente debe estar ligado al concepto formulado por otras aplicaciones. La mayoría de sistemas existentes contemplan la medición de una única cantidad física, lo cual no se ajusta a las necesidades reales de un *SHM*, por lo que debe de ser considerado un protocolo de comunicación que sea capaz de manejar diferentes tipos de medición, es decir que sea transparente a la cantidad física medida. Otra consideración importante es que la tasa de transferencia de la red de comunicación subyacente, debe poder manejar la enorme cantidad de datos generados por los nodos sensores comunes en *SHM*, tal y como se desarrolló en [68] (Ver cuadro 2.3).

Por otro lado y a pesar de los grandes beneficios que acarrea utilizar una *WSN* en *SHM*, la naturaleza inalámbrica de la red plantea un nuevo y complejo conjunto de desafíos para los investigadores; latencia, seguridad de la comunicación, concurrencia, enrutamiento efectivo y escalabilidad de la red, son algunos de ellos. Debido a que los sensores deben ser alimentados por una fuente de energía externa, un gran desafío en la red inalámbrica es la escasez de energía, que se utiliza para la detección, el procesamiento y la comunicación[68].

Cuadro 2.3: Requerimientos de tasa de transferencia en nodos sensor, para toma de datos de valores físicos comunes en SHM. Tomado de [68]

Medición	Tasa de Transferencia
Acceleración	9,6kbps
Tensión	9,6bps
Desplazamiento	6,4kbps
Temperatura	1,6kbps
Velocidad del viento	3,2kbps

La monitorización de la salud estructural debe tener consideraciones tales como sensores comunes, parámetros comúnmente medidos y detección de daños y algoritmos de localización. Los principales desafíos son la escalabilidad, la sincronización de tiempo, la optimización de ubicación de sensores y el procesamiento de datos. Es de suma importancia tomar en cuenta que, el monitoreo de salud estructural es una aplicación de WSN con requisitos de sistema únicos[8][68].

2.5.2.2. Requerimientos y criterios para SHM en implementación como WSN

Tal y como se comenta en la sección “Consideraciones de un SHM con WSN”, los requisitos de sistemas “wired” desarrollados para trabajar en SHM, han de ser considerados también para plantear los protocolos de comunicación en un WSN para SHM, sin embargo en la actualidad se sabe que la comunicación inalámbrica para comunicación de gama baja (tráfico relativamente bajo de datos, ej.: nodo sensor) brinda amplios beneficios en términos de facilidad y aplicabilidad, permitiendo así su instalación en estructuras catalogadas como antiguas, así como en estructuras modernas.

Emergen además nuevos desafíos y retos con este paradigma de comunicación inalámbrica para los nodos sensor, tales como: latencia y seguridad de comunicación, acceso justo al medio, efectividad de enrutamiento y escalabilidad de la red. En una red inalámbrica, los nodos sensor no están físicamente atados a la red eléctrica del resto del sistema, por lo que requieren de una fuente de poder independiente para su energización y por lo tanto otro gran desafío, que además afecta negativamente la capacidad de procesamiento y el tamaño de la memoria de un nodo sensor inalámbrico.

En redes de sensores inalámbricos se utilizan bandas de radio frecuencia que forman parte del ISM (*Industrial, Scientific and Medical*) de licencia gratuita, que es ampliamente utilizado para fines industriales y de investigación. Un protocolo de comunicación debe incluir medidas que consideren la inminente interferencia causada por otras fuentes cercanas de radio frecuencia que trabajen en bandas ISM[68].

2.5.2.3. Mejores prácticas para SHM en implementación como WSN

El procesamiento de los datos a nivel de nodo y la compresión de datos son técnicas aplicadas para reducir la cantidad de transmisiones de datos, que están directamente asociadas al consumo energético del nodo sensor, además de facilitar el manejo de la cantidad de datos que se generan en los sistemas SHM.

Las topologías basadas en *cluster* con comunicación *multi-hop* (en varios saltos), han sido documentadas e implementadas en redes de espacio distribuido con el fin de realizar monitorización estructural a gran escala, el procesamiento de datos queda entonces distribuido a nivel de *cluster*. Para el manejo del acceso al medio común, se utilizan protocolos basados en reservas que eliminan el *Idle listening*, mejorando la transmisión de datos, el consumo energético y reduciendo las retransmisiones. Los problemas de interferencia se afrontan con el uso de tecnología de amplio espectro (*Spread spectrum*), es decir con un ancho de banda deliberadamente más amplio de lo común[68].

2.5.2.4. Implementaciones reales de tecnologías de sensores /redes inalámbricas

Diversos diseños, prototipos y experimentos con redes de sensores inalámbricos (*WSN*) para *SHM* se han propuesto, implementado y llevado a cabo, tanto en el ámbito académico, como en el comercial. Se tiene registro de hace más de 14 años donde por ejemplo se diseñó una red de sensores inalámbricos llamada *Wisden*; para la adquisición de datos estructurales en una estructura de prueba. Este sistema se utilizó luego en dos entornos reales; una estructura de prueba sísmica y un edificio de oficinas abandonado en Los Ángeles, California, EE.UU. [52][62].

Un sistema *WSN* que se implementó en el puente Golden Gate en San Francisco, EE.UU, es también un ejemplo de implementación de *WSN* para *SHM*[60]. Además el sistema *WSN* llamado *WISAN* (Sistema Inteligente Inalámbrico Sensor y Actuador) para detección de daños y su localización en estructuras [25]. Un sistema sobre el que también se tiene información, es el implementado para probar un puente de hormigón pretensado, en Malasia. Otro diseño integral para *SHM* es *BriMon*; donde se tiene el diseño de un sistema de sensores inalámbricos, para el monitoreo de puentes ferroviarios a largo plazo[62].

2.5.3. Redes colaborativas de sensores

Por sus siglas en el idioma inglés una red inalámbrica colaborativa de sensores (*CWSN*), es una tecnología de colaboración en *WSN* para hacer que los módulos en un nodo y los nodos de sensado en las redes, se comuniquen y cooperen entre sí, para llevar a cabo una tarea específica. Debido a que en una red inalámbrica de sensores, los nodos tienen recursos limitados de energía, capacidad de cómputo y de medición de variables; la colaboración se torna muy importante para poder llevar a cabo aplicaciones con éxito. Las investigaciones en este área pueden ser estrategias de colaboración, arquitecturas, algoritmos y aplicaciones, entre otros[67]. En esta investigación se busca idear una estrategia de colaboración basada en un nuevo protocolo de comunicación, que permita cubrir las necesidades de *SHM* para puentes en Costa Rica.

La clave en este tipo de investigaciones es cómo diseñar modos de colaboración de acuerdo con las ventajas de *WSN* en la detección distribuida (adquisición de datos de variables) y los recursos de red; de modo tal que las tareas puedan asignarse de manera razonable y todos los recursos -incluidos los nodos- se puedan sincronizar. Existe tres tipos de *CWSN* desde el punto de vista de su colaboración; el tipo cooperación, que significa que todos los nodos funcionan de manera cooperativa según sus propios grados de contribución a los objetivos, como por ejemplo la detección cooperativa; tipo competencia, que significa que un recurso participa en la colaboración de acuerdo con su fuerza competitiva, como la programación de recursos; y tipo auto-organización, lo que significa que el estado cooperativo está controlado y mejorado por la detección instantánea en un entorno especial[67]. Este enfoque perfila una investigación tipo cooperación[53] [71].

2.5.3.1. Modelo de servicio Web colaborativo

Para describir una colaboración de enfoque general, debemos asumir que cada servicio Web tiene uno o más puertos en cada una de las categorías que se muestran

en la figura 2.9. Aprovechamos el hecho de que el estado y los resultados de cada servicio web están completamente definidos por los mensajes en estos puertos. La primera categoría son los puertos (orientados a recursos) que suministran la información necesaria para definir el estado del servicio web; dichos puertos pueden ser aumentados, de forma tal que el usuario no tenga problema en pasar la información de control a la aplicación. Los puertos de salida, que son orientados al usuario, proporcionan la información necesaria para construir la interfaz de usuario[29].

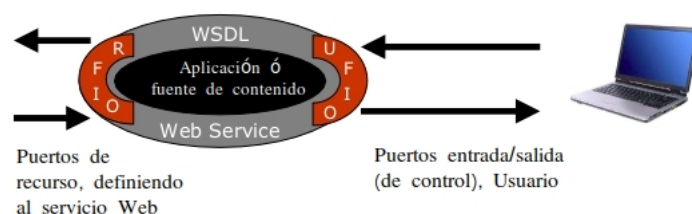


Figura 2.9: Puertos de usuario y de recursos de un servicio web[29].

En este modelo, la aplicación reside tanto el maestro como los otros clientes y los archivos se comparten, dichos archivos se descargan en los hosts de los clientes de antemano, y se implementan en directorios consistentes entre el cliente maestro y los participantes. El cliente maestro captura los mensajes de eventos. Envía estos mensajes de eventos a todos los clientes participantes para su respectiva actualización mientras hace uso de los mismos. Todos pueden compartir los archivos de forma de forma síncrona[29].

2.5.3.2. Requerimientos y criterios para el diseño del protocolo de comunicación entre nodos sensor en una red colaborativa

Al referirse a modelos y protocolos colaborativos, se debe tener claro su diferencia respecto a modelos y protocolos que no lo son. En un modelo no colaborativo los nodos transmiten su información directamente al destino por medio de algún método de división de recursos, como lo pueden ser *TDMA*(*Time Division Multiple Access*, *TDMA* por sus siglas en inglés) ó *FDMA*(*Frequency Division Multiple Access*, *FDMA* por sus siglas en inglés). Mientras que en un modelo de protocolo colaborativo un nodo transmite su mensaje a otro nodo que lo decodifica, añade su propio mensaje y lo vuelve a codificar para hacerle *broadcast* [58].

La colaboración entre nodos ha demostrado aumentar la tasa de transferencia, disminuir el gasto energético ó reducir la necesidad de recursos. En una red colaborativa la mayoría de nodos tienen información propia que enviar al resto de la red. Las aplicaciones de ambientes colaborativos modernos están basadas en tecnología inalámbrica, categorizadas en dos tipos: redes de sensores inalámbricos basados en colaboración, que utilizan la teoría y el método de colaboración para los problemas clave de *WSN* incluido el protocolo de comunicación; y colaboración basada en *WSN*, que investiga cómo usar *WSN* para monitorear y controlar objetos de forma cooperativa[68][58].

El protocolo de comunicación de los nodos de un *WSN* se debe diseñar considerando un ambiente distribuido, dinámico y complejo. Teórica y metodológicamente la filosofía de colaboración debe cumplir con requisitos especiales para ser aplicada

en WSN. Cada nodo debe ser pequeño y eficiente para que necesite menos recursos informáticos y de memoria, además debe de ser consciente de la energía para evitar el consumo de energía adicional y debe ser adaptable a los cambios de la topología de WSN y del entorno[58][67].

2.5.3.3. Mejores prácticas en una red colaborativa de sensores

Las teorías de colaboración tradicionales[58] [70], no se pueden trasplantar a WSN tal cuales. Se debe considerar que la red escala en gran medida, la topología es dinámica y la comunicación es estocástica. Además, el código de cada nodo debe ser muy compacto y eficiente. Tener alguna estrategia de consenso permite el control de los nodos, mientras que la inteligencia de enjambre (*Swarm intelligence*) implícita, donde los nodos son simples y con algún nivel de inteligencia y autonomía, permite que la red tenga gran nivel de inteligencia global. Los algoritmos de enjambre son al final métodos de colaboración, que se encuentran bajo investigación como soluciones de adaptabilidad en WSN.

El problema de localización de nodos desconocidos en una WSN se afronta con la implementación de algoritmos colaborativos, explicado a fondo en [67]. *Energy aware* es la propia consciencia de los nodos acerca de su nivel de energético, en su diseño se incluye una optimización en el número de transmisiones; transmitir y procesar un mensaje según su importancia y el nivel energético de un nodo. Los mensajes importantes se enviarán a todos los nodos, pero los mensajes menores solo se transmitirán a los nodos con alta energía o al concentrador de datos directamente. Colaboración en tareas de la red implica descripción, descomposición, asignación, calendarización y ejecución de las tareas entre todos los nodos de la red[67][68][29][58].

2.5.3.4. Redes Grid y P2P

Las redes tipo *Grids* y *P2P* están asociadas con entornos de colaboración. Las redes P2P comenzaron con comunidades *ad-hoc*, como las que comparten archivos MP3; mientras que las *Grids* iniciaron como soporte de virtual para empresas u organizaciones. Hoy en día es posible discutir sin ambigüedad sus diferencias, similitudes y lo que significa integrarlas en ambientes colaborativos, para lograr objetivos específicos y con enfoques colaborativos; ambas son adaptables a ambientes controlados por sensores[29].

2.5.4. Servicios Web en redes colaborativas de sensores

Con el fácil acceso a la nube y la mayor capacidad de procesamiento de los servidores en la nube o algún tipo de servidor por medio de internet, es posible recopilar datos de redes de sensores en diferentes ubicaciones y procesar grandes cantidades de datos. El monitoreo y control de dispositivos físicos y sensores a través de Internet, es cada vez más popular, debido a factores tales como: recientes mejoras de conectividad a Internet(Servicios Web), dispositivos electrónicos de bajo consumo energético y los métodos de comunicación inalámbrica; uno de los protocolos de comunicación más utilizados en dichas redes es el denominado protocolo de aplicación restringida (*CoAP* que hace que la comunicación entre sensores inalámbricos a través de Internet sea más interactiva[43]).

Existen dos paradigmas principales de servicios web: arquitectura orientada a servicios (*Service-Oriented Architecture*, *SOA* por sus siglas en inglés) y arquitectura orientada a recursos (*ROA*, donde el URI es único para los recursos e información de atributos[63]). El paradigma *SOA* se basa en protocolos de servicios web estándar, como el Protocolo simple de acceso a objetos (SOAP), el Lenguaje de descripción de servicios web (WSDL) y el Lenguaje de ejecución de procesos comerciales (BPEL). Debido a su confiabilidad y a su seguridad a nivel de mensaje, *SOA* es el paradigma que más se usa en aplicaciones empresariales; sin embargo, la complejidad de los protocolos lo hace menos atractivo para la integración básica y *ad-hoc* de servicios. Por otro lado *ROA*, se basa en *REST* de facto [63][48]. *REST* se ha convertido en un enfoque preferible debido a su arquitectura simple y liviana, fácil acceso y escalabilidad. Los servicios web RESTful (servicios web basados en REST) se desarrollan para proporcionar interfaces estándar[22][35][64][59].

Los servicios web se consideran un "sistema de software diseñado para admitir la interacción interoperable de máquina a máquina a través de una red". Se caracterizan por permitir el intercambio de datos y la integración de aplicaciones a través de la red, dando lugar a poder soportar el uso de datos de varios dispositivos (por ejemplo: en la nube, computadora local, micro-computadora y dispositivos móviles) y plataformas (por ejemplo: diferentes sistemas operativos). Es importante emplear un protocolo de servicio web ampliamente adoptado y soportar además consultas condicionales ya que estas son inherentes en aplicaciones de *SHM* (por ejemplo, consulta de rango para datos de series de tiempo).

REST significa transferencia de estado representacional en el idioma español, pero toma su nombre por sus siglas el idioma inglés. *REST* es una arquitectura basada en estándares Web y utiliza protocolo HTTP para la transferencia de datos. Además mejora la escalabilidad mediante la distribución del estado de la aplicación de los servidores a los clientes. El estilo de la arquitectura *REST* tiene seis restricciones. Estas restricciones, aplicadas a la arquitectura, fueron originalmente comunicadas por Roy Fielding en su disertación doctoral (*Architectural Styles and the Design of Network-based Software Architectures*). Las restricciones dadas son interfaz uniforme, sin estado, cacheable, cliente-servidor, sistema en capas y código bajo demanda[65][64][36].

Los servicios web RESTful tienen un rendimiento rápido y una gran escalabilidad, lo cual sugiere que su implementación permite cumplir con los requisitos de un sistema *SHM*. Cabe señalar que existen otros protocolos de comunicación livianos, como el protocolo MQTT; si bien dichos protocolos tienen ventajas sobre la comunicación en tiempo real mediante mensajes de publicación/suscripción; RESTful presenta grandes ventajas en capacidades de consulta que manejan eficientemente los patrones de uso de datos típicos comúnmente vistos en aplicaciones *SHM* (por ejemplo: consulta de rango para datos de series de tiempo)[59][50].

Con el propósito de estandarizar los protocolos de comunicación a utilizar y que los mismos sean de carácter gratuito y abierto, se busca utilizar durante la ejecución de esta investigación el uso de servicios Web RESTful, debido a que se adaptan bien a sistemas *SHM*[35][43].

2.5.4.1. Generalidades de la interacción Web

La Web moderna utiliza *HTTP/TCP/IP* como base de su protocolo fundamental. El protocolo *HTTP* además de llevar datos de contenido, define una gama amplia

de meta-datos para solicitudes y respuestas. Como parte del protocolo de mensajería, los meta-datos son transportados en los encabezados *HTTP*, además pueden ser: información sobre la propia aplicación del cliente, información acerca de los idiomas de contenido aceptables y codificaciones de los datos.

Muchas de las aplicaciones Web utilizan el paradigma de transacción *REST*. Una solicitud *REST* completa es autónoma, es decir que contiene la información necesaria para que un servicio procese la solicitud. Sin el contexto del cliente, un servicio *RESTful* podría no ser capaz de funcionar en absoluto. Una parte muy importante del protocolo *HTTP* es el *cookie HTTP*, que es una forma simple de contexto distribuido, donde el servicio lo utiliza para transmitir tokens -a menudo indiferentes para sus clientes-, los tokens son típicamente únicos para cada cliente, permitiendo al servicio asociar múltiples solicitudes de un cliente determinado.

La utilidad de las aplicaciones distribuidas depende de la latencia³ entre la solicitud de acción y la representación de un resultado, la Web está compuesta de múltiples servicios altamente distribuidos por lo que la latencia tiene una importancia considerable en el diseño de un sistema de este tipo. Navegadores modernos y otras aplicaciones han evolucionado para ser cada vez más eficientes en la forma en que usan los datos, almacenando en caché el contenido localmente y reutilizando la información del *DNS* y las conexiones *TCP*.

La comunicación *HTTP/RESTful* implica que: los clientes tengan datos que enviar, las solicitudes de clientes sean únicas y estén íntimamente ligadas a los datos de contexto de los clientes asociados consigo mismos, y que se tome en cuenta la latencia como medida de la eficiencia y de la capacidad de respuesta[36].

2.5.4.2. Algoritmos y generalidades de implementaciones

Se esta sección se describen algunas implementaciones de *REST* sobre redes colaborativas. *MANET* es un modelo para nodos de redes móviles *ad-hoc* y actuadores en redes inalámbricas de sensores de colaboración para optimizar el procesamiento de datos en áreas donde ocurren eventos, tales como lugares donde se requiere prevenir incendios. Para este modelo se desarrolló un algoritmo llamado Distribuido.

En el algoritmo distribuido cada nodo necesita información global y centralizada sobre las ubicaciones de todos los demás nodos; así como las rutas más cortas entre sí. Además, también se requiere información sobre los eventos. Sin embargo, en los escenarios de red reales, dicha información no está disponible, por lo que el algoritmo heurístico distribuido hace que cada nodo *MANET* recopile información parcial a través de las comunicaciones, logrando el mismo objetivo. Se utilizan dos fases: configuración y negociación.

La fase de configuración se utiliza para distribuir el número de nodos *MANET* entre los eventos detectados en las áreas correspondientes, logrando un valor proporcional ponderado max-min (*WPMM*), como criterio de equidad. Se busca: un número fijo de nodos, y el tiempo de procesamiento para cada evento detectado que es proporcional a su importancia y escala. Las redistribuciones posteriores de los nodos en cada ciclo utilizan el algoritmo en la fase de negociación, que es basada en una técnica de mercadeo, es decir, un nodo tiende a moverse hacia el área más cercana para procesar los eventos detectados, mientras que otro nodo indica a los

³Tiempo de viaje ida y vuelta dentro del transporte o el protocolo de aplicación, de forma más general es el tiempo de retardo entre la causa y el efecto.[36].

más cercanos que se muevan a su área para realizar el procesamiento de eventos detectados y la comunicación entre nodos es realizada por medio de servicios Web *REST*[56].

Con un problema muy similar al del estado de los puentes en Costa Rica, en la universidad de Carleton, Ottawa, CANADA; se realiza la investigación de una arquitectura *Middleware*⁴ para el manejo de la infraestructura de puentes basado en sensores[61], donde en 2005 el *US bridges [ASC 2005]* reportó un 27% de puentes en estado estructuralmente deficiente, además de tener un 40% de la totalidad de puentes en Canadá con más de 40 años de edad. Antes del proyecto el problema fue abordado por separado, creando un entorno fragmentado y una gran complejidad, el enfoque propuesto pretende la integración basada en la red para conectar las piezas.

El proyecto de *Middleware* se basa en una arquitectura configurable, donde diferentes tipos de servicios se pueden gestionar, de diferentes maneras. Es posible tener un enfoque centralizado para algunos nodos, mientras que a su vez se puede tener un enfoque distribuido para otros nodos. Cada uno de los sitios que participan en la administración de la infraestructura del puente forma un nodo y todos los nodos están interconectados por la red para formar un sistema distribuido con N nodos.

La arquitectura además posee módulos cuya función es la de desacoplar a otros módulos sin afectación en el desempeño de la red, permitiendo la integración de nuevos servicios sin interrumpirse entre sí. Algunos recursos están asociados con dos tipos de servicio, uno que es basado en *SOAP*, mientras que el otro tipo de recurso utiliza *REST* de peso ligero. Los servicios Web *RESTful* utilizan un protocolo de peso ligero y, por lo tanto, son utilizados en aplicaciones que requieren baja latencia. Para aplicaciones más generales, se utilizan servicios Web basados en *SOAP*, debido a que *REST* no tiene protocolos estandarizados para funciones como la seguridad[61].

2.6. Proyecto e-Bridge

En general el estado actual de la infraestructura vial en Costa Rica es una debilidad de la economía del país. Los puentes representan un componente fundamental de la infraestructura vial de cualquier país. Daños y deficiencias en puentes pueden tener un impacto directo en la ejecución de actividades socio-económicas de la población en general, la infraestructura vial es una de las tres debilidades más fuertes que el país viene arrastrando desde hace más de trece años[20][18].

Un porcentaje significativo de puentes nacionales muestra una condición promedio o crítica respecto a algunas partes de su estructura, específicamente 62% muestra condición regular y un 34% condición crítica (Ver imagen de la figura 2.10), razón por la cual nace e-Bridge como proyecto de investigación en el Centro de Investigación en Vivienda y Construcción (CIVCO) del Instituto Tecnológico de Costa Rica[16][15]. Esta tesis es desarrollada en el contexto de investigación del proyecto e-Bridge, para dar apoyo en la generación de prototipos de redes de sensores que permiten la toma y gestión de datos en puentes.

⁴Software especialista que actúa como conexión entre un sistema operativo o base de datos y aplicaciones, especialmente en una red.[40].

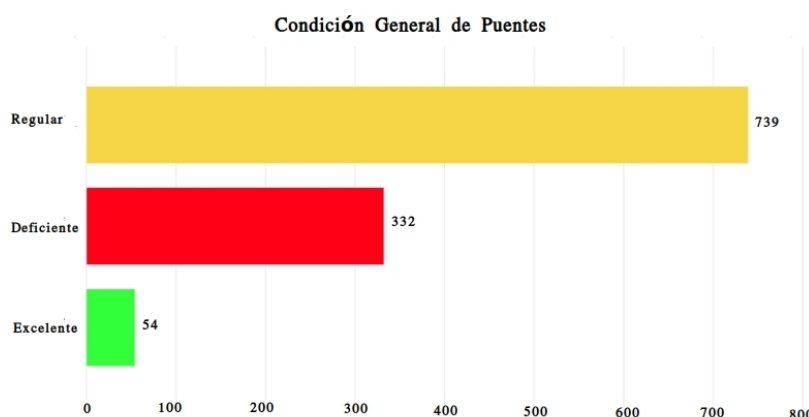


Figura 2.10: Condición General de los puentes en Costa Rica - proyecto e-Bridge.

2.6.1. Enfoque e impacto de e-Bridge

El enfoque de investigación del proyecto e-Bridge es el de la predicción remota de fallas en puentes. Surge como un primer proyecto nacional en el área y cuenta con el aporte económico del TEC y del Consejo Nacional para Investigaciones Científicas y Tecnológicas (CONICIT), además del soporte en el suministro de información por parte del MOPT y el Consejo Nacional de Vialidad (CONAVI). El objetivo principal de este proyecto es el de diseñar y desarrollar soluciones tecnológicas para el monitoreo de salud estructural de puentes en Costa Rica.

La infraestructura de carreteras y puentes es crucial para la economía costarricense, ya que promueve actividades como el turismo y el comercio. La planificación adecuada y la priorización de proyectos de infraestructura es de gran importancia para las instituciones gubernamentales relacionadas con e-Bridge[17].

2.6.2. Etapas y actualidad del proyecto

El proyecto de investigación e-Bridge, está definido por varias etapas (Ver imagen de la figura 2.11). Inicialmente e-Bridge 1.0, 2011-2012; destinada principalmente al desarrollo de capacidades básicas de *SHM* en el ITCR. Posteriormente E-Bridge 2.0, 2013-2015, donde el objetivo primordial fue la integración de información. E-Bridge 3.0, 2016-2017, da lugar al diseño de varios prototipos *SHM* para puentes con metodologías de evaluación, análisis de riesgos ambientales, características de diagnóstico y técnicas de inteligencia empresarial, para respaldar el análisis de desempeño del puente[17] y así apoyar en la toma de decisiones por parte de diversos actores e instituciones en el área de gestión vial.

Actualmente, resultados de las investigaciones de las diferentes etapas de e-Bridge están siendo aplicados y transferidos por medio del proyecto de extensión “Priorización de intervención de estructuras de puentes mediante el uso de indicadores de desempeño. Plan piloto: Municipalidad de El Guarco de Cartago” del ITCR. Este proyecto tiene como objetivo brindar a una municipalidad particular herramientas de apoyo a la priorización de las intervenciones requeridas para los puentes que deben monitorear y mantener.

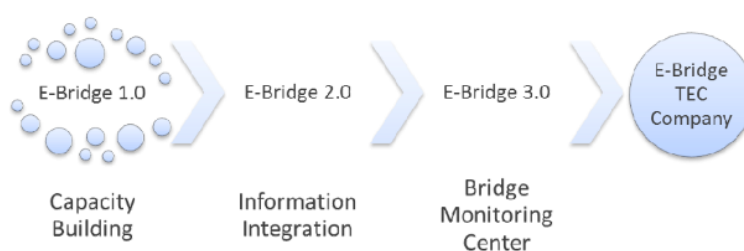


Figura 2.11: Diagrama de desarrollo general del proyecto e-Bridge.

2.6.3. Logros del proyecto

Varias publicaciones, entre ellas las expuestas en [46][17][18][14][19][47][20][15][16][30][41] y algunos prototipos de sistema, son parte del palmarés de e-Bridge. De los prototipos construidos algunos son utilizados para obtener mediciones de diversas variables relacionadas a salud estructural de puentes y otros son para redes de comunicación.

Como antecedente del presente trabajo, en el año 2016, a manera de proyecto exploratorio y con un alcance más limitado, el Centro de Investigaciones en Computación (CIC) y el grupo de investigación e-Bridge 3.0., en conjunto; realizan una investigación preliminar para la utilización de un dispositivo empotrado que gestiona un conjunto de sensores para la obtención de datos, el trabajo realizado contempla variables tales como: vibración, deformación y desplazamiento de puentes. Las capacidades de dicho dispositivo son de conexión inalámbrica y ejecución de una versión especializada para dispositivos embebidos del sistema operativo GNU/Linux.

Dicho dispositivo posee la capacidad de realizar una conexión a un servidor de tipo FTP, para colocar datos o bien ejecutar servicios web en un servidor de aplicaciones, al que tienen acceso ingenieros encargados de interpretar los datos y con el fin de apoyarse en los datos para la toma de decisiones. Además cuenta con un servicio web por el cual las personas interesadas en el proyecto pueden tener un acceso directo al dispositivo para su configuración, para la visualización de notificaciones y para el despliegue y tratamiento de la información generada por el dispositivo empotrado[44][15][16].

El presente trabajo contempla una revisión exhaustiva del estado del arte del enfoque que se le da al monitoreo de salud estructural (*SHM*) en la actualidad, visto desde una perspectiva colaborativa que fue utilizada en los 2000 para suplir necesidades comerciales de empresas como paradigma computacional [13] y que por sus cualidades filosóficas y funcionales, hoy en día se aplica a varios ámbitos de situaciones humanas, como por ejemplo desastres naturales. Además se integra a la investigación, tecnología de plataforma abierta de software plasmada en paradigmas computacionales de tendencia como lo son *IoT* y *RESTful*, tecnología inalámbrica para sensores (*WSN*); y tal y como se abarcó en [41], hardware de bajo costo como lo son algunos sensores digitales, y dispositivos empotrados con sistemas operativos Linux que son plataformas de alta eficiencia, estos últimos postrados como una tendencia en la solución de problemas cotidianos. Dicha recopilación representa un aporte teórico importante y no disponible hasta el momento, es una base bien fundamentada para la presente investigación y para la generación de trabajo futuro.

Capítulo 3

Análisis de Requerimientos

En este capítulo se presenta la definición de requerimientos que fueron establecidos mediante reuniones con el Dr. Cesar Garita sobre lo que debe cumplir el protocolo de comunicación y la red colaborativa para monitoreo de salud estructural de puentes en Costa Rica. Se incluye; un diagrama de contexto que describe el funcionamiento general del sistema diseñado e implementado, los requerimientos funcionales del sistema mediante diagramas de casos uso y de secuencia, el modelo de los datos y finalmente los requerimientos técnicos de los componentes que lo conforman.

3.1. Diagrama de contexto

En la Figura 3.1, se muestra un diagrama de contexto de la solución implementada. El sistema consiste de un conjunto de sensores que son colocados en las distintas estructuras puente, cada sensor es a su vez conectado a un dispositivo embebido que se encarga de su gestión. Un dispositivo que se enlaza por primera vez a la red, lo hace mediante una pre-configuración que le permite auto-identificarse con el resto de la red y así definir su propia dirección IP, la configuración de arranque utiliza los parámetros definidos por el usuario para realizar lecturas de la información proveniente de los sensores.

Los nodos de red realizan en primera instancia un análisis constante de los datos obtenidos de los sensores, en busca de definir en cual de los rangos estipulados en la pre-configuración se encuentra la alerta (verde, amarillo ó rojo), las alarmas son explicadas con más detalles posteriormente; en segunda instancia cada cierto periodo de tiempo realiza una consulta al servidor web ó a través del nodo maestro (dependiendo del rol del nodo) en búsqueda de actualizaciones, además de enviar el promedio de los datos estables para su registro. El dispositivo embebido maestro se encarga de enviar y solicitar información por medio de una conexión a internet.

El usuario puede interactuar con la red por medio de la interfaz web ó directamente a la base de datos; en ambos casos están disponibles los datos de cada nodo sensor, la configuración actual bajo la que trabajan los nodos sensor, la definición de redes colaborativas, la definición de puentes, el estado actual de alerta de cada red colaborativa y el tiempo de propagación de alarma entre nodos; incluyendo a los

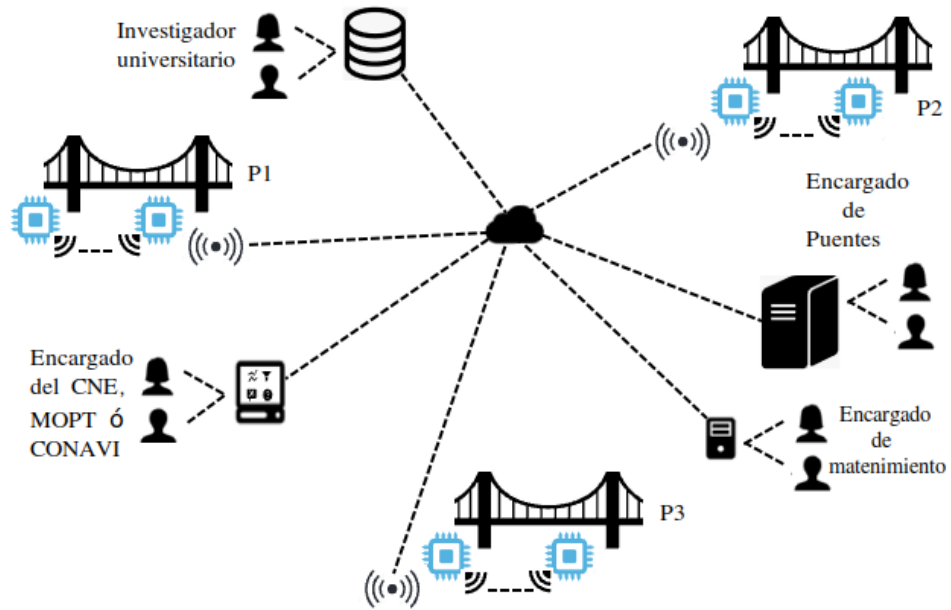


Figura 3.1: Diagrama de contexto.

que componen redes locales. Una opción mucho más limitada, es el uso de algunas consultas predefinidas, que básicamente permiten saber como están definidas las redes colaborativas en el sistema. Los roles de usuario se definen por medio de su función, por ejemplo un acceso al servidor donde reside el servidor de base de datos permite un análisis de *raw data* para investigación, por otro lado la interfaz web permite al encargado del CNE/MOPT/CONAVI tomar una decisión inmediata, con base a lo que puede ver desde su teléfono celular, computador personal ò cualquier dispositivo con conexión a internet.

El protocolo de red contempla diferentes niveles de colaboración entre nodos, el nivel más básico de colaboración es el local (por ejemplo P1 en la figura 3.1), donde todos los nodos sensor se hablan entre sí para definir un estado unificado de cada puente, solamente uno de ellos transmite el dato de alerta del puente a la red colaborativa a la que pertenece y los datos de cada nodo al servidor de base de datos. La red colaborativa, es un nivel más alto donde se colabora entre puentes, que comparten entre sí su estado de alerta, permitiendo unificar un estado de alerta por red colaborativa, la cuál toma como estado global el el peor de los casos de los puentes que la conforman; a manera de ejemplo en la figura 3.1 podría definirse a P1 y a P2 como una red colaborativa 1 y a su vez P3 podría pertenecer a otra red colaborativa.

3.2. Requerimientos funcionales del prototipo desarrollado

En términos generales esta investigación genera como principal resultado el diseño e implementación del protocolo y prototipo de una red de sensores colaborativas para monitorizar puentes en Costa Rica, con el objetivo de prevenir que el efecto de los desastres naturales en afecte en gran medida a dichas estructuras y a sus usuarios. En la figura 3.2 se muestra el diagrama de casos de uso general para lograr dicho objetivo.

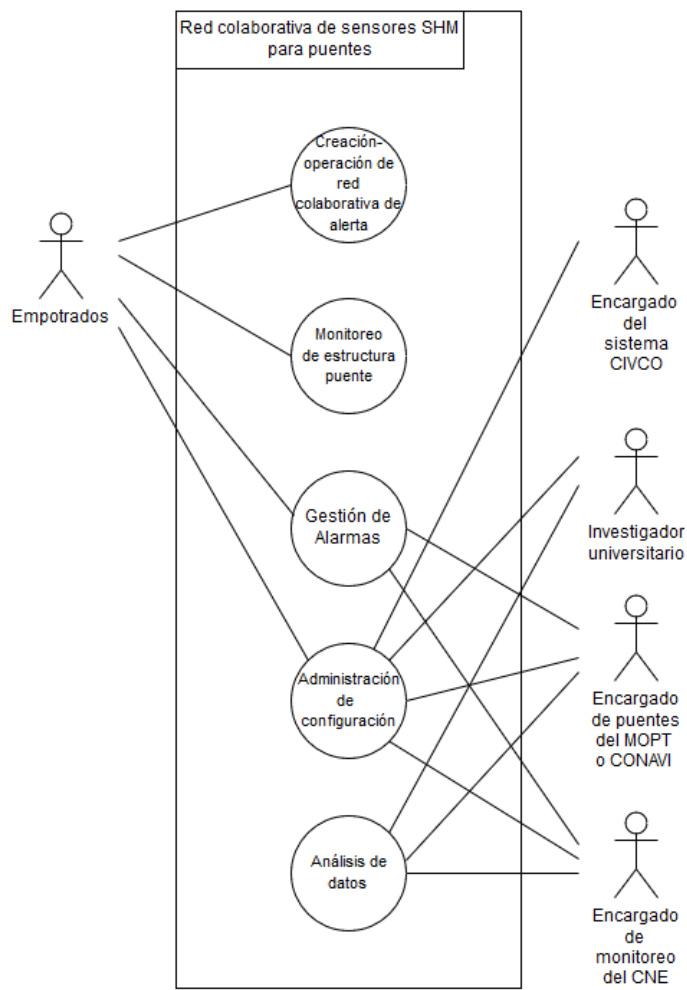


Figura 3.2: Diagrama de casos de uso para la red colaborativa de prevención de desastres.

El caso de Creación y Operación generaliza la preparación de la red, que incluye parámetros de funcionamiento, la creación de la bases de datos, la carga de datos y la generación y definición de su ontología. Monitorizar comprende toda acción que

toma la red para inquirir datos y utilizarlos en la definición de estados de alerta en el caso de Gestión de Alarmas ó para envío de información interpretada a la base de datos en el caso Análisis de Datos. Cada elemento que participa en la red requiere del caso Administración de Configuración como vía autenticación y rol de desempeño, que a su vez es utilizado por todo aquel usuario que requiera de hacer cambios en el modo de funcionamiento ya sea de los nodos o redes colaborativas.

3.2.1. Creación y operación de la red colaborativa de alerta

El flujo de los datos mostrado en la figura 3.3, se utiliza para la determinación del estado de alarma, tanto local (en una estructura puente determinada), como global (varias estructuras puente con algún vínculo entre sí), se inicia con la toma de datos, que para el prototipo de prueba es mediante un sensor infrarrojo que determina el nivel del agua bajo el puente. En este punto, la complejidad de cada estructura puente, estará en función de cuantos nodos sensor sean requeridos para poder realizar la determinación de la variable medida. A mayor número de unidades requeridas, más compleja será la red colaborativa local, de la cual saldrá un único punto de conexión a la red telefónica, para enlazarse a la estructura a la red colaborativa global.

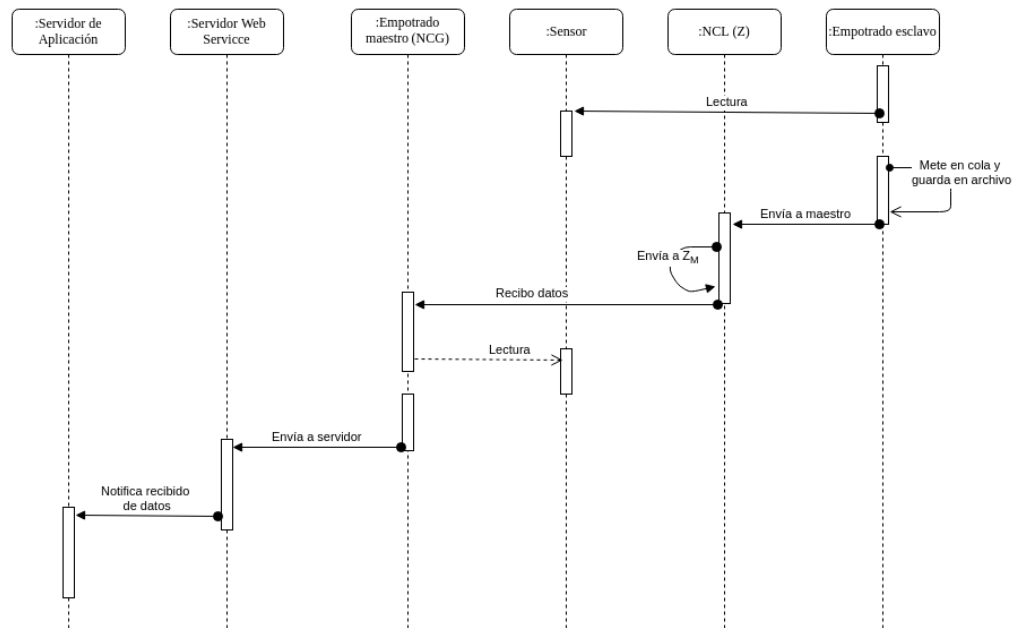


Figura 3.3: Diagrama de secuencia para la creación y operación de la red colaborativa de sensores.

A nivel de red colaborativa local, se define tanto el concepto de nodo sensor maestro, como el de nodo sensor esclavo. Un nodo sensor es aquel que toma mediciones de la(s) variable(s) bajo escrutinio (ej: nivel del agua bajo el puente), hace un análisis estadístico que corrige la toma de mediciones fuera de rango (propias de los sensores de bajo costo) y finalmente transmite su dato final a otro nodo, ya sea al

más cercano de los nodos esclavo o al nodo maestro, dicha comunicación se hace a través del nodo de comunicación local (NCL(Z)), comunicación que es realizada con tecnología que permite trasegar datos a nivel local. Por otro lado un nodo maestro realiza las mismas funciones que un nodo esclavo, pero además determina mediante una comparación entre los datos de todos los nodos locales, si existe un falso positivo de parte de alguno de los puntos de toma de datos, incluso respecto a sí mismo. Cada nodo sensor promedia los últimos diez valores más estables que ha tomado, antes de transmitir su propio dato a la red colaborativa local. Finalmente el nodo maestro transmite los datos de todos los nodos de la estructura al servidor de aplicaciones y su nivel de alerta a todas las redes colaborativas globales a las que pertenece.

Un falso positivo se refiere a un valor estadísticamente determinado por un nodo sensor, que no es cercano al de sus vecinos; a nivel de red colaborativa local el nodo maestro es quien se encarga de determinar si un nodo está o no reportando un falso positivo. Para el caso específico de la altura del nivel del agua, podría indicar que hay algún objeto bajo el puente que obstruye el paso del agua en un sector y que es el que genera una medición diferente respecto a los demás puntos de medición, por lo que sería descartable para determinar la alerta local, en el caso de variables físicas diferentes al nivel del agua, dicha funcionalidad podría ser muy útil, más su análisis sería diferente. La detección de un falso positivo no altera las alarmas globales de la red, genera una notificación y depende completamente de tener más de un punto de medición en una sola estructura puente para poder funcionar correctamente, en el caso de tener únicamente una unidad de medición no será posible determinar un falso positivo.

Como se ha mencionado antes en este documento, cada nodo sensor maestro tiene la tarea de ser el punto de conexión con el resto de la red, tanto a nivel de servidor como de redes colaborativas globales, por lo tanto debe de transmitir un dato final de la(s) variable(s) física(s) medida(s) por cada nodo, al servidor web y a su vez debe de enviar su estado o nivel de alerta a todos los nodos que formen parte de su(s) red(es) colaborativa(s). Un puente puede formar parte de más de una red colaborativa, ya que puede inferir en mas de una cuenca, pero solamente a una ruta. De esta forma todos los nodos maestro de una red colaborativa se comunican con el fin de generalizar el estado de alerta, que afecta a puentes diferentes del que da origen al disparo de la alarma, incluso podría propagarse a otra red colaborativa en caso de que un puente pertenezca a más de una red. Una vez que el dato y el estado de alarma es recibido por el servidor web, este se encarga de actualizar al servidor de aplicación, que a su vez incorpora ó actualiza la base de datos. El resultado final, que se compone de datos medidos y estado de las alarmas, puede ser visualizado en corto tiempo en la interfaz web del sistema.

3.2.1.1. Servidor de Aplicaciones

En el *Backend* se atienden las solicitudes realizadas al servidor de aplicaciones. Una solicitud debe de hacerse a través del Servidor Web y puede realizarse tanto por los nodos maestro, como por el *Frotend* desde la página web. La solicitud bien puede ser para adquirir información almacenada en la base de datos ó bien para solicitar el almacenamiento de nueva información en la misma. Cada solicitud *HTTP* es manejada por el *Backend* y respondida hacia el *Frotend* o nodo maestro (ver figura 3.4).

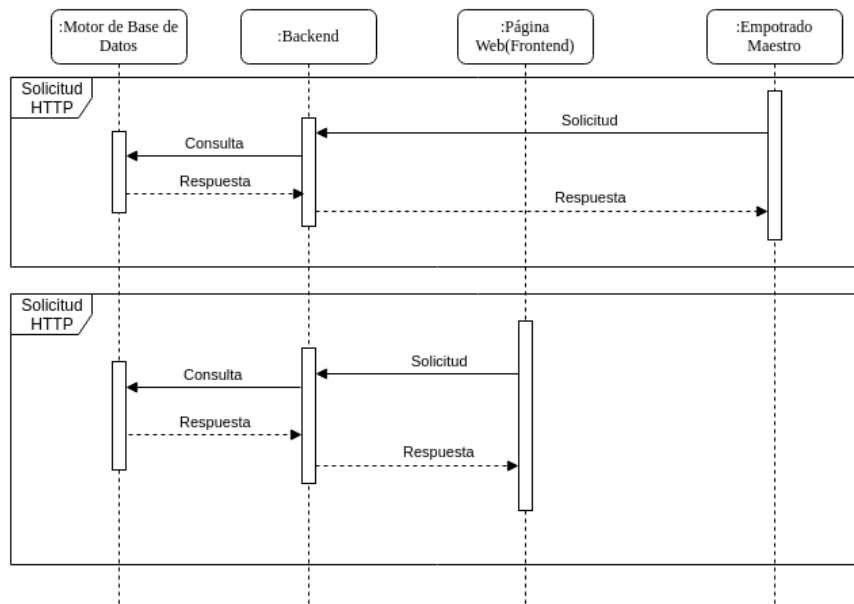


Figura 3.4: Diagrama de secuencia del Servidor de Aplicaciones.

Una solicitud que proviene de un nodo maestro, utiliza "Petición" para interactuar con el servidor de aplicaciones y obtener ya sea una actualización de su configuración o ingresar datos de mediciones tomadas a la base de datos. La información de la red colaborativa que tiene que ver con el estado de alerta, también será almacenada, pero no necesariamente como una prioridad ya que se priorizará propagarla a otros nodos de la red global. En la figura 3.5 se puede apreciar que toda solicitud, independientemente de quien la haya realizado, se realiza a través de "Consulta Base de Datos".

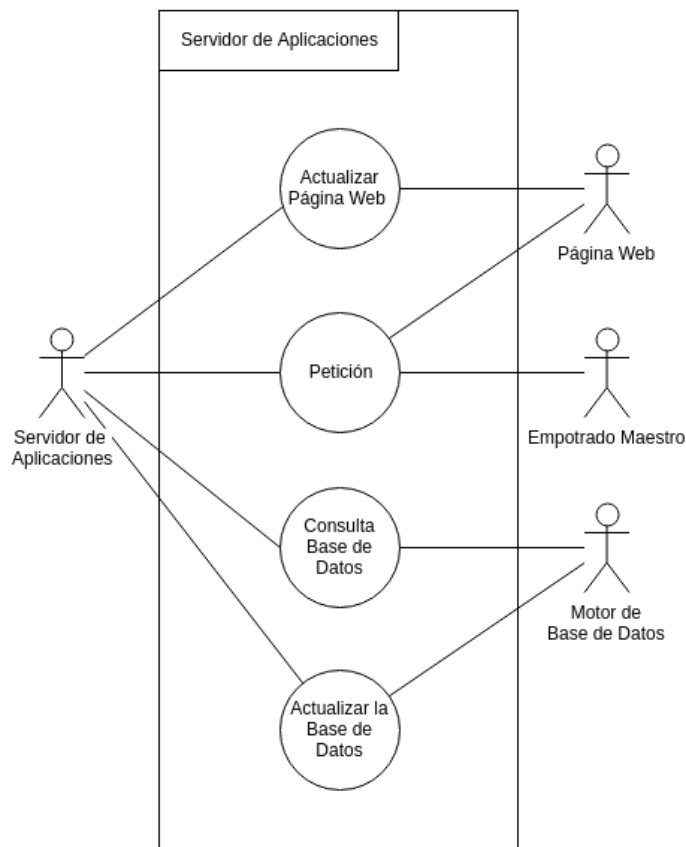


Figura 3.5: Casos de uso del Servidor de Aplicaciones.

3.2.2. Monitorización de la estructura puente

En la figura 3.6 se detalla la parte del protocolo que se encarga de la toma de datos. En el diagrama de secuencia de los datos se indica que los empotrados solicitan una lectura y el sensor responde de vuelta con el dato medido en ese momento; el nodo maestro procesa el dato, determina si existe algún falso positivo y transmite un paquete de datos al Servidor Web que interpreta los datos para que el Servidor de Aplicaciones los utilice. La toma del dato es procesada por el empotrado de forma tal que se eliminan los valores fuera de rango, que son causados por utilizar sensores de bajo costo, sin embargo es una medida que no afecta a sensores de mayor calidad, debido a que lo que se hace es una desestimación de los valores más lejanos a los más estables que se han tomado, hasta contar diez.

Dentro de los requerimientos de diseño del sistema que han sido especificados por los expertos en el área de estructuras tipo puente del TEC, se debe tomar en cuenta en el procesamiento de datos, el uso de recursos (incluyendo sensores) de bajo costo y dispositivos de código abierto, lo cual favorece su instalación en estructuras pequeñas sin incurrir en un alto costo, pero aún siendo capaces de dar una buena medición de las variables físicas evaluadas. Como parte también del proce-

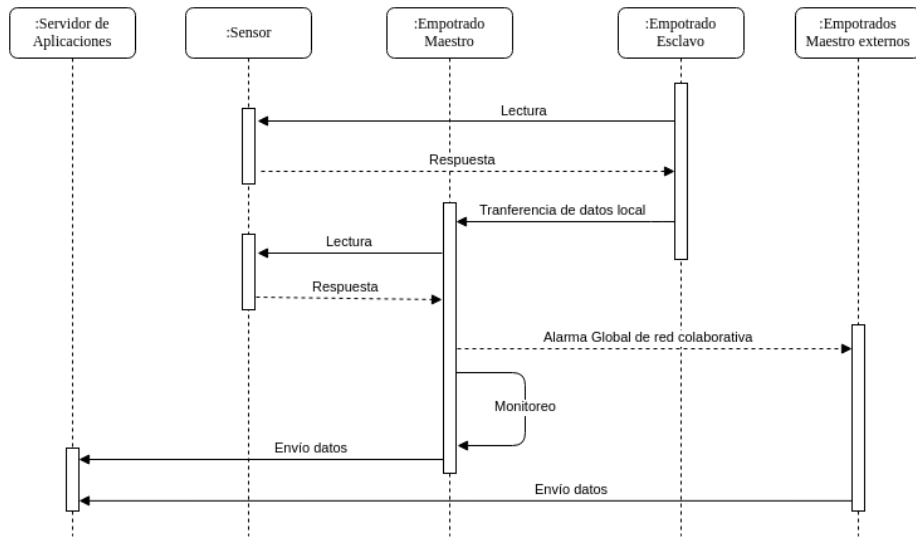


Figura 3.6: Diagrama de secuencia para monitorizar la estructura puente.

samiento de los datos en el nodo maestro, se hace un análisis sobre el estado local de alarmas y en caso de que haya un disparo de alarma, se envía el nuevo estado a todos los nodos que pertenecen a la misma red colaborativa a la que pertenece el nodo que generó el disparo.

Principalmente se requiere prevenir desastres naturales, razón por la cual se impregna en el diseño el sentido colaborativo entre estructuras lejanas (dos o más puentes distintos que mantienen algún vínculo territorial), es decir que un nodo maestro actualiza a otro nodo maestro (maestro externo), con los cambios de estado de alarma y el maestro externo toma el nuevo estado de alarma como suyo en una variable global que a su vez tiene una prioridad más alta que el estado de alerta local; el sistema se caracteriza por seguir la misma estructura de alarmas que define la Comisión Nacional de Emergencias (CNE), es decir estado verde, amarillo o rojo; la definición de cada alarma utilizada en la implementación del prototipo se aprecia en el cuadro 3.1.

Cuadro 3.1: Definición de alarmas de emergencia para nivel de agua bajo el puente

Alarma	Rango del nivel de la altura libre inferior del puente (%)
Verde	50-100
Amarillo	25-50
Rojo	0-25

Los casos de uso para la obtención de datos por parte de los nodos sensor complementan al diagrama de secuencia y se especifican en la figura 3.7. Entre los casos definidos para el sistema, se encuentran “Monitoreo de estructura”, que permite tanto a “Subir Datos al Servidor” como a “Actualizar Alarma Global”, para que puedan ser utilizados para tomar acciones inmediatas de prevención o de clausura de estructuras en riesgo inminente.

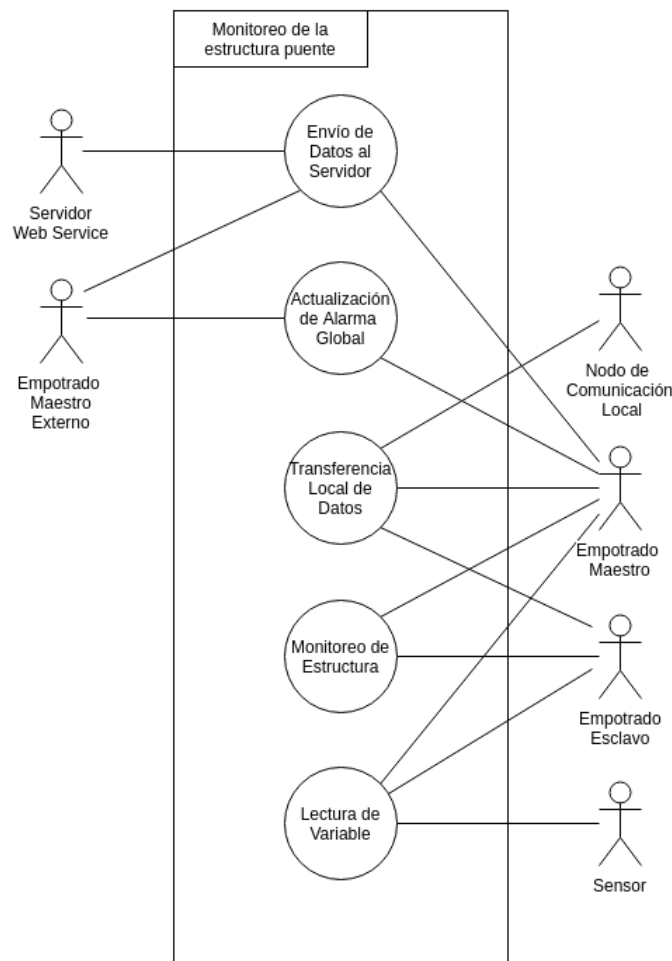


Figura 3.7: Casos de uso para la obtención de los datos en los nodos sensor.

3.2.3. Gestión de alarmas

La generación de alarmas habilita la toma de decisiones a mediano o corto plazo, e incluso de forma inmediata. Clausurar un puente en uso, podría salvar vidas. En la figura 3.8 se muestra el diagrama de secuencia de la forma en que se generan las alarmas como parte del protocolo de emergencias definido.

Un nodo sensor, ya sea esclavo o maestro puede generar un cambio de estado en las alarmas después de haber promediado los diez valores más estables y si el valor final altera el estado actual de la alarma local de la estructura (ver cuadro 3.1), entonces dicho nodo propondrá a los demás nodos sensor un cambio de estado de alerta, sin embargo ese dato deberá ser validado por medio de comparación con sus propios valores por el resto de nodos y sentenciado por el nodo maestro, quien será al final quien determine si se pronuncia un nuevo estado de alerta al resto de la(s) red(es) cooperativa(s) a las que pertenece dicho puente, o por el contrario reporta al

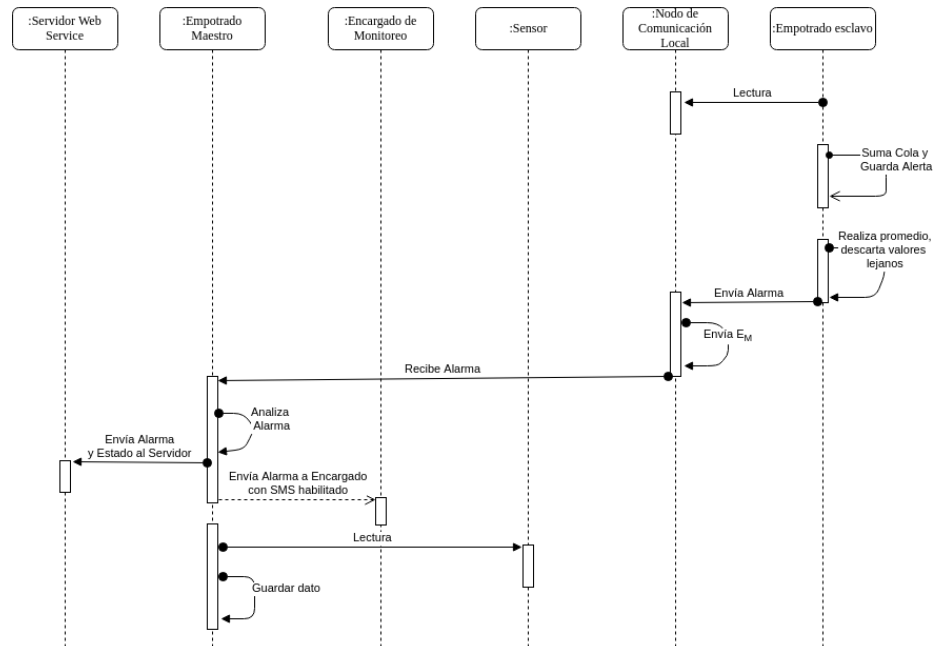


Figura 3.8: Diagrama de secuencia para la gestión de alarmas.

servidor web que persiste un falso positivo en el nodo que reporta un dato diferente al resto de nodos de la estructura puente.

En el diagrama de la figura 3.9 se observa que los casos de “Reporta Alarma a Maestro” y “Lectura de Variable” son los que le permiten a un nodo esclavo determinar que debe de reportar una alarma al maestro (cumpliendo con la premisa de realizar algún nivel de procesamiento en los nodos establecida en el análisis del estado del arte). Mientras que el caso de “Analiza Alarma” es quien decidirá si debe enviar un mensaje que actualice tanto al servidor como a sus nodos maestros vecinos. La notificación vía mensaje de texto (*SMS*) queda sujeta al modo de operación del empotrado, será posible notificar si se trabaja a bajo nivel, es decir con comandos AT, este escenario de prueba será profundizado en capítulos posteriores.

3.2.4. Administración de la configuración

Cada nodo que sea añadido a la red requiere de una pre-configuración antes de su instalación, pero además puede ser actualizado por parte del usuario mediante la interfaz Web, dentro de los parámetros que pueden ser modificados se encuentran: el puente al que pertenece un nodo, el rol del nodo (maestro-esclavo), el tiempo de muestreo, el tipo de variable a medir y las redes colaborativas a las que el nodo pertenece. Se ha dotado a su vez al sistema con una funcionalidad de auto-medición de la intensidad de señal telefónica por parte de cada nodo sensor, proveyendo al encargado de monitorización o encargado de puentes con una herramienta muy útil para la instalación y ubicación a lo largo de una estructura puente para los nodos sensor.

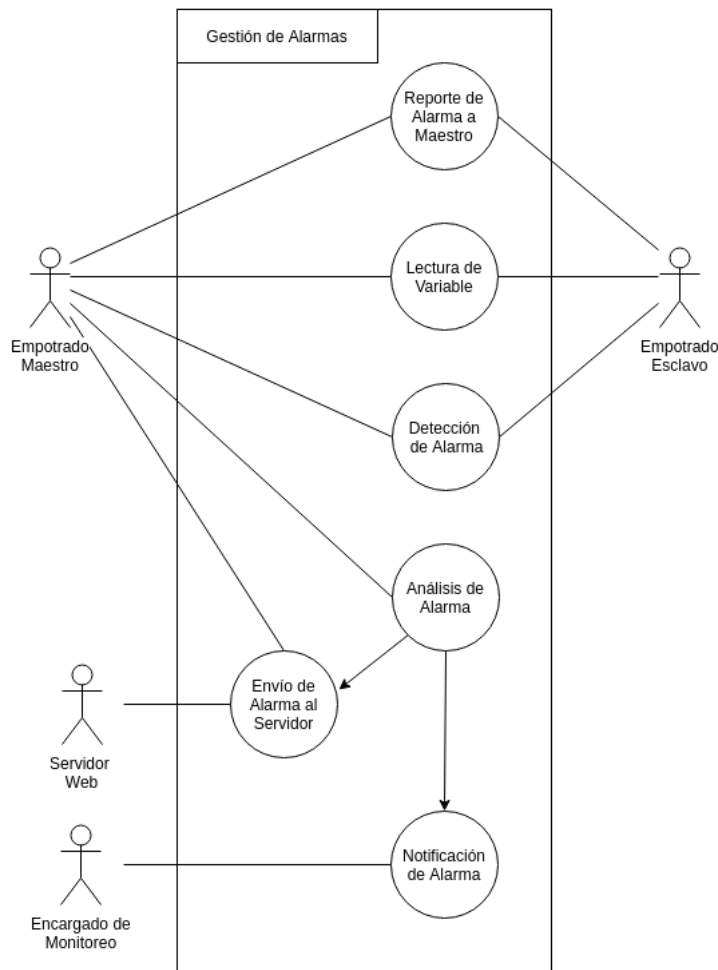


Figura 3.9: Casos de uso de la gestión de alarmas.

En la figura 3.10 se muestra el diagrama de casos de uso para realizar el manejo de la configuración de la red, donde se incluye nodos, puentes y redes colaborativas. El caso Petición Web es el caso que permite acceder a la base datos, ya sea para hacer un cambio en la configuración por parte del encargado de puentes o del encargado de monitorización o para notificar a los nodos acerca de cambios en la configuración. Por otro lado el caso de Notificación de Cambio es quien indica a los nodos sensor que deben de recibir una actualización para modificar su configuración.

3.2.4.1. Determinación de nodo maestro en una estructura puente

Es común que en una misma estructura puente se requiera de más de un punto de toma de datos, donde probablemente algunos de ellos no son buenos puntos para conectarse a la red telefónica. Es posible realizar una verificación del nivel de la señal que determine cual de las unidades está mejor ubicada para convertirse en el punto

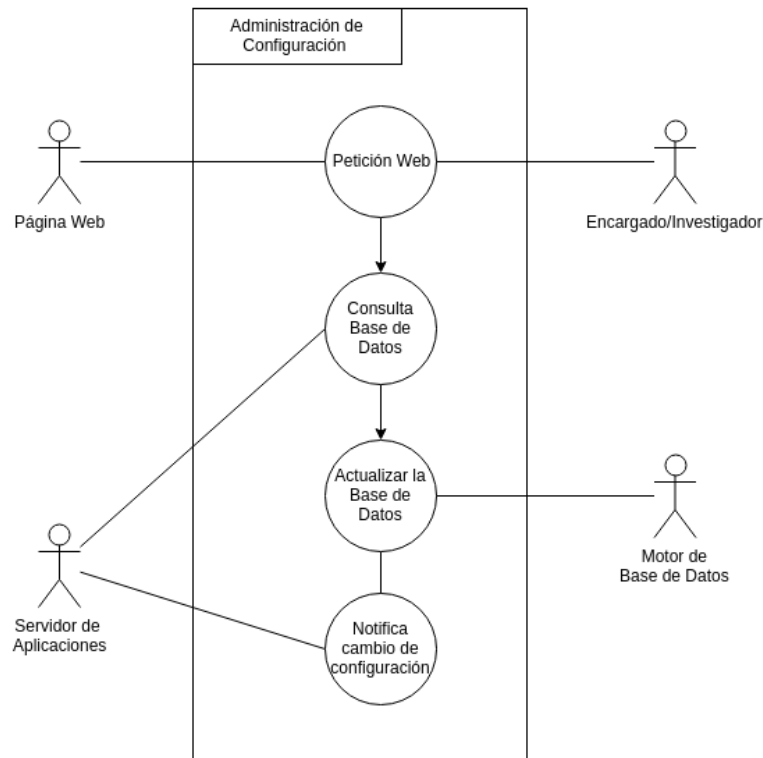


Figura 3.10: Casos de uso para la administración de la configuración.

de integración a la red, sin embargo dicha funcionalidad esta limitada al número de entradas/salidas del empotrado, si el puerto utilizado para adquirir internet cubre las disponibles, entonces no será posible medir la intensidad simultáneamente.

En el caso de estar trabajando como servidor, no hay acceso al control de bajo nivel del hardware (Comandos AT para un Raspberry), entonces deberá limitarse a una asignación manual y aleatoria del dispositivo maestro en una sola estructura, alternativamente puede realizarse la medición con el empotrado funcionando sin ser servidor para saber cual posición esta mejor ubicada y posteriormente hacer una reconfiguración del dispositivo en campo. La figura 3.11 muestra el diagrama de casos de uso de la determinación del nodo máster por medio de medición de señal de conexión a la red telefónica.

En el diagrama de secuencia de la figura 3.12 se despliega la asignación automática del nodo maestro en una estructura puente, acción que se realiza por medio de la medición de la intensidad de la señal percibida por los dispositivos. La asignación automática del nodo maestro utiliza al mejor postor.

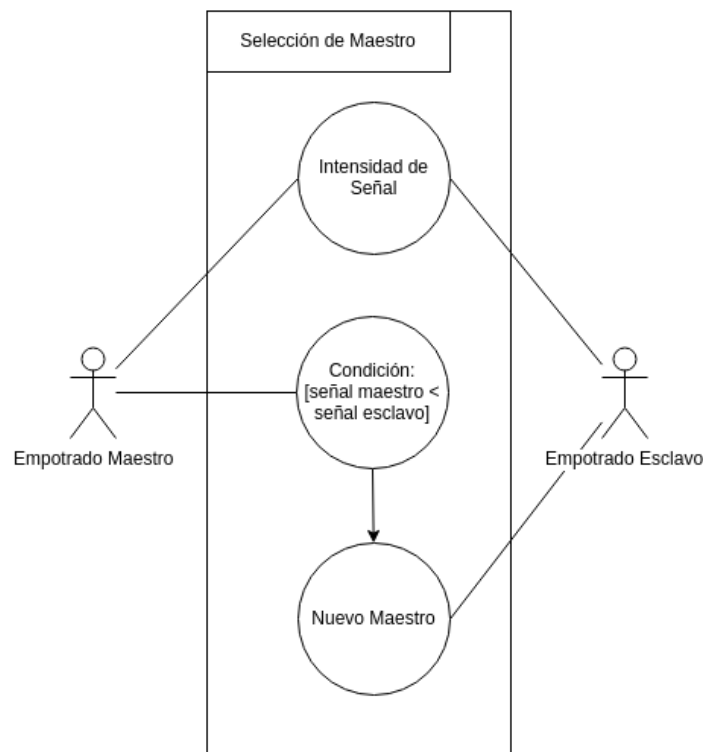


Figura 3.11: Casos de uso para selección del nodo de enlace a la red colaborativa.

3.2.5. Análisis de los datos del sistema

La autonomía por estructura puente que ofrece la filosofía cooperativa de la red del sistema, permite tomar acciones automáticas con variables físicas como el nivel de agua del puente por ejemplo cerrar una aguja de paso o controlar un semáforo de acceso al puente (no implementado en el prototipo); sin embargo uno de los requerimientos más importantes establecidos y que además es conocido que debe de estar disponible para sistemas *SHM*, es que sea posible realizar análisis de datos, ya que existen variables físicas que generan alteraciones en estructuras a mediano y largo plazo, como por ejemplo la vibración.

El encargado de monitorización, es el actor más importante en el análisis de datos, ya que es quien hace uso de las herramientas que provee el sistema. En la figura 3.13, se puede observar que el encargado utiliza el caso "Petición Web" para interactuar con la estructura de despliegue de datos, que le permite hacer análisis de los mismos. Por medio de la página web y "Consulta HTTP" se utiliza el servidor de aplicaciones para consultar la base de datos. Algunas herramientas como gráficos, o selecciones inteligentes de datos, podrían implementarse como parte de la interfaz Web actual, en el caso de necesitar los datos para su uso en Excel, se cuenta con una herramienta de exportación del *Raw Data* en dicho formato.

La secuencia para realizar una petición web y obtener información desde la base de datos por medio del motor de la misma es mostrada en la figura 3.14. Una peti-

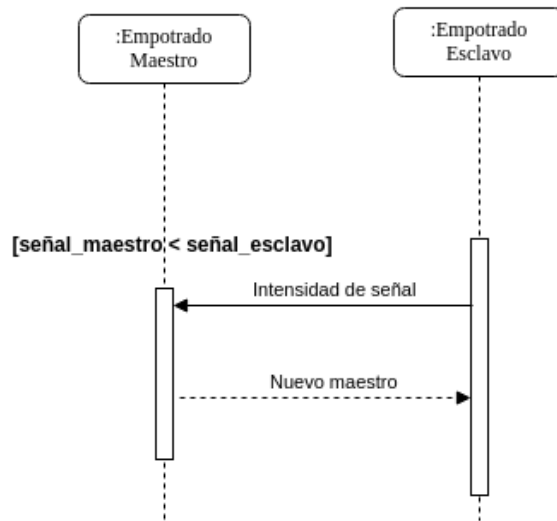


Figura 3.12: Diagrama de secuencia de la selección del nodo de enlace a la red colaborativa.

ción web es trasladada a una consulta *HTTP* por parte de la página web, que a su vez interactúa con el servidor de aplicaciones para hallar el dato solicitado ó actualizar uno existente en la base de datos.

3.3. Modelo de datos

El motor de la base de datos soporta y contiene a la base de datos del sistema, cuya estructura es mostrada en la figura 3.15.

En la tabla *Devices* se almacenan los datos acerca de nodos sensor, donde se almacena además que tipo de nodo sensor es cada uno; esclavo o maestro, así como una relación a los respectivos archivos de configuración, ubicados en la tabla *Configurations*, tales datos pueden ser modificados por el usuario mediante la interfaz web, dando entonces una gran flexibilidad y adaptabilidad a cualquier cambio requerido en la toma de datos o situación de emergencia, tal y como se estipula en la sesión de requerimientos para sistemas basados en *IoT*.

La independencia de la red local, que a su vez es colaborativa también, independiza el funcionamiento del puente en caso de perder comunicación con el resto de la(s) red(es) colaborativa(s), una re-conexión del nodo a la red permite la actualización de los datos que son almacenados en la tabla *Reports*, misma tabla donde es almacenada la información regularmente en comunicación constante con la red de sensores. Los datos relacionados con puentes se almacenana en la tabla *Structures*.

Los datos utilizados en la lógica de redes colaborativas residen en las tablas *Nets* y *DeviceInNets*. En la primera se definen las redes existentes y en la segunda se definen las asociaciones de pertenencia de los nodos en cada red colaborativa, toda configuración almacenada en estas tablas puede ser ingresada y modificada por medio

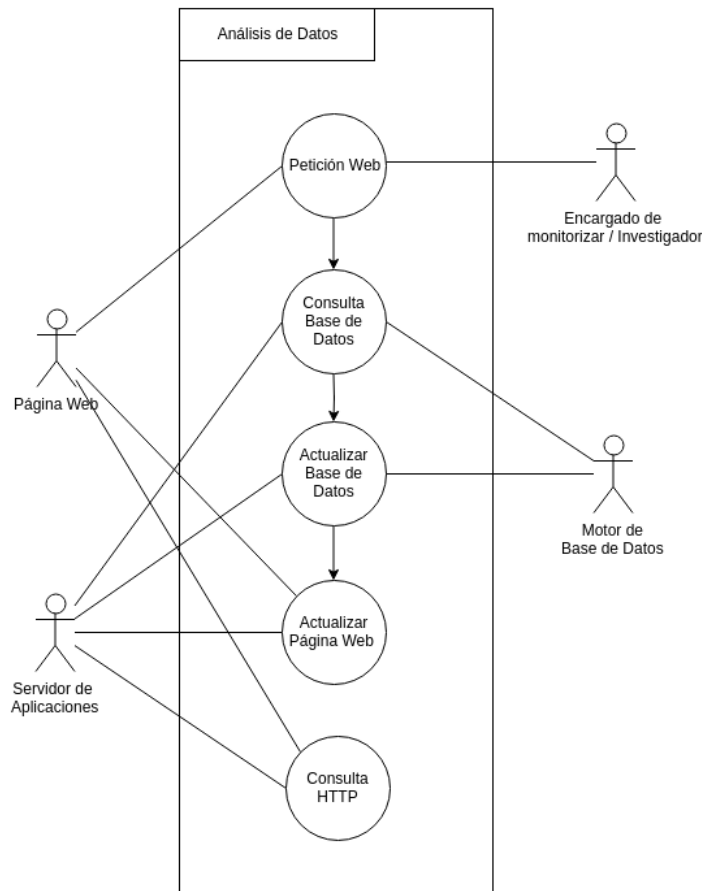


Figura 3.13: Casos de uso para el análisis de los datos.

de la interfaz gráfica. Para su consulta se pre-definen algunas consultas que le darán al usuario una forma amigable de saber cual es la configuración actual.

Las estructuras de datos dados en *Routes* y *Basins* son dadas para la recolección de información que puede ser importante para investigación por parte de los encargados de puentes y de monitorización.

3.3.1. Mediciones

En la Figura 3.16 se muestra la estructura de la tabla de mediciones, donde se muestran los datos tomados por todos los nodos sensor, sin importar el tipo de nodo sensor que sea.

Cada columna de la tabla contiene un dato diferente de cada medición. Iniciando de izquierda a derecha, "Reporte" organiza los datos del más reciente al más antiguo, "Dispositivo" y "Puente" identifican al nodo sensor y al puente al que pertenece la medición respectivamente, posteriormente se dan la fecha y hora de la toma del dato, "Distancia" indica el valor medido del sensor que en este caso es distancia y está

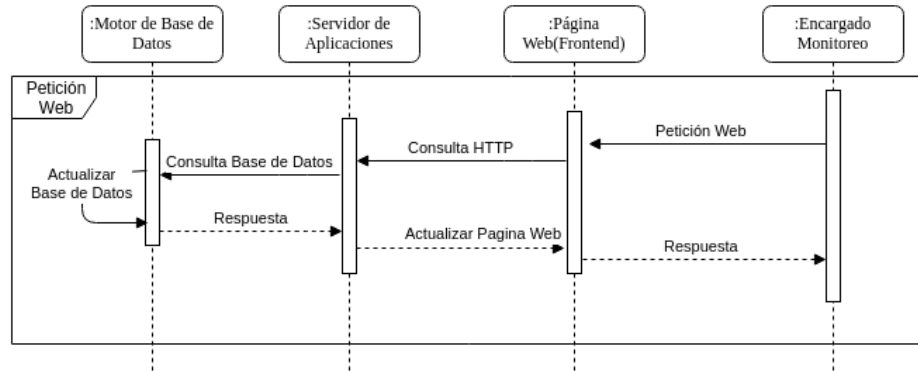


Figura 3.14: Diagrama de secuencia para el análisis de los datos del sistema.

dado en centímetros, “Alerta” indica el estado del puente reportado de acuerdo al criterio establecido por criterio experto, “Alerta Global” indica el estado de alarma de la red colaborativa correspondiente y finalmente uno de los datos que revela buena información en referencia al desempeño de la red es el “Tiempo de propagación”, que determina el momento en el que el nodo sensor hace su cambio de estado, ya sea por inducción de la red colaborativa a la que pertenece o por sus propias mediciones.

3.4. Requerimientos técnicos de componentes del prototipo

Los componentes del prototipo se subdividen en cinco secciones: sensor, empotrado, módulo de conexión local, módulo de conexión a la red global y servidor concentrador de datos. A continuación se presentan los requerimientos ordenados bajo esta división.

3.4.1. Sensor

El sensor debe ser un dispositivo que permita medir la variable física de distancia a bajo costo. Además debe permitir la corrección de malas mediciones por medio de software, mediante técnicas estadísticas de promedio y comparación contra valores previos. La ganancia se debe poder ajustar y al igual que su rango de ubicación. Así, se puede encontrar una ganancia que pueda medir distancias que cubran el 75% de diez metros, que es la distancia máxima establecida para las estructuras a evaluar con el prototipo.

3.4.2. Empotrado

Se escoge el sistema empotrado basándose en las siguientes métricas de diseño:

- Bajo consumo de potencia

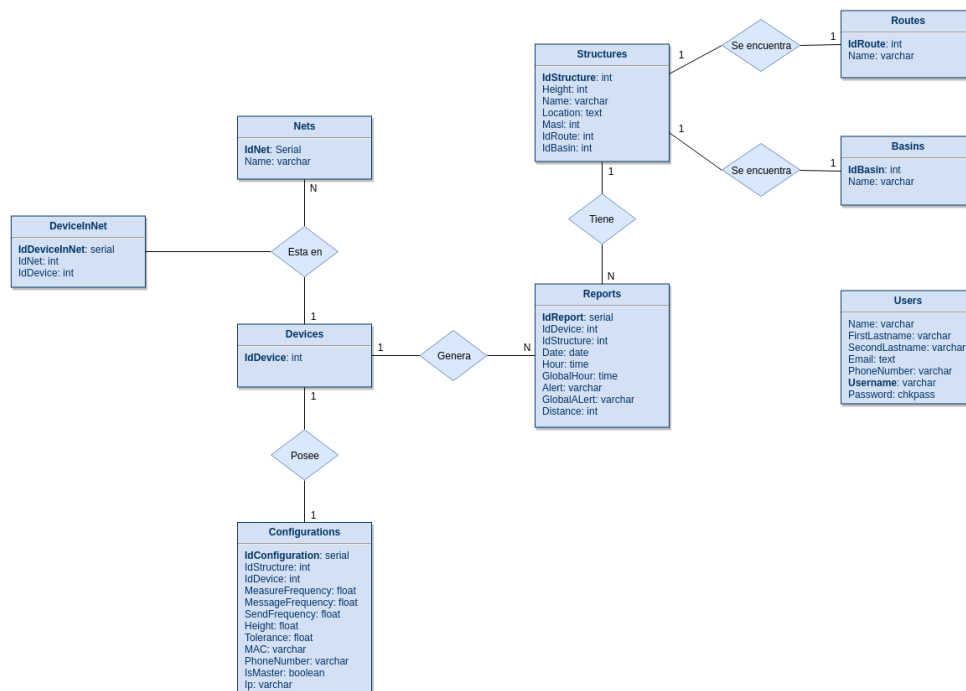


Figura 3.15: Estructura del esquema de la base de datos del sistema.

- Capacidad para USB
- Capacidad para HDMI o VGA
- Rendimiento del procesador
- Capacidad de conectividad
- Documentación
- Licencia del hardware
- Soporte para el hardware
- SO soportado
- Aspectos diferenciadores
- Limitaciones
- Disponibilidad de puertos
- Conexión a internet por Ethernet o Wifi

Tabla de Mediciones								
Contiene la información recopilada en los puentes								
Reporte	Dispositivo	Puente	Hora	Fecha	Distancia	Alerta	Alerta Global	Hora Propagacion
1933	22	tec1	12:05:49	28-03-19	31	Verde	Verde	11:03:46
1932	22	tec1	12:05:31	28-03-19	30	Verde	Verde	11:03:46
1931	22	tec1	12:05:12	28-03-19	31	Verde	Verde	11:03:46
1930	22	tec1	12:04:54	28-03-19	31	Verde	Verde	11:03:46
1929	22	tec1	12:04:30	28-03-19	31	Verde	Verde	11:03:46
1928	20	tec1	12:04:07	28-03-19	310	Falso Positivo	Verde	11:59:12
1927	22	tec1	12:04:20	28-03-19	31	Verde	Verde	11:03:46
1926	22	tec1	12:04:10	28-03-19	31	Verde	Verde	11:03:46
1925	22	tec1	12:03:51	28-03-19	31	Verde	Verde	11:03:46
1924	22	tec1	12:03:33	28-03-19	31	Verde	Verde	11:03:46

Figura 3.16: Tabla de Mediciones.

3.4.3. Módulo de conexión local

Se debe utilizar alguna tecnología de corto alcance que permita la conexión punto a punto, en la cual la información es transmitida desde un nodo sensor a otro configurado como coordinador. La conexión entre ambos dispositivos se debe realizar ajustando algún parámetro de identificación, tal como la dirección MAC, donde el coordinador se conozca en la configuración.

3.4.4. Módulo de conexión a la red global

Este módulo debe de ser un protocolo de comunicación de nivel 3, de acuerdo al modelo interconexión de sistemas abiertos por capas de OSI, que permita la conexión a la red telefónica para comunicar estructuras que se encuentren muy lejanas entre sí y que no pueden ser cubiertas por tecnologías de corto alcance, además de no generar dependencia de alguna estructura de comunicación en radio frecuencia que esta fuera del alcance del proyecto de investigación.

3.4.5. Servidor

El servidor físico es donde residen: el Servidor de Aplicaciones, el Servidor Web y el concentrador datos, para lo cual se establece que los requerimientos mínimos son los siguientes:

- SO soportado en plataforma Linux
- Capacidad para USB
- Capacidad para HDMI o VGA
- Varios Cores
- RAM mayor a 8GB

- HD no menor a 128GB
- Licencia del hardware
- Soporte
- Máquina virtual o física
- Conexión a internet por Ethernet o Wifi

3.4.6. Red inalámbrica

Se considera red inalámbrica a toda tecnología que tenga como función la transferencia de datos que utilice la red colaborativa. Tanto la comunicación punto a punto de una estructura puente, como la colaborativa global requieren del envío de información relacionada al protocolo de prevención de desastres. La red utiliza tecnología de corto alcance hasta donde es posible, sin embargo el trasiego de datos entre estructuras es posible con la estructura actual por medio de la red de telefonía celular, por lo que se requiere que dicha red forme parte de la red inalámbrica del prototipo, permitiendo aprovechar el acceso a internet para la implementación de una estructura RESTful.

Capítulo 4

Diseño

En este capítulo se describe la fase de diseño del prototipo de red de sensores y el servidor de recolección de datos; el servidor recibe reportes de los nodos sensor maestro de cada estructura puente, o los de la red colaborativa a la que pertenece. Esta fase tiene como fin definir el diseño del prototipo antes de su desarrollo.

Un proceso de investigación exhaustivo y de pruebas preliminares ha permitido realizar la implementación de un sistema mínimo de red colaborativa que cumple con las especificaciones para *SHM* y redes inalámbricas de sensores, con la intención de sobrellevar las situaciones de emergencia en estructuras tipo puente.

La fase de diseño tiene como propósito definir los planos de construcción del sistema de sensores, el protocolo de comunicación entre ellos y el envío de datos hacia la estructura en la nube del servidor principal. Es muy importante que el diseño esté bien definido, logrando cubrir las necesidades actuales y futuras de los entes relacionados, que al final son los usuarios del entorno. Esta fase es de gran impacto y determina la efectividad de un sistema como producto final del diseño.

La composición de este capítulo está basada en:

- Diseño de la arquitectura de la red.
- Diseño del protocolo.
- Principales funciones y algoritmos.
- Configuración del sistema.
- Wireframe de la página principal.
- Mapa o plano general del sitio.

4.1. Diseño de la arquitectura de la red

La arquitectura mostrada en la figura 4.1 muestra el diagrama de diseño general de la red de sensores, donde se aprecia que pueden darse diversas configuraciones de nodo sensor en cada estructura puente. La cantidad de nodos sensor en un puente es la que determina su configuración local. A nivel de semántica de la red, se tiene una combinación entre dos modelos: cliente-servidor y sistema distribuido.

En la red todo nodo sensor es un cliente que realiza peticiones al servidor en la nube, el servidor las procesa y luego envía la respuesta al cliente; las peticiones pueden ser con el objetivo de obtener actualizaciones para su configuración o bien de las redes colaborativas; estructura que se replica entre nodos sensor maestros, para compartir estados de alerta.

Periódicamente, cada nodo sensor maestro realiza peticiones para actualizar datos de mediciones de cada puente, en la base de datos del servidor en la nube. Los conceptos de diseño del nodo sensor maestro y nodo sensor esclavo, se desarrollan durante el capítulo con una mayor profundidad.

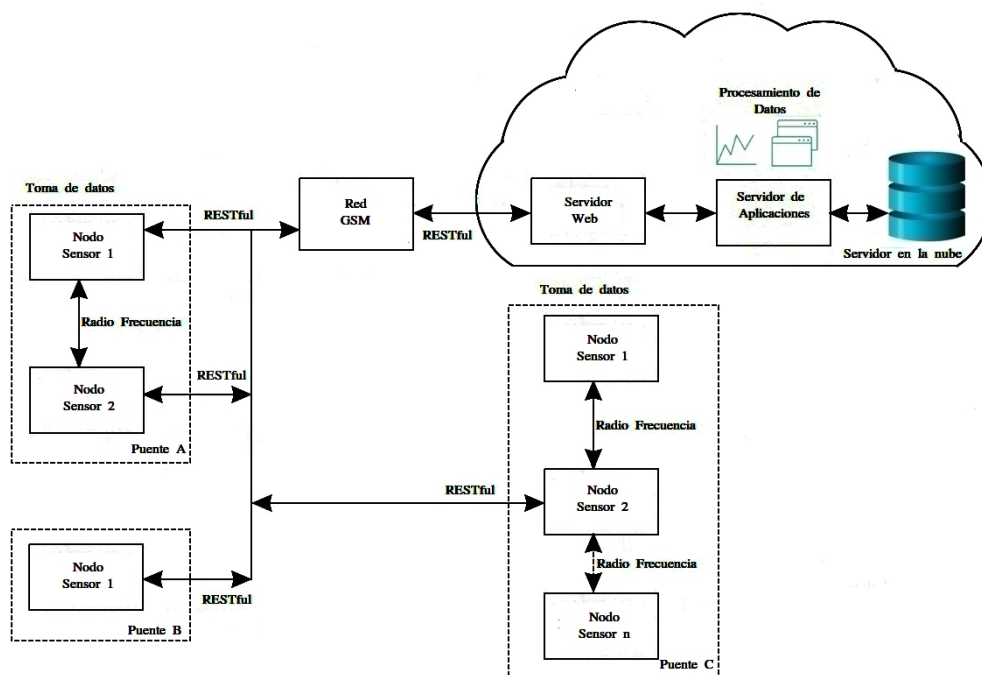


Figura 4.1: Diseño general de la red.

La filosofía de red colaborativa en esta aplicación, se plasma en la colaboración que realizan los nodos sensor entre sí, la cual se efectúa de dos formas (local y global), tal y como se ha mencionado en capítulos anteriores. Cada nodo sensor, en un mismo puente, colabora entre sí para determinar un valor y estado general, cuya interacción da como resultado un estado de alerta general confiable.

Por otro lado, cada nodo sensor definido como maestro comparte su estado general de alerta con los nodos sensor maestro de otros puentes, para definir así el estado de alerta global de la red colaborativa, todo por medio de la estructura *RESTful*.

La aplicación es de ejecución continua y los datos en tiempo real pueden ser visualizados a través de un navegador web, así como un cambio de configuración puede ser efectuado también por medio de la interfaz. Otro factor importante dentro de la arquitectura es la versatilidad del servidor en la nube, al cual se le puede distribuir aun más y a quien el uso de una estructura *RESTful* faculta con una escalabilidad e interoperabilidad amplia.

A continuación se presenta con más detalle cada uno de los componentes de la

arquitectura de la red de sensores de la figura 4.1:

4.1.1. Configuraciones de puentes

La representación del “puente A” de la figura 4.1, muestra una de las posibilidades de configuración que se vislumbra como la opción a utilizar mayoritariamente, es decir, con un nodo maestro y un nodo esclavo; lo anterior debido a que se ha notado que su interacción es suficiente para determinar un estado de alerta.

En el mejor de los casos, todo nodo sensor, sin importar su rol, podría tener acceso a la red global para solicitar actualizaciones para sí mismo al servidor en la nube, por lo que la configuración del “puente A” podría tener la ventaja de poder realizar actualizaciones de configuraciones por medio de cualquiera de sus nodos; a su vez presenta la desventaja de ser más costoso que una configuración donde no todos los nodos tengan dicha capacidad.

Una comparación entre los resultados de un “Filtro de mediana móvil” mitiga la necesidad de instalar más de dos unidades por estructura puente para obtener una medición aceptable en términos de precisión, y a su vez seguir siendo capaces de detectar “falsos positivos” con dos unidades (comunicación local Xbee).

La configuración de “puente B” y la de “puente C”(figura 4.1) son posibles pero menos recomendadas. En orden de prioridad, se prefiere tener la configuración plasmada en “puente C”(dos esclavos y un maestro) ya que permite implementar detección de “falsos positivos”. La configuración del “puente B” (un solo nodo maestro) es la de mejor esfuerzo,

4.1.2. Arquitectura del servidor en la nube

El servidor permite el correcto funcionamiento de la red de la figura 4.1, ya que además de almacenar los datos crudos, actúa como canal de agregación a la red global. En él, también reside la interfaz web que permite que los usuarios generen sus credenciales de sistema, visualicen los datos y realicen ajustes a redes colaborativas existentes o creen nuevas redes colaborativas.

Cabe resaltar que la interfaz web maneja concurrencia de usuarios, las peticiones o acciones de cada usuario se verán reflejadas en la red en función del tiempo de propagación promedio para servidor-nodo, que será definido en el capítulo de validación.

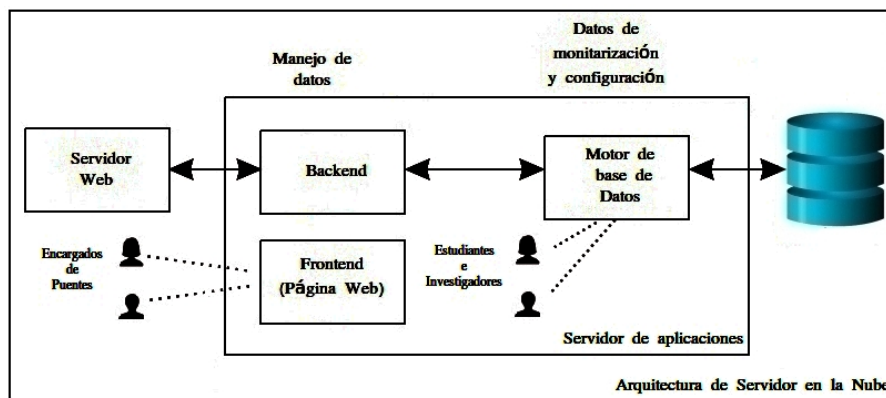


Figura 4.2: Servidor en la nube.

En la Figura 4.2 se muestra el diseño del servidor en la nube, cuya arquitectura tiene como requisito ser simple, de código abierto y extensible. Para cumplir con estos requerimientos se utiliza el entorno de ejecución de JavaScript, Node.js, para las secuencias del lado del *Back End* del servidor de aplicaciones, integradas posteriormente del lado del cliente en el navegador web, que constituye parte del *Front End*.

El *Front End* es manejado por medio del *framework Express.js*, que es diseñado para aplicaciones web y creación de APIs. Se plantea la creación de un API mediante *framework RESTful* en el servidor de la nube que permita relacionar datos con los nodos sensor. La estructura *RESTful* es lo suficientemente escalable y de desarrollo en Java.

En el servidor residen los datos, las configuración y parte del protocolo distribuido de comunicación. El último bloque de la arquitectura del servidor en la nube es el motor de base de datos, que se alinea a los requisitos planteados para todo el servidor. El sistema utilizado es PostgreSQL, que también es de código abierto y además es un sistema muy estable con más de tres décadas de desarrollo activo y buen desempeño. Todos los datos crudos son almacenados por medio del motor de PostgreSQL y se dividen en datos de mediciones y datos de configuración de la red.

4.1.3. Nodo sensor de medición y conexión a la red

En la figura 4.3 se muestra la arquitectura de un nodo sensor estándar como componente más básico de la arquitectura de la figura 4.1, cuya configuración determina su rol en las redes colaborativas.

La configuración del nodo sensor en la interfaz gráfica, cuenta con un *checkbox* que se marca para definirlo como nodo maestro, lo que significa que el nodo que sea definido como tal, será el encargado de transferir los datos de todos los demás nodos sensor de un puente, al servidor de aplicaciones.

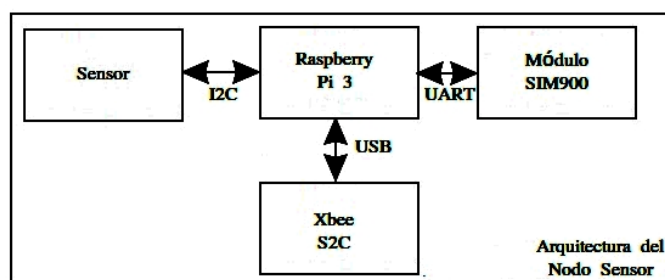


Figura 4.3: Nodo sensor Maestro-Esclavo.

Un nodo sensor maestro actúa como concentrador de datos en un puente, lo que le permite determinar el estado de alarma del puente y mediante el módulo SIM900 realiza la transferencia del mensaje a sus redes colaborativas. Solamente se define un nodo sensor como maestro por estructura puente, mientras que es posible definir más de un nodo esclavo por estructura; un nodo esclavo también debe ejecutar el muestreo de datos y además debe transferir sus datos de medición al nodo maestro por medio del módulo Xbee S2C.

4.2. Diseño del protocolo

En esta sección se presentan los detalles de diseño de la red colaborativa de sensores. Inicialmente se describe la estructura de los datos, posteriormente se desarrolla el API creado para la red de sensores e-Bridge, en la siguiente sub-sección se describe la composición del hardware a nivel de protocolos y su mensajería, finalizando con el protocolo que se encarga de manejar las alertas.

4.2.1. Estructura de datos

La forma en que se organizan los datos ha sido considerada dentro de la etapa de diseño también debido a los requerimientos del sistema, el principal requerimiento es el de la capacidad de manejar relaciones entre grupos de puentes, razón por la cual se incorpora el concepto de redes colaborativas globales.

A nivel de base de datos se manejan asociaciones de puentes por medio de redes colaborativas. Existe una configuración tanto de puentes como de dispositivos, que permiten su manejo como entes independientes que pueden o no tener relación a una red colaborativa.

Por la naturaleza de la investigación, se incluyen tablas para reportes, que son esenciales para el análisis de los datos.

A continuación se describen los modelos conceptual y de entidad-relación de la base de datos utilizada.

4.2.1.1. Modelo Conceptual de la base de datos

El diseño conceptual parte de las especificaciones de requisitos de usuario, resultando en el esquema conceptual de la base de datos. El diagrama conceptual es una descripción de alto nivel de la estructura de la base de datos, a este nivel no

importa cual motor de base de datos se utilice, se debe mantener la estructura de los datos. En esta sección se ilustran y comentan únicamente las entidades o tablas principales.

En la figura 4.4 se muestra el esquema conceptual del sistema como un conjunto de representaciones lingüísticas y gráficas tomadas a partir del diagrama del esquema entidad-relación dado en la figura 4.5.

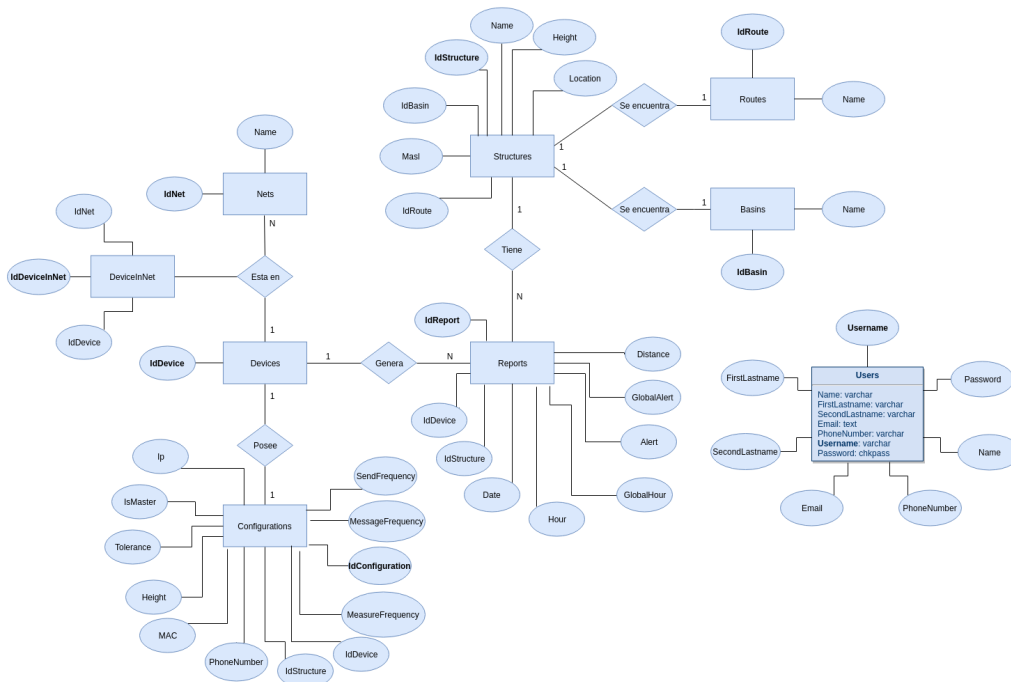


Figura 4.4: Diagrama conceptual de la base de datos.

Toda tabla se representa en forma de bloques, cuyos atributos se representan por óvalos y su relación de contenido mediante rombos. Es notorio como la tabla *Device* es una de las entidades más importantes, puesto que entrelaza mediante alguna relación a *Configurations*, *Nets* y *Reports*.

4.2.1.2. Modelo Entidad-Relación de la base de datos

Todos los datos del sistema son representados en tablas, tal y como se muestra en el diagrama relacional de la base de datos de la figura 4.5. *DeviceinNet* contiene las relaciones de redes y dispositivos que pertenecen a una red colaborativa definida, la relación entre tablas se establece por medio de *IdNet* e *IdDevice*. En esta sección se ilustra y comenta a los entes más importantes únicamente.

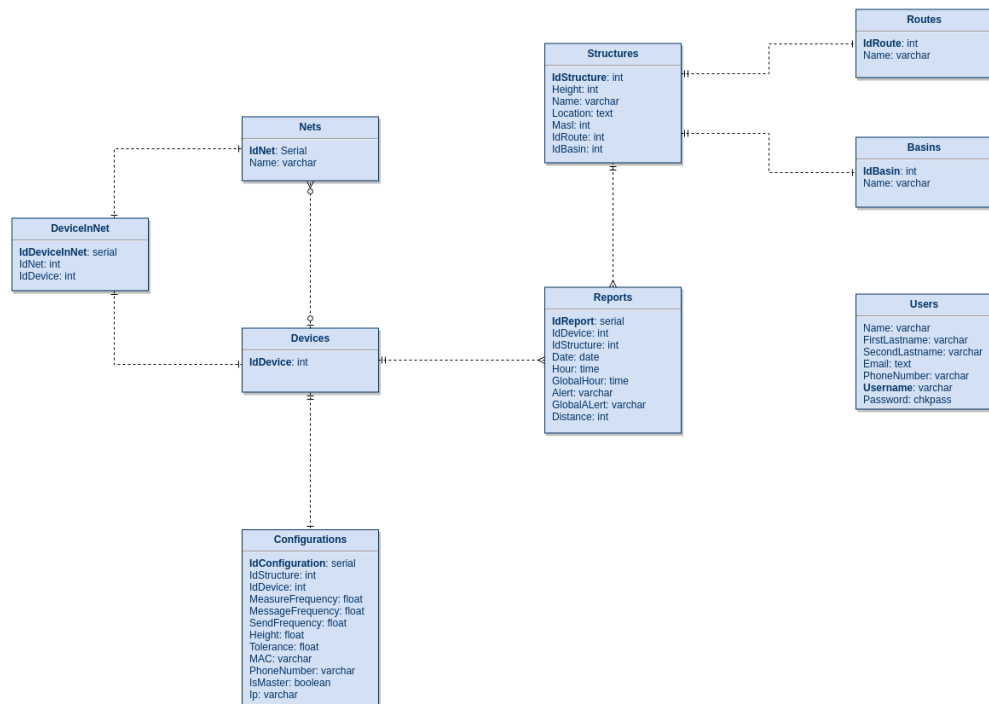


Figura 4.5: Diagrama relacional de la base de datos - Notación Crowsfoot.

Configurations almacena los datos ingresados por el usuario para determinar la forma de ingresar tanto estructuras puente como dispositivos, la entidad relación se hace por medio de *IdStructure* e *IdDevice*. Por otro lado *Reports* utiliza las mismas relaciones para la extracción de información respecto a la operación de dispositivos y mediciones. *Users* es la única tabla sin relación con los datos de la red, en vista de que es la que almacena credenciales para ingresar a la interfaz web.

4.2.2. API e-Bridge

Como parte de la definición del protocolo de red, se ha definido un API con el objetivo de que las aplicaciones definidas accedan a las funciones o datos y así estandarizar todo tipo de acción. En el proyecto se sigue el modelo general post/get/put/delete de webservices, basado en una estructura *RESTful*.

Cada módulo tiene una función asociada en Javascript, que funciona como una interfaz de relación ente el API ebridge diseñado para este proyecto y cada acción es manejada como un reporte a nivel de base de datos. Si un cliente solicita actualizar sus mediciones al servidor, se maneja como un reporte.

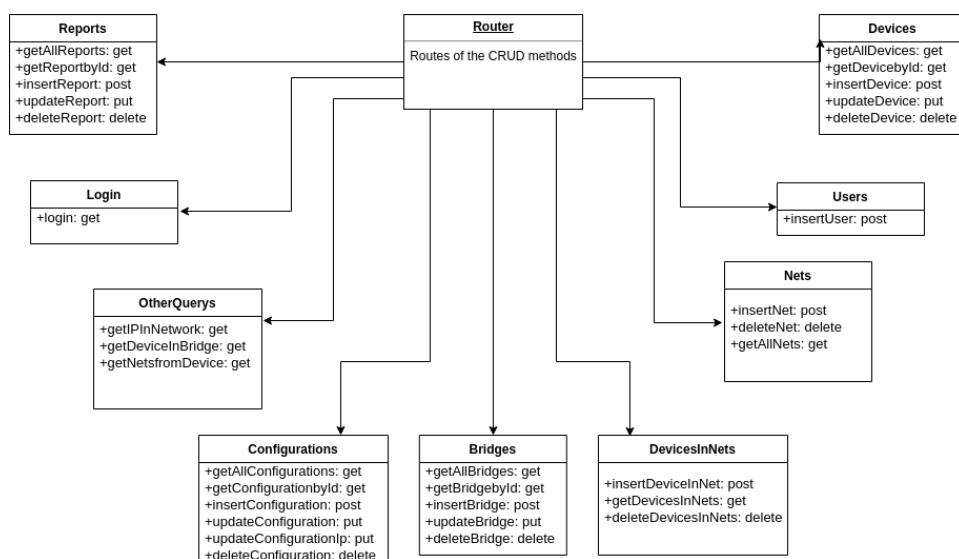


Figura 4.6: Diagrama de módulos del API.

En el diagrama de módulos de la figura 4.6, se muestra la estructura del API desarrollado durante este proyecto, cada módulo muestra las acciones CRUD que es posible ejecutar con el módulo, por ejemplo en *Configuration* es posible realizar, toda acción (POST, GET, PUT, DELETE).

En las siguientes sub-secciones se detallan los módulos más importantes del diagrama de la figura 4.6, entre las cuales están: petición y respuesta del API por login, registro, configuración, puente, dispositivo y reporte.

4.2.2.1. Login

Cada usuario requiere identificarse para entrar a la interfaz web, para lo cual se ha definido tanto una forma estándar de solicitud de entrada como una respuesta en el API. A continuación se desarrolla la parte de Petición en el *Login*.

Cuadro 4.1: Método y URL en el login

Método	URL
POST	api/ebridge/login/

Cuadro 4.2: Parámetros por método en el Login

Tipo	Parámetros	Valores
POST	username	string
POST	password	string

Respuesta en el *Login*.

Cuadro 4.3: Respuesta por código en el Login

Código	Tipo	Respuesta
200	POST	{ "status": 'success', "data": <auth_key>, "message": 'Login successful' }
400	POST	{"error": 'Please provide username.'}
400	POST	{"error": 'Please provide password.'}
401	POST	{"error": 'Incorrect username or password.'}

4.2.2.2. Registro

El sistema permite la creación de usuarios nuevos mediante un registro, la definición de una forma estándar de solicitud de registro como una respuesta en el API se presentan a continuación. Petición en el Registro:

Cuadro 4.4: Método y URL en el Registro

Método	URL
POST	api/ebridge/user/

Cuadro 4.5: Parámetros por método en el Registro

Tipo	Parámetros	Valores
POST	name	string
POST	firstLastName	string
POST	secondLastName	string
POST	email	string
POST	phone	string
POST	username	string
POST	password	string

Respuesta en el Registro:

Cuadro 4.6: Respuesta por código en el Registro

Código	Tipo	Respuesta
200	POST	{ "status": "success", "message": "Inserted one user" }
400	POST	{"error": "Incomplete data."}

4.2.2.3. Configuración

Todo dispositivo, red colaborativa y puente, debe de ser previamente configurada en el sistema para poder operar correctamente dentro del protocolo de la red, petición y respuesta también han sido contemplados dentro del API. A continuación se desarrolla petición en la Configuración:

Cuadro 4.7: Método y URL en la Configuración

Método	URL
GET	api/ebridge/ configuration/
POST	api/ebridge/ configuration/
PUT	api/ebridge/ configuration/

Cuadro 4.8: Parámetros por método en la Configuración

Tipo	Parámetros	Valores
POST/PUT/GET	device	integer
POST/PUT/GET	bridge	integer
POST/PUT/GET	measureFrequency	float
POST/PUT/GET	messageFrequency	float
POST/PUT/GET	sendFrequency	float
POST/PUT/GET	height	float
POST/PUT/GET	tolerance	float

Respuesta en la Configuración:

Cuadro 4.9: Respuesta por código en la Configuración

Código	Tipo	Respuesta
200	GET	{ "status": "success", "data": <configuration>, "message": "Configuration retrieved" }
200	POST	{ "status": "success", "message": "Configuration inserted" }
200	PUT	{ "status": "success", "message": "Configuration updated" }
400	GET/PUT/POST	{ "error": "Unknown device.", }
401	PUT	{ "error": "Invalid data.", }
401	POST	{ "error": "Invalid data.", }

4.2.2.4. Puentes

Cada puente debe de ser identificado como único en el sistema para poder operar correctamente dentro del protocolo de la red, petición y respuesta también han sido contemplados dentro del API. En los cuadros 4.10 y 4.11 se desarrolla petición.

Cuadro 4.10: Método y URL para Puentes

Método	URL
GET	api/ebridge/bridge/
POST	api/ebridge/bridge/

Cuadro 4.11: Parámetros por método para Puentes

Tipo	Parámetros	Valores
POST/GET	bridge	integer
POST/GET	name	string
POST/GET	height	float
POST/GET	location	string

La respuestas se desarrollan en el cuadro 4.12.

Cuadro 4.12: Respuesta por código para Puentes

Código	Tipo	Respuesta
200	GET	{ "status": "success", "data": <bridge>, "message": "Bridge retrieved" }
200	POST	{ "status": "success", "message": "Bridge inserted" }
400	GET	{ "error": "Unknown bridge." }
401	POST	{ "error": "Invalid data." }

4.2.2.5. Dispositivo

La petición para dispositivos ha sido definida en los cuadros 4.13 y 4.14.

Cuadro 4.13: Método y URL para Dispositivos

Método	URL
GET	api/ebridge/device/
POST	api/ebridge/device/

Cuadro 4.14: Parámetros por método para Dispositivos

Tipo	Parámetros	Valores
POST/GET	device	integer
POST/GET	bridge	integer

La respuestas se desarrollan en en el cuadro 4.15.

Cuadro 4.15: Respuesta por código para dispositivos

Código	Tipo	Respuesta
200	GET	{ "status": "success", "data": <device>, "message": "Device retrieved" }
200	POST	{ "status": "success", "message": "Device inserted" }
400	GET	{ "error": "Unknown device." }
401	POST	{ "error": "Invalid data." }

4.2.2.6. Reportes

Un sistema de información para prevención de desastres debe de contar con un sistema de reportes, el API E-Bridge contempla la petición y respuesta para reportes. La petición se desarrolla en los cuadros 4.16 y 4.17.

Cuadro 4.16: Método y URL para Reportes

Método	URL
GET	api/ebridge/report/
POST	api/ebridge/report/

Cuadro 4.17: Parámetros por método para Reportes

Tipo	Parámetros	Valores
POST/GET	device	integer
POST/GET	bridge	integer
POST/GET	hour	datetime
POST/GET	date	datetime
POST/GET	state	string
POST/GET	distance	float

La respuestas se desarrollan en en el cuadro 4.18.

Cuadro 4.18: Respuesta por código para reportes

Código	Tipo	Respuesta
200	GET	{ "status": "success", "data": <report>, "message": "Report retrieved" }
200	POST	{ "status": "success", "message": "Report inserted" }
400	GET	{ "error": "Unknown report." }
401	POST	{ "error": "Invalid data." }

4.2.3. Protocolo y mensajería del Hardware

Esta sub-sección describe tanto la estructura del mensaje entre piezas de hardware como el protocolo utilizado.

4.2.3.1. Comunicación entre Raspberry Pi 3 y sensor SRF-08

El envío se realiza por medio de comunicación serial con el protocolo I^2C (*Inter-Integrated Circuit, I2C*), en la dirección 0x70. La transmisión de mensajes se realiza con el protocolo en su forma estándar, utilizando únicamente la dirección dada anteriormente, se sabe que inicia con una condición de *Start*, luego un bit de lectura-

escritura, dos *Data Frames* separados por bits de *acknowledge(ack/nack)* y un bit de parada para cerrar el mensaje.

La escritura al dispositivo se realiza por medio de un controlador que actualiza el valor del archivo `/dev/serial0` y es leído por el programa en cada medición.

4.2.3.2. Comunicación entre Raspberry Pi 3 y módulo xBee S2C

El envío entre estos elementos también se realiza por medio de comunicación serial, pero con el protocolo SPI (*Serial Peripheral Interface*, SPI por sus siglas en inglés). Dicho protocolo en su forma estándar, establece una red de un sólo máster y un solo esclavo de memoria a memoria para transmitir el mensaje, la transmisión es sincronizada por *SCLK(Serial Clcok)*.

La escritura al dispositivo se realiza por medio de un controlador que actualiza el valor del archivo `/dev/ttyUSB0` y es leído por el módulo xBee para enviarlo al dispositivo correspondiente. Para la comunicación entre módulos Xbee se utiliza el formato JSON con los mensajes que se muestran en el cuadro 4.19.

Cuadro 4.19: Parámetros para comunicación de hardware

Objetivo	Parámetros	Valores
Envío de Mediciones del dispositivo esclavo al maestro	type state hour date bridge device distance	integer string dateTime dateTime integer integer float
Consulta de cuál dispositivo tiene mejor señal móvil para ser el dispositivo maestro	type id signal	integer integer float
Envío a todos los dispositivos para indicar el nuevo maestro	type master	integer integer

4.2.4. Sistema de alertas

Una parte fundamental del protocolo de red colaborativa para prevención de desastres es la definición sobre como funciona el sistema de alertas, como se ha explicado en capítulos anteriores para esta investigación se ha definido “Verde”, “Amarillo” y “Rojo” como los tres tipos de estados de alarma, más solamente dos de ellos son alertas, es decir, “Amarillo” y “Rojo”. La alerta “Amarillo” en el caso de la variable nivel de agua bajo el puente es la primera alerta de que no se está en condiciones normales y fue definida en el capítulo de requerimientos en un 50% por criterio experto, por otro lado la alerta “Rojo” se disparará cuando el nivel del agua alcance un 75% del valor nominal de la estructura puente (Nivel del agua en condiciones normales).

Dicho lo anterior el protocolo define dos formas de propagación de alertas a través de la red de sensores, Las “Alertas Locales” y las “Alertas Globales”.

4.2.4.1. Alertas Locales

Se refieren al resultado de la interacción entre nodos sensor de un solo puente, donde de forma conjunta toman la decisión sobre cual es el estado de alerta de un puente, la decisión de cambio de estado debe de ser una decisión unánime, con algún porcentaje de tolerancia definido por criterio experto y que es configurable (ver sub-sección “Configuración de dispositivos”). En el caso de que la decisión no sea unánime, se valora la presencia de un falso positivo (ver sub-sección de “4.3.2. Falsos Positivos” en la sección de “4.3. Principales funciones y algoritmos”).

La decisión funciona como una función “AND” excepto para falsos positivos; es decir:

$$AlertaLocal = Puente_1 AND Puente_2 AND \dots AND Puente_n$$

Algunos escenarios de decisión entre nodos sensor, sin importar quien tiene el rol de maestro, muestran que son decisiones con unanimidad (ver cuadro 4.20).

Cuadro 4.20: Algunos escenarios del Protocolo para Alertas Locales

Nodo Sensor	Alerta de Nodo	Alerta de Puente
1 2 n	Verde Verde Verde	Verde
1 2 n	Amarillo Amarillo Amarillo	Amarillo
1 2 n	Rojo Rojo Rojo	Rojo
1 2 n	Verde Rojo Verde	Falso Positivo en 2 Verde

4.2.4.2. Alertas Globales

Para las alertas globales primero es necesario definir el concepto de red colaborativa global, que es un grupo de puentes determinado estratégicamente por criterio experto, como puentes que tienen alguna inferencia sobre los demás, ya sea por que se ubican en la misma ruta o son parte aguas abajo de la misma cuenca tal y como se ilustra en el caso hipotético de la figura 4.7.

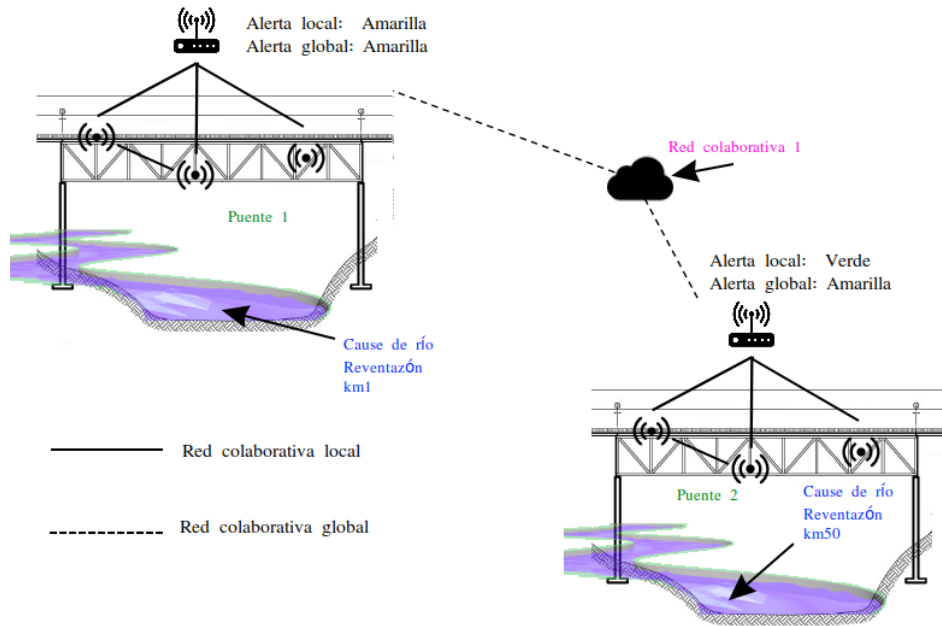


Figura 4.7: Ejemplo de alerta global.

Puede existir más de una red colaborativa global y debe de ser configurada mediante la interfaz web (ver sub-sección “4.5.2. Configuración de redes” en la sección de “4.5. Configuración del sistema”).

Una “Alerta Global” se refiere al resultado de la interacción entre nodos sensor maestro de una sola red colaborativa global, donde de forma conjunta toman la decisión sobre cual es el estado de alerta de un puente. El criterio de decisión entre nodos sensor, esta dado en el cuadro.

La decisión funciona por prioridades; es decir:

$$Alerta_Global = Alerta_Puente_n_mayor_prioridad$$

donde:

$$Alerta_Puente_n = Verde <$$

$$Alerta_Puente_n = Amarillo <$$

$$Alerta_Puente_n = Rojo$$

Los escenarios de decisión entre nodos sensor maestro, con una o más mediciones de peso en una red colaborativa, cambia(n) el estado de la red, en el cuadro 4.21) se muestran algunos escenarios hipotéticos de prueba para fines ilustrativos, donde cada fila es un escenario. Por ejemplo en la primera fila del cuadro, se muestra el caso en que el nodo sensor n presenta una alarma de estado “Rojo” en la columna “Alerta de Puente”, lo cual altera la alerta global de la red colaborativa a la que pertenece, que es la que se presenta en la última columna del cuadro (Alerta de Red).

Cuadro 4.21: Algunos escenarios del Protocolo para Alertas Globales

Puente de Red	Alerta de Puente	Alerta de Red
1	Verde	
2	Amarillo	Rojo
..		
n	Rojo	
1	Amarillo	
2	Verde	Amarillo
..		
n	Amarillo	
1	Verde	
2	Verde	Verde
n	Verde	

4.3. Principales funciones y algoritmos

A continuación se presenta con más detalle cada uno de los componentes funcionales y algorítmicos del protocolo:

4.3.1. Filtro de mediana móvil

Como parte de los requisitos del sistema se debía de trabajar con sensores de bajo costo, que siempre presentan espurios debidos al ruido inherente de los ambientes de medición. Dicho requerimiento se cumple en la implementación y se ha tratado de amortiguar su impacto en los datos medidos, mediante el uso de un método numérico-algorítmico conocido, Filtro de mediana móvil.

Dicho método no considera a valores extremos causados por el sensor, es un instrumento muy útil y ampliamente empleado en electrónica. Muy utilizados en sistemas de adquisición de datos y captación de sensores para la eliminación de ruido. Para su utilización se define una ventana de N elementos, que recolecta las últimas N mediciones, descartando a las que están muy alejadas de la mediana estadística. Como parte del método empleado, el empujado reporta su medición hasta cumplir con los requisitos del filtro y obtener diez (ventana definida) mediciones cercanas a la mediana.

4.3.2. Falsos positivos

Un falso positivo se define como el escenario en que uno de los nodos sensor, de una estructura puente, genera una medición distinta al resto de nodos sensor instalados en ese puente. Sugiriendo así, estar en presencia de un objeto bajo el puente que no refleja el nivel del agua bajo el mismo.

Mediante una comparación constante entre las mediciones de los nodos sensor de una misma estructura puente y a su vez una comparación contra el histórico de

mediciones del nodo sensor en cuestión (buscando un cambio abrupto), se determina si el valor que se está midiendo es o no es un valor real de altura del puente. Para este algoritmo es importante hacer ver que el parámetro de configuración llamado Tolerancia, es el que determina si el cambio o la diferencia de mediciones es o no abrupta.

4.3.3. Determinación de mejor señal

Mediante el uso del comando 'commandAT('AT+CSQ',100,2)' que genera de respuesta por parte del SIM900, el nivel de potencia de su señal de comunicación a la red telefónica, la Raspberry Pi comparte dicho dato por medio del módulo Xbee entre los nodos sensor de una misma estructura puente, se hace la comparación y quien tenga la señal más alta, gana la disputa.

Mediante variables del programa se estipula cual de los nodos sensor llevará el rol de maestro en la estructura, y se le declara al resto de nodos como nodos esclavos, cuya función principal será la de muestreo de datos. La configuración también ofrece la opción de configurar a alguno de los nodos directamente como nodo maestro, para ofrecer versatilidad al sistema.

4.4. Configuración del sistema

Todas las opciones de configuración del sistema se muestran en la Figura 4.8. A continuación se desarrolla cada una de las opciones por dispositivo, red colaborativa y puente.



Figura 4.8: Interfaz web.

4.4.1. Configuración de dispositivos

Inicialmente se debe crear el dispositivo para luego poder asociarlo tanto a un puente, como a una red colaborativa. En la Figura 4.9 se muestra la interfaz que le permite al usuario crear un dispositivo, únicamente requiere de asignarle un identificador y pulsar el botón de aceptar para almacenarlo en la base de datos.

Figura 4.9: Interfaz para crear dispositivos

Para configurar a un dispositivo utilizable en la red de sensores, debe de existir previamente un puente en la base de datos a la cual ligarlo, además debe existir el dispositivo per se, y se requiere agregar la frecuencia de tiempo con la que se realizará el muestreo de medición (dado en minutos), el periodo de tiempo con el que se quiere enviar que en la mayoría de los casos se utiliza igual que el periodo de muestreo, la altura nominal del puente, la tolerancia de medición con que se permitirá medir y por último la dirección MAC del dispositivos Xbee a utilizar.

Figura 4.10: Configuración de dispositivos.

4.4.2. Configuración de redes

Todo dispositivo debe de pertenecer a alguna red colaborativa. En la Figura 4.11 se muestra la interfaz que le permite al usuario crear una red colaborativa, requiere de asignar un identificador y pulsar el botón de aceptar para que la red sea almacenada en la base de datos.

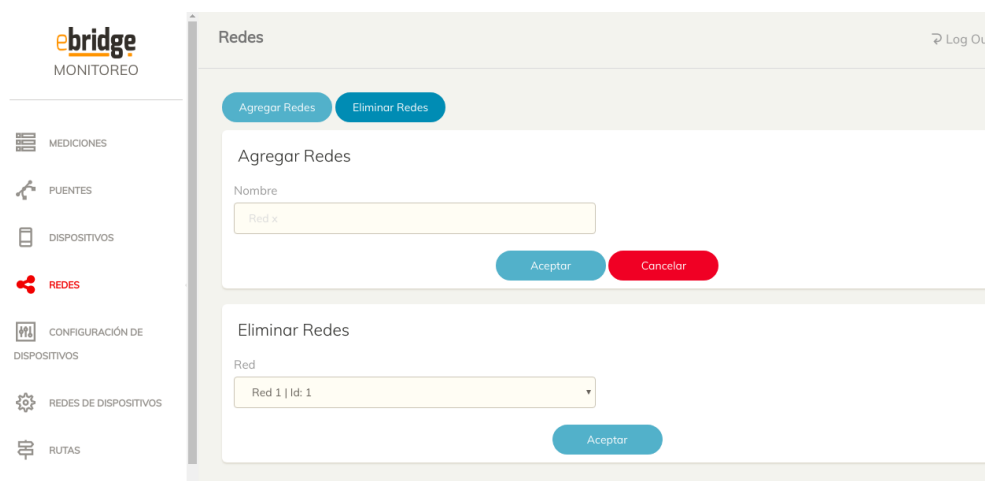


Figura 4.11: Interfaz para crear redes

La configuración de redes colaborativas se hace por medio de la interfaz de la Figura 4.11, al igual que para dispositivos debe de existir previamente la red per se en la base de datos, además debe existir el dispositivo para ligarlo a la red colaborativa mediante el botón de aceptar, un mensaje de confirmación aparece en forma de *popup* para su confirmación.



Figura 4.12: Configuración de redes.

4.4.3. Creación de puentes

En la Figura 4.13 se muestra la interfaz mediante la cual el usuario puede crear puentes que pertenecen a una red colaborativa, requiere de asignar un identificador, definir su altura nominal, el nombre del puente y por último su ubicación, al pulsar el botón de aceptar se almacena en la base de datos.

The screenshot shows the 'Puentes' management interface. It includes a sidebar with navigation options and a main content area with two sections: 'Agregar Puentes' and 'Eliminar Puentes'. The 'Agregar Puentes' section has input fields for 'Identificador', 'Nombre', 'Altura', and 'Ubicación', along with 'Aceptar' and 'Cancelar' buttons. The 'Eliminar Puentes' section has a dropdown menu for selecting a bridge and an 'Aceptar' button.

Figura 4.13: Interfaz para crear puentes

4.5. Mapa o plano general del sitio

En esta sección se define la navegación en el sitio web del sistema de monitorización, además se aclara hasta donde se puede llegar desde la página principal. También es importante mencionar que no se tiene un componente de búsqueda ya que al ser un sitio de visualización de datos en tiempo real, se trata de un sitio de actualización constante, para lo cual se ha facultado con la posibilidad de bajar los datos a un archivo con formato de Excel.

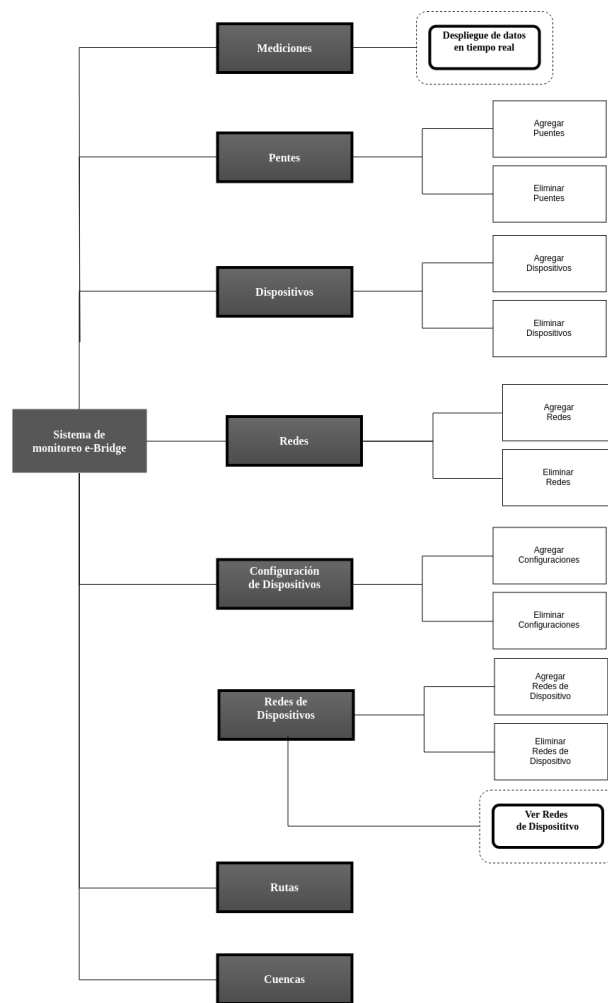


Figura 4.14: Plano general del Sitio.

Por otro lado el sitio contiene componentes de configuración de la red, por lo que se trata de tener todo disponible a un solo clic, mientras que la parte de visualización de datos en tiempo real fue diseñada para mostrar los datos mas recientes de forma inmediata y pertenece a la plantilla principal donde todas los demás páginas se conectan. La Figura 4.14 ilustra el plano general del sitio, cuya notación ha sido tomada de [54].

4.6. Wireframe de la página principal

En el *Wireframe* de la interfaz de mediciones para e-Bridge, se pretende dar a conocer el prototipo o boceto, donde se representa visualmente, de una forma muy sencilla y esquemática, la estructura de la interfaz web. Dicho *Wireframe* se presenta en la figura 4.15.

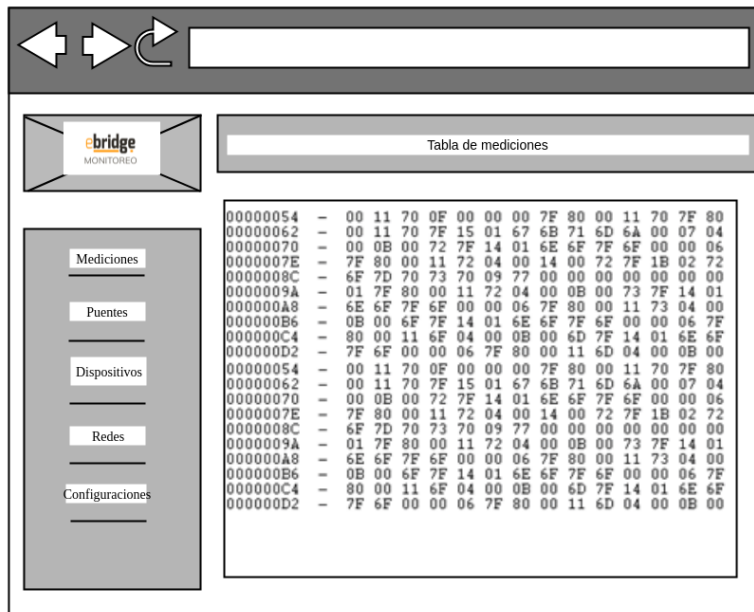


Figura 4.15: Boceto de la interfaz web, Wireframe. Notación tomada de [54].

El objetivo este tipo de boceto es definir el contenido y la posición de los diversos bloques, los menús de navegación, bloques de contenido, etc. En la columna de la izquierda se presenta todas las opciones de acceso directo a otra sub-página diferente a la principal. El diseño pretende que de entrada el sistema despliegue los datos mas recientes ingresados a la base de datos por parte de las redes colaborativas y los nodos sensor.

Capítulo 5

Implementación

En este capítulo se presentan los detalles de programación y sus ambientes, además los sistemas operativos utilizados en el desarrollo de la aplicación. Se muestra también, la implementación del diseño de la red colaborativa de sensores como prototipo preliminar, para el cual se implementan tanto nodos sensor como el servidor en la nube, y a su vez el entorno de comunicación entre ellos.

Este capítulo se estructura de la siguiente forma:

- Descripción de los ambientes de programación y software.
- Servidor en la nube, módulos y clases.
- Nodos Sensor.
- Repositorio en GitHub.

5.1. Descripción de los ambientes de programación y software

El proyecto mantiene una filosofía colaborativa a través de toda la implementación. A continuación se describe la implementación de cada parte del proyecto, en función de los elementos de software que las componen.

5.1.1. Sistemas operativos

Con una distribución basada en Debian, Ubuntu 16.04.6 ofrece un sistema operativo efectivo para ambientes de desarrollo y programación. Además es gratuito, con naturaleza de código abierto y con amplio soporte.

```

$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:   Ubuntu 16.04.6 LTS
Release:      16.04
Codename:     xenial
$ hostnamectl
  Static hostname: ebridge
           Icon name: computer-vm
           Chassis: vm
           Machine ID: 05c64f700b5045af9115305d439f910f
           Boot ID: 1b618d628b894de39c7c5d3cc0df76d2
           Virtualization: vmware
           Operating System: Ubuntu 16.04.6 LTS
           Kernel: Linux 4.4.0-142-generic
           Architecture: x86-64

```

Figura 5.1: Respuesta a comandos *lsb_release -a* y *hostnamectl*.

En la figura 5.1 se muestran los detalles de versión del sistema operativo utilizado en la máquina virtual del servidor en la nube. La información mostrada fue obtenida por medio de la Terminal de Ubuntu, mediante un acceso remoto por ssh y los comandos *lsb_release -a* y *hostnamectl*.

```

pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 9 (stretch)"
NAME="Raspbian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
pi@raspberrypi:~ $ █

```

Figura 5.2: Versión de Raspbian para dispositivos empotrados.

La selección del Raspberry como dispositivo empotrado para la implementación de nodos sensor, se utiliza Raspbian Stretch with desktop, que es una plataforma basada en Debian, hecha a la medida para manejar todo el hardware de un Raspberry Pi. En la figura 5.2 se muestra la versión de sistema operativo utilizada en todos los dispositivos empotrados que actúan como nodos sensor de medición de la red.

5.1.2. Lenguajes de programación

Para cada elemento que compone al sistema, se ha utilizado un lenguaje de programación que se adapta a cada circunstancia, además de que la implementación de las diferentes partes de la red, se realiza en función de sus requerimientos.

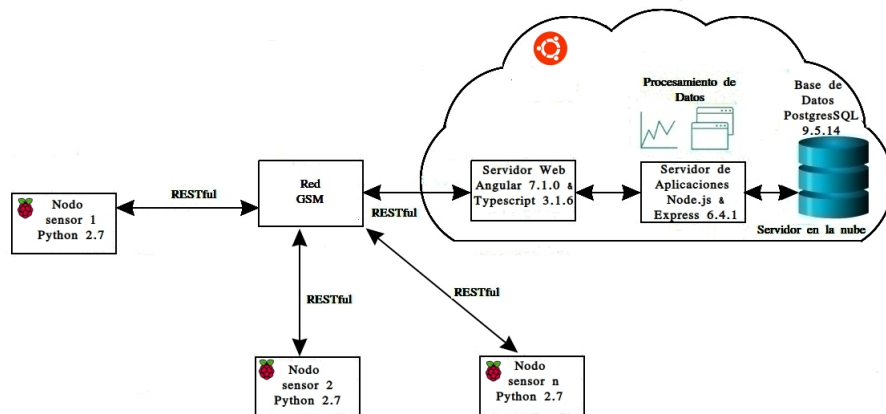


Figura 5.3: Arquitectura general del sistema a base de componentes de software.

En la figura 5.3 se muestra la arquitectura del sistema en términos de componentes de software, que esencialmente se puede subdividir en cuatro partes: servidor de aplicación, servidor web, base de datos y nodo sensor. Cuya implementación es brevemente descrita a continuación y profundizada a través del capítulo.

```

anibal@anibal-Satellite-S55t-B:~$ ssh -p 24 dgomez@201.207.53.225
dgomez@201.207.53.225's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

24 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

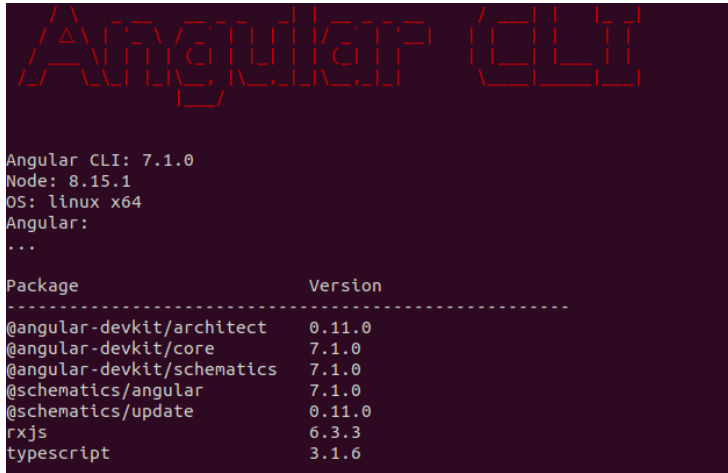
*** System restart required ***
Last login: Sat Apr 20 08:37:04 2019 from 186.26.117.246
$ node -v
v8.15.1
$ npm -v
6.4.1
$ express --version
4.16.0

```

Figura 5.4: Versión de servidor de Node.js, NPM y Express.

El *BackEnd* del servidor de aplicación ha sido implementado en el entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js [27] y el framework Express 6.4.1 para la creación de APIs y enlace de aplicación web. La versión de Node.js utilizada es parte de V8 tal y como se muestra en la figura 5.4, la documentación

de V8 se puede encontrar en [28].



```

Angular CLI: 7.1.0
Node: 8.15.1
OS: linux x64
Angular:
...
Package      Version
-----
@angular-devkit/architect 0.11.0
@angular-devkit/core      7.1.0
@angular-devkit/schematics 7.1.0
@schematics/angular       7.1.0
@schematics/update        0.11.0
rxjs                    6.3.3
typescript              3.1.6

```

Figura 5.5: Versión de servidor de Angular y Typescript.

El servidor Web del cual forma parte el *FrontEnd* del servidor, está implementado mediante Angular 7.1.0 y su parte funcional en Typescript 3.1.6 tal y como se muestra en la figura 5.5. La página web puede ser accedida por medio del url: <http://201.207.53.225:4200/login> (ver figura 5.6)



Figura 5.6: Sitio web de e-Bridge.

La base de datos está implementada en PostgreSQL 9.5.14 y el programa de nodos sensor se desarrolló en Python 2.7. Ambos componentes de la red, son profun-
dizados en el desarrollo del capítulo.

5.1.3. Scripts de arranque de empotrados

La naturaleza de la red y su posible implementación en ambientes reales, genera la necesidad de que los nodos sensor sean capaces de inicializarse de forma automática en campo. Cada nodo sensor debe ser capaz de convertirse en nodo maestro

de ser requerido, por lo que su conexión a internet debe de ser también automática, su implementación les permite funcionar como módem a través de pppd.

```

GNU nano 2.7.4
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi
sudo pon rnet
#echo hola anibal
#mount
sudo ifconfig wlan0 down
sleep 45
sudo /usr/bin/python /home/pi/TestV2/RaspberryCode/device.py > /home/pi/Desktop/log.txt 2>&1 &
# sudo echo 17 > /sys/class/gpio/export
# sudo echo out > /sys/class/gpio/gpio17/direction
# sudo echo 1 > /sys/class/gpio/gpio17/value
exit 0

```

Figura 5.7: Script de arranque de los dispositivos empotrados(Nodo Sensor).

En la figura 5.7 se muestra la implementación del script de arranque, mediante el archivo rc.local del sistema operativo Raspbian, dicho archivo es parte de los archivos de arranque y es tradicionalmente ejecutado después de que todos los servicios normales del sistema operativo han sido inicializados [57].

Mediante experimentación de campo, se ha determinado la necesidad de un tiempo de sleep al arrancar al dispositivo empotrado (45 segundos o más, ver la figura 5.7). Un tiempo de espera, permite al dispositivo inicializar correctamente el pppd, dicha tarea se ejecuta por medio del comando sudo pon rnet.

Como detalle relevante de instalación de los nodos sensor, se debe mencionar que el módulo GPRS debe de ser manualmente encendido antes de que el sistema operativo del Raspberry inicie por completo, dicha tarea recae por completo en el ingeniero de campo o encargado de puentes, quien debe de asegurarse que los leds presetan iluminación de color verde; uno de ellos fijo y el otro intermitente.

Una vez establecida la conexión al internet de la red telefónica, se arranca el programa device.py del nodo sensor y se crea un archivo llamado log.txt que permite dar seguimiento a la operación del nodo.

5.2. Servidor en la nube, módulos y clases

La implementación del servidor estuvo principalmente definida por el manejo de Node.js y su *framework* Express, Angular y TypeScript, respectivamente para Bac-

kend y Frontend.

La base de datos por su parte tiene se conecta con el entorno por medio de Angular y esta construida completamente con *PostgreSQL*, debido a dos razones esencialmente; como primera razón no se manejan meta-datos(por el momento); como segunda razón se desea mantener compatibilidad con otras bases de datos de desarrollos que forman parte del proyecto e-Bridge; sin embargo no se descarta una migración como trabajo futuro.

5.2.1. Servidor de aplicaciones

El diagrama de clases que se muestra en la figura 5.8 corresponde a la implementación del servidor de aplicaciones. Sus principales funciones son la del manejo de datos tanto de medición de todas las redes colaborativas (por nodo sensor), como datos de configuración de nodos sensor y redes colaborativas; también se administran los datos por estructura puente y sus respectivas configuraciones.

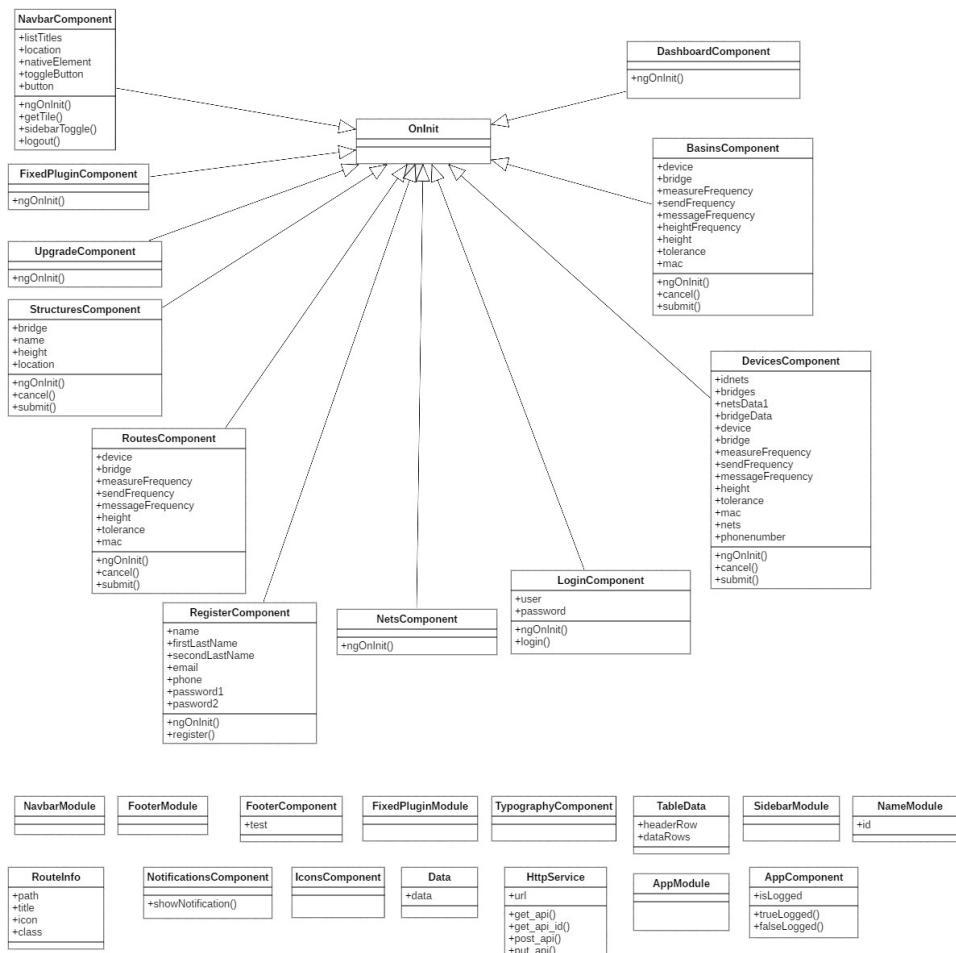


Figura 5.8: Diagrama de clases del servidor de aplicaciones.

Todo elemento de la red, en algún momento debe de transferir sus datos y probablemente los de otros elementos al servidor de aplicaciones, para que estos sean ingresados a la base de datos. Su independencia como sistema distribuido se da a la hora de toma de decisiones, donde los nodos sensor tienen capacidad de procesamiento a la hora de enfrentar estados de emergencia y lo hacen a través de su filosofía colaborativa inherente.

5.2.2. Servidor Web

El servidor en la nube también alberga su propia interfaz de servidor web, utilizada para la comunicación con nodos sensor, tanto para configuración, como para recolección de datos cuando los mismos trabajan en estructura RESTful.

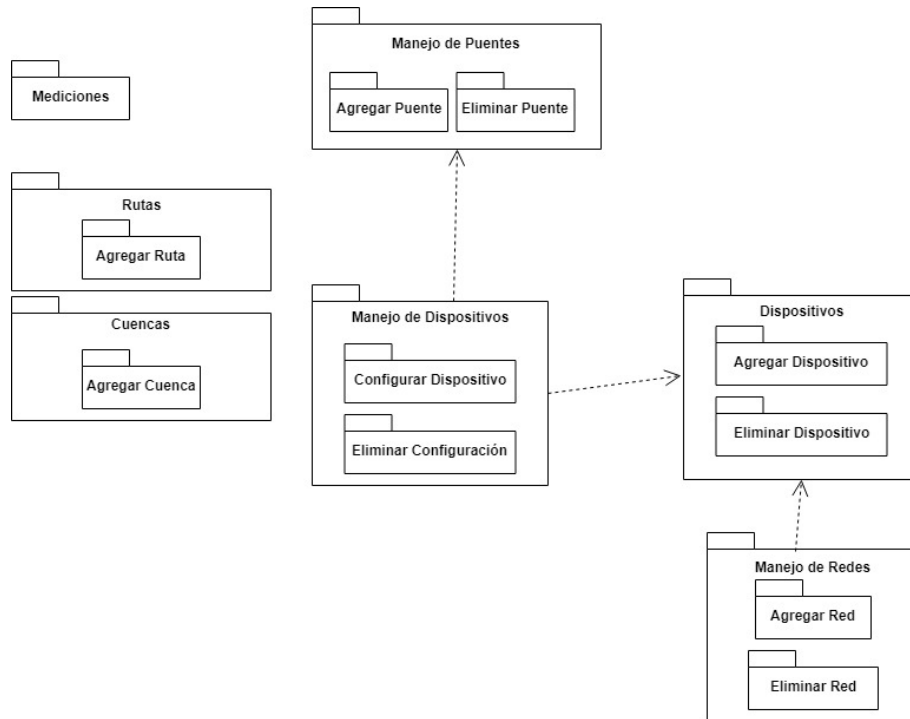


Figura 5.9: Diagrama de módulos de la página web.

Una de las principales funciones del servidor web es atender solicitudes de la página web, portal que se utiliza para varios propósitos, entre ellos configurar nodos sensor, puentes, redes colaborativas y para definición de roles; ya sea de nodo esclavo o de nodo maestro. En la figura 5.9 se proporciona una descripción por módulos del sitio web implementado, que está disponible en la url:<http://201.207.53.225:4200>.

5.2.3. Base de datos

La base de datos en el servidor de aplicaciones tiene el nombre de “ebridge” y está compuesta por un total de 9 tablas, mencionadas a continuación: *Routes*, *Basins*,

Structures, Devices, Reports, Configurations, Users, Nets y *DeviceInNet*, y su implementación está realizada en la versión 9.5.14 de PostgreSQL.

La creación de la base de datos “ebridge” en el servidor en la nube, se realiza por medio de un archivo llamado “bridge.sql”, el cual se encuentra disponible en el repositorio *git* en la ruta: “Anibalru/E-Bridge”. El *script* de creación de “ebridge” se muestra en el figura 5.10.

```
1  /*CREATION OF THE DATABASE 'ebridge' */
2
3  drop database if exists ebridge;
4
5  create database ebridge;
6  \c ebridge;
7
8  CREATE TABLE Routes(
9  IdRoute int NOT NULL,
10 Name varchar NULL,
11 CONSTRAINT PK_Routes PRIMARY KEY (IdRoute));
12
13 CREATE TABLE Basins(
14 IdBasin int NOT NULL,
15 Name varchar NULL,
16 CONSTRAINT PK_Basins PRIMARY KEY (IdBasin));
17
18 CREATE TABLE Structures(
19 IdStructure int NOT NULL,
20 Height int NOT NULL,
21 Name varchar NULL,
22 Location text NULL,
23 Masl int NOT NULL,
24 IdRoute int NOT NULL,
25 IdBasin int NOT NULL,
26 CONSTRAINT PK_Structures PRIMARY KEY (IdStructure));
27
28 CREATE TABLE Devices(
29 IdDevice int NOT NULL,
```

Figura 5.10: Script de deployment de la base de datos “ebridge”.

5.3. Nodo sensor

A continuación se presenta con más detalle cada uno de los componentes funcionales y algorítmicos del protocolo desde la perspectiva del nodo sensor. La descripción completa de diagrama de flujo, clase y paquetes, además se detalla la implementación del protocolo de alarmas y su manejo en la toma de decisiones, en la sub-sección de protocolo de comunicación se describe a todos los niveles la forma de interacción entre elementos de la red de sensores, finalizando con la implementación del filtro de mediciones.

5.3.1. Flujo, clases y paquetes

En la figura 5.11 se establece el diagrama de flujo que siguen los dispositivos nodo sensor dentro de la red. Se aprecia que antes de iniciar el proceso de medición y generación de reportes, se inicializa un hilo para la temporización, otro hilo para la comunicación con el módulo xBee, además se valida el rol de maestro o esclavo y se hace la conexión a la red telefónica; es en esta parte del proceso que se inicializan las mediciones de campo.

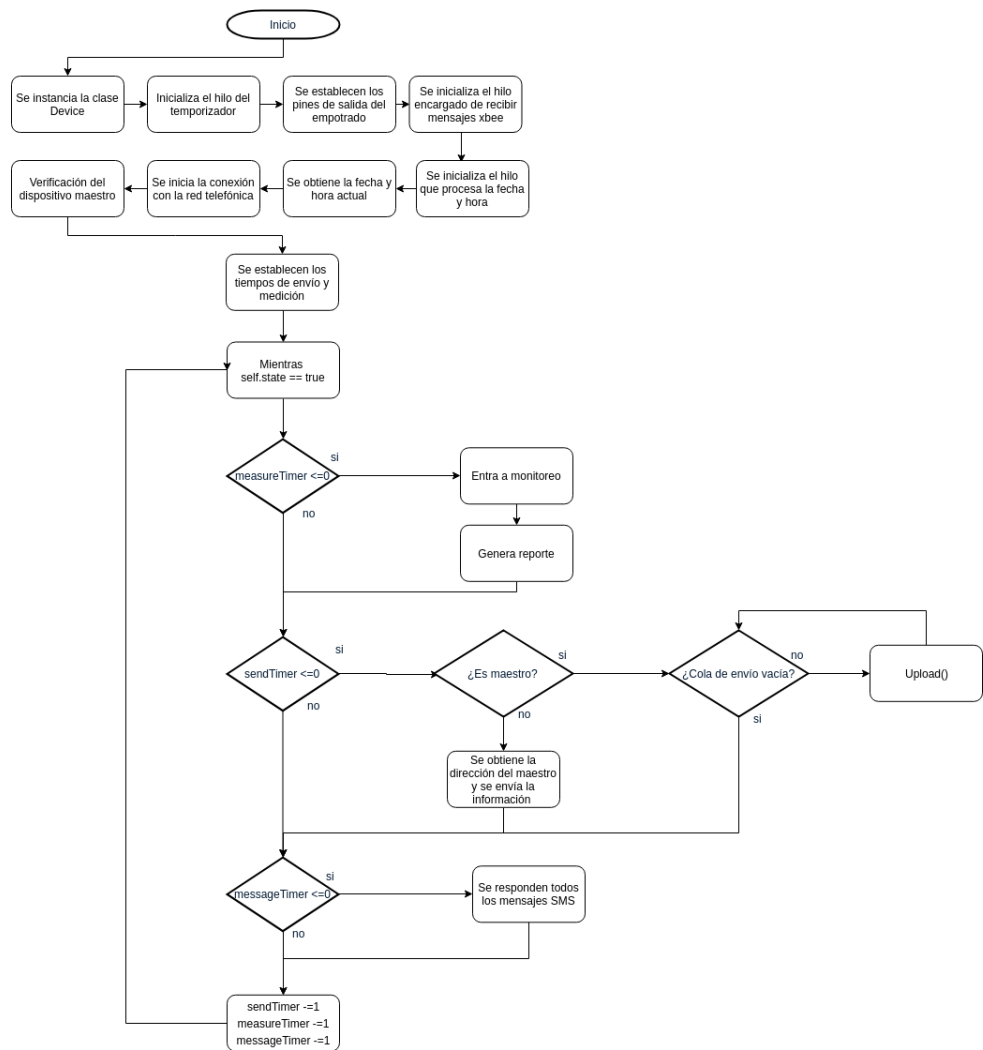


Figura 5.11: Diagrama de flujo de empotrados.

A nivel de clases se describe al nodo sensor como se muestra en la figura 5.12, donde es claro que únicamente se definen dos clases, una de manejo del dispositivo llamada *device* y otra para el manejo de solicitudes por http.

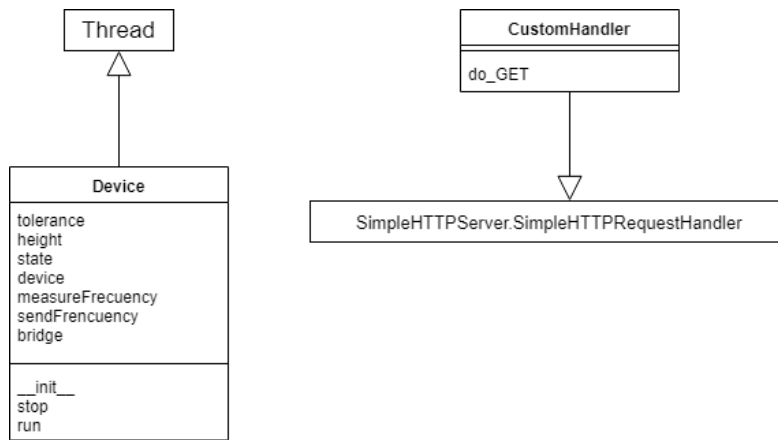


Figura 5.12: Diagrama de clases del empotrados.

El manejo de paquetes de datos a nivel de los nodos sensor es bastante interesante, ya que es una de las partes donde se refleja la mezcla entre un sistema distribuido y un sistema maestro-esclavo. En un solo puente se podría tener el escenario donde varios nodos, sean nodos sensor esclavo y haya comunicación entre ellos, incluso transmisión de datos de un nodo esclavo a otro nodo esclavo, para realizar una re-transmisión al nodo maestro. El diagrama de paquetes se muestra en la figura 5.13.

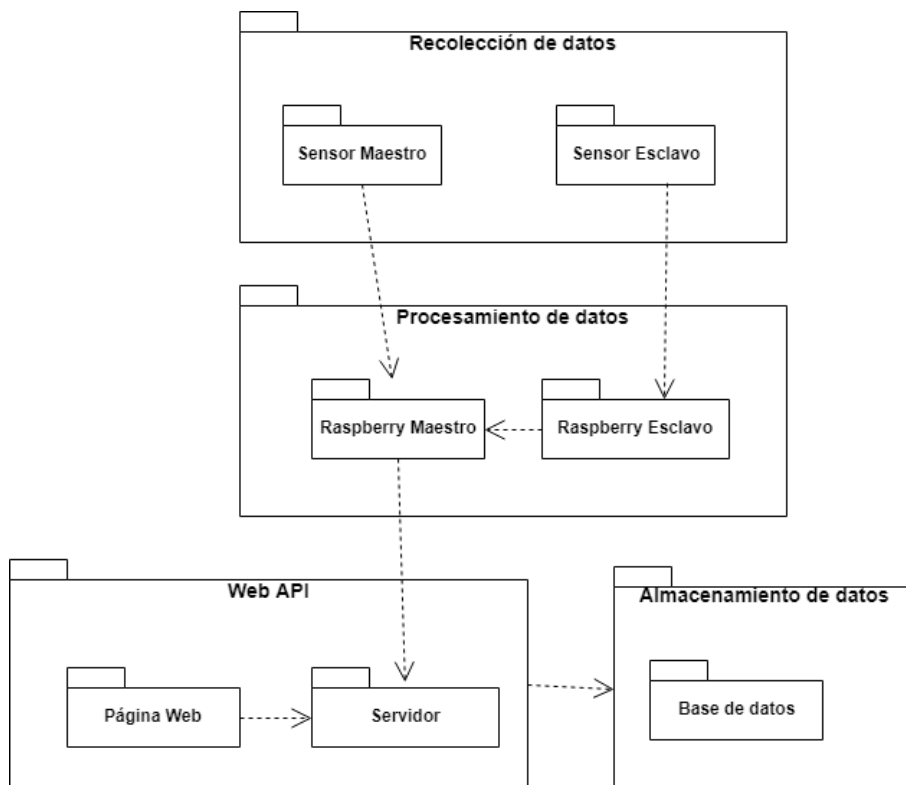


Figura 5.13: Diagrama de paquetes del empotrados.

5.3.2. Protocolo de alertas

Las alertas contemplan el estado tanto de la red local como de la red global a la que pertenece la estructura, tal y como se ha descrito en el capítulo anterior. En la figura 5.14 se muestra su implementación en Python, la cual es ejecutada de la misma forma en todos los nodos que componen la red colaborativa de sensores.

```
def checkAlert(idDevice,state):
    global globalId, globalState, globalEmergency, dTime, globalHour
    if(state > 1 and globalEmergency == False):
        globalHour = str(dTime.time())
        globalEmergency = True
        globalId = idDevice
        globalState = state
        print("EMERGENCIA EN: ", globalId, globalState)
    elif(state > globalState and globalEmergency == True):
        globalHour = str(dTime.time())
        globalId = idDevice
        globalState = state
        print("EMERGENCIA EN: ", globalId, globalState)
    else:
        if(idDevice == globalId):
            if(state == 1):
                print("TERMINA EMERGENCIA EN: ", globalId, globalState)
                globalHour = str(dTime.time())
                globalEmergency = False
                globalId = -1
                globalState = 1
            elif(state == 2):
                print("EMERGENCIA PASA A AMARILLO: ", globalId, globalState)
                globalHour = str(dTime.time())
                globalState = state
    return "OK"
```

Figura 5.14: Implementación del protocolo de alertas.

5.3.3. Protocolo de comunicación

En un nodo sensor de maneja comunicación de tres formas distintas; por radio frecuencia con otros nodos sensor locales (mediante el módulo XbeeS2C), mediante mensajes de texto SMS, ó por medio de la estructura de servidor web, que maneja a cada nodo sensor en estructura RESTful.

5.3.3.1. Servicios Web

REST es esencialmente un conjunto de convenciones útiles para estructurar una API web. Es decir una API con la que se interactúa a través de HTTP, utilizando esta estructura se realizan solicitudes a direcciones URL específicas, que a menudo son contestadas con datos relevantes [55]. En la figura 5.15 se muestra la implementación en Python del servicio web de los nodos sensor.

```

class CustomHandler(SimpleHTTPServer.SimpleHTTPRequestHandler):
    def do_GET(self):
        global devicesInNet
        if None != re.search('/ebridge/alert/*', self.path):
            alert = int(self.path.split('/')[1])
            device = int(self.path.split('/')[2])
            print self.path.split('/')
            #This URL will trigger our sample function and send what it returns back to the browser
            self.send_response(200)
            self.send_header('Content-type', 'text/html')
            self.end_headers()
            self.wfile.write(checkAlert(device,alert).encode()) #call sample function here
            return
        if None != re.search('/ebridge/device/*', self.path):
            ip = self.path.split('/')[1]
            print self.path.split('/')
            devicesInNet.append(ip)
            #This URL will trigger our sample function and send what it returns back to the browser
            self.send_response(200)
            self.send_header('Content-type', 'text/html')
            self.end_headers()
            self.wfile.write("ok".encode()) #call sample function here
            return
        else:
            #serve files, and directory listings by following self.path from
            #current working directory
            SimpleHTTPServer.SimpleHTTPRequestHandler.do_GET(self)

```

Figura 5.15: Implementación de servicios web en estructura RESTful.

5.3.3.2. Comunicación entre nodos sensor por medio de SMSs

En la actualidad los módulos GPRS se han tornado muy comunes en proyectos de investigación, debido a que la tecnología SMS ha bajado significativamente de costo. Se ha planteado la comparación de la eficiencia entre tecnologías, mediante una implementación que permita trabajar bajo condiciones muy similares e incluso con la misma arquitectura.

```

def getAllSMS():
    return commandAT('AT+CMGL="ALL"',20000,2)

def sendSMS(message, phoneNumber):
    commandAT('AT+CMGF=1',100,2)
    commandAT('AT+CNMI = 2,1,0,0,0',100,2)
    commandAT('AT+CMGS= "{}"'.format(phoneNumber),100,2)
    commandAT(message,len(message),5)
    commandAT('\x1A',100,5)

```

Figura 5.16: Implementación de funciones para mensajería SMS entre dispositivos de diferentes estructuras puente.

Mediante el uso de comandos AT (parte de la tecnología GSM) y una conexión a través de un puerto serial del Raspberry, se ha implementado la comunicación para la red global (redes colaborativas definidas por el usuario) a través de mensajes de texto plano (SMS), con el fin de realizar una comparación de desempeño contra una estructura meramente implementada en estructura REST. Los principales metodos implementados se muestran en el figura

5.3.3.3. Xbee

En el caso de comunicación local, se realiza el manejo de comunicación e identificación por medio de la dirección MAC del dispositivo Xbee, que es tratado como una variable del programa y obtenido como parte de la configuración del dispositivo que envía el servidor en la nube a cada dispositivo nodo.

```
def recibir(height, device, bridge, tolerance):
    global master, devicesInBridge

    print("Esperando informacion...")

    data_count = 0
    data_json = ""
    json_parts = 2

    while True:
        contenido = xbee.wait_read_frame()
        #print(contenido)
        if contenido != None:
            print("LLEGO ALGO")
            contenido = decodificar(contenido)
            data_json += contenido
```

Figura 5.17: Implementación de Xbee.

En la figura 5.17 se muestra el método de recepción de datos que forma parte de la comunicación Xbee, que se ejecuta en uno de los hilos del programa. Es importante mencionar que el código fuente debe incluir la biblioteca `xbee.python2to3`, que ya proporciona una API que facilita el trasiego de información entre dispositivos. Una vez recibida la información de algún vecino, se decodifica la información para ser tratada a nivel de alarma local.

5.3.4. Filtro y toma de mediciones

Como se ha mencionado en capítulos anteriores, el uso de sensores de bajo costo implica un tratamiento de los datos medidos por software. El objetivo es el de mitigar el efecto especialmente de los espurios o el ruido sal y pimienta (puntos que van al mínimo o máximo posible) que ofrecen los datos cuando un sensor no es de alta calidad, sin embargo mediante la implementación de técnicas conocidas como el filtro de media y mediana móvil, es posible realizar mediciones bastante aceptables estadísticamente hablando.


```

# Get the mean of the measures
def getMeasures(tolerance):
    measureslist = []
    while (len(measureslist) != 10):
        distance = measure()
        if distance > 0 and distance < 10000:
            measureslist.append(distance)
    measuresMean = numpy.mean(measureslist)
    measureslist = checkDeviation(measureslist, measuresMean, tolerance)
    if (len(measureslist) <= 5):
        return getMeasures(tolerance)
    else:
        return numpy.mean(measureslist)

```

Figura 5.18: Filtrado de los datos mediante media y mediana móvil.

En la figura 5.18 se muestra el método utilizado tanto para la implementación del filtro tanto mediana como media móvil, los cambios de uno respecto al otro, serían, la ventana (N par para media móvil, N impar para mediana móvil) y la función utilizada (mean, median).

5.4. Repositorio en GitHub

La red colaborativa de sensores del proyecto e-Bridge posee su propio repositorio en Github, donde se mantiene y da seguimiento de toda versión y cambio en el código de cada módulo del proyecto. En la figura 5.19 se muestra la estructura del repositorio, que se divide principalmente en servidor, página web y código de los dispositivos empotrados.

The screenshot shows the GitHub repository page for 'Anibalru / E-Bridge'. The repository has 154 commits, 2 branches, 0 releases, and 3 contributors. The latest commit is by 'stevensg97' on Feb 25. The file list includes:

File	Commit Message	Time Ago
.cache	[Add] Database	5 months ago
.config/configstore	[Add] changes@	4 months ago
.npm	[Add] changes	4 months ago
EBridgeServer	[Add] IsMaster column	4 months ago
EBridgeWebPage	[Add] Delete configurations	2 months ago
RaspberryCode	[Add] Ultima version David 1	5 months ago
.bash_logout	[Add] Database	5 months ago
.bashrc	[Add] Database	5 months ago
.gitconfig	[Add] Database	5 months ago

Figura 5.19: Repositorio principal de la red colaborativa de sensores de e-Bridge.

El repositorio del servidor se muestra en la figura 5.20. Entre los principales archivos está el `app.js` que es el archivo de configuración principal para una aplicación elaborada en Express, en él se define la url y puerto del servidor del sitio web. Además se definen los queries a utilizar en el archivo `queries.js`, tales como insertar, crear, borrar, actualizar, entre otras acciones.

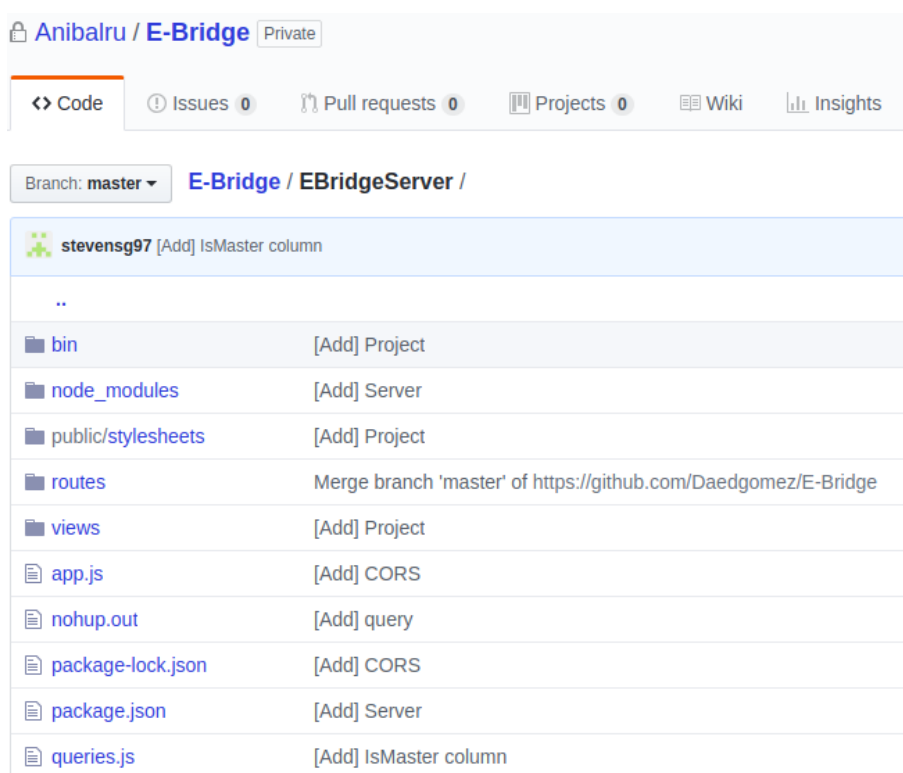
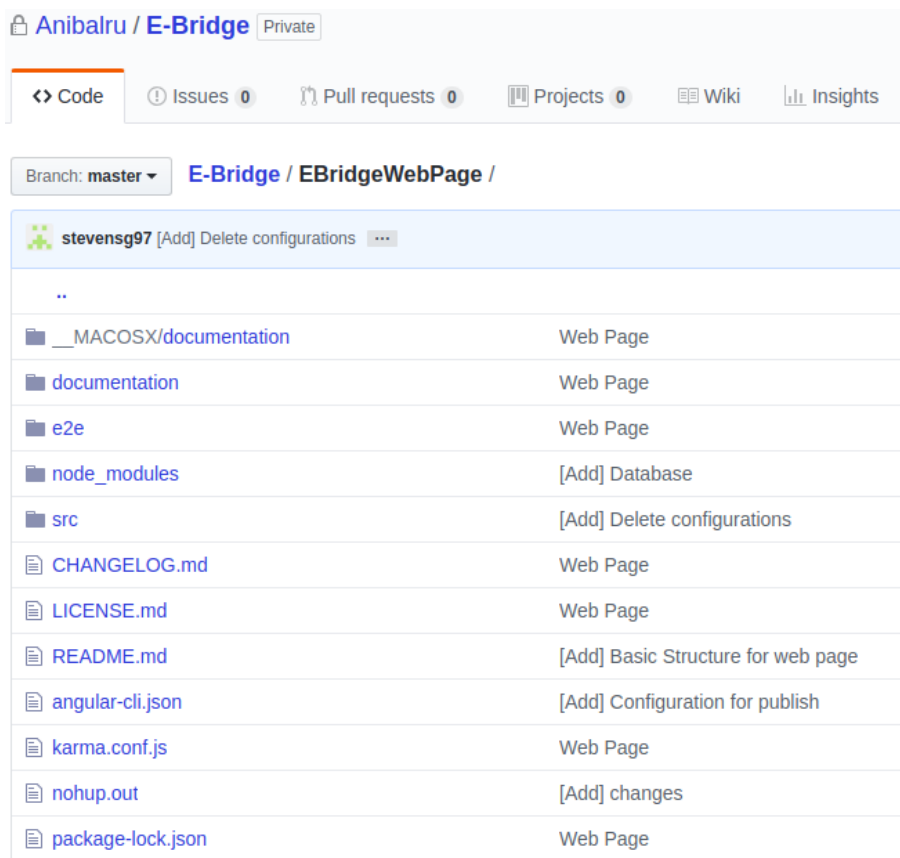


Figura 5.20: Repositorio del servidor de aplicación de la red de sensores e-Bridge.

Los archivos de configuración y código agregado al servidor y sitio web, se encuentran en el repositorio `EBridgeWebPage` como se muestra en la figura 5.21.



The screenshot shows the GitHub interface for a repository named 'E-Bridge' by user 'Anibalru'. The repository is marked as 'Private'. The main navigation bar includes 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', and 'Insights'. Below this, the current branch is 'master' and the repository path is 'E-Bridge / EBridgeWebPage /'. The file list shows a directory structure with folders like 'documentation', 'e2e', 'node_modules', and 'src', and files like 'CHANGELOG.md', 'LICENSE.md', 'README.md', 'angular-cli.json', 'karma.conf.js', 'nohup.out', and 'package-lock.json'. Each item has a status indicator, such as 'Web Page' or '[Add] Database'.

Item	Status
..	
__MACOSX/documentation	Web Page
documentation	Web Page
e2e	Web Page
node_modules	[Add] Database
src	[Add] Delete configurations
CHANGELOG.md	Web Page
LICENSE.md	Web Page
README.md	[Add] Basic Structure for web page
angular-cli.json	[Add] Configuration for publish
karma.conf.js	Web Page
nohup.out	[Add] changes
package-lock.json	Web Page

Figura 5.21: Repositorio del servidor web de la red de sensores e-Bridge.

Capítulo 6

Validación y Pruebas

En este capítulo se presenta la especificación de pruebas realizadas al prototipo de sistema, el proceso de validación aplicado ha sido definido a partir de varias pautas, entre las cuales destacan: pruebas piloto durante la implementación, criterio experto del personal del CIVCO, investigación de trabajos relacionados, prácticas comunes en redes de sensores, pruebas de filosofía colaborativa, entre otras.

El propósito principal de las pruebas de validación del prototipo, es el de verificar el funcionamiento correcto de acuerdo a lo esperado en el diseño. A través del capítulo se incluyen cuadros que muestran los resultados obtenidos y el análisis de los mismos. La variable medida de puentes para el proceso de validación es la altura libre inferior de cada estructura puente.

Dentro de los datos tomados más importantes, están: el escenario de colaboración entre nodos sensor de un solo puente, el escenario de colaboración entre nodos sensor de varios puentes y los datos obtenidos al comparar al prototipo contra si mismo; este último, utilizando dos tecnologías de comunicación distintas entre nodos sensor de redes colaborativas y definidas por el usuario, es decir entre varios puentes. Para un caso se utiliza estructura RESTful y para el otro, mensajes SMS.

Para el escenario de varios puentes, los nodos sensor transmiten mensajes entre sí para compartir el estado de alerta de cada estructura puente; en caso de emergencia, un puente en riesgo debe influir sobre el estado de todos los puentes relacionados por una red colaborativa.

6.1. Escenarios de prueba

Con base en los requisitos generales establecidos en capítulos anteriores y la manera en que la plataforma del prototipo construido los aborda; este capítulo pretende demostrar y validar sus resultados. Debido a que se tienen dos tipos de colaboración entre redes, en el proceso se han establecido dos escenarios de prueba para el prototipo.

Como resultado de la investigación en ciencia del diseño[2], se han definido los escenarios que permiten abordar un problema organizativo importante. Con el objetivo de demostrar su aplicabilidad, se han tomado datos de su salida ante un grupo de entradas definidas, es decir pruebas de artefacto o prototipo como caja negra.

El primer escenario que es para redes locales en un sólo puente, tiene como función validar el correcto funcionamiento de la red colaborativa local en una estructura puente, para lo cual se define a uno de sus nodos sensor, como nodo maestro y al resto, se les define como nodos sensor tipo esclavo. En una estructura es posible instalar n nodos sensor.

Por otro lado, el segundo escenario es más bien el que busca validar el protocolo de comunicación y alarmas entre estructuras distintas, donde los puentes están relacionados entre sí y dicha relación es determinada por la red colaborativa común a la que pertenecen.

Es importante hacer notar que en general, de escenario a escenario el hardware no varía mayormente, y específicamente para el escenario de demostración para redes de varios puentes; todas las pruebas se realizan sin ninguna variación de hardware.

Durante el desarrollo del capítulo se realiza la validación del funcionamiento del protocolo de alertas para prevención de desastres en ambos escenarios; por otro lado y para el escenario de varios puentes, se emplean dos tecnologías de comunicación distintas entre sí, que implican protocolos de comunicación diferentes para la red colaborativa global, es importante resaltar que la configuración de los nodos maestros es la que varía.

6.1.1. Escenario de demostración para redes en un puente

La figura 6.1 muestra la parte de la arquitectura del prototipo, que representa al escenario para redes locales de cada estructura puente; donde en una red local cada puente debería de contar con al menos los elementos que se presentan en la figura, es decir: n nodos sensor, un enlace de comunicación entre los nodos del puente y un enlace al exterior.

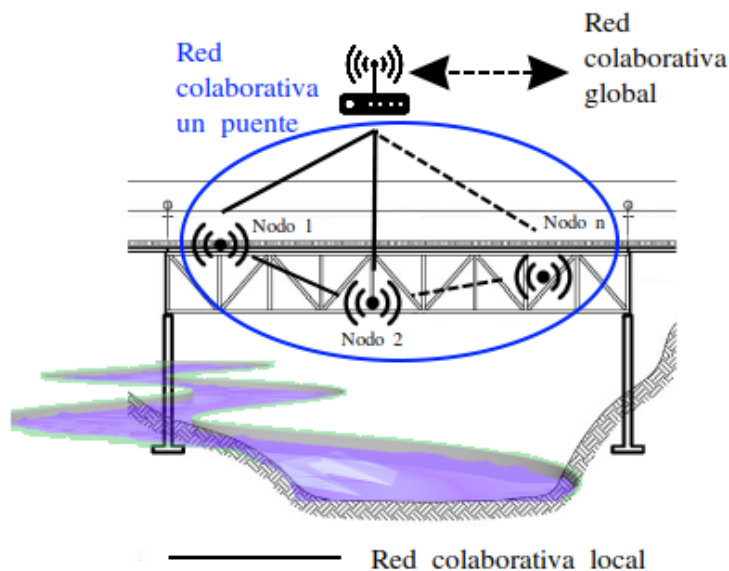


Figura 6.1: Escenario de demostración del protocolo para una estructura puente.

Para la realización de experimentos en este escenario se tomó el puente vehicular de la entrada norte del TEC y se instrumentó con dos nodos sensor, las pruebas se hicieron con cada uno de los nodos sensor midiendo la distancia entre el puente y el nivel del agua, cerca de uno de los extremos del puente. Todo en condiciones soleadas durante aproximadamente dos horas.

▼ 11:		▼ 10:	
idconfiguration:	35	idconfiguration:	36
idstructure:	806	idstructure:	806
iddevice:	100	iddevice:	102
measurefrequency:	0.5	measurefrequency:	0.5
messagefrequency:	0.5	messagefrequency:	0.5
sendfrequency:	0.5	sendfrequency:	0.5
height:	450	height:	450
tolerance:	20	tolerance:	20
mac:	"1"	mac:	"1"
phonenummer:	"1"	phonenummer:	"1"
ismaster:	true	ismaster:	false
ip:	"10.196.132.236"	ip:	"10.230.78.58"

Figura 6.2: Configuración de nodos para escenario de un solo puente.

Una configuración común de entradas para el escenario de un solo puente, se muestra en la figura 6.2, para el cual la red es conformada de manera local mediante la definición de uno de los empotrados como nodo sensor maestro, mientras que el resto de los nodos sensor son parte de la red, pero sin dicha etiqueta. El puente es previamente medido de forma manual (ver imagen 6.8) para definir el parámetro altura.

6.1.2. Escenario de demostración para redes de varios puentes

En la figura 6.4 se presenta el escenario de demostración del protocolo en funcionamiento para varias estructuras puente, que es utilizado para la validación del protocolo de red global. Una red colaborativa de puentes puede traslaparse con otra red distinta, y formar una gran red de puentes.

▼ 10:		▼ 11:		▼ 8:	
idconfiguration:	23	idconfiguration:	24	iddeviceinnet:	23
idstructure:	805	idstructure:	806	idnet:	8
iddevice:	60	iddevice:	62	iddevice:	60
measurefrequency:	0.5	measurefrequency:	0.5	▼ 9:	
messagefrequency:	0.5	messagefrequency:	0.5	iddeviceinnet:	24
sendfrequency:	0.5	sendfrequency:	0.5	idnet:	8
height:	372	height:	454	iddevice:	62
tolerance:	20	tolerance:	20		
mac:	"1"	mac:	"1"		
phonenummer:	"1"	phonenummer:	"1"		
ismaster:	true	ismaster:	true		
ip:	"10.195.2.128"	ip:	"10.138.42.52"		

Figura 6.3: Configuración de nodos para escenario de varios puentes.

Para ejemplificar el escenario, podría definirse una "Red colaborativa 1" para analizar un escenario de riesgo suponiendo que dos puentes están cerca de una ciudad

que podría quedar incomunicada; se define otra “Red colaborativa 2” para otro escenario donde hay riesgo de sobrecarga de capacidad hidráulica de puentes y de subsecuente inundación de áreas a lo largo del cauce de un río, por lo que se quiere monitorizar el nivel de agua con el fin de poder generar alertas a comunidades río abajo.

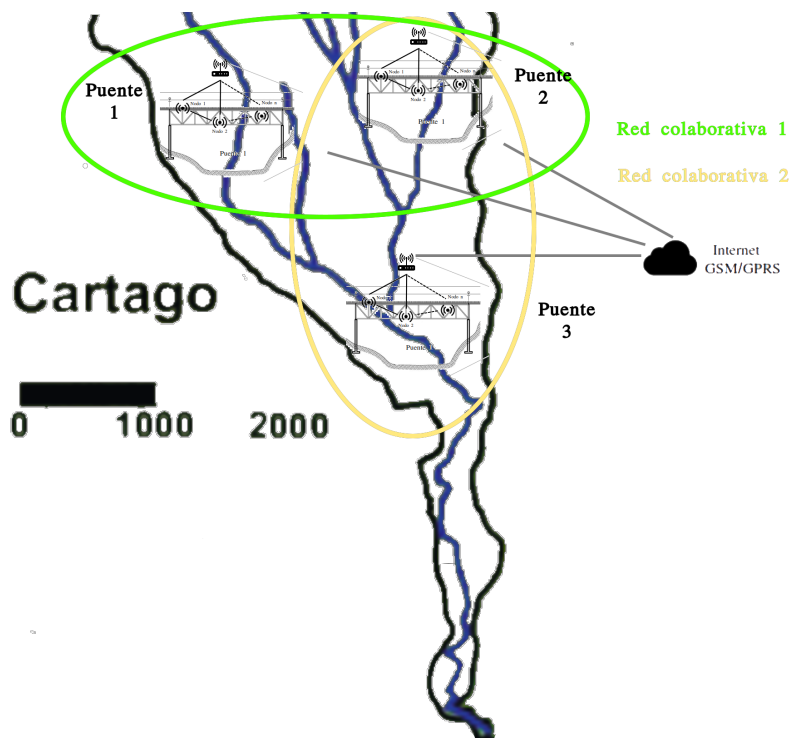


Figura 6.4: Escenario de demostración del protocolo global para varios puentes.

La configuración típica de este escenario con entradas definidas se muestra en la figura 6.3, donde ambos nodos pertenecen a la misma red colaborativa global (Red 8) y a diferentes puentes, que a su vez son medidos manualmente con alturas diferentes (ver imagen 6.8).

6.2. Pruebas funcionales

En esta sección se presentan los resultados obtenidos a partir del criterio de caja negra, donde se especifica cuales son las salidas ante entradas específicas del sistema. Los casos de uso son puestos a prueba mediante el planteamiento de resultados bajo condiciones definidas en los casos.

6.2.1. Pruebas de altura libre para escenario de un puente

Las mediciones tomadas con el caso de uso descrito en la sección 6.1.1. son mostradas en el cuadro 6.1 (SA se debe interpretar como el estado “Sin Alarma”).

Cuadro 6.1: Mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
100	PuenteEntrada	14:38:34	22/5/2019	433	SA	SA
102	PuenteEntrada	14:37:51	22/5/2019	422	SA	SA
100	PuenteEntrada	14:38:06	22/5/2019	434	SA	SA
102	PuenteEntrada	14:37:09	22/5/2019	423	Falso-Positivo	SA
100	PuenteEntrada	14:37:40	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:37:30	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:36:28	22/5/2019	420	Falso-Positivo	SA
100	PuenteEntrada	14:36:50	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:36:23	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:35:45	22/5/2019	421	Falso-Positivo	SA
100	PuenteEntrada	14:35:58	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:35:34	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:35:04	22/5/2019	423	Falso-Positivo	SA
100	PuenteEntrada	14:35:16	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:34:46	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:34:08	22/5/2019	421	Falso-Positivo	SA
100	PuenteEntrada	14:34:31	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:33:58	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:33:12	22/5/2019	429	Falso-Positivo	SA
100	PuenteEntrada	14:33:34	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:33:06	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:32:30	22/5/2019	422	Falso-Positivo	SA
100	PuenteEntrada	14:32:42	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:32:19	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:31:46	22/5/2019	426	Falso-Positivo	SA
100	PuenteEntrada	14:31:58	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:31:31	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:31:03	22/5/2019	423	Falso-Positivo	SA
100	PuenteEntrada	14:31:15	22/5/2019	68	Rojo	Rojo
100	PuenteEntrada	14:30:44	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:30:22	22/5/2019	420	Falso-Positivo	SA
100	PuenteEntrada	14:30:34	22/5/2019	67	Rojo	Rojo
102	PuenteEntrada	14:29:40	22/5/2019	420	Falso-Positivo	SA
100	PuenteEntrada	14:29:52	22/5/2019	193	Amarillo	Amarillo
100	PuenteEntrada	14:29:42	22/5/2019	198	Amarillo	Amarillo
102	PuenteEntrada	14:28:57	22/5/2019	421	Falso-Positivo	SA
100	PuenteEntrada	14:29:32	22/5/2019	207	Amarillo	Amarillo
102	PuenteEntrada	14:28:12	22/5/2019	421	SA	SA
100	PuenteEntrada	14:29:21	22/5/2019	434	SA	SA
100	PuenteEntrada	14:28:53	22/5/2019	432	SA	SA
102	PuenteEntrada	14:27:29	22/5/2019	423	SA	SA
100	PuenteEntrada	14:28:13	22/5/2019	440	SA	SA
100	PuenteEntrada	14:27:47	22/5/2019	437	SA	SA
102	PuenteEntrada	14:26:47	22/5/2019	421	SA	SA
100	PuenteEntrada	14:27:09	22/5/2019	433	SA	SA
102	PuenteEntrada	14:25:52	22/5/2019	420	SA	SA
100	PuenteEntrada	14:26:34	22/5/2019	434	SA	SA
100	PuenteEntrada	14:26:31	22/5/2019	436	SA	SA
102	PuenteEntrada	14:25:11	22/5/2019	420	SA	SA
100	PuenteEntrada	14:25:50	22/5/2019	432	SA	SA

Continúa en la siguiente página

Cuadro 6.1 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
102	PuenteEntrada	14:24:28	22/5/2019	421	SA	SA
100	PuenteEntrada	14:25:32	22/5/2019	431	SA	SA
100	PuenteEntrada	14:25:18	22/5/2019	438	SA	SA
102	PuenteEntrada	14:23:44	22/5/2019	422	SA	SA
100	PuenteEntrada	14:24:44	22/5/2019	432	SA	SA
102	PuenteEntrada	14:23:01	22/5/2019	421	SA	SA
100	PuenteEntrada	14:24:17	22/5/2019	435	SA	SA
100	PuenteEntrada	14:24:08	22/5/2019	434	SA	SA
102	PuenteEntrada	14:22:19	22/5/2019	423	SA	SA
100	PuenteEntrada	14:23:34	22/5/2019	439	SA	SA
102	PuenteEntrada	14:21:36	22/5/2019	422	SA	SA
100	PuenteEntrada	14:23:16	22/5/2019	434	SA	SA
100	PuenteEntrada	14:23:01	22/5/2019	439	SA	SA
102	PuenteEntrada	14:20:53	22/5/2019	421	SA	SA
100	PuenteEntrada	14:22:30	22/5/2019	437	SA	SA
102	PuenteEntrada	14:20:08	22/5/2019	421	SA	SA
100	PuenteEntrada	14:22:06	22/5/2019	439	SA	SA
100	PuenteEntrada	14:21:52	22/5/2019	437	SA	SA
102	PuenteEntrada	14:19:26	22/5/2019	421	SA	SA
100	PuenteEntrada	14:21:22	22/5/2019	431	SA	SA
102	PuenteEntrada	14:18:42	22/5/2019	422	SA	SA
100	PuenteEntrada	14:21:05	22/5/2019	436	SA	SA
100	PuenteEntrada	14:20:37	22/5/2019	432	SA	SA
102	PuenteEntrada	14:17:56	22/5/2019	421	SA	SA
100	PuenteEntrada	14:20:16	22/5/2019	437	SA	SA
102	PuenteEntrada	14:17:14	22/5/2019	421	SA	SA
100	PuenteEntrada	14:19:54	22/5/2019	430	SA	SA
102	PuenteEntrada	14:16:30	22/5/2019	425	SA	SA
100	PuenteEntrada	14:19:36	22/5/2019	430	SA	SA
100	PuenteEntrada	14:19:23	22/5/2019	438	SA	SA
102	PuenteEntrada	14:15:47	22/5/2019	424	SA	SA
100	PuenteEntrada	14:19:19	22/5/2019	435	SA	SA
102	PuenteEntrada	14:15:00	22/5/2019	423	SA	SA
100	PuenteEntrada	14:18:38	22/5/2019	430	SA	SA
100	PuenteEntrada	14:18:16	22/5/2019	440	SA	SA
102	PuenteEntrada	14:14:16	22/5/2019	420	SA	SA
100	PuenteEntrada	14:17:56	22/5/2019	438	SA	SA
102	PuenteEntrada	14:13:33	22/5/2019	421	Falso-Positivo	SA
100	PuenteEntrada	14:14:19	22/5/2019	443	SA	SA
102	PuenteEntrada	14:12:50	22/5/2019	421	SA	SA
100	PuenteEntrada	14:13:57	22/5/2019	438	SA	SA
100	PuenteEntrada	14:13:46	22/5/2019	437	SA	SA
102	PuenteEntrada	14:12:07	22/5/2019	426	SA	SA
100	PuenteEntrada	14:12:27	22/5/2019	433	SA	SA
102	PuenteEntrada	14:11:25	22/5/2019	423	SA	SA
100	PuenteEntrada	14:12:11	22/5/2019	439	SA	SA
100	PuenteEntrada	14:12:10	22/5/2019	436	SA	SA
102	PuenteEntrada	14:10:43	22/5/2019	423	SA	SA
100	PuenteEntrada	14:11:34	22/5/2019	433	SA	SA
100	PuenteEntrada	14:11:07	22/5/2019	433	SA	SA
102	PuenteEntrada	14:09:57	22/5/2019	422	SA	SA

Continúa en la siguiente página

Cuadro 6.1 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
100	PuenteEntrada	14:10:27	22/5/2019	436	SA	SA
100	PuenteEntrada	14:10:12	22/5/2019	439	SA	SA
102	PuenteEntrada	14:09:12	22/5/2019	423	SA	SA
100	PuenteEntrada	14:09:36	22/5/2019	435	SA	SA
100	PuenteEntrada	14:09:09	22/5/2019	435	SA	SA
102	PuenteEntrada	14:08:30	22/5/2019	423	SA	SA
100	PuenteEntrada	14:08:45	22/5/2019	438	SA	SA
102	PuenteEntrada	14:07:46	22/5/2019	427	SA	SA
100	PuenteEntrada	14:08:23	22/5/2019	431	SA	SA
100	PuenteEntrada	14:07:57	22/5/2019	437	SA	SA
102	PuenteEntrada	14:07:02	22/5/2019	425	SA	SA
100	PuenteEntrada	14:07:41	22/5/2019	436	SA	SA
102	PuenteEntrada	14:06:19	22/5/2019	7	Falso-Positivo	Rojo
100	PuenteEntrada	14:07:21	22/5/2019	434	SA	SA
100	PuenteEntrada	14:06:58	22/5/2019	433	SA	SA
100	PuenteEntrada	14:05:59	22/5/2019	7	Rojo	Rojo
102	PuenteEntrada	14:05:37	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	14:05:49	22/5/2019	7	Rojo	Rojo
102	PuenteEntrada	14:04:56	22/5/2019	10	Rojo	Rojo
100	PuenteEntrada	14:05:08	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	14:04:55	22/5/2019	5	Rojo	Rojo
102	PuenteEntrada	14:04:14	22/5/2019	14	Rojo	Rojo
100	PuenteEntrada	14:04:33	22/5/2019	8	Rojo	Rojo
102	PuenteEntrada	14:03:21	22/5/2019	429	Falso-Positivo	SA
100	PuenteEntrada	14:04:23	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	14:04:07	22/5/2019	7	Rojo	Rojo
102	PuenteEntrada	14:02:35	22/5/2019	429	SA	SA
100	PuenteEntrada	14:03:14	22/5/2019	435	SA	SA
100	PuenteEntrada	14:02:51	22/5/2019	439	SA	SA
102	PuenteEntrada	14:01:53	22/5/2019	421	SA	SA
100	PuenteEntrada	14:02:11	22/5/2019	433	SA	SA
102	PuenteEntrada	14:01:07	22/5/2019	426	SA	SA
100	PuenteEntrada	14:01:53	22/5/2019	436	SA	SA
100	PuenteEntrada	14:01:43	22/5/2019	430	SA	SA
102	PuenteEntrada	14:00:20	22/5/2019	422	SA	SA
100	PuenteEntrada	14:00:50	22/5/2019	435	SA	SA
100	PuenteEntrada	14:00:33	22/5/2019	434	SA	SA
102	PuenteEntrada	13:59:37	22/5/2019	426	SA	SA
100	PuenteEntrada	13:59:55	22/5/2019	435	SA	SA
100	PuenteEntrada	13:59:35	22/5/2019	435	SA	SA
102	PuenteEntrada	13:58:51	22/5/2019	424	Falso-Positivo	SA
100	PuenteEntrada	13:59:02	22/5/2019	7	Rojo	Rojo
102	PuenteEntrada	13:58:01	22/5/2019	428	Falso-Positivo	SA
100	PuenteEntrada	13:58:22	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	13:58:21	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	13:57:32	22/5/2019	8	Rojo	Rojo
102	PuenteEntrada	13:57:10	22/5/2019	423	Falso-Positivo	SA
100	PuenteEntrada	13:57:22	22/5/2019	8	Rojo	Rojo
102	PuenteEntrada	13:56:25	22/5/2019	428	Falso-Positivo	SA
100	PuenteEntrada	13:56:45	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	13:56:35	22/5/2019	9	Rojo	Rojo

Continúa en la siguiente página

Cuadro 6.1 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
102	PuenteEntrada	13:55:35	22/5/2019	423	Falso-Positivo	SA
100	PuenteEntrada	13:56:25	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	13:55:42	22/5/2019	433	SA	SA
102	PuenteEntrada	13:54:52	22/5/2019	422	SA	SA
100	PuenteEntrada	13:55:15	22/5/2019	436	SA	SA
102	PuenteEntrada	13:54:10	22/5/2019	423	SA	SA
100	PuenteEntrada	13:54:52	22/5/2019	434	SA	SA
100	PuenteEntrada	13:54:39	22/5/2019	431	SA	SA
102	PuenteEntrada	13:53:19	22/5/2019	426	SA	SA
100	PuenteEntrada	13:53:58	22/5/2019	432	SA	SA
100	PuenteEntrada	13:53:42	22/5/2019	433	SA	SA
102	PuenteEntrada	13:52:34	22/5/2019	422	SA	SA
100	PuenteEntrada	13:52:55	22/5/2019	435	SA	SA
100	PuenteEntrada	13:52:18	22/5/2019	434	SA	SA
102	PuenteEntrada	13:51:49	22/5/2019	426	SA	SA
100	PuenteEntrada	13:52:06	22/5/2019	438	SA	SA
102	PuenteEntrada	13:51:02	22/5/2019	429	SA	SA
100	PuenteEntrada	13:51:26	22/5/2019	434	SA	SA
100	PuenteEntrada	13:51:02	22/5/2019	437	SA	SA
102	PuenteEntrada	13:50:19	22/5/2019	422	SA	SA
100	PuenteEntrada	13:50:41	22/5/2019	437	SA	SA
100	PuenteEntrada	13:50:06	22/5/2019	436	SA	SA
102	PuenteEntrada	13:49:32	22/5/2019	424	SA	SA
100	PuenteEntrada	13:49:56	22/5/2019	437	SA	SA
102	PuenteEntrada	13:48:48	22/5/2019	423	SA	SA

En el cuadro 6.2 se muestran los datos obtenidos mediante los cálculos realizados por medio del *script* llamado “LatexTableMakerV10.py”, cuya entrada es el archivo “DatosEscenrio1PuenteCajaNegra.txt”.

Cuadro 6.2: Variables estadísticas de los datos tomados para el escenario de red colaborativa de un sólo puente, cuadro 6.1

Red	Variable estadística	Cálculo
5	STD	6.562047855279957
	Moda	421
	Promedio	429.5839416058394
	Media	431.0

Las variables moda, promedio y media son agregadas para su valoración y análisis, que además son los resultados de la implementación de filtro de mediana móvil dado en el capítulo de implementación.

La desviación estándar es de 6.56, ignorando cualquier valor menor a 420 cm, puesto que son discontinuidades provocadas para alterar las entradas del sistema y generar cambios de alerta.

La distribución normal de los datos se muestra en la figura 6.5, donde se aprecia que el prototipo realiza una medición de distancia donde la mayor parte de los datos se concentra entre -1σ (424,437952145) y 1σ (437,562047855).

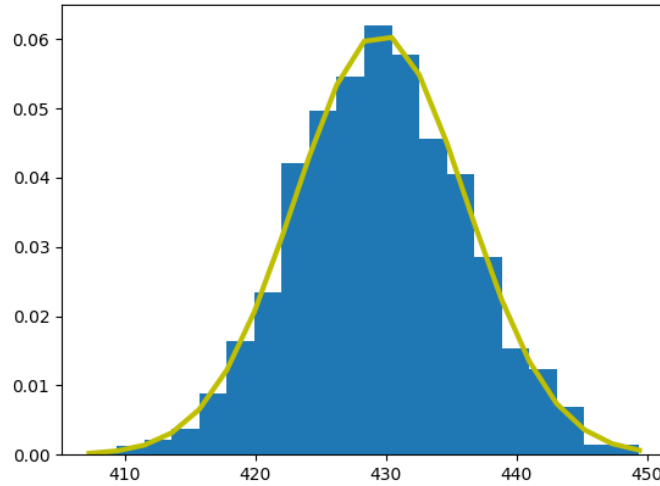


Figura 6.5: Gráfico de la distribución normal de los datos del cuadro 6.1.

Es posible concluir de esta prueba que el artefacto o caja negra, responde ante estímulos externos que varían la distancia medida entre la estructura y el nivel del agua que pasa bajo el puente, respondiendo así con la activación del protocolo de alertas para emergencias, lo anterior se aprecia con el cambio de SA a otro estado en el cuadro 6.1.

6.2.2. Pruebas de altura libre para escenario de varios puentes

Al igual que para el escenario de un solo puente, las pruebas realizadas consisten en evaluar el escenario de varios puentes como artefacto [2] por método de observabilidad; estas pruebas sugiere un apoyo desde otra arista a la hipótesis planteada al inicio de la investigación, donde la filosofía de colaboración entre nodos se usa para detectar desastres naturales.

Los resultados de medición de altura libre para el escenario para varios puentes se muestran en el cuadro 6.3. Para esta prueba los resultados corresponden a la toma de datos del escenario ante entradas definidas de la misma forma que se muestra en la configuración típica dada en 6.1.2.

Cuadro 6.3: Mediciones de red global - redes de varios puentes

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
62	PuenteEntrada	13:40:28	18/5/2019	427	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:39:20	18/5/2019	428	Sin-Alarma	Sin-Alarma

Continúa en la siguiente página

Cuadro 6.3 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
62	PuenteEntrada	13:38:14	18/5/2019	429	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:37:16	18/5/2019	427	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:36:23	18/5/2019	426	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:35:10	18/5/2019	426	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:34:07	18/5/2019	425	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:33:12	18/5/2019	426	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:32:16	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:31:33	18/5/2019	395	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:31:09	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:30:45	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:30:06	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:29:56	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:29:18	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:29:13	18/5/2019	394	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:28:29	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:28:21	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:27:34	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:27:25	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:26:44	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:26:19	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:25:44	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:24:46	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:24:54	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:24:02	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:24:03	18/5/2019	428	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:23:14	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:23:09	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:21:21	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:22:10	18/5/2019	428	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:21:07	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:20:01	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:20:20	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:19:11	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:19:20	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:18:23	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:18:28	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:17:35	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:17:26	18/5/2019	429	Sin-Alarma	Rojo
60	PuenteResidencias	13:16:48	18/5/2019	60	Rojo	Rojo
62	PuenteEntrada	13:16:33	18/5/2019	427	Sin-Alarma	Rojo
60	PuenteResidencias	13:15:59	18/5/2019	60	Rojo	Rojo
60	PuenteResidencias	13:15:08	18/5/2019	60	Rojo	Rojo
62	PuenteEntrada	13:14:33	18/5/2019	426	Sin-Alarma	Rojo
60	PuenteResidencias	13:14:21	18/5/2019	60	Rojo	Rojo
60	PuenteResidencias	13:13:32	18/5/2019	60	Rojo	Rojo
62	PuenteEntrada	13:13:24	18/5/2019	431	Sin-Alarma	Rojo
60	PuenteResidencias	13:12:46	18/5/2019	61	Rojo	Rojo
60	PuenteResidencias	13:11:58	18/5/2019	61	Rojo	Rojo
62	PuenteEntrada	13:12:01	18/5/2019	426	Sin-Alarma	Amarillo
60	PuenteResidencias	13:11:02	18/5/2019	152	Amarillo	Amarillo
62	PuenteEntrada	13:11:00	18/5/2019	426	Sin-Alarma	Amarillo

Continúa en la siguiente página

Cuadro 6.3 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
60	PuenteResidencias	13:10:13	18/5/2019	151	Amarillo	Amarillo
62	PuenteEntrada	13:10:05	18/5/2019	427	Sin-Alarma	Amarillo
60	PuenteResidencias	13:09:24	18/5/2019	171	Amarillo	Amarillo
60	PuenteResidencias	13:08:39	18/5/2019	188	Verde	Verde
62	PuenteEntrada	13:08:37	18/5/2019	426	Sin-Alarma	Amarillo
60	PuenteResidencias	13:07:52	18/5/2019	151	Amarillo	Amarillo
62	PuenteEntrada	13:07:41	18/5/2019	426	Sin-Alarma	Amarillo
60	PuenteResidencias	13:07:06	18/5/2019	152	Amarillo	Amarillo
62	PuenteEntrada	13:06:46	18/5/2019	426	Sin-Alarma	Amarillo
60	PuenteResidencias	13:06:16	18/5/2019	170	Amarillo	Amarillo
62	PuenteEntrada	13:05:44	18/5/2019	426	Sin-Alarma	Amarillo
60	PuenteResidencias	13:05:29	18/5/2019	147	Amarillo	Amarillo
60	PuenteResidencias	13:04:43	18/5/2019	222	Verde	Verde
62	PuenteEntrada	13:04:43	18/5/2019	427	Sin-Alarma	Verde
60	PuenteResidencias	13:03:55	18/5/2019	222	Verde	Verde
62	PuenteEntrada	13:03:52	18/5/2019	426	Sin-Alarma	Verde
60	PuenteResidencias	13:03:08	18/5/2019	221	Verde	Verde
62	PuenteEntrada	13:02:52	18/5/2019	425	Sin-Alarma	Verde
60	PuenteResidencias	13:02:21	18/5/2019	221	Verde	Verde
62	PuenteEntrada	13:02:00	18/5/2019	425	Sin-Alarma	Verde
60	PuenteResidencias	13:01:34	18/5/2019	222	Verde	Verde
62	PuenteEntrada	13:01:03	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:00:46	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	13:00:03	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	13:00:01	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:59:20	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:58:54	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:58:32	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:57:50	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:57:49	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:57:06	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:57:04	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:56:23	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:56:11	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:55:40	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:55:13	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:54:53	18/5/2019	394	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:54:05	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:53:18	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:53:22	18/5/2019	426	Sin-Alarma	Rojo
60	PuenteResidencias	12:52:31	18/5/2019	10	Rojo	Rojo
62	PuenteEntrada	12:52:22	18/5/2019	427	Sin-Alarma	Rojo
60	PuenteResidencias	12:51:44	18/5/2019	11	Rojo	Rojo
62	PuenteEntrada	12:51:28	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:50:57	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:50:29	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:50:13	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:49:30	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:49:30	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:48:47	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:48:23	18/5/2019	426	Sin-Alarma	Sin-Alarma

Continúa en la siguiente página

Cuadro 6.3 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
60	PuenteResidencias	12:48:04	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:47:16	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:46:59	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:46:27	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:45:50	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:45:44	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:44:56	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:44:44	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:44:13	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:43:31	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:43:30	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:42:47	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:42:41	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:42:04	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:41:44	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:41:21	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:40:50	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:40:33	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:39:50	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:39:47	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:39:07	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:38:55	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:38:24	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:38:02	18/5/2019	428	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:37:40	18/5/2019	393	Sin-Alarma	Rojo
62	PuenteEntrada	12:36:58	18/5/2019	5	Rojo	Rojo
60	PuenteResidencias	12:36:53	18/5/2019	393	Sin-Alarma	Rojo
60	PuenteResidencias	12:36:10	18/5/2019	393	Sin-Alarma	Rojo
62	PuenteEntrada	12:35:56	18/5/2019	6	Rojo	Rojo
60	PuenteResidencias	12:35:14	18/5/2019	393	Sin-Alarma	Rojo
62	PuenteEntrada	12:35:02	18/5/2019	7	Rojo	Rojo
60	PuenteResidencias	12:34:26	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:34:20	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:33:34	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:33:35	18/5/2019	426	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:32:43	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:32:41	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:31:52	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:31:55	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:31:04	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:31:06	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:30:21	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:30:20	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:29:33	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:29:29	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:28:50	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:28:32	18/5/2019	428	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:28:07	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:27:19	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:27:19	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:26:36	18/5/2019	393	Sin-Alarma	Sin-Alarma

Continúa en la siguiente página

Cuadro 6.3 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
62	PuenteEntrada	12:26:24	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:25:48	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:25:29	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:25:05	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:24:40	18/5/2019	429	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:24:17	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:23:49	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:23:29	18/5/2019	391	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:22:59	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:22:46	18/5/2019	392	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:22:04	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:22:00	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:21:21	18/5/2019	391	Sin-Alarma	Rojo
62	PuenteEntrada	12:21:03	18/5/2019	10	Rojo	Rojo
60	PuenteResidencias	12:20:39	18/5/2019	393	Sin-Alarma	Rojo
62	PuenteEntrada	12:20:15	18/5/2019	10	Rojo	Rojo
60	PuenteResidencias	12:19:56	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:19:18	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:19:08	18/5/2019	393	Sin-Alarma	Amarillo
60	PuenteResidencias	12:18:24	18/5/2019	391	Sin-Alarma	Amarillo
62	PuenteEntrada	12:17:39	18/5/2019	180	Amarillo	Amarillo
60	PuenteResidencias	12:17:42	18/5/2019	391	Sin-Alarma	Amarillo
60	PuenteResidencias	12:16:54	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:16:42	18/5/2019	429	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:16:11	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:15:55	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:15:23	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:14:46	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:14:40	18/5/2019	393	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:13:57	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:13:57	18/5/2019	428	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:13:14	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:13:06	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:12:32	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:12:20	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:11:44	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:11:32	18/5/2019	428	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:10:54	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:10:42	18/5/2019	426	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:10:06	18/5/2019	392	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:09:55	18/5/2019	425	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:09:23	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:08:55	18/5/2019	431	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:08:37	18/5/2019	393	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:08:06	18/5/2019	430	Sin-Alarma	Rojo
60	PuenteResidencias	12:07:50	18/5/2019	81	Rojo	Rojo
62	PuenteEntrada	12:07:14	18/5/2019	427	Sin-Alarma	Rojo
60	PuenteResidencias	12:07:03	18/5/2019	81	Rojo	Rojo
62	PuenteEntrada	12:06:24	18/5/2019	426	Sin-Alarma	Rojo
60	PuenteResidencias	12:06:15	18/5/2019	81	Rojo	Rojo
62	PuenteEntrada	12:05:08	18/5/2019	426	Sin-Alarma	Rojo

Continúa en la siguiente página

Cuadro 6.3 – continuación de mediciones de red local - Un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
60	PuenteResidencias	12:05:18	18/5/2019	82	Rojo	Rojo
60	PuenteResidencias	12:04:31	18/5/2019	82	Rojo	Rojo
62	PuenteEntrada	12:04:21	18/5/2019	426	Sin-Alarma	Rojo
60	PuenteResidencias	12:03:44	18/5/2019	82	Rojo	Rojo
62	PuenteEntrada	12:03:36	18/5/2019	426	Sin-Alarma	Rojo
60	PuenteResidencias	12:02:49	18/5/2019	78	Rojo	Rojo
62	PuenteEntrada	12:02:45	18/5/2019	427	Sin-Alarma	Verde
60	PuenteResidencias	12:01:59	18/5/2019	278	Verde	Verde
62	PuenteEntrada	12:01:48	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:00:56	18/5/2019	394	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:00:48	18/5/2019	429	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	12:00:13	18/5/2019	281	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	12:00:01	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	11:59:25	18/5/2019	280	Sin-Alarma	Sin-Alarma
62	PuenteEntrada	11:59:14	18/5/2019	427	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	11:58:39	18/5/2019	290	Sin-Alarma	Sin-Alarma
60	PuenteResidencias	11:57:55	18/5/2019	273	Verde	Verde

Las mediciones, mostradas en el cuadro 6.3 se analizan de forma separada por puente. De la misma forma que se hizo para el escenario de un puente, se cuenta con una forma automática de cálculo de las variables estadísticas de los datos, mediante el script: “LatexTableMakerV10.py”.

Para el puente denominado en la configuración como “PuenteEntrada”, se muestra la distribución normal de sus mediciones en la figura 6.6. Para el puente “PuenteResidencias” es en la figura 6.7 que se proporciona la gráfica normal de los datos adquiridos en el escenario de varios puentes.

En el cuadro 6.4 se muestran los datos obtenidos mediante el *script* llamado “LatexTableMakerV10.py”, cuya entrada es el archivo “DatosEscenarioVariosPuentes-CajaNegra.txt”. Para ambos puentes se tiene que el prototipo realiza mediciones con distribución normal, la mayor parte de los datos se concentra entre -1σ (424,762 para PuenteEntrada y 392,273 para PuenteResidencias) y 1σ (427,238 para PuenteEntrada y 393,727 para PuenteResidencias).

Cuadro 6.4: Variables estadísticas de los datos tomados para el escenario de varios puentes, cuadro 6.3

Red/Puente	Variable estadística	Cálculo
8 PuenteEntrada	STD	1.2377260793743967
	Moda	426
	Promedio	426.5612244897959
	Media	426.0
8 PuenteResidencias	STD	0.7267503628932
	Moda	393
	Promedio	392.9647058823529
	Media	393.0

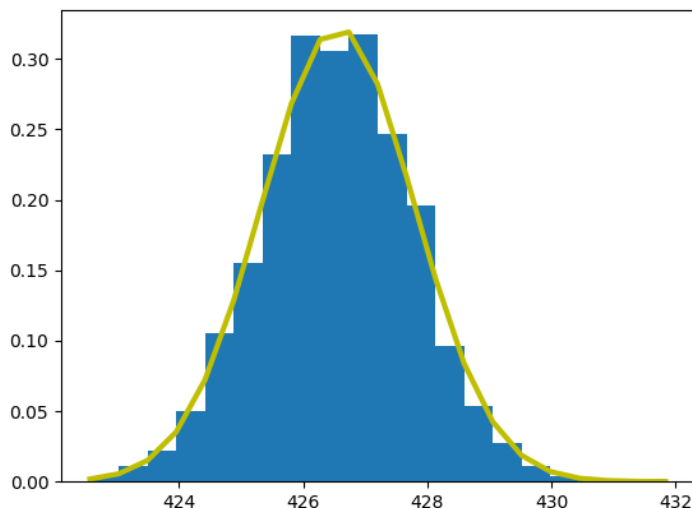


Figura 6.6: Gráfico de la distribución normal de los datos del cuadro 6.3 para el puente llamado "PuenteEntrada".

Cada puente fue medido manualmente como se muestra en la figura 6.8, con el fin de establecer su distancia entre la estructura y el nivel del agua del río al menos una vez con el fin de establecer una referencia por puente y así poder configurar el parámetro de altura para cada puente.

La medición manual se ha tomado durante las pruebas para la configuración del parámetro de altura de los puentes, que a su vez es utilizado por el sistema para realizar los cálculos del 25%, 50% y 75% que define a los estados de alerta de cada puente, tal y como se solicitó por parte del grupo de expertos del CIVCO.

Es posible esperar respecto a esta prueba sobre el artefacto, que también hay una respuesta ante estímulos externos de variación de la altura libre de los puentes, que activa el protocolo de alertas y a su vez hay una propagación mediante el cambio de estado de Alerta Global en puentes ajenos al disparo de alerta local pero que conforman la misma red colaborativa (ver cuadro 6.3).

6.2.3. Protocolo de comunicación para estructuras - local

Esta prueba, se realiza con el objetivo de comprobar el funcionamiento del protocolo de comunicación entre nodos sensor de una sola estructura puente. Todos los nodos sensor instalados y configurados en un puente, deben compartir sus mediciones, así también como sus estados de alerta individuales, de forma tal que se pueda establecer una alerta global conjunta para el puente.

La prueba se basa en trasegar el envío de información entre puntos de medición, además de la comprobación de una toma de decisiones conjunta entre los mismos. El envío de la información en una red local, se realiza por medio de los módulos Xbee.

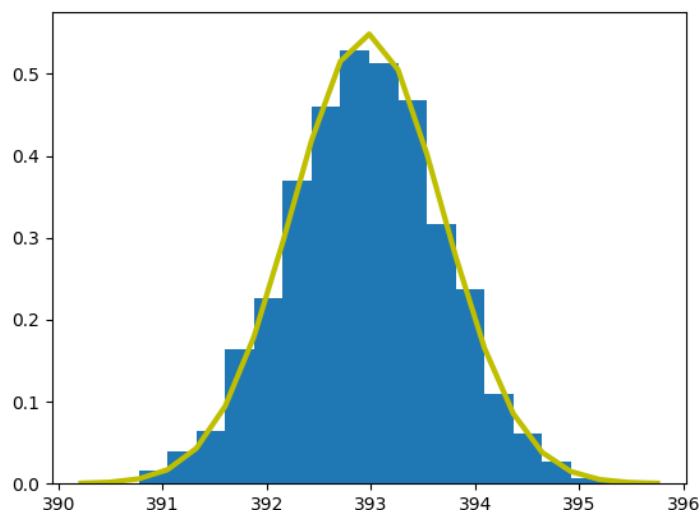


Figura 6.7: Gráfico de la distribución normal de los datos del cuadro 6.3 para el puente llamado "PuenteResidencias".

En el cuadro 6.5 se muestra la secuencia de estados de alerta, transmitidos a través de la comunicación de red local (Xbee), junto con su hora de propagación. Esta última es de suma importancia, puesto que es la que indica el momento en que un nodo sensor recibe un mensaje por parte de otro. El lector debe interpretar los datos de la columna H.P. como "Hora de Propagación"

Cuadro 6.5: Mensajes transmitidos entre nodos sensor maestro y esclavo para el escenario de un solo puente y su hora de propagación

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global	H.P.
100	PuenteEntrada	14:37:40	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:37:30	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:36:50	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:36:23	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:35:58	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:35:34	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:35:16	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:34:46	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:34:31	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:33:58	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:33:34	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:33:06	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:32:42	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:32:19	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:31:58	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:31:31	22/5/2019	67	Rojo	Rojo	14:30:34

Continúa en la siguiente página

Cuadro 6.5 – continuación de mensajes transmitidos - un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global	H.P.
100	PuenteEntrada	14:31:15	22/5/2019	68	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:30:44	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:30:34	22/5/2019	67	Rojo	Rojo	14:30:34
100	PuenteEntrada	14:29:52	22/5/2019	193	Amarillo	Amarillo	14:29:32
100	PuenteEntrada	14:29:42	22/5/2019	198	Amarillo	Amarillo	14:29:32
100	PuenteEntrada	14:29:32	22/5/2019	207	Amarillo	Amarillo	14:29:32
102	PuenteEntrada	14:06:19	22/5/2019	7	Falso-Positivo	Rojo	14:04:14
100	PuenteEntrada	14:05:59	22/5/2019	7	Rojo	Rojo	14:04:07
102	PuenteEntrada	14:05:37	22/5/2019	8	Rojo	Rojo	14:04:14
100	PuenteEntrada	14:05:49	22/5/2019	7	Rojo	Rojo	14:04:07
102	PuenteEntrada	14:04:56	22/5/2019	10	Rojo	Rojo	14:04:14
100	PuenteEntrada	14:05:08	22/5/2019	7	Rojo	Rojo	14:04:07
100	PuenteEntrada	14:04:55	22/5/2019	5	Rojo	Rojo	14:04:07
102	PuenteEntrada	14:04:14	22/5/2019	14	Rojo	Rojo	14:04:14
100	PuenteEntrada	14:04:33	22/5/2019	8	Rojo	Rojo	14:04:07
100	PuenteEntrada	14:04:23	22/5/2019	7	Rojo	Rojo	14:04:07
100	PuenteEntrada	14:04:07	22/5/2019	7	Rojo	Rojo	14:04:07
100	PuenteEntrada	13:59:02	22/5/2019	7	Rojo	Rojo	13:56:25
100	PuenteEntrada	13:58:22	22/5/2019	7	Rojo	Rojo	13:56:25
100	PuenteEntrada	13:58:21	22/5/2019	7	Rojo	Rojo	13:56:25
100	PuenteEntrada	13:57:32	22/5/2019	8	Rojo	Rojo	13:56:25
100	PuenteEntrada	13:57:22	22/5/2019	8	Rojo	Rojo	13:56:25
100	PuenteEntrada	13:56:45	22/5/2019	8	Rojo	Rojo	13:56:25
100	PuenteEntrada	13:56:35	22/5/2019	9	Rojo	Rojo	13:56:25
100	PuenteEntrada	13:56:25	22/5/2019	8	Rojo	Rojo	13:56:25

De acuerdo a los resultados del cuadro 6.5 se puede afirmar que la comunicación de red local se da entre nodos sensor, puesto que hay una toma de decisión consensuada por ambos nodos sensor tanto para determinar el estado de Alerta Global del puente, como para un Falso-Positivo. Es importante entender que en este caso, Alerta Global es el estado definitivo del puente.

6.2.4. Protocolo de comunicación para redes de estructuras- global

Esta prueba se realiza con el objetivo de comprobar el funcionamiento del protocolo de comunicación entre nodos sensor maestros. Los estados de alerta deben ser compartidos entre puentes por medio de su nodo sensor maestro.

En el caso de uso de varios puentes, la prueba consiste en hacer un rastreo del trasiego de mensajes entre nodos sensor maestro de cada puente, donde se espera que los puentes que sean parte de una misma red colaborativa sean los que compartan información y modifiquen el estado de los demás en función de la generación de alertas.



Figura 6.8: Medición manual de la altura de cada puente.

Cuadro 6.6: Mensajes transmitidos entre nodos sensor maestro para el escenario de varios puentes y su hora de propagación

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global	H.P.
62	PuenteEntrada	13:17:26	18/5/2019	429	SA	Rojo	13:12:03
60	PuenteResidencias	13:16:48	18/5/2019	60	Rojo	Rojo	13:11:58
62	PuenteEntrada	13:16:33	18/5/2019	427	SA	Rojo	13:12:03
60	PuenteResidencias	13:15:59	18/5/2019	60	Rojo	Rojo	13:11:58
60	PuenteResidencias	13:15:08	18/5/2019	60	Rojo	Rojo	13:11:58
62	PuenteEntrada	13:14:33	18/5/2019	426	SA	Rojo	13:12:03
60	PuenteResidencias	13:14:21	18/5/2019	60	Rojo	Rojo	13:11:58
60	PuenteResidencias	13:13:32	18/5/2019	60	Rojo	Rojo	13:11:58
62	PuenteEntrada	13:13:24	18/5/2019	431	SA	Rojo	13:12:03
60	PuenteResidencias	13:12:46	18/5/2019	61	Rojo	Rojo	13:11:58
60	PuenteResidencias	13:11:58	18/5/2019	61	Rojo	Rojo	13:11:58
62	PuenteEntrada	13:12:01	18/5/2019	426	SA	Amarillo	13:09:31
60	PuenteResidencias	13:11:02	18/5/2019	152	Amarillo	Amarillo	13:09:24
62	PuenteEntrada	13:11:00	18/5/2019	426	SA	Amarillo	13:09:31
60	PuenteResidencias	13:10:13	18/5/2019	151	Amarillo	Amarillo	13:09:24
62	PuenteEntrada	13:10:05	18/5/2019	427	SA	Amarillo	13:09:31
60	PuenteResidencias	13:09:24	18/5/2019	171	Amarillo	Amarillo	13:09:24
60	PuenteResidencias	13:08:39	18/5/2019	188	Verde	Verde	13:08:39
62	PuenteEntrada	13:08:37	18/5/2019	426	SA	Amarillo	13:05:34
60	PuenteResidencias	13:07:52	18/5/2019	151	Amarillo	Amarillo	13:05:29
62	PuenteEntrada	13:07:41	18/5/2019	426	SA	Amarillo	13:05:34
60	PuenteResidencias	13:07:06	18/5/2019	152	Amarillo	Amarillo	13:05:29
62	PuenteEntrada	13:06:46	18/5/2019	426	SA	Amarillo	13:05:34
60	PuenteResidencias	13:06:16	18/5/2019	170	Amarillo	Amarillo	13:05:29
62	PuenteEntrada	13:05:44	18/5/2019	426	SA	Amarillo	13:05:34
60	PuenteResidencias	13:05:29	18/5/2019	147	Amarillo	Amarillo	13:05:29
60	PuenteResidencias	13:04:43	18/5/2019	222	Verde	Verde	13:01:34
62	PuenteEntrada	13:04:43	18/5/2019	427	SA	Verde	13:04:01

Continúa en la siguiente página

Cuadro 6.6 – continuación de mensajes entre de varios puentes y su hora de propagación

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global	H.P.
60	PuenteResidencias	13:03:55	18/5/2019	222	Verde	Verde	13:01:34
62	PuenteEntrada	13:03:52	18/5/2019	426	SA	Verde	13:03:13
60	PuenteResidencias	13:03:08	18/5/2019	221	Verde	Verde	13:01:34
62	PuenteEntrada	13:02:52	18/5/2019	425	SA	Verde	13:02:26
60	PuenteResidencias	13:02:21	18/5/2019	221	Verde	Verde	13:01:34
62	PuenteEntrada	13:02:00	18/5/2019	425	SA	Verde	13:01:40
60	PuenteResidencias	13:01:34	18/5/2019	222	Verde	Verde	13:01:34
62	PuenteEntrada	12:53:22	18/5/2019	426	SA	Rojo	12:51:50
60	PuenteResidencias	12:52:31	18/5/2019	10	Rojo	Rojo	12:51:44
62	PuenteEntrada	12:52:22	18/5/2019	427	SA	Rojo	12:51:50
60	PuenteResidencias	12:51:44	18/5/2019	11	Rojo	Rojo	12:51:44
60	PuenteResidencias	12:37:40	18/5/2019	393	SA	Rojo	12:35:10
62	PuenteEntrada	12:36:58	18/5/2019	5	Rojo	Rojo	12:35:02
60	PuenteResidencias	12:36:53	18/5/2019	393	SA	Rojo	12:35:10
60	PuenteResidencias	12:36:10	18/5/2019	393	SA	Rojo	12:35:10
62	PuenteEntrada	12:35:56	18/5/2019	6	Rojo	Rojo	12:35:02
60	PuenteResidencias	12:35:14	18/5/2019	393	SA	Rojo	12:35:10
62	PuenteEntrada	12:35:02	18/5/2019	7	Rojo	Rojo	12:35:02
60	PuenteResidencias	12:21:21	18/5/2019	391	SA	Rojo	12:20:17
62	PuenteEntrada	12:21:03	18/5/2019	10	Rojo	Rojo	12:20:15
60	PuenteResidencias	12:20:39	18/5/2019	393	SA	Rojo	12:20:17
62	PuenteEntrada	12:20:15	18/5/2019	10	Rojo	Rojo	12:20:15
60	PuenteResidencias	12:19:08	18/5/2019	393	SA	Amarillo	12:17:41
60	PuenteResidencias	12:18:24	18/5/2019	391	SA	Amarillo	12:17:41
62	PuenteEntrada	12:17:39	18/5/2019	180	Amarillo	Amarillo	12:17:39
60	PuenteResidencias	12:17:42	18/5/2019	391	SA	Amarillo	12:17:41
62	PuenteEntrada	12:08:06	18/5/2019	430	SA	Rojo	12:02:53
60	PuenteResidencias	12:07:50	18/5/2019	81	Rojo	Rojo	12:02:49
62	PuenteEntrada	12:07:14	18/5/2019	427	SA	Rojo	12:02:53
60	PuenteResidencias	12:07:03	18/5/2019	81	Rojo	Rojo	12:02:49
62	PuenteEntrada	12:06:24	18/5/2019	426	SA	Rojo	12:02:53
60	PuenteResidencias	12:06:15	18/5/2019	81	Rojo	Rojo	12:02:49
62	PuenteEntrada	12:05:08	18/5/2019	426	SA	Rojo	12:02:53
60	PuenteResidencias	12:05:18	18/5/2019	82	Rojo	Rojo	12:02:49
60	PuenteResidencias	12:04:31	18/5/2019	82	Rojo	Rojo	12:02:49
62	PuenteEntrada	12:04:21	18/5/2019	426	SA	Rojo	12:02:53
60	PuenteResidencias	12:03:44	18/5/2019	82	Rojo	Rojo	12:02:49
62	PuenteEntrada	12:03:36	18/5/2019	426	SA	Rojo	12:02:53
60	PuenteResidencias	12:02:49	18/5/2019	78	Rojo	Rojo	12:02:49
62	PuenteEntrada	12:02:45	18/5/2019	427	SA	Verde	12:02:06
60	PuenteResidencias	12:01:59	18/5/2019	278	Verde	Verde	12:01:59

La comunicación en este caso de uso, es mediante estructura RESTful, que utiliza el protocolo http para establecer la red de nodos sensor. La hora de propagación, al igual que en el caso de uso de la sección anterior, indica el momento en que un nodo sensor maestro recibe un mensaje por parte de otro nodo sensor maestro ubicado en otro puente de su misma red colaborativa. Los datos para este escenario se localizan en el cuadro 6.6

Se puede concluir de esta prueba que el protocolo de comunicación se estable-

ce entre nodos sensor maestro. Cada nodo sensor maestro propaga el estado local del puente al que pertenece, para así determinar conjuntamente cual será la Alerta Global de la red colaborativa.

6.2.5. Generación de alertas para estructuras

El parámetro de altura de cada puente, que se define en “Configuración de dispositivos” por medio de la página web, es el que determina la forma en que se generan las alarmas por estructura. La medición de una estructura indica un estado “Sin_Alarma” cuando se encuentra entre, el parámetro definido y su 25%, mientras que una alerta “Verde” es la que se encuentra entre 25% – 50%, “Amarilla” entre 50% – 75% y “Roja” entre 75% – 100%.

En el cuadro 6.7 se muestran varios casos de alerta en las pruebas del escenario de un solo puente, con un parámetro de altura definido en 450cm.

Todas las mediciones mostradas en el cuadro fueron provocadas intencionalmente de forma manual durante la operación del artefacto como caja negra, acortando la distancia ya sea con objetos o moviendo los nodos sensor hacia los extremos del puente que tienen menor distancia respecto a la estructura, para así emular un incremento del nivel del agua.

Cuadro 6.7: Mediciones de red local con alarma - un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
100	PuenteEntrada	14:37:40	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:37:30	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:36:50	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:36:23	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:35:58	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:35:34	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:35:16	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:34:46	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:34:31	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:33:58	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:33:34	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:33:06	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:32:42	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:32:19	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:31:58	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:31:31	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:31:15	22/5/2019	68	Rojo	Rojo
100	PuenteEntrada	14:30:44	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:30:34	22/5/2019	67	Rojo	Rojo
100	PuenteEntrada	14:29:52	22/5/2019	193	Amarillo	Amarillo
100	PuenteEntrada	14:29:42	22/5/2019	198	Amarillo	Amarillo
100	PuenteEntrada	14:29:32	22/5/2019	207	Amarillo	Amarillo
102	PuenteEntrada	14:06:19	22/5/2019	7	Falso-Positivo	Rojo
100	PuenteEntrada	14:05:59	22/5/2019	7	Rojo	Rojo
102	PuenteEntrada	14:05:37	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	14:05:49	22/5/2019	7	Rojo	Rojo
102	PuenteEntrada	14:04:56	22/5/2019	10	Rojo	Rojo

Continúa en la siguiente página

Cuadro 6.7 – continuación de mediciones de red local con alarma - un solo puente

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
100	PuenteEntrada	14:05:08	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	14:04:55	22/5/2019	5	Rojo	Rojo
102	PuenteEntrada	14:04:14	22/5/2019	14	Rojo	Rojo
100	PuenteEntrada	14:04:33	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	14:04:23	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	14:04:07	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	13:59:02	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	13:58:22	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	13:58:21	22/5/2019	7	Rojo	Rojo
100	PuenteEntrada	13:57:32	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	13:57:22	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	13:56:45	22/5/2019	8	Rojo	Rojo
100	PuenteEntrada	13:56:35	22/5/2019	9	Rojo	Rojo
100	PuenteEntrada	13:56:25	22/5/2019	8	Rojo	Rojo

Se puede decir que el protocolo de alarmas está funcionando como se esperaba, puesto que toma el valor configurado para el dispositivo como altura (450cm) y gestiona su sistema de alertas con base en el porcentaje que tiene asignado cada uno de los estados.

6.2.6. Protocolo de alertas para redes de estructuras

En el cuadro 6.8 se muestra la secuencia de alertas trasegadas entre los puentes “PuenteEntrada” y “PuenteResidencias” que definen la forma en que interactúan los nodos sensor maestro de estructuras que pertenecen a una misma red colaborativa global.

Cuadro 6.8: Mediciones de red global con alarma - varios puentes

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
62	PuenteEntrada	13:17:26	18/5/2019	429	SA	Rojo
60	PuenteResidencias	13:16:48	18/5/2019	60	Rojo	Rojo
62	PuenteEntrada	13:16:33	18/5/2019	427	SA	Rojo
60	PuenteResidencias	13:15:59	18/5/2019	60	Rojo	Rojo
60	PuenteResidencias	13:15:08	18/5/2019	60	Rojo	Rojo
62	PuenteEntrada	13:14:33	18/5/2019	426	SA	Rojo
60	PuenteResidencias	13:14:21	18/5/2019	60	Rojo	Rojo
60	PuenteResidencias	13:13:32	18/5/2019	60	Rojo	Rojo
62	PuenteEntrada	13:13:24	18/5/2019	431	SA	Rojo
60	PuenteResidencias	13:12:46	18/5/2019	61	Rojo	Rojo
60	PuenteResidencias	13:11:58	18/5/2019	61	Rojo	Rojo
62	PuenteEntrada	13:12:01	18/5/2019	426	SA	Amarillo
60	PuenteResidencias	13:11:02	18/5/2019	152	Amarillo	Amarillo
62	PuenteEntrada	13:11:00	18/5/2019	426	SA	Amarillo
60	PuenteResidencias	13:10:13	18/5/2019	151	Amarillo	Amarillo
62	PuenteEntrada	13:10:05	18/5/2019	427	SA	Amarillo

Continúa en la siguiente página

Cuadro 6.8 – continuación de medición de red global con alarma - varios puentes

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
60	PuenteResidencias	13:09:24	18/5/2019	171	Amarillo	Amarillo
60	PuenteResidencias	13:08:39	18/5/2019	188	Verde	Verde
62	PuenteEntrada	13:08:37	18/5/2019	426	SA	Amarillo
60	PuenteResidencias	13:07:52	18/5/2019	151	Amarillo	Amarillo
62	PuenteEntrada	13:07:41	18/5/2019	426	SA	Amarillo
60	PuenteResidencias	13:07:06	18/5/2019	152	Amarillo	Amarillo
62	PuenteEntrada	13:06:46	18/5/2019	426	SA	Amarillo
60	PuenteResidencias	13:06:16	18/5/2019	170	Amarillo	Amarillo
62	PuenteEntrada	13:05:44	18/5/2019	426	SA	Amarillo
60	PuenteResidencias	13:05:29	18/5/2019	147	Amarillo	Amarillo
60	PuenteResidencias	13:04:43	18/5/2019	222	Verde	Verde
62	PuenteEntrada	13:04:43	18/5/2019	427	SA	Verde
60	PuenteResidencias	13:03:55	18/5/2019	222	Verde	Verde
62	PuenteEntrada	13:03:52	18/5/2019	426	SA	Verde
60	PuenteResidencias	13:03:08	18/5/2019	221	Verde	Verde
62	PuenteEntrada	13:02:52	18/5/2019	425	SA	Verde
60	PuenteResidencias	13:02:21	18/5/2019	221	Verde	Verde
62	PuenteEntrada	13:02:00	18/5/2019	425	SA	Verde
60	PuenteResidencias	13:01:34	18/5/2019	222	Verde	Verde
62	PuenteEntrada	12:53:22	18/5/2019	426	SA	Rojo
60	PuenteResidencias	12:52:31	18/5/2019	10	Rojo	Rojo
62	PuenteEntrada	12:52:22	18/5/2019	427	SA	Rojo
60	PuenteResidencias	12:51:44	18/5/2019	11	Rojo	Rojo
60	PuenteResidencias	12:37:40	18/5/2019	393	SA	Rojo
62	PuenteEntrada	12:36:58	18/5/2019	5	Rojo	Rojo
60	PuenteResidencias	12:36:53	18/5/2019	393	SA	Rojo
60	PuenteResidencias	12:36:10	18/5/2019	393	SA	Rojo
62	PuenteEntrada	12:35:56	18/5/2019	6	Rojo	Rojo
60	PuenteResidencias	12:35:14	18/5/2019	393	SA	Rojo
62	PuenteEntrada	12:35:02	18/5/2019	7	Rojo	Rojo
60	PuenteResidencias	12:21:21	18/5/2019	391	SA	Rojo
62	PuenteEntrada	12:21:03	18/5/2019	10	Rojo	Rojo
60	PuenteResidencias	12:20:39	18/5/2019	393	SA	Rojo
62	PuenteEntrada	12:20:15	18/5/2019	10	Rojo	Rojo
60	PuenteResidencias	12:19:08	18/5/2019	393	SA	Amarillo
60	PuenteResidencias	12:18:24	18/5/2019	391	SA	Amarillo
62	PuenteEntrada	12:17:39	18/5/2019	180	Amarillo	Amarillo
60	PuenteResidencias	12:17:42	18/5/2019	391	SA	Amarillo
62	PuenteEntrada	12:08:06	18/5/2019	430	SA	Rojo
60	PuenteResidencias	12:07:50	18/5/2019	81	Rojo	Rojo
62	PuenteEntrada	12:07:14	18/5/2019	427	SA	Rojo
60	PuenteResidencias	12:07:03	18/5/2019	81	Rojo	Rojo
62	PuenteEntrada	12:06:24	18/5/2019	426	SA	Rojo
60	PuenteResidencias	12:06:15	18/5/2019	81	Rojo	Rojo
62	PuenteEntrada	12:05:08	18/5/2019	426	SA	Rojo
60	PuenteResidencias	12:05:18	18/5/2019	82	Rojo	Rojo
60	PuenteResidencias	12:04:31	18/5/2019	82	Rojo	Rojo
62	PuenteEntrada	12:04:21	18/5/2019	426	SA	Rojo
60	PuenteResidencias	12:03:44	18/5/2019	82	Rojo	Rojo
62	PuenteEntrada	12:03:36	18/5/2019	426	SA	Rojo
60	PuenteResidencias	12:02:49	18/5/2019	78	Rojo	Rojo

Continúa en la siguiente página

Cuadro 6.8 – continuación de medición de red global con alarma - varios puentes

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global
62	PuenteEntrada	12:02:45	18/5/2019	427	SA	Verde
60	PuenteResidencias	12:01:59	18/5/2019	278	Verde	Verde

Se puede decir que el protocolo de alertas toma el estado más crítico, de entre los estados de puentes que forman parte de la red colaborativa, para determinar cual es el estado global de la misma.

6.3. Pruebas de rendimiento

Como parte del proceso de validación de un sistema o metodología, se requiere analizar y caracterizar el rendimiento del mismo. Con el objetivo de generar datos que permitan analizar su viabilidad y mejor enfoque, mediante parámetros considerados importantes ante eventualidades naturales, se realiza una comparación entre tecnologías de comunicación para nodos maestro.

En esta sección se muestran datos que se han tomado, para dos casos del escenario de varios puentes. El primer caso es en el cual se propaga el mensaje de cambio de alerta entre nodos maestro, por medio de mensajes de texto *SMS*(Short Message Service, *SMS* por sus siglas en inglés).

Con el fin de medir el impacto de utilizar una arquitectura completamente basada en servicios web, se plantea el segundo caso de uso, cuya arquitectura se basa completamente en servicios web, de protocolo *RESTful* para el trasiego de mensajes de alerta entre nodos sensor tipo maestro.

Entre los parámetros a comparar entre ambos casos, se encuentran: tasas de transferencia, tiempo de propagación de las alertas, costo de transmisión por unidad de tiempo (KB usados en el mismo periodo de funcionamiento) y duración de la batería. Los parámetros mencionados permiten definir si hay beneficios sustanciales de rendimiento al utilizar una estructura *REST* para comunicación con el servidor y entre nodos maestros.

Además de comparar el desempeño contra una arquitectura similar, pero sin servicios web, se busca caracterizar al prototipo, para lo cual se propone añadir mediciones de: comunicación nodo-nodo, comunicación nodo-servidor, configuración de operación desde interfaz web (nodo maestro, nodo esclavo) y el protocolo de prevención de emergencias.

6.3.1. Tiempos de propagación de alertas

El escenario de varios puentes se utilizó con ambas tecnologías, ambas pruebas fueron realizadas en laboratorio y ambas se cronometraron en una 1h con 40 minutos. Los resultados obtenidos son utilizados, tanto en esta sub-sección como en la posterior.

La comparación realizada en esta prueba, fue para determinar que tan rápido se trasiegan los mensajes entre nodos sensor maestro de estructuras que pertenecen a una misma red colaborativa global, utilizando mensajes de texto plano o *SMS*, contra *REST*.

El cuadro 6.9 muestra los resultados obtenidos para el escenario de varios puentes, donde el caso de uso establece un rango de latencia de propagación de mensajes utilizando estructura SMS. Se puede observar que el mayor tiempo de propagación alcanzado es de 00:02:53 en la medición de las 21:36:56.

Los datos resaltados en color azul para la columna H.P., representan el momento en que el nodo sensor detecta la emergencia, mientras que los datos en color verde representan el momento en que se propaga el mensaje de alerta en el nodo sensor maestro que se localiza en otro puente que pertenece a la misma red colaborativa.

Cuadro 6.9: Mensajes transmitidos entre puentes por medio de tecnología SMS

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global	H.P.
31	PuenteDelTec	22:27:01	14/5/2019	82	SA	Rojo	22:24:25
30	PuentePrueba1	22:25:27	14/5/2019	13	Rojo	Rojo	22:23:28
31	PuenteDelTec	22:25:16	14/5/2019	82	SA	Rojo	22:24:25
30	PuentePrueba1	22:23:28	14/5/2019	13	Rojo	Rojo	22:23:28
30	PuentePrueba1	22:11:30	14/5/2019	83	SA	Verde	22:07:13
30	PuentePrueba1	22:09:47	14/5/2019	83	SA	Verde	22:07:13
31	PuenteDelTec	22:08:40	14/5/2019	53	Verde	Verde	22:06:33
30	PuentePrueba1	22:08:05	14/5/2019	83	SA	Verde	22:07:13
31	PuenteDelTec	22:06:33	14/5/2019	53	Verde	Verde	22:06:33
31	PuenteDelTec	21:41:06	14/5/2019	50	Verde	Verde	21:36:56
30	PuentePrueba1	21:40:41	14/5/2019	81	SA	Verde	21:39:49
31	PuenteDelTec	21:39:01	14/5/2019	50	Verde	Verde	21:36:56
31	PuenteDelTec	21:36:56	14/5/2019	50	Verde	Verde	21:36:56
30	PuentePrueba1	21:16:44	14/5/2019	81	SA	Rojo	21:12:23
31	PuenteDelTec	21:15:01	14/5/2019	16	Rojo	Rojo	21:10:47
30	PuentePrueba1	21:14:59	14/5/2019	83	SA	Rojo	21:12:23
31	PuenteDelTec	21:12:54	14/5/2019	16	Rojo	Rojo	21:10:47
30	PuentePrueba1	21:13:15	14/5/2019	83	SA	Rojo	21:12:23
30	PuentePrueba1	21:11:30	14/5/2019	83	SA	Verde	21:07:14
31	PuenteDelTec	21:10:47	14/5/2019	16	Rojo	Rojo	21:10:47
30	PuentePrueba1	21:09:48	14/5/2019	83	SA	Verde	21:07:14
31	PuenteDelTec	21:08:40	14/5/2019	42	Verde	Verde	21:00:17
30	PuentePrueba1	21:08:05	14/5/2019	83	SA	Verde	21:07:14
31	PuenteDelTec	21:06:33	14/5/2019	42	Verde	Verde	21:00:17
30	PuentePrueba1	21:06:21	14/5/2019	83	SA	Verde	21:05:29
31	PuenteDelTec	21:04:28	14/5/2019	42	Verde	Verde	21:00:17
30	PuentePrueba1	21:04:37	14/5/2019	83	SA	Verde	21:03:45
30	PuentePrueba1	21:02:52	14/5/2019	83	SA	Verde	21:02:00
31	PuenteDelTec	21:02:23	14/5/2019	42	Verde	Verde	21:00:17
31	PuenteDelTec	21:00:17	14/5/2019	42	Verde	Verde	21:00:17
31	PuenteDelTec	20:56:46	14/5/2019	85	SA	Amarillo	20:50:40
31	PuenteDelTec	20:55:01	14/5/2019	85	SA	Amarillo	20:50:40
30	PuentePrueba1	20:53:46	14/5/2019	41	Amarillo	Amarillo	20:49:49
31	PuenteDelTec	20:53:16	14/5/2019	85	SA	Amarillo	20:50:40
30	PuentePrueba1	20:51:47	14/5/2019	41	Amarillo	Amarillo	20:49:49
31	PuenteDelTec	20:51:32	14/5/2019	85	SA	Amarillo	20:50:40

En el cuadro 6.10 se muestran los resultados obtenidos para la prueba realizada al

escenario de varios puentes, en contra-parte a la mencionada anteriormente; este caso de uso establece un rango de latencia de propagación de mensajes entre nodos sensor utilizando estructura SMS de servicios web. Se puede observar que el evento que más tiempo tardó es de 5 segundos, en el evento de las 14:21:29 de la columna H.P.

Cuadro 6.10: Mensajes transmitidos entre puentes por medio de tecnología REST

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global	H.P.
22	PuentePrueba2	14:41:17	6/5/2019	88	SA	Rojo	14:21:34
25	PuentePrueba1	14:40:42	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:40:05	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:39:25	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:38:54	6/5/2019	88	SA	Rojo	14:21:34
25	PuentePrueba1	14:38:08	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:37:42	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:36:52	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:36:31	6/5/2019	88	SA	Rojo	14:21:34
25	PuentePrueba1	14:35:34	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:35:19	6/5/2019	88	SA	Rojo	14:21:34
25	PuentePrueba1	14:34:19	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:34:02	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:33:03	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:32:50	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:31:47	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:31:38	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:30:30	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:30:26	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:29:10	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:29:10	6/5/2019	88	SA	Rojo	14:21:34
25	PuentePrueba1	14:27:53	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:27:58	6/5/2019	88	SA	Rojo	14:21:34
22	PuentePrueba2	14:26:46	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:26:36	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:25:34	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:25:20	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:24:23	6/5/2019	88	SA	Rojo	14:21:34
25	PuentePrueba1	14:24:04	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:23:11	6/5/2019	88	SA	Rojo	14:21:34
25	PuentePrueba1	14:22:46	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:21:59	6/5/2019	89	SA	Rojo	14:21:34
25	PuentePrueba1	14:21:29	6/5/2019	6	Rojo	Rojo	14:21:29
22	PuentePrueba2	14:20:47	6/5/2019	89	SA	Amarillo	14:19:01
25	PuentePrueba1	14:20:13	6/5/2019	21	Amarillo	Amarillo	14:18:57
22	PuentePrueba2	14:19:35	6/5/2019	88	SA	Amarillo	14:19:01
25	PuentePrueba1	14:18:57	6/5/2019	21	Amarillo	Amarillo	14:18:57
22	PuentePrueba2	14:18:23	6/5/2019	88	SA	Verde	14:17:46
25	PuentePrueba1	14:17:40	6/5/2019	46	Verde	Verde	14:15:08
22	PuentePrueba2	14:17:11	6/5/2019	88	SA	Verde	14:16:29
25	PuentePrueba1	14:16:24	6/5/2019	46	Verde	Verde	14:15:08
22	PuentePrueba2	14:15:59	6/5/2019	88	SA	Verde	14:15:12
25	PuentePrueba1	14:15:08	6/5/2019	46	Verde	Verde	14:15:08

Continúa en la siguiente página

Cuadro 6.10 – continuación de mensajes transmitidos entre puentes por medio de tecnología REST

Nodo	Puente	Hora	Fecha	M(cm)	Alerta	Al. Global	H.P.
25	PuentePrueba1	14:04:54	6/5/2019	19	Rojo	Rojo	14:03:33
25	PuentePrueba1	14:03:33	6/5/2019	19	Rojo	Rojo	14:03:33
25	PuentePrueba1	14:02:16	6/5/2019	39	Amarillo	Amarillo	14:02:16
22	PuentePrueba2	13:28:30	6/5/2019	87	SA	Verde	13:27:41
25	PuentePrueba1	13:27:39	6/5/2019	53	Verde	Verde	13:26:23
22	PuentePrueba2	13:27:18	6/5/2019	87	SA	Verde	13:26:25
25	PuentePrueba1	13:26:23	6/5/2019	53	Verde	Verde	13:26:23

6.3.2. Costos operativos entre SMS vs REST por unidad de tiempo

Esta sección expone la comparación realizada de costo de operación, donde se tiene datos, tanto para la estructura SMS, como la de RESTful. Las pruebas fueron realizadas bajo casi las mismas condiciones para ambas tecnologías y durante un tiempo aproximado de 1 hora y 40 minutos para cada caso de uso.

A fin de tener referencias de costos de operación por mensaje transmitido entre los nodos sensor maestro y para cada uno de los casos de uso; se realiza una revisión actual de los precios por mensaje de texto y precio por kB consumido para compañía telefónica costarricense, Kolbi; cuyos datos se exponen en la figura 6.9.



Tarifas servicio de mensajería y video llamada excedente

Servicio	Tarifa con I.I
SMS	€3
Video Llamada	€50.50

Internet por Kbyte

Si contás con un servicio móvil prepago activo, o vas a adquirir uno, ahora podrás navegar en kolbi 4.5G sin costo adicional, en la modalidad de cobro por Kbyte. Solo pagarás por la transferencia de datos que realicés.

Características:

Velocidad 4G hasta	Velocidad 3G hasta	Precio por Kbyte (I.V.I)
20 Mbps	6 Mbps	€0.008588

Datos tomados de:

https://www.kolbi.cr/wps/portal/kolbi_dev/personas/postpago/informacion-postpago/tarifas-postpago

Fecha: 24 de mayo de 2019, Hora: 9:53pm

Figura 6.9: Costo de kB vs SMS para la operadora Kolbi.

En el cuadro 6.11. En la estructura SMS el costo se calcula por mensaje, dado las compañías telefónicas establecen un precio por mensaje en la mayoría de los casos. En REST se establecen los precios por kB consumido.

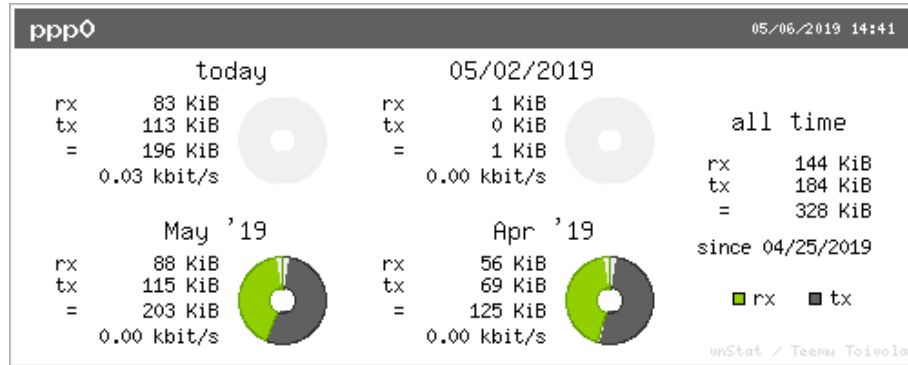


Figura 6.10: Datos de consumo de datos del dispositivo 22.

En la figura 6.10 se aprecia que durante el periodo de funcionamiento de la prueba, el empotrado tuvo un consumo de 196 kbytes totales, durante el mismo periodo de una hora con cuarenta minutos.

Cuadro 6.11: Costos de operación para las redes colaborativas con diferentes tecnologías

Variable	Cantidad	Costo por unidad	Total (1h y 40 minutos)
Mensaje	103	3	109 colones
kByte	193	0,008588 colones	1.6575 colones

6.3.3. Tiempos de autonomía con baterías ICR18650 4400mAh 3.7V

Mediante el uso de baterías de Ion de Litio de 4400mAh se ha establecido un periodo de duración de aproximadamente dos horas de duración para un nodo sensor en estructura REST en constante comunicación con otro nodo sensor.

La prueba se hizo en laboratorio y se simuló un situación similar a la del escenario de varios puentes, donde se provocan varias situaciones de cambio de estado de las alertas (ver figura 6.11).

6.4. Consulta y validación con expertos

Para validar el funcionamiento general del prototipo con expertos del CIVCO, su nivel de adaptabilidad y utilidad; se siguió una metodología de consulta participativa involucrando usuarios y expertos tanto internos como externos al proyecto.

En particular, se organizó una demostración de funcionamiento con 6 personas expertas en el tema de puentes. En la imagen de la figura 6.12 se da un resumen

759	42	PuenteEntrada	13:52:58	16-05-19	113	Rojo	Rojo	13:50:32
758	42	PuenteEntrada	13:51:19	16-05-19	113	Rojo	Rojo	13:50:32
757	42	PuenteEntrada	13:50:32	16-05-19	113	Rojo	Rojo	13:50:32
756	40	PuenteResidencias	13:50:29	16-05-19	181	Amarillo	Rojo	12:28:31
755	42	PuenteEntrada	13:49:40	16-05-19	114	Amarillo	Amarillo	13:49:40
754	40	PuenteResidencias	13:49:34	16-05-19	181	Amarillo	Rojo	12:28:31
753	42	PuenteEntrada	13:48:54	16-05-19	113	Rojo	Rojo	13:41:49
752	40	PuenteResidencias	13:48:44	16-05-19	181	Amarillo	Rojo	12:28:31
530	42	PuenteEntrada	11:59:42	16-05-19	62	Rojo	Rojo	11:58:23
529	40	PuenteResidencias	11:59:08	16-05-19	62	Rojo	Rojo	11:56:55
528	42	PuenteEntrada	11:59:00	16-05-19	63	Rojo	Rojo	11:58:23
527	40	PuenteResidencias	11:58:20	16-05-19	62	Rojo	Rojo	11:56:55

Figura 6.11: Medición de duración de batería 4400mHh.

de las respuestas obtenidas, donde el instrumento de consulta se basa en los ítemes mostrados a la izquierda de la imagen y a su derecha barras que indican el promedio de calificación obtenido.

Se aprecia que a criterio de expertos, resta trabajo específicamente en términos de rendimiento de la red y claridad de funcionalidad. Rendimiento es referido al tiempo que tarda en conectarse un nodo sensor de red local y en claridad de funcionalidad al factor de como usar el sistema.

Respecto a rendimiento se proyecta mejorar el tiempo de conexión de redes colaborativas locales por debajo de los umbrales de timeout, ya sea por medio de Xbee, como se encuentra actualmente, o incluso realizar la valoración de adaptabilidad en otra tecnología. Lo anterior puede ser en modalidad de tesis o trabajo de graduación.

Para mejor entendimiento de la funcionalidad y uso se ha generado un manual de configuración, que incluyen los dos escenarios disponibles para este proyecto y una breve explicación de los mismos.

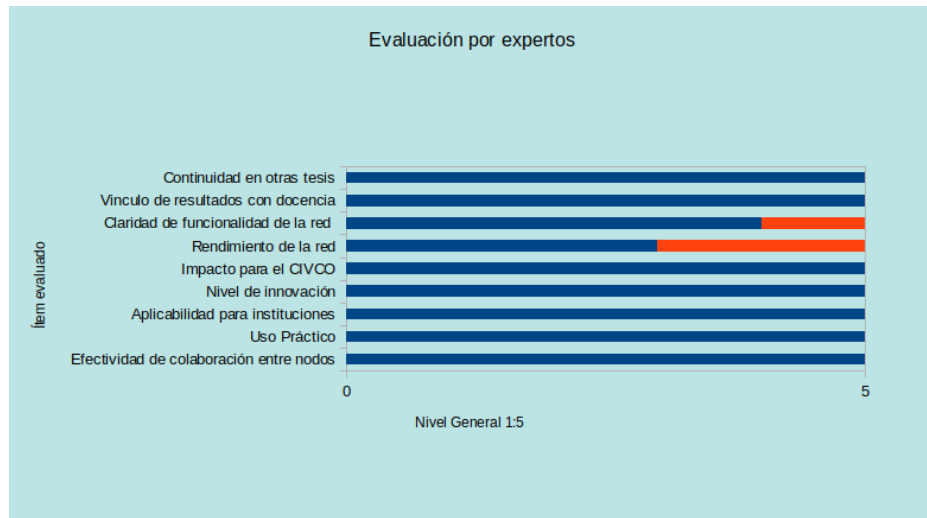


Figura 6.12: Resumen de las respuestas obtenidas en la consulta a expertos.

6.5. Análisis de resultados

En las secciones anteriores se han compilado datos como parte del proceso de validación de un sistema o metodología, que permite analizar y caracterizar el rendimiento del mismo.

Se proporcionan los resultados de funcionalidad y rendimiento obtenidos para dos diferentes escenarios y varios casos de uso descritos, con el objetivo de generar datos que permitan analizar su viabilidad y mejor enfoque; lo anterior, mediante parámetros considerados importantes ante eventualidades naturales, apoyando así a la hipótesis planteada.

Es importante analizar que los tiempos de respuesta son un factor a tomar en cuenta a la hora de trabajar con sistemas *SHM*, puesto que está relacionado de forma prácticamente directa con el rescate de vidas e incluso de estructuras.

Entre los parámetros más interesantes se plantean los tiempos de propagación de los mensajes de alerta a través de las redes colaborativas, así como los costos de operación de los dos casos de uso planteados, donde su filosofía de operación es la misma pero la tecnología de transmisión es una diferente.

En el cuadro 6.12 se encuentra un resumen de los datos de comparación entre tecnologías que sugiere que una estructura REST es buena opción para implementar un sistema SHM para prevención de desastres en una red colaborativa de sensores.

Cuadro 6.12: Comparación entre tecnologías para las redes colaborativas con diferentes tecnologías

Tecnología	Costo	#Mensajes	Tiempo Propagación
SMS	54.5 col/nodo	Menos mensajes	Máximo 00:02:53
REST	3.315 col/nodo	N Mensajes	Máximo 00:00:07

Los resultados sugieren un mejor desempeño de los sistemas basados en estructura RESTful tanto en tiempo de respuesta como en costo de operación. Lo anterior se considera de alto valor, puesto que son datos obtenidos bajo circunstancias muy similares, más no determinantes; se recomienda realizar variantes filosóficas e incluso tecnológicas en las pruebas, que permitan validar el comportamiento obtenido en esta investigación.

Por otro lado la validación de los protocolos tanto local como global de colaboración, muestran el comportamiento esperado al menos en los escenarios planteados. Donde se propaga un evento anormal, en un tiempo relativamente corto de 7 segundos como máximo en tecnología y estructura de servicios web, RESTful. Para ver el detalle referirse a los cuadros 6.10 y tab:MedRedGlobalAlarmasYPropagacion. Sin embargo el escenario es limitado y se propone para trabajo futuro la implementación de un escenario más complejo.

El uso de filosofía colaborativa tanto en diseño como en implementación de sistemas distribuidos, tomando en cuenta los datos de esta investigación; sugiere ser una excelente opción para abordar problemas de interés público, donde con alguna capacidad de procesamiento de datos y algún nivel de autonomía de ejecución en puntos estratégicos, es posible tomar decisiones inmediatas que pueden cambiar el cause de grandes problemas, como lo es por ejemplo un desastre natural.

Lo anterior establece un precedente para la hipótesis planteada, donde se sugiere la posibilidad de utilizar una red colaborativa para detectar desastres naturales en estructuras puente utilizando RESTful.

La funcionalidad y aceptabilidad del prototipo construido después del proceso de investigación de esta tesis, sugiere además, que faltan pruebas de estrés al protocolo puesto que en los datos no se ven puntos de falla en transmisión e interpretación de alertas.

Capítulo 7

Conclusiones

En este capítulo se presentan las principales conclusiones que se han obtenido durante la investigación realizada, dirigidas al cumplimiento de los objetivos planteados desde la propuesta de tesis. Estas conclusiones se distribuyen en dos grupos:

- Conclusiones generales.
- Trabajo Futuro.

A continuación se incluyen las conclusiones generales del proyecto.

7.1. Conclusiones

En esta sección se describen las conclusiones generales más relevantes que se obtuvieron durante la realización de los objetivos del proyecto. Las conclusiones se agrupan de acuerdo a las fases que se realizaron en la metodología de desarrollo de la información seguida en esta tesis.

7.1.1. Conclusiones sobre trabajos relacionados

Estas conclusiones se refieren al capítulo 2 de trabajos relacionados y antecedentes, donde se obtuvo la base de trabajo de la tesis, que permite posicionarse en el contexto de los temas desarrollados. Las conclusiones más generales del capítulo a continuación:

- Las implementaciones exitosas de sistemas *SHM* actuales requieren de planificación estructural previa en los puentes para el trazado de la estructura de comunicación de los sensores.
- Se prevé que las *WSN* para *SHM* en el futuro no sean aisladas sino que más bien se integren a estructuras inteligentes de computadores *IoT*.
- Para manejar la cantidad de sensores *IoT* se propone un enfoque ligero de servicio web en estructura *RESTful* que permita la gestión de dispositivos de sensores inalámbricos.

- Para controlar el crecimiento y manejo de IPs se propone un método de diseño e implementación de gateways para las WSN con estructura *RESTful* utilizando un framework basado en Javascript.
- Los sistemas *SHM* deben contemplar tres etapas como mínimo: detección y adquisición de datos, gestión de plataforma y acceso y recuperación de datos.
- Para realizar el monitorización en una zona particular, una *CN* debe contar con una red de entidades autónomas, distribuidas y heterogéneas, que consisten de organizaciones, entre ellos los sensores de variables con algún nivel de capacidad de procesamiento.
- En una *CN* las entidades colaboran estrechamente, para ofrecer productos y servicios por parte de la red colaborativa, por ejemplo la detección de desastres natural en estructuras puente.
- Conjuntar *CN*, *WSN* y *SHM* requiere del desarrollo de un protocolo de comunicación flexible por medio de que permita el crecimiento a gran escala de nodos sensor, por ejemplo *HTTP* en estructura *REST*.
- En una *CWSN* los módulos en un nodo y los nodos de recolección de datos en las redes deben comunicarse y cooperar entre sí, para llevar a cabo una tarea específica.

7.1.2. Conclusiones sobre el análisis de los requerimientos

El análisis de requerimientos realizado en el capítulo 3, también ha aportado una serie de conclusiones generales, dadas a continuación:

- El protocolo de red requerido contempla al menos dos niveles de colaboración para poder cubrir las necesidades de *SHM*, tanto a nivel de la estructura como de la red de estructuras.
- Un nivel básico de poder de toma de decisiones, da una autonomía necesaria a cada estructura puente en caso de emergencia, que es manejada por un nodo sensor maestro.
- La generación y gestión de alarmas a un nivel local, posibilita la toma de decisiones a mediano o corto plazo, e incluso de forma inmediata. De la misma forma la capacidad de discernimiento entre una alarma real y un falso positivo, permite evitar incurrir en un estado de alerta innecesario.
- Es importante que el encargado de monitorización, el investigador, el estudiante; o cualquier tipo de usuario, pueda realizar configuración de dispositivos, redes o puentes de forma simple y ágil, incluso desde su móvil.

7.1.3. Conclusiones sobre el diseño

En la fase de diseño plasmada en el capítulo 4 de la tesis, se concluye que:

- La estructura del prototipo permite realizar validación del proceso en el que se conjuntan varios conceptos de las etapas de investigación y de análisis de requerimientos, lo cual permite ubicar aspectos de tecnologías conocidas y además la tropicalización de un problema.

- La filosofía de colaboración en las redes que definen al diseño, son quienes refutan o aprueban su viabilidad como posible solución a la temática de puentes en Costa Rica. La arquitectura experimental definida permite crear y modificar redes de colaboración tanto locales como globales para estructuras puente.
- Es posible reutilizar el prototipo creado en futuras validaciones. Los códigos fuente y documentación sobre configuración y software están disponibles en el proyecto GitHub de la investigación.
- La filosofía colaborativa del enfoque de solución propuesto para esta tesis, fomenta la característica de división de tareas a través de la red de sensores que resuelve un problema de mayor escala como lo son los desastres naturales.
- El uso de redes colaborativas a dos niveles en un sistema distribuido con servicios web para *SHM*, agrega un factor de innovación alto, capaz de dar un enfoque *IoT* a la solución de un problema de carácter público.

7.1.4. Conclusiones sobre la implementación

Respecto al capítulo 5 de implementación, se concluye que:

- Se logra satisfacer el objetivo de implementar un prototipo de red inalámbrica colaborativa enfocado a la monitorización de salud en estructuras puente, mediante el uso de herramientas de software libre, hardware de toma de datos de bajo costo y dispositivos empujados de alta disponibilidad y costo moderado.
- Las herramientas de software libre favorecen la realización del proyecto a bajo costo con una alta capacidad de procesamiento de datos. Raspbian; provee todas las características de una plataforma Debian en un dispositivo pequeño y manejable como lo es una Raspberry. Python y sus bibliotecas de manejo de servicios web y matemática, proveen a cada nodo sensor, de un gran poder de comunicación y procesamiento de datos a un bajo costo, energético y computacional.
- El servidor del prototipo se compone de un conglomerado de paquetes de software libre. Iniciando por el *Backend* con Node.js y Express, que permitieron la creación del API de ebridge, cuya principal característica es la disponibilidad a través de cualquier navegador, facilitando así el uso y análisis de los datos. Angular y Typescript dan vida al *Frontend*, que también proveen al usuario de una herramienta de monitorización y configuración, que hace que el prototipo sea utilizable y accesible.

7.1.5. Conclusiones sobre las pruebas

Del capítulo 6 de referente a validación y pruebas, se extrae que:

- La validación de ambos escenarios; el de un solo puente y el de varios puentes, en ambos casos tanto el protocolo de comunicación entre nodos sensor, como el protocolo de alertas, sugieren ser viable en términos de transmisión

de mensajes, factibilidad, adaptabilidad. En ambos casos se obtuvieron respuestas y un funcionamiento de caja negra esperado y dentro de un rango de tiempo que podría adaptarse al enfrentamiento de una situación real de desastre natural a nivel de estructuras puente.

- Es claro que para una implementación real debe haber de por medio, un proceso de pruebas de escalamiento a nivel de nodos sensor, además de corrección de situaciones inesperadas y un mejoramiento estructural de los dispositivos de toma de datos, es decir que sean adaptables a las condiciones adversas del clima y robo por su disposición en lugares públicos.
- Los tiempos de propagación obtenidos en este proyecto para una implementación en estructura RESTful dan un indicio muy positivo, respecto a la posibilidad de ser un sistema efectivo en términos de tiempo de respuesta ante posibles eventualidades naturales en puentes.
- El trasiego de datos entre nodos sensor sugiere ser muy alentador, no se registran pérdidas de datos en las pruebas realizadas, ni a nivel de laboratorio, ni a nivel de campo hasta el momento.
- Entre las debilidades de este tipo de tecnología se encuentra su dependencia a la conectividad a internet por algún medio, a pesar de su alta disponibilidad en la actualidad podría ser ese aspecto uno de los que ofrezcan mayor desafío a la hora de garantizar efectividad ante una eventualidad a gran escala.
- En términos energéticos se puede decir que el sistema es ineficiente ya que su nivel de autonomía eléctrica demostró ser insuficiente para sostenerse en el tiempo como un sistema que requiera poca o nula intervención humana para desempeñarse correctamente durante un espacio prolongado.
- Respecto a la consulta con expertos del CIVCO se tiene un nivel de aceptación general y tomando todos los rubros valorados del 93.3%, donde uno de los ítems a mejorar alcanzó un 3/5 y el otro un 4/5 en promedio, el resto son 5/5.
- Como conclusión general de la investigación y posterior a la implementación del prototipo, el desarrollo metodológico de una arquitectura de redes colaborativas con más de un nivel de colaboración, sugiere tener grandes impactos en una organización como el MOPT, COSEVI, municipios e incluso Gobierno central de la república, en vista de que no se cuenta con algún sistema similar al propuesto, con un alto grado de adaptabilidad tanto social como económica.

7.2. Trabajo Futuro

Como elementos de trabajo futuro se contemplan los siguientes:

- A nivel de sistema se debe haber mejoras de implementación, el servicio principal de muestra de datos carece de un refrescamiento automático de los datos entrantes.

- La implementación de una estructura Gateway para el manejo, asignación y re-asignación de IPs, es una recomendación establecida para otras aplicaciones que podría ser muy útil en este trabajo. Un enfoque más simplista es el de limpiar simplemente los IPs por dispositivo a nivel de base de datos, sin embargo y a pesar de ser bastante útil también le resta dinamismo a la reincorporación de nodos sensor desconectados.
- Surge un posible proyecto de graduación para un(a) estudiante de la carrera de Diseño Industrial, para la creación y constante mejora del cobertor para exteriores que utiliza un nodo sensor. Lo anterior representa un desafío interesante puesto que debe proteger la circuitería pero a su vez no debe interferir entre las comunicaciones inalámbricas inherentes del dispositivo. Adicionalmente se le podría agregar una característica de permitir su instalación en lugares de difícil acceso en puentes para evitar su robo.
- Surge un posible proyecto de graduación para un(a) estudiante de la(s) carrera(s) de Ingeniería Electrónica, Mecatrónica o Computadores, que caracterice y optimice la comunicación de la red local en puentes, además de considerar y comparar otras opciones de comunicación además de Xbee, con el fin de determinar cual posee una mejor adaptabilidad al problema. No debe de omitirse la caracterización e incluso optimización del módulo de software envuelto.
- Se recomienda la realización de pruebas de alto estrés para la red colaborativa global, con un número más alto de nodos sensor maestro, puentes y redes colaborativas globales; que permitan determinar e identificar los límites de la misma, que incluyan número de nodos máximo soportado por el protocolo y número máximo de redes colaborativas que se pueden traslapar.
- Se recomienda la caracterización del tamaño máximo del reporte enviado en estructura RESTful entre nodos, de forma tal que se identifique información que eventualmente pueda ser suprimida de los reportes enviados entre nodos sensor; en función del tráfico de datos de la red y con el objetivo de enviar únicamente la información estrictamente necesaria en momentos de alta demanda, manteniendo el correcto funcionamiento de la red colaborativa de sensores.

Capítulo 8

Referencias

Bibliografía

- [1] C. Jr. Tokognon A., Gao B., and Yun Tian G. *Structural Health Monitoring Framework Based on Internet of Things: A Survey*. IEEE, 2017.
- [2] Hevner A., March S., and Park J. *Design Science In Information System Research*. Mis Quarterly, 2004.
- [3] Islam A., Belqasmi F., Glitho R., and Khendek F. Implementing oma restful location services in wireless sensor environments. In *IEEE Symposium on Computers and Communications (ISCC)*, number ISBN 978-1-4673-2713-8 in Computer Science, pages 617–620, Cappadocia, Turkey, July 2012. IEEE.
- [4] Obando A. and Garita C. Diseño general de una red inalámbrica para monitoreo de salud de puentes. In *Convención IEEE de Estudiantes de Centroamérica y Panamá - CONESCAPAN XXXIV*, number 655 in -, San Salvador, El Salvador, Setiembre 22-25 2015.
- [5] Weis S. A. Rfid (radio frequency identification): Principles and applications.
- [6] T. M. Ahlborn, R. Shuchman, L. L. Sutter, C. N. Brooks, D. K. Harris, J. W. Burns, K. A. Endsley, D. C. Evans, K. Vaghefi, and R. C. Oats. *The State-of-the-Practice of Modern Structural Health Monitoring for Bridges: A Comprehensive Review*. Michigan Technological University, 2010.
- [7] GSM Association. Mobile nfc infrastructure, 2012.
- [8] Noel B., Abdaoui a., Elfouly t., Ahmed H., Badawy A., and Shehata S. *Structural Health Monitoring Using Wireless Sensor Networks: A Comprehensive Survey*. IEEE Communcations Survey and Tutorials, 2017.
- [9] Fatih M.and Turgut C. Design and implementation of restful wireless sensor network gateways using node.js framework. In *European Wireless 2014; 20th European Wireless Conference*, number ISBN 978-3-8007-3621-8 in Wireless European, pages 1–6, Barcelona, Spain, May 2014. VDE.
- [10] Garita C. Federated information management for virtual enterprises. In *Informatics Institute*, page 228, Amsterdam, 2001. The Netherlands: University of Amsterdam.
- [11] Garita C. A case study of vo education in costa rica. In *5th IFIP Working Conference on Virtual Enterprises*, Virtual Enterprises and Collaborative Networks, pages 589–596, Toulouse, France, August 2004. PRO-VE.

- [12] Garita C. Enfoques de integración de información para sistemas de monitoreo de salud estructural de puentes. *Tecnología en Marcha*, 29(641), 2016.
- [13] Garita C., Kaletas E.C., Afsarmanesh H., and Hertzberger L.O. A service interface definitions catalogue for virtual enterprises in tourism. In *IFIP — The International Federation for Information Processing*, volume 101 of *IFIPAICT*, pages 97–108, Amsterdam, The Netherlands, August 2002. University of Amsterdam, Springer Science+Business.
- [14] Garita C. and Ortiz G. Towards a workflow management approach for health monitoring of bridges. In *15th IFIP International Working Conference on Virtual Enterprises PRO-VE*, number 629 in 1, pages 489–497, Amsterdam, The Netherlands, 6-8 October 2014. Springer.
- [15] Garita C. and Ortiz G. The e-bridge 2.0 approach for shm of bridges in costa rica. In *10th International Workshop on Structural Health Monitoring - IWSHM 2015*, number 646 in -, University of Stanford, California, September 1-3 2015.
- [16] Garita C. and Ortiz G. e-bridge 3.0: A strategic approach to structural health monitoring of bridges in costa rica. In F. J. Mata and A. Pont, editors, *6th IFIP World Information Technology Forum, WITFOR 2016*, number 698 in 481, San Jose, Costa Rica, September 12-16 2016. Springer-Verlag.
- [17] Garita C. and Ortiz G. The e-bridge project approach for shm of bridges in costa rica: Maintenance bridge tools. In *7th Workshop on Civil Structural Health Monitoring - CSHM-7 2017*, number 724 in -, Medellín, Colombia, June 22-23 2017.
- [18] Garita C. and Ortiz G. Integrando información estratégica para monitoreo de puentes nacionales. In *III Jornadas Costarricenses de Computación e Informática - JoCICI 2017*, number 725 in -, Cartago, Costa Rica, Julio 23-24 2017.
- [19] Garita C. and Ortiz G. Development of a business intelligence prototype for bridge health monitoring. In *International Conference on Information Systems and Computer Science (INCISCOS)*, number ISBN 978-1-5386-7612-7 in Computer Science, pages 337–342, Quito, Ecuador, Ecuador, December 2018. IEEE.
- [20] Garita C., Ortiz G., and Mora J. Análisis de requerimientos para un sistema nacional de monitoreo de puentes. *Revista Tecnología en Marcha*, -(706), 2018.
- [21] Garita C., Afsarmanesh H., and Hertzberger L.O. A survey of distributed information management approaches for virtual enterprise infrastructures. In *Managing Virtual Web Organizations in the 21st Century: Issues and Challenges*, number ISBN 978-1-930708-24-2 in -, page 19, Amsterdam, The Netherlands, 2002. University of Amsterdam, IGIglobal.
- [22] Pautasso C. Bpel for rest. In *International Conference on Business Process Management, 2008*, page 278–293, Lugano, Switzerland, January 2008. Faculty of Informatics, University of Lugano.
- [23] G. Contreras. Lanammeucr genera primer inventario de puentes de costa rica.

- [24] Kotsopoulos S. D., Casalegno F., Recasens A., and Graybill W. Developing a restful communication protocol and an energy optimization algorithm for a connected sustainable home. In *URB-IOT '14: Proceedings of the First International Conference on IoT in Urban Space*, number ISBN 978-1-63190-037-2 in Computer Communication Networks, pages 62–68, Rome, Italy, October 2014. urb-iot.
- [25] Sazonova E., Janoyan K., and Jhac R. *Wireless intelligent sensor network for autonomous structural health monitoring*. SPIE, 2004.
- [26] Navarro F. and Ureña D. *Mediciones y análisis de vibraciones en el puente del Virilla*. Investiga TEC, 2015.
- [27] Node.js Foundation. About node.js.
- [28] Node.js Foundation. V8 documentation.
- [29] Fox G, Pallickara S., and Wang M. A demonstration of collaborative web services and peer-to-peer grids. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, number ISBN:0-7695-2108-8/04 in -, page 62, Washington, DC, USA, April 2004. SCS, IEEE Computer Society.
- [30] Ortiz G. *eBridge Predicción Remota de fallas en puentes*. eBridge, Civco, TEC, 2012.
- [31] Y. Gao and B. Spencer. *Department of Civil and Environmental Engineering University of Illinois at Urbana-Champaign*. San Val, 1995.
- [32] Afsarmanesh H., Garita C., Camarinha-Matos L., and Lima C. Workflow support for management of information of prodnet ii. In *5th International Workshop on Intelligent Manufacturing Systems - IMS*, pages 41–46, Gramado, Brazil, 1998.
- [33] Brumley B. H., Deines K. L., Cabrera R. G., and Terray E. A. Broadband acoustic doppler current profiler, 1997.
- [34] Tschofenig H., Arkko J., Thaler D., and McPherson D. *Architectural Considerations in Smart Object Networking*. Internet Architecture Board (IAB), 2015.
- [35] Zhao H. and Doshi P. Towards automated restful web service composition. In *IEEE International Conference on Web Services, 2009*, number 30602 in -, pages 189–196, Los Angeles, CA, January 2009. Faculty of Informatics, University of Lugano. IEEE.
- [36] Moiseenko I., Stapp M., and Oran David. Communication patterns for web interaction in named data networking in named data networking. In *ICN'14, September 24–26, 2014, Paris, France.*, number ACM 978-1-4503-3206-4/14/09 in -, pages–, Paris, France, September 2014. ACM.
- [37] Westerweel J. *Fundamentals of digital particle image velocimetry*, 1997.
- [38] Camarinha-Matos L. and Afsarmanesh H. Collaborative networks: a new scientific discipline. *Journal of Intelligent Manufacturing*, -(16):439–452, 2005.

- [39] Coz J. L., Hauet A., Pierrefeu G., Dramais G., and Camenen B. Performance of image-based velocimetry (Ispiv) applied to flash-flood discharge measurements in mediterranean rivers. In *Journal of Hydrology, 2010*, volume 394 of -, pages 42–52, University of Stanford, California, November 2010. Elsevier.
- [40] Hughes L. *Introduction to Middleware: Web Services, Object Components, and Cloud Computing*. CRC Press, 2017.
- [41] Chaves M., Ruiz A., Garita C., and Ortiz G. Preliminary design of a low-cost water level monitoring system for bridges. In *IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII)*, number ISBN 978-1-5386-6122-2 in Computer Communication Networks, pages 1–6, San Salvador, El Salvador, El Salvador, December 2018. IEEE.
- [42] Enckell M. *Lessons Learned in Structural Health Monitoring of Bridges Using Advanced Sensor Technology*. Doctorate Ph.D. Thesis, Royal Institute of Technology, Sweden, 2011.
- [43] Fatih M. and Turgut C. *Design and Implementation of RESTful Wireless Sensor Network Gateways Using Node.js Framework*. VDE VERLAG, 2014.
- [44] Gutierrez M. *Desarrollo de un prototipo de red inalámbrica para monitoreo de salud de puentes*. eBridge, Civco, TEC, 2016.
- [45] Gutierrez M. and Garita C. Prototype development of a wireless embedded system for bridge monitoring. In *37th Central America and Panama Convention (CONCAPAN XXXVII)*, number 733, Managua, Nicaragua, November 15-17 2017. IEEE. An optional note.
- [46] Gutierrez M. and Garita C. Prototype development of a wireless embedded system for bridge monitoring. In *IEEE 37th Central America and Panama Convention (CONCAPAN XXXVII)*, number 733 in CONCAPAN, Managua, Nicaragua, November 15-17 2017. IEEE. An optional note.
- [47] Gutierrez M. and Garita C. Prototype development of a wireless embedded system for bridge monitoring. In *IEEE 37th Central America and Panama Convention (CONCAPAN XXXVII)*, number ISBN 978-1-5386-3509-4 in Computer Science, pages 1–6, Managua, Nicaragua, November 2017. IEEE.
- [48] Hüffmeyer M. and Schreier U. Restacl - an access control language for restful services. In *ABAC'16, March 11 2016, New Orleans, LA, USA (ACM)*, number ISBN 978-1-4503-4079-3/16/03 in 1, page 76, New Orleans, LA, USA, March 2016. ACM.
- [49] C. Marsan. *The Internet of Things: An Overview*. Internet Society, 2015.
- [50] G. Mulligan and D Gracanin. A comparison of soap and rest implementations of a service based interaction independence middleware framework. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1423–1432, Austin, Texas, July 2009. IEEE.
- [51] Shahid N. and Aneja S. Internet of things: Vision, application areas and research challenges. In *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, number ISBN 978-1-5090-3243-3 in -, pages 583–587, Palladam, India, February 2017. IEEE.

- [52] Xu N., Rangwala S., Chintalapudi K., Ganesan D., Broad A., Govindan R., and Estrin D. *A Wireless Sensor Network For Structural Monitoring*. ACM, 2004.
- [53] Hideki O., Mitran P., Poor V., and Tarokh V. *Collaborative Beamforming for Distributed Wireless Ad Hoc Sensor Networks*. IET Digital Library, 2007.
- [54] Morville P. and Rosenfeld L. *Information Architecture for the World Wide Web: Designing Large-Scale Web Sites, 3rd Edition*. O'REILLY, 2006.
- [55] Real Python. Api integration in python.
- [56] Le-Trung Q. and Kotsis G. A network model for manet nodes and actors collaboration to optimize processing in event areas. In *ICN'14, September 24–26, 2014, Paris, France.*, number ACM 978-1-59593-808-4/07/0010 in -, pages-, Crete Island, Greece, October 2007. ACM.
- [57] raspberrypi.org. rc.local.
- [58] Akhavan S. and Gazor S. Collaborative communications: Joint relay and protocol selection. In *11th Canadian Workshop on Information Theory*, number ISBN 978-1-4244-3400-8 in Information Theory, pages 25–28, Ottawa, ON, Canada, June 2009. IEEE.
- [59] Jeonga S., Houb R., Lynchb J., Sohnc H., and Lawa K.H. *A Scalable Cloud-based Cyberinfrastructure Platform for Bridge Monitoring*. Department of Civil and Environmental Engineering, Stanford University, USA, 2017.
- [60] Kim S., Pakzad S., Culler D., Demmel J., Fenves G., Glaser S., and Turon M. *Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks*. University of Western Sydney, Australia, 2007.
- [61] Majumdar S., Asif M., Melendez J.O., Kanagasundaram R., Lau D.T., Nandy B., Zaman M., Srivastava P., and Goel N. Middleware architecture for sensor-based bridge infrastructure management. In *Proceedings of the 15th Communications and Networking Simulation Symposium*, number ISBN: 978-1-61839-785-0 in -, pages-, Orlando, Florida, USA, March 2012. SCS, ACM.
- [62] Wijetunge S., Gunawardana U., and Liyanapathirana R. *Wireless Sensor Networks for Structural Health Monitoring: Considerations for Communication Protocol Design*. University of Western Sydney, Australia, 2010.
- [63] Won S. and Soo H. Decentralized access permission control using resource-oriented architecture for the web of things. In *16th International Conference on Advanced Communication Technology*, number ISBN 978-89-968650-3-2 in Advanced Communication Technology, pages 749–753, Pyeongchang, South Korea, February 2014. IEEE.
- [64] Fielding T. *Architectural Styles and the Design of Network-based Software Architectures*. University of California Irvine, 2000.
- [65] Fredrich T. *RESTful Service Best Practices Recommendations for Creating Web Services*. RestApiTutorial, 2012.
- [66] Dargie W. and Poellabauer C. *Fundamentals of Wireless Sensor Networks Theory and Practice*. Wiley, 2010.

- [67] Li W., Bao J., and Shen W. *Collaborative Wireless Sensor Networks: A Survey*. IEEE, 2011.
- [68] S. Wijetunge, U. Gunawardana, and R. Liyanapathirana. *Wireless Sensor Networks for Structural Health Monitoring: Considerations for Communication Protocol Design*. University of Western Sydney, Australia, International Conference on Telecommunications, 2010.
- [69] Dong Y., Song R., and Liu H. *Bridges Structural Health Monitoring and Deterioration Detection Synthesis of Knowledge and Technology*. AUTC, 2010.
- [70] Lei Y. Research and implementation of wsn-based data acquisition and analysis system using agent collaboration. In *2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, number ISBN 978-1-4244-4544-8 in PEITS, pages 235–237, Shenzhen, China, February 2010. IEEE.
- [71] Han Z. and Vincent H. *Lifetime Improvement in Wireless Sensor Networks via Collaborative Beamforming and Cooperative Transmission*. IET Digital Library, 2007.