Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Dirección General de Estudios de Posgrado

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

Sistema de medición inteligente de consumo de agua en hogares usando IoT y Cloud Computing

TESIS

Para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software

AUTOR

Henry Gustavo FUENTES VILLANUEVA

ASESOR

David Santos MAURICIO SÁNCHEZ

Lima, Perú

2021

Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Fuentes, H. (2021). *Sistema de medición inteligente de consumo de agua en hogares usando IoT y Cloud Computing*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Henry Gustavo Fuentes Villanueva
Tipo de documento de identidad	DNI
Número de documento de identidad	PE / 44444568
URL de ORCID	https://orcid.org/0000-0001-6208-1465
Datos de asesor	
Nombres y apellidos	David Santos Mauricio Sánchez
Tipo de documento de identidad	DNI
Número de documento de identidad	PE / 06445495
URL de ORCID	https://orcid.org/0000-0001-9262-626X
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Dr. Ciro Rodriguez Rodriguez
Tipo de documento	DNI
Número de documento de identidad	PE / 06020241
Miembro del jurado 1	
Nombres y apellidos	Mg. Cayo Víctor León Fernández
Tipo de documento	DNI
Número de documento de identidad	PE / 07001405
Miembro del jurado 2	
Nombres y apellidos	Mg. Carlos Ernesto Chavez Herrera
Tipo de documento	DNI
Número de documento de identidad	PE / 40788722
Datos de investigación	

Línea de investigación	C.0.3.22. Ingeniería de software
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Latitud: -12.055926 Longitud: -77.083469
Año o rango de años en que se realizó la investigación	2018 - 2020
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Vicedecanato de Investigación y Posgrado
Unidad de Posgrado

ACTA DE SUSTENTACIÓN VIRTUAL DE TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON MENCIÓN EN INGENIERÍA DE SOFTWARE

A los once (11) días del mes de noviembre de 2021, siendo las 4:00 pm., se reunieron en la sala virtual https://meet.google.com/sid-hfsj-hsg el Jurado de Tesis conformado por los siguientes docentes:

Dr. Ciro Rodriguez Rodriguez (Presidente)
Mg. Carlos Ernesto Chávez Herrera (Miembro)
Mg. Cayo Víctor León Fernández (Miembro)
Dr. David Santos Mauricio Sánchez (Miembro Asesor)

*Se inició la Sustentación invitando al candidato a Magíster **HENRY GUSTAVO FUENTES VILLANUEVA,** para que realice la exposición oral y virtual de la tesis para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software, siendo la Tesis intitulada:*

"SISTEMA DE MEDICIÓN INTELIGENTE DE CONSUMO DE AGUA EN HOGARES USANDO IOT Y CLOUD COMPUTING"

Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

(17) Diecisiete – Muy Bueno

*Por tanto, el presidente del Jurado, de acuerdo al Reglamento General de Estudios de Posgrado, otorga al Bachiller **HENRY GUSTAVO FUENTES VILLANUEVA** el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software.*

Siendo las. ……17:35…… horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.

Dr. Ciro Rodriguez Rodriguez
(Presidente)

Mg. Carlos Ernesto Chávez Herrera
(Miembro)

Mg. Cayo Víctor León Fernández
(Miembro)

Dr. David Santos Mauricio Sánchez
(Miembro Asesor)

DEDICATORIA

Este trabajo está dedicado a mi esposa y familiares por haber sido mi apoyo a lo largo de toda mi carrera de maestría y a lo largo de mi vida.

A todas las personas especiales que me acompañaron en esta etapa, aportando a mi formación tanto profesional y como ser humano.

AGRADECIMIENTOS

Al profesor Dr. David Santos Mauricio Sánchez, por la orientación, revisión y dedicación para que este trabajo cumpla con los objetivos trazados.

A todos los docentes del programa de maestría, por sus grandes enseñanzas que sirvieron de mucho en la elaboración de la presente tesis.

Al jurado revisor, por sus revisiones y recomendaciones de la presente tesis.

A todas aquellas personas que indirectamente me ayudaron para culminar este trabajo y que muchas veces constituyen un invalorable apoyo.

Y por encima de todo doy gracias a Dios.

ÍNDICE GENERAL

DEDICATORIA iv

AGRADECIMIENTOS v

ÍNDICE GENERAL vi

LISTA DE TABLAS x

LISTA DE FIGURAS xi

RESUMEN xiv

ABSTRACT xv

CAPÍTULO 1: INTRODUCCIÓN 16

1.1. Situación problemática 16

1.2. Formulación del problema 17

1.2.1. Problema general 17

1.2.2. Problemas específicos 17

1.3. Justificación teórica 18

1.4. Justificación práctica 18

1.5. Motivación 19

1.6. Objetivos 20

1.6.1. Objetivo general 20

1.6.2. Objetivos específicos 20

1.7. Propuesta 21

1.8. Organización de la tesis 21

CAPÍTULO 2: MARCO TEÓRICO 23

2.1. Internet Of Things (IoT) 23

2.2. Dispositivos IoT 25

2.3. Node-RED 26
2.4. Medidor inteligente de agua 27
2.5. Cloud Computing 29
2.6. Métodos de detección de fuga 31
CAPÍTULO 3: ESTADO DEL ARTE 32
3.1. Metodología de la investigación 32
3.2. Planificación de la revisión 32
3.3. Desarrollo de la revisión 34
3.4. Resultados de la revisión 35
3.5. Análisis 39
3.5.1. Q1: ¿Qué arquitecturas existen para un sistema de medición inteligente? 39
3.5.2. Q2: ¿Qué funcionalidades tiene un sistema medición inteligente? 44
3.5.3. Q3: ¿Qué tecnologías inalámbricas usa un dispositivo inteligente? 46
3.5.4. Q4: ¿Qué medidas de seguridad tiene un sistema medición inteligente? 48
3.5.5. Q5: ¿Qué algoritmos o reglas existen para detectar una fuga de agua? 49
CAPÍTULO 4: SISTEMA PROPUESTO DE MEDICIÓN INTELIGENTE DEL CONSUMO DE AGUA 52
4.1. Diagrama en alto nivel 52
4.2. Componentes 53
4.2.1. Adquisición de datos 53
4.2.2. Edge Gateway 53
4.2.3. Mecanismo de anti – manipulación (anti-tampering) 53

4.2.4. Cloud 54
4.2.5. Algoritmo de detección de fugas 54
CAPÍTULO 5: IMPLEMENTACIÓN DEL SOFTWARE 57
5.1. Arquitectura del software 57
5.2. Requisitos 63
5.3. Metodología de desarrollo 65
5.4. Diagrama de base de datos 67
5.5. Limitación 69
5.6. Pruebas de software 69
5.7. Funcionamiento 72
CAPÍTULO 6: VALIDACIÓN 78
6.1. Caso de estudio 78
6.2. Aplicación 78
6.3. Discusión de resultados 86
CAPÍTULO 7: CONCLUSIONES 89
7.1. Conclusiones 89
7.2. Recomendaciones 90
7.3. Trabajos futuros 90
REFERENCIAS BIBLIOGRÁFICAS 91
ANEXOS 98
ANEXO A. Prototipos 98
ANEXO B. Manual de Usuario 102
ANEXO C. Producción Científica sobre la investigación 108
ANEXO C.1. Carta de Aceptación de artículo 108

ANEXO C.2. Resumen del artículo 109

ANEXO C.3. Ranking scimagojr 110

LISTA DE TABLAS

Tabla 1. Criterios de selección y exclusión 33

Tabla 2. Lista de artículos escogidos 36

Tabla 3. Arquitecturas para un sistema de medición inteligente 42

Tabla 4. Funcionalidades en sistemas de medición inteligente 45

Tabla 5. Tecnologías inalámbricas 47

Tabla 6. Medidas de seguridad en sistemas de medición inteligente 49

Tabla 7. Reglas de predicción de fuga de agua 50

Tabla 8. Lista de iteraciones del desarrollo del software 65

Tabla 9. Campos de la tabla de mediciones 67

Tabla 10. Campos de la tabla de alertas 68

Tabla 11. Métrica número de alertas recibidas 70

Tabla 12. Matriz de confusión 82

Tabla 13. Escenarios de consumo normal de agua 83

Tabla 14. Escenarios de consumo anómalo de agua 85

Tabla 15. Comparación de consumo real de agua vs. consumo registrado por el sistema 87

Tabla 16. Resultados de la matriz de confusión por algoritmo 88

LISTA DE FIGURAS

Figura 1. Ahorro usando sistemas inteligentes 19

Figura 2. Secuencia de lectura de la tesis 22

Figura 3. Arquitectura de tres y 5 capas en IoT 25

Figura 4. ESP8266 25

Figura 5. Raspberry Pi 3 Model B+ 26

Figura 6. Node-RED 27

Figura 7. Características de un medidor inteligente 28

Figura 8. Sensor de flujo de agua G3/4 29

Figura 9. Proceso de búsqueda de artículos 34

Figura 10. Cantidad artículos escogidos por años de publicación 35

Figura 11. Cantidad artículos escogidos por base de datos 36

Figura 12. Arquitectura Integrada IoT 40

Figura 13. Arquitectura SWaMM 40

Figura 14. Arquitectura Smart Water Quality Monitoring System (SWQMS) 41

Figura 15. Arquitectura Wireless Water Quality Monitoring System (WWQMS) 42

Figura 16. Marco de trabajo basado en la geo-localización inteligente 45

Figura 17. Diagrama a alto nivel del sistema de medición inteligente del consumo de agua propuesto 53

Figura 18. Algoritmo de detección de fugas 56

Figura 19. Vista física de la medición inteligente del consumo de agua 58

Figura 20. Vista tecnológica de la medición inteligente del consumo de agua ..58

Figura 21. Medidor inteligente de agua 59

Figura 22. Raspberry Pi (Edge Gateway) 60

Figura 23. Node-RED – Flujograma del proceso de transformación y almacenamiento de datos 60

Figura 24. Node-RED – Flujograma del proceso de envío de datos desde el componente “Edge Gateway” 60

Figura 25. IBM stream Flow analysis 61

Figura 26. Vista de ubicación de dispositivos 62

Figura 27. Vista consumo: (a) Tiempo Real; (b) histórico 63

Figura 28. Diagrama de Base de datos del sistema 69

Figura 29. Alertas generadas 71

Figura 30: Número de alertas recibidos 72

Figura 31: Ejecución – Vista inicio de sesión 73

Figura 32: Ejecución – Vista localización de dispositivos 74

Figura 33: Ejecución – Vista consumo histórico 75

Figura 34: Ejecución – Vista Node-RED – Gateway recibe consumo en tiempo real 75

Figura 35: Ejecución – Vista consumo en tiempo real 76

Figura 36: Ejecución – Vista “IBM Flow Stream” 77

Figura 37: Ejecución – Alerta celular 77

Figura 38. Maqueta de medidor inteligente 79

Figura 39. Prototipo – Vista inicio de sesión 98

Figura 40. Prototipo – Vista Pantalla inicial 98

Figura 41. Prototipo – Vista filtro dispositivos 99

Figura 42. Prototipo – Vista búsqueda usuarios consumo 99

Figura 43. Prototipo – Vista detalle usuario – dispositivo 100

Figura 44. Prototipo – Vista histograma consumo histórico 101

Figura 45. Prototipo – Vista tiempo real consumo.....101
Figura 46. Manual de usuario – Vista inicio de sesión 102
Figura 47. Manual de usuario – Vista bienvenida 103
Figura 48. Manual de usuario – Vista localización de dispositivos 104
Figura 49. Manual de usuario – Vista búsqueda de usuarios consumo 105
Figura 50: Manual de usuario – Vista detalle usuario consumo 105
Figura 51: Manual de usuario – Vista consumo histórico 106
Figura 52: Manual de usuario – Vista consumo tiempo real.....106
Figura 53: Manual de usuario – Alerta en celular.....107
Figura 54: Manual de usuario – Vista completa de alerta en celular 107

RESUMEN

En la actualidad, en varios lugares del mundo, el consumo de agua no es medido y visualizado en tiempo real, además, las fugas de agua no son detectadas a tiempo y con alta precisión, por lo que se genera un desperdicio innecesario de agua. Es por ello que, en esta tesis, se presenta la implementación de un sistema de medición inteligente del consumo de agua bajo una arquitectura, con alto desacoplamiento e integración de diversas tecnologías, que permite visualizar, en tiempo real, estos consumos, además, se propone un algoritmo de detección de fugas basado en reglas, contexto histórico y ubicación del usuario que logra cubrir 10 posibles escenarios de consumo de agua entre consumo normal y anómalos. El sistema permite recopilar los datos desde un medidor inteligente, el cual es preprocesado en un servidor local (Gateway) y enviado a la Cloud cada cierto tiempo para ser analizado por el algoritmo de detección de fugas y, simultáneamente, ser visualizado desde una interfaz web. Los resultados muestran 100 % de Exactitud (Accuracy), Exhaustividad (Recall), Precisión (Precision) y Valor F1 (F1 Score) en el algoritmo para detectar fugas, superior a otros trabajos, y un margen de error de 4.63 % en el registro de la cantidad de agua consumida.

Palabras clave: *Smart Metering; Water Consumption; Leak Detection Algorithm; IoT Devices; IoT Architecture; Cloud Computing*

ABSTRACT

Presently, in several parts of the world, water consumption is not measured or visualized in real time, in addition, water leaks are not detected in time and with high precision, generating unnecessary waste of water. That is why, this thesis presents the implementation of a smart water measurement consumption system under an architecture design, with high decoupling and integration of various technologies, which allows real-time visualizing the consumptions, in addition, a leak detection algorithm is proposed based on rules, historical context, and user location that manages to cover 10 possible water consumption scenarios between normal and anomalous consumption. The system allows data to be collected by a smart meter, which is preprocessed by a local server (Gateway) and sent to the Cloud from time to time to be analyzed by the leak detection algorithm and, simultaneously, be viewed on a web interface. The results show that the algorithm has 100% Accuracy, Recall, Precision, and F1 score to detect leaks, far better than other procedures, and a margin of error of 4.63% recorded by the amount of water consumed.

Keywords: *Smart Metering; Water Consumption; Leak Detection Algorithm; IoT Devices; IoT Architecture; Cloud Computing*

CAPÍTULO 1: INTRODUCCIÓN

1.1. Situación problemática

El agua es el recurso natural más importante para los seres humanos, por lo que la Organización Mundial de la Salud (OMS) recomienda que una persona debería consumir un promedio de 100 litros por día para atender todas sus necesidades (United Nations, 2014), pero en las principales capitales de Sudamérica, se excede a lo recomendado, por ejemplo, La Paz consume 120 litros, seguido de Bogotá con 168 litros, luego Santiago con 200 litros, Quito con 220 litros y, finalmente, Lima con 250 litros, cuyo excedente equivale a más de 77 mil piscinas olímpicas de agua al año (SUNASS, 2017). Por otro lado, en EE.UU, dentro de una casa, el consumo diario de agua es, aproximadamente, 138 galones (522 litros) , donde la descarga del inodoro constituye el mayor gasto (24 %), seguido de las cañerías (20 %), duchas (20 %), lavadora de ropa (16 %), fugas(13 %), bañeras (3 %), lavadora de platos (2 %) y otros (3 %) (DeOreo et al., 2016). Debido al consumo excesivo del agua, es necesario buscar mecanismos para reducirla, y es notorio que las fugas es uno de los principales problemas que generan gastos innecesarios.

Estudios revelan que la cantidad de agua desperdiciada por fugas varía extensamente entre diferentes países. En países desarrollados de Europa, es aproximadamente 15 % (Francia: 27 %, UK: 21 %, Holanda: 5 %) (GrowingBlue, 2011), y, en EE. UU,. se estima el 13 % (DeOreo et al., 2016); por otro lado, algunos países en vías de desarrollo presentan un elevado índice que oscila entre 20 % a 70 % (Sharma y Vairavamoorthy, 2009).

Es bien sabido que, en varios países, no se mide en tiempo real el consumo de agua, por lo que el consumidor debe esperar hasta el siguiente mes para recibir el estado de su consumo, que, en general, es obtenido a través de mediciones "manuales" que se hacen a cada medidor. Esto genera un desconocimiento diario del consumo del agua y la imposibilidad de detectar a tiempo una fuga no visible que trae como consecuencia un despilfarro de este recurso y pérdidas

económicas inclusive para la empresa proveedora cuando este recurso es subvencionado.

Actualmente, existen varios sistemas de medición inteligente que además ayudan a detectar, predecir y alertar alguna fuga o exceso de agua, sin embargo, ninguno de estos estudios propone una arquitectura IoT, que permite visualizar, en tiempo real, el consumo de agua, e integra a la vez la ubicación, la data histórica, y reglas generales, para detectar posibles fugas de agua, y así mejorar la tasa de precisión del 98%.

1.2. Formulación del problema

1.2.1. Problema general

¿De qué manera un sistema de medición inteligente basado en una arquitectura de IoT y Cloud Computing permite detectar fugas de agua en hogares con una alta precisión, y, además, visualizar los consumos de agua en tiempo real?

1.2.2. Problemas específicos

- **PE1:** ¿Qué investigaciones científicas abordan el ecosistema de un sistema de medición inteligente, arquitecturas de IoT y Cloud computing y algoritmos de detección de fuga?
- **PE2**: ¿Es posible proponer e implementar una arquitectura, para sistemas de medición inteligente de consumo de agua, que permita visualizar consumos de agua en tiempo real?
- **PE3**: ¿Es posible registrar los consumos de agua con un margen de error menor a 5%?
- **PE4**: ¿Es posible ejecutar un algoritmo capaz de detectar posibles fugas con una precisión mayor al 98%?
- **PE5:** ¿Es posible realizar experimentos numéricos con un dataset de prueba para determinar la fiabilidad y precisión del algoritmo propuesto?

1.3. Justificación teórica

La presente tesis se realiza con el propósito de aportar al conocimiento existente sobre una arquitectura IoT para sistemas de medición inteligente del consumo de agua, que permite visualizar, en tiempo real, el consumo, e integra a la vez la ubicación, la data histórica, y reglas generales, para detectar posibles fugas de agua, y así mejorar la tasa de precisión con respecto a otros estudios similares.

1.4. Justificación práctica

El uso de sistemas de medición inteligentes pueden generar ahorro de agua del 2.8 % al 10.0 % (Liu y Mukheibir, 2018), esto se puede visualizar en la Figura 1 , y así se evita tener mediciones equivocadas generadas por error humano, y crea la confianza en los usuarios en cuanto a los recibos de consumo de agua y a los pagos (Joo et al., 2015), además, permite informar sobre alguna fuga de agua, lo que reduce la probabilidad hasta en 50 % de que esta vuelva a ocurrir (Schultz et al., 2018).

Cabe mencionar que en el Perú, según la SUNASS (2018), un caño malogrado desperdicia 22 mil litros de agua, lo que representa un gasto adicional en el recibo de S/. 55 al mes, sin embargo, si se tuviera esto monitoreado, se podría obtener un ahorro económico.

Feedback dimension	Savings Range	Mean	Median
Medium:			
In Home Display (IHD)	4.0%– 8.5%	6.4%	6.8%
In Home Display (IHD) & Portal	5.0%	–	–
Paper Report	6.6% –8.0%	7.5%	9.0%
Post Card	7.9%	–	–
Online Portal*	(−1%) 3.0%– 10.0%	5.8%	5.3%
Online Portal & Paper report	2.8%– 4.6%	4.4%	4.8%
Letter & Phone call	6.9%	–	–
Smartphone App**	0.0%	0.0%	0.0%
Content:			
(Near) real-time data	4.0%– 10.0%		
End-use data	7.9%– 8.0%		
Leak data	4.0%– 10.0%		
Comparative use data	2.8% –10.0%		
Duration:			
< 2 months	2.8% –3.0%	2.9%	2.9%
4 ≤ 6 months	4.1%–8.0%	5.9%	7.25%
12–14 months	4.2%– 8.5%	5.9%	5.2%
24 months	4.0%		
Frequency:			
Once-off	24.1%		
Bi-annually	8.0%		
Bi-monthly	2.8%– 8.0%	5.3%	4.9%
Monthly	7.9%		
Weekly	3.0%		
Daily	4.2% – 6.8%	5.5%	5.5%
Real-time	4.0% – 10.0%	5.5%	5.3%
Program Scale (number of households):			
< 50	3.0% – 8.0%		
50–99	4.2% – 6.8%		
100–149	(−1.0%)		
150–199	6.6% – 8.5%		
200–999	4.0% – 10.0%		
1,000–2000	2.8% – 4.1%		
> 2000	4.6% – 8.0%		

*Notes: *Most online portals typically come with the option to sign up for usage alerts sent via email and/or SMS, thereby utilising multiple feedback mediums.*

Figura 1. **Ahorro usando sistemas inteligentes**

Fuente. (Liu y Mukheibir, 2018)

1.5. Motivación

Actualmente, existen varios sistemas de medición inteligente que, además, ayudan a detectar, predecir y alertar a tiempo alguna fuga o exceso de agua. A través del uso de las reglas Water Balance, Minimun Night Flow (MNF) y métodos estadísticos, se revela que se ha llegado a predecir con 97 % de precisión la fuga

de agua (Farah y Shahrour, 2017), por otro lado, con métodos de machine learning se alcanzó 74 % de precisión (Patabendige et al., 2018). También, con un algoritmo que integra reglas de MNF y Continuous Non-Zero Water Consumption (CNZ), se logró alcanzar 98 % de precisión (Luciani et al., 2019). Sin embargo, ninguno de estos estudios implementa una arquitectura IoT, que permite visualizar, en tiempo real, el consumo de agua, e integra a la vez la ubicación, la data histórica, reglas como MNF, CNZ y otras, para detectar posibles fugas de agua, además, aún la tasa de precisión se puede mejorar.

1.6. Objetivos

1.6.1. Objetivo general

Implementar un sistema de medición inteligente basada en una arquitectura de IoT y Cloud Computing que permita detectar fugas de agua en hogares con una alta precisión, y, además, visualizar los consumos de agua en tiempo real.

1.6.2. Objetivos específicos

Para lograr el objetivo general, es necesario alcanzar los siguientes objetivos específicos:

- **OE1:** Revisar artículos científicos sobre el ecosistema de un sistema de medición inteligente, para estudiar y seleccionar arquitecturas de IoT y Cloud computing, funcionalidades, tecnología inalámbrica y seguridad que debe haber en esta. Además, investigar sobre reglas, modelos estadísticos y de machine learning, usados para detectar posibles fugas de agua.
- **OE2:** Proponer e implementar una arquitectura, para sistemas de medición inteligente de consumo de agua, que permita visualizarlos en tiempo real.
- **OE3:** Registrar los consumos de agua con un margen de error menor a 5%.
- **OE4**: Ejecutar un algoritmo capaz de detectar posibles fugas con una precisión mayor al 98%.
- **OE5:** Realizar experimentos numéricos con un dataset de prueba para determinar la fiabilidad y precisión del algoritmo propuesto.

1.7. Propuesta

La integración de diversos aspectos desarrollados para la detección de fugas de agua se complementa en muchas situaciones, por ejemplo, la detección basada en data histórica se orienta al comportamiento del consumo y no contempla escenarios particulares que solo pueden ser detectados a través de reglas, como CNZ. Por ello, en este trabajo se propone un sistema de medición del consumo y detección de fuga de agua que integra la ubicación del usuario, data histórica y reglas, además, está basada en una arquitectura de IoT y Cloud Computing. La ubicación permite identificar una posible fuga en caso de que el usuario no se encuentre en casa, la data histórica del consumo de agua del usuario permite hallar un patrón de consumo diario, la arquitectura IoT permite la captura y el preprocesamiento de los datos de consumo de agua obtenidos a través de los medidores inteligentes, en tiempo real, y, por último, en la Cloud, se realiza el análisis y visualización de estos datos.

1.8. Organización de la tesis

Esta tesis se encuentra organizada en siete (7) capítulos:

En el Capítulo 1, se hace la introducción a la investigación, se indica el problema, la justificación, la motivación, el objetivo, y la propuesta.

En el Capítulo 2, se describe las bases teóricas en donde se tocan los conceptos sobre sistemas de medición inteligente del consumo de agua y detección de fugas.

En el Capítulo 3, se presenta el estado del arte donde se realiza una revisión sistemática de la literatura sobre los diferentes aportes en sistemas de medición inteligente de consumo de agua que ayudan a responden las siguientes seis preguntas de investigación: Q1) ¿Qué arquitecturas existen para un sistema de medición inteligente? Q2) ¿Qué funcionalidades tiene un sistema medición inteligente? Q3) ¿Qué tecnologías inalámbricas usa un dispositivo inteligente? Q4) ¿Qué medidas de seguridad tiene un sistema medición inteligente? Q5) ¿Qué algoritmos o reglas existen para detectar una fuga de agua?

En el Capítulo 4, se describe el aporte propuesto con sus principales componentes.

En el Capítulo 5, se presenta la arquitectura, requisitos, metodología, diagrama de base de datos, limitaciones, pruebas y funcionamiento del software que implementa el aporte.

En el Capítulo 6, se muestra la validación a través de experimentos numéricos y los resultados obtenidos del aporte planteado.

En el Capítulo 7, se describe las conclusiones y los trabajos futuros.

Además, con el fin de facilitar la lectura de la presente tesis, en la Figura 2, se propone la secuencia de lectura. Para una mejor comprensión de la tesis se sugiere al lector dos alternativas: (i) Leer el resumen, luego, continuar con el Capítulo 1 y seguir leyendo en orden los demás capítulos (2, 3, 4, 5, 6 y 7). (ii) O leer el resumen, luego el Capítulo 1 y continuar leyendo el capítulo de su interés.

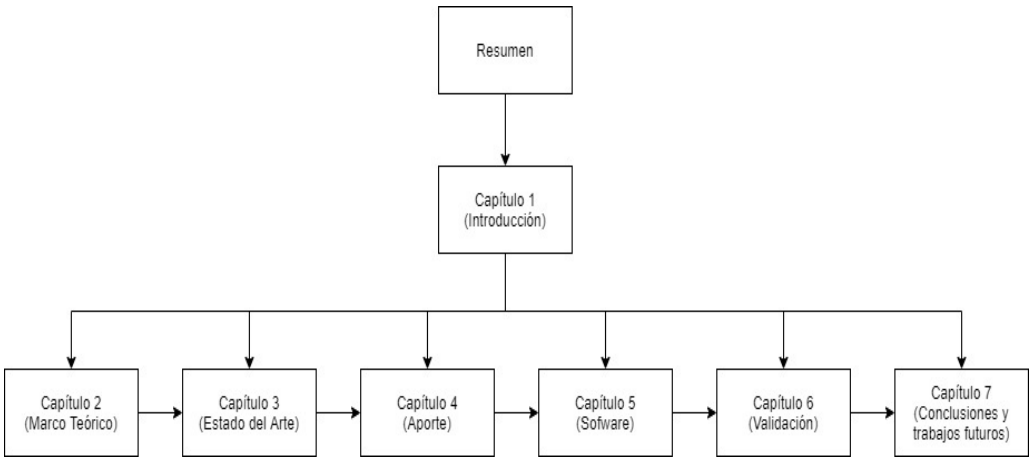


Figura 2. **Secuencia de lectura de la tesis**
Fuente. Elaboración propia

CAPÍTULO 2: MARCO TEÓRICO

En el presente capítulo, se define los siguientes conceptos que complementan la presente investigación: Internet Of Things (IoT), dispositivos IoT, Node-RED, medidor inteligente de agua, Cloud Computing y métodos de detección de fuga.

2.1. Internet Of Things (IoT)

Para Perumal et al.(2015) IoT es un habilitador que vincula objetos integrados que rodea el medio ambiente y realiza algún tipo de intercambio de mensajes entre ellos. Por otro lado, Oracle lo describe como una red de objetos físicos "Things" que están embebidos con sensores, software, y otras tecnologías con el propósito de conectar e intercambiar datos con otros dispositivos y sistemas a través del internet (Oracle, s. f.).

IoT no solo trata de conectar dispositivos, sino también de convertir los datos obtenidos en conocimiento, y este conocimiento en acción con el objetivo de solucionar los problemas de cualquier negocio usando su propia data (Microsoft, s. f.-c). Entre los usos que IoT brinda, se puede mencionar los siguientes:

- **Monitoreo remoto:** el hecho de manera constante o con ciertos intervalos cualquier tipo de activos (maquinaria, vehículos, entre otros) puede ayudar a reducir los costos del servicio, crear una mejor experiencia a los clientes y saber dónde se encuentran los recursos a cualquier momento.
- **Mantenimiento predictivo:** IoT, con el uso de software de machine learning, se puede predecir posibles fallas antes de que estas ocurran, lo cual permite que los técnicos puedan intervenir antes de que los clientes se den cuenta de que existe algún problema.
- **Gestión de instalaciones:** el hecho de monitorear las instalaciones, infraestructura y otros espacios de los edificios permite mejorar el ahorro en el uso de luz, agua y la utilización de espacios.
- **Eficiencia en manufacturación:** Con los datos obtenidos de los dispositivos y sensores que intervienen en la cadena de producción, se puede identificar

cuellos de botella y, de esta manera, mejorar el proceso de producción, también, se puede reducir los tiempos muertos causados por fallas de equipos, además, se pueden eliminar errores en los productos antes de que estos sean completados.

Sethi y Sarangi (2017) menciona que la arquitectura más básica para IoT es la de tres capas, tal como se muestra en la Figura 3(A). La *Perception layer* es la capa física en la cual se encuentran los sensores para obtener información del medio ambiente. Luego, la *Network layer* es la responsable de conectar los dispositivos y servidores que están en una red, incluso, es usada para transmitir y procesar los datos del sensor. Finalmente, se tiene la *Application layer,* que define varias aplicaciones en la cual IoT puede ser desplegado, por ejemplo, casas o ciudades inteligentes.

Por otro lado, Sethi y Sarangi (2017) menciona que también existe una arquitectura más completa que consta de cinco capas, tal como se muestra en la Figura 3(B). La *Perception* y *Application layer* tienen las mismas funciones que se detallaron en la arquitectura de tres capas. La *Transport layer* se encarga de trasferir la data de los sensores desde la *Perception layer* a la *Processing layer* , y viceversa, a través de redes como Wireless 3G, LAN, Bluetooth, entre otras. Finalmente, la *Business layer* se encarga de gestionar todo el sistema IoT, incluyendo las aplicaciones, modelos de negocio, y la privacidad del usuario.

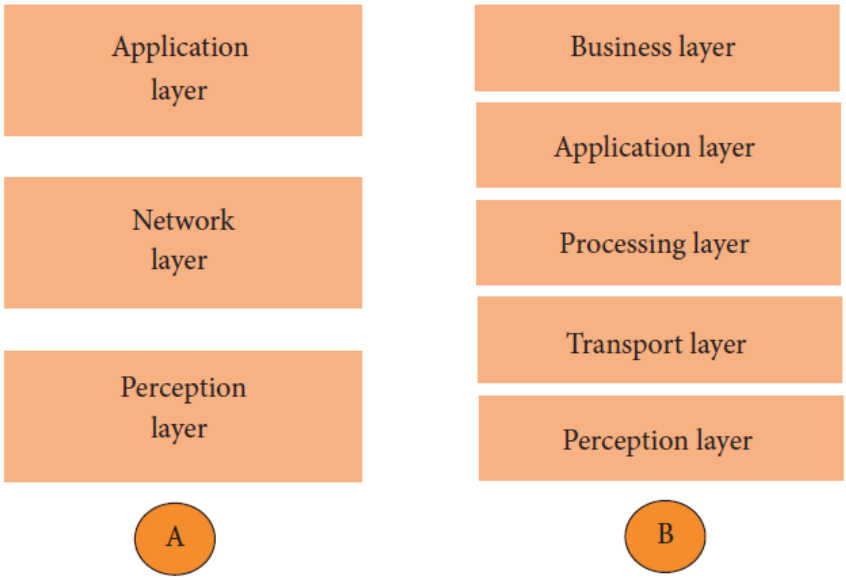


Figura 3. **Arquitectura de tres y 5 capas en IoT**
Fuente. (Sethi y Sarangi, 2017)

2.2. Dispositivos IoT

Un dispositivo muy común de bajo costo usado en proyectos de IoT es el ESP8266 (ver Figura 4. **ESP8266**), el cual es un chip integrado con conexión Wi-Fi capaz de ejecutar aplicaciones instaladas en ella. Este contiene un conector USB, el cual mediante un micro cable USB puede conectarse a una computadora y, de esta manera, se puede instalar sin ningún problema algún software (Handson Technology, s. f.).

Figura 4. **ESP8266**
Fuente.(Handson Technology, s. f.)

Otro dispositivo usado en IoT es la Raspberry Pi (ver Figura 5), el cual es una computadora de bajo costo que se ejecuta bajo el sistema operativo Linux, también provee un set de GPIO (general purpose input/output) pines que permite el control electrónico de sus componentes físicos.

Figura 5. **Raspberry Pi 3 Model B+**
Fuente.(RaspberryPi, s. f.)

2.3. Node-RED

Es una herramienta de programación que se usa para conectar dispositivos físicos (hardware), APIs y servicios en línea. Node-RED provee un editor web (ver Figura 6) que facilita la conexión de flujos entre sí utilizado una amplia gama de nodos que se pueden implementar en tiempo de ejecución con tan solo un clic (Node-RED, s. f.).

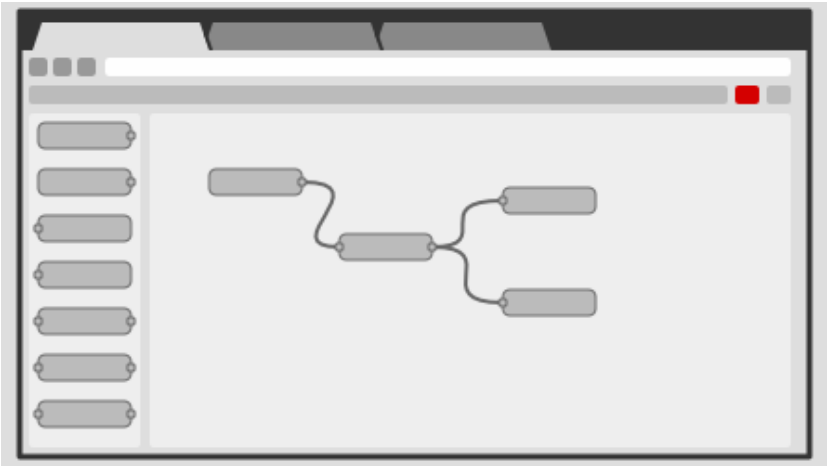


Figura 6. **Node-RED**
Fuente.(Node-RED, s. f., p.)

2.4. Medidor inteligente de agua

Es un medidor normal de agua vinculado a un dispositivo que permite la lectura y visualización electrónica continua del consumo de agua (Hauber-Davidson y Idris, 2006). Para Lloret et al.(2016), un sistema con medidores inteligentes permite que las empresas proveedoras de agua puedan leer y registrar los consumos en intervalos de tiempo, o tener al menos reportes diarios, un monitoreo constante o la facturación mensual.

La Figura 7 muestra una lista de características que hay que considerar en los medidores inteligentes con respecto al consumo de energía, medios de comunicación, datos, alarmas, costo y mantenimiento de estos.

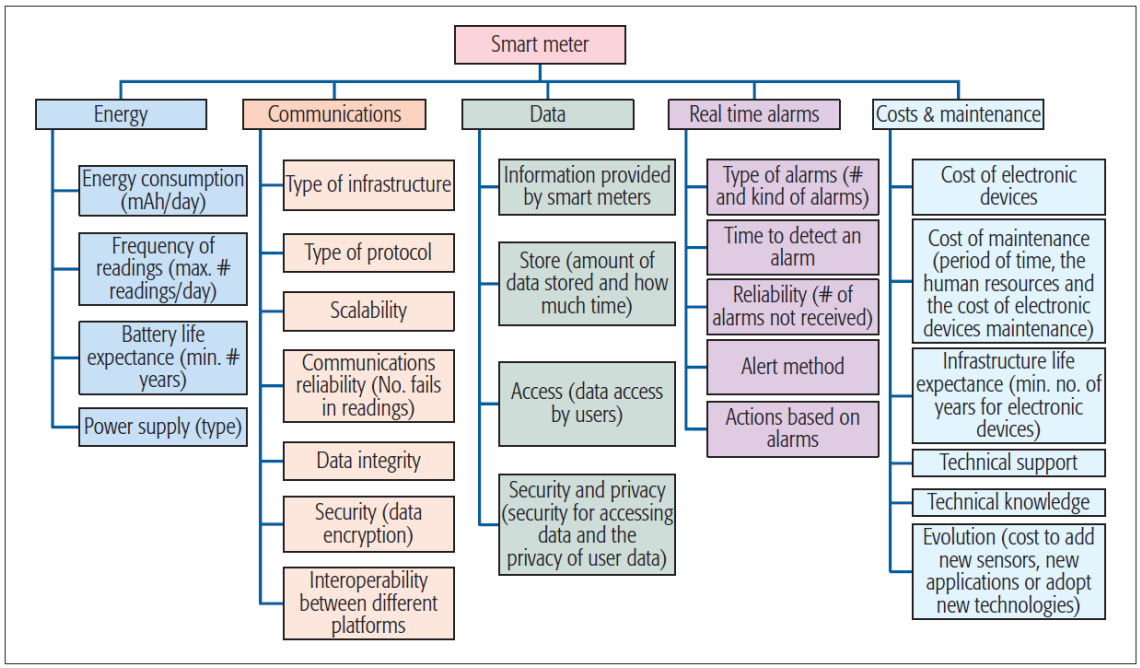


Figura 7. **Características de un medidor inteligente**
Fuente. (Lloret et al., 2016)

En IoT, existen varios proyectos que usan el sensor de pulsos de agua G3/4 (ver Figura 8) como una alternativa para medir el consumo de agua. Este está hecho con una válvula de plástico, un rotor de agua, y un sensor con efecto “hall”. Cuando el agua fluye a través del rotor, este rotor empieza a girar, y su velocidad cambia conforme el flujo de agua. El sensor con efecto “hall” es quien genera las señales de pulso para medir el consumo de agua (seedstudio, s. f.).

Figura 8. **Sensor de flujo de agua G3/4**
Fuente.(seedstudio, s. f.)

2.5. Cloud Computing

Microsoft (s. f.-a) lo define como la entrega de servicios computacionales, incluyendo servidores, almacenamiento, bases de datos, redes, software, análisis e inteligencia artificial, a través de la internet "la cloud" para ofrecer una innovación más rápida, recursos flexibles y economías de escala.

Para Microsoft (s. f.-a), los principales beneficios del Cloud computing son los siguientes:

- Ahorro en costo: Debido a que se elimina la compra de hardware, software y la configuración y ejecución de centro de datos.
- Escalabilidad: Se refiere a la capacidad de aumentar los recursos computacionales en cualquier momento de forma permanente o en momentos específicos.
- Seguridad: Muchos proveedores Cloud ofrecen un amplio set de políticas, tecnologías y controles que ayudan a proteger los datos, aplicaciones e infraestructura de potenciales amenazas.

- Velocidad: Esta característica permite que uno mismo pueda adquirir a demanda cualquier recurso computacional en pocos minutos haciendo tan solo unos cuantos clics.

Por otro lado, según Microsoft (s. f.-a), los tipos de servicios cloud son los siguientes:

- **Infrastructure as a service (IaaS):** Con este tipo de servicio se renta infraestructura IT como servidores, máquinas virtuales, sistemas operativos y se paga por su uso.
- **Platform as a service (PaaS):** Se refiere a los servicios cloud que son proveídos a demanda para el desarrollo, testeo, entrega y gestión de aplicaciones de software, sin tener que preocuparse por la administración o configuración de la infraestructura de los servidores, almacenamiento, la red y base de datos necesarios para el desarrollo.
- **Serverless computing:** Este servicio permite enfocarse en la construcción de aplicaciones funcionales sin perder tiempo en la gestión de los servidores e infraestructura requerida para esto. La arquitectura serverless es altamente escalable y dirigida por eventos, donde solo se usan los recursos cuando una función o desencadenador específico es llamada.
- **Software as a service (SaaS):** Consiste en brindar al usuario aplicaciones software a través del internet a demanda y es típicamente usado pagando una subscripción.

Es bien sabido que los dispositivos IoT generan una gran cantidad de datos constantemente, pero estos de manera aislada no son muy útiles, y es ahí donde el Cloud computing entra a tallar para proveer la infraestructura y recursos, de tal modo que estos datos sean procesados y analizados de manera eficiente para ofrecer un mayor beneficio al negocio.

Otro concepto que es inevitable tocar cuando se habla de Cloud computing e IoT es “Edge computing”. Este se refiere a una infraestructura de computación distribuida que acerca las aplicaciones a los orígenes de datos, como son los dispositivos de IoT o servidores locales (IBM, s. f.-b). Esto permite que los datos puedan ser analizados o procesados dentro del mismo dispositivo o un servidor local y, de esta manera, se reduce el tiempo de latencia, lo cual mejora el tiempo de respuesta y experiencia del usuario.

2.6. Métodos de detección de fuga

Existe una diversidad de métodos o reglas usadas para detectar fugas dentro de una red distribuida de agua, a continuación, se menciona las más usadas:

- **Minimum Night Flow (MNF):** Regla que asume que cualquier consumo de agua existente entre las 2:00 a. m. y las 4:00 a. m. son indicios de una posible fuga (Farah y Shahrour, 2017).
- **Continuous Non-Zero Water Consumption(CNZ):** Regla que dice que si no existe algún consumo cero dentro de un rango de 24 horas, es considerado como una posible fuga (Luciani et al., 2019).
- **K - Nearest Neighbors(K-NN):** Es un algoritmo básico de machine learning definido como un método robusto para identificar anomalías basado en la distancia con sus vecinos más cercanos. Este método puede ser adaptado para detectar anomalías en el consumo de agua al compararlo con los consumos históricos (Patabendige et al., 2018).

CAPÍTULO 3: ESTADO DEL ARTE

En el presente capítulo, se realiza una revisión sistemática de la literatura sobre los diferentes aportes en sistemas de medición inteligente del agua y algoritmos de detección de fugas. Para ello, se seleccionará artículos que responden las preguntan planteadas para la investigación.

3.1. Metodología de la investigación

Se realizó una revisión sistemática de la literatura considerando las pautas utilizadas por Kitchenham y Charters (2007), las cuales han sido adoptadas para desarrollar la tesis y esta implica las siguientes fases:

- **Planificación de la revisión:** En la cual se plantean las preguntas de investigación y el protocolo de revisión es definido.
- **Desarrollo de la revisión:** En la cual el plan se ejecuta y se seleccionan los estudios primarios de acuerdo con los criterios de inclusión y exclusión establecidos para el efecto.
- **Resultados de la revisión:** En la cual las estadísticas y el análisis de los documentos encontrados son presentados.

3.2. Planificación de la revisión

Para responder a la pregunta de investigación principal, es necesario formular las siguientes preguntas sobre sistemas de medición inteligentes de consumo de agua

- Q1: ¿Qué arquitecturas existen para un sistema de medición inteligente?
- Q2: ¿Qué funcionalidades tiene un sistema medición inteligente?
- Q3: ¿Qué tecnologías inalámbricas usa un dispositivo inteligente?
- Q4: ¿Qué medidas de seguridad tiene un sistema medición inteligente?
- Q5: ¿Qué algoritmos o reglas existen para detectar una fuga de agua?

Se ha realizado una revisión de la literatura usando el motor de búsqueda “Web of Science” debido a que facilita el acceso a la meta data de un conjunto de bases

de datos en las que aparecen artículos de revistas científicas que pertenecen a bases de datos de renombre internacional como ELSEVIER, IEE, SPRINGER y MDPI. La revisión cubre artículos escritos entre enero de 2015 y noviembre de 2019.

La cadena de búsqueda usada fue (“water meter" OR "smart meter" OR "smart water" OR "smart grid" OR "smart metering" OR "water meter" OR "water metering" OR "smart water metering" OR "water-energy metering" "smart energy" OR "measuring water" OR "water monitoring" OR "environmental monitoring" OR "real time water") OR ("anomalous water" OR "water consumption" OR "leakage detection" OR "leak detection") , además, se consideró de selección y exclusión que aparecen en la Tabla 1:

Tabla 1. **Criterios de selección y exclusión**

Criterio de selección	Criterio de exclusión
Periodo: desde enero de 2015 hasta noviembre de 2019	El idioma del artículo es diferente al inglés
Tipo de publicación: artículo de journal	Libros, publicaciones tipo conferencia y revistas que no tengan renombre internacional
Artículos que contestan directamente las preguntas de investigación	Artículos que mencionan medición inteligente de agua o métodos de detección de fuga, pero que no contribuyen a responder las preguntas planteadas
Artículos que presenten modelos, tecnologías, frameworks sobre medición inteligente, IoT y cloud computing y métodos de detección de fuga de agua	Artículos que no están dentro del área de computación

Fuente. Elaboración propia

3.3. Desarrollo de la revisión

Los posibles artículos identificados en el proceso de búsqueda, de acuerdo con la estrategia propuesta, fueron sometidos a un proceso de selección considerando los criterios de inclusión y exclusión establecidos. Muchos de los artículos fueron rechazados debido a que no correspondían al área de ingeniería de software o ciencia de la computación, pero se hizo algunas excepciones con el área de la ingeniería electrónica debido a que IoT es un tema relacionado a esa rama de la ingeniería. Cabe resaltar que hubo otros artículos que no fueron considerados porque a pesar de tener relación con el tema no llegaron a responder las preguntas elaboradas. El proceso aplicado es representado en la Figura 9.

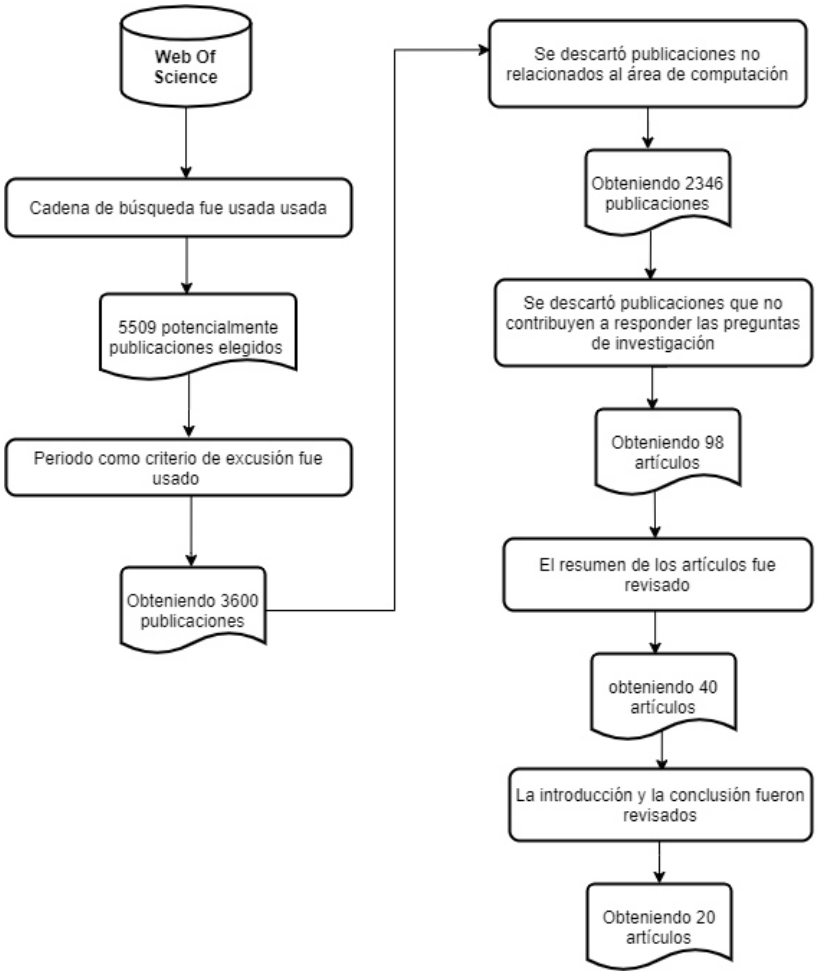


Figura 9. **Proceso de búsqueda de artículos**

Fuente. Elaboración propia

Los artículos tipo conferencia no fueron incluidos en nuestra búsqueda final, pero se realizaron al inicio 15 lecturas de estas con el objetivo de tener un panorama general del tema. Estos fueron encontrados usando el motor de búsqueda IEEE Xplore.

3.4. Resultados de la revisión

El desarrollo de la revisión sistemática dio lugar a 5509 publicaciones, y 20 de estos fueron analizados y seleccionados según los criterios de inclusión y exclusión para dar respuesta a las preguntas formuladas.

Además, en la Figura 10, se muestra que la mayoría de estos artículos seleccionados (10) han sido publicados durante el año 2018.

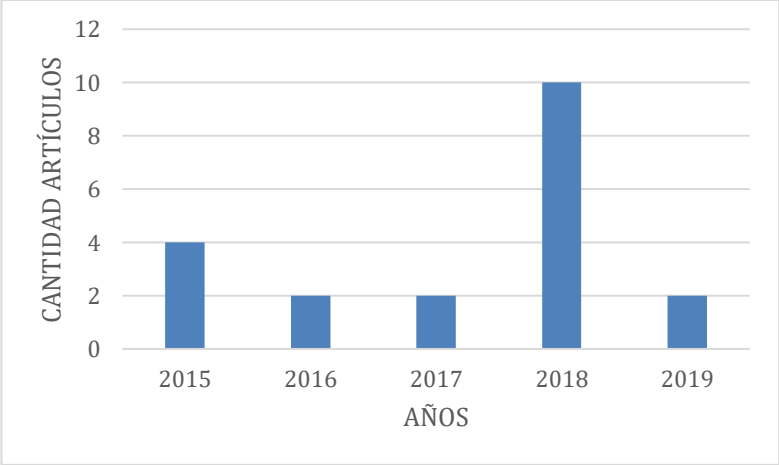


Figura 10. **Cantidad artículos escogidos por años de publicación**
Fuente. Elaboración propia

Por otro lado, en la Figura 11, es evidente que estos artículos seleccionados en su mayoría pertenecen a las bases de datos de ELSEVIER (6), SPRINGER (5) y la IEEE (3).

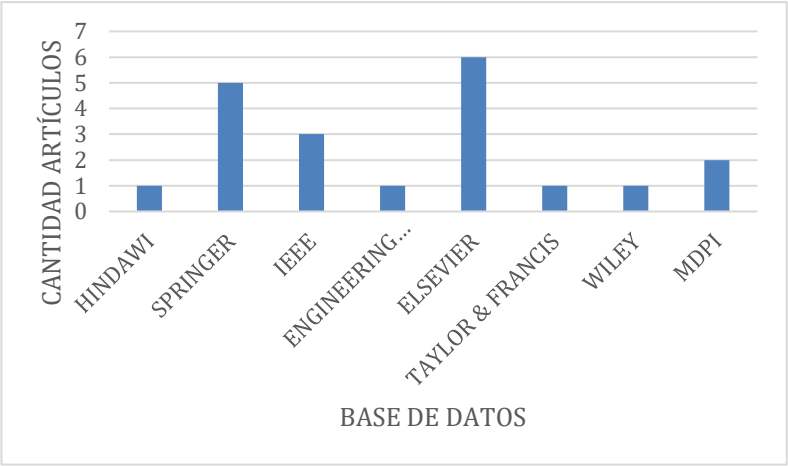


Figura 11. **Cantidad artículos escogidos por base de datos**
Fuente. Elaboración propia

Finalmente, en la Tabla 2, se muestra la lista de artículos seleccionados con sus respectivos títulos, base de datos, año de publicación y referencias.

Tabla 2. **Lista de artículos escogidos**

Título	Base Datos	Año	Referencias
A Review of the Topologies Used in Smart Water Meter Networks: A Wireless Sensor Network Application	HINDAWI	2016	(Marais et al., 2016)
A Smart Meter and Smart House Integrated to an IdM and Key-based Scheme for Providing Integral Security for a Smart Grid ICT	SPRINGER	2018	(Abreu et al., 2018)
A survey of smart water quality monitoring system	SPRINGER	2015	(Dong et al., 2015)

An Integrated IoT Architecture for Smart Metering	IEEE	2016	(Lloret et al., 2016)
An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments	IEEE	2015	(Pan et al., 2015)
An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service	ENGINEERING TECHNOLOGY & APPLIED SCIENCE RESEARCH	2018	(Zafar et al., 2018)
Big Data Mining of Users' Energy Consumption Patterns in the Wireless Smart Grid	IEEE	2018	(Zhu et al., 2018)
Detection and interpretation of anomalous water use for non-residential customers	ELSEVIER	2018	(Patabendige et al., 2018)
Digital metering feedback and changes in water consumption - A review	ELSEVIER	2018	(Liu y Mukheibir, 2018)
Field application of waterworks automated meter reading systems and analysis of household water consumption	TAYLOR & FRANCIS	2015	(Joo et al., 2015)
Integrated intelligent water-energy metering systems and informatics:	ELSEVIER	2018	(Stewart et al., 2018)

Visioning a digital multi-utility service provider			
Leakage Detection Using Smart Water System: Combination of Water Balance and Automated Minimum Night Flow	SPRINGER	2017	(Farah y Shahrour, 2017)
Measuring water use, conservation, and differences by gender using an inexpensive, high frequency metering system	ELSEVIER	2017	(Horsburgh et al., 2017)
Real-time water quality monitoring using Internet of Things in SCADA	SPRINGER	2018	(Saravanan et al., 2018)
Remote water quality monitoring in wide area	ELSEVIER	2015	(Chung y Yoo, 2015)
Smart Water Meters and Data Analytics Decrease Wasted Water Due to Leaks	WILEY	2018	(Schultz et al., 2018)
Smart water technology for leakage detection: feedback of large-scale experimentation	SPRINGER	2018	(Farah y Shahrour, 2018)
Water quality monitoring in smart city: A pilot project	ELSEVIER	2018	(Chen y Han, 2018)
Wireless Middleware Solutions for Smart Water Metering	MDPI	2019	(Alvisi et al., 2019)

Green Smart Technology for Water (GST4Water): Water Loss Identification at User Level by Using Smart Metering Systems	MDPI	2019	(Luciani et al., 2019)

Fuente. Elaboración propia

3.5. Análisis

3.5.1. Q1: ¿Qué arquitecturas existen para un sistema de medición inteligente?

Las soluciones tecnológicas, para la medición del consumo de agua, se soportan sobre una arquitectura de IoT, esta se refiere al diseño de las capas del sistema que permitirán la comunicación entre dispositivos inteligentes, juntamente con el análisis y toma de decisiones basado en los datos recolectados de estos (Lloret et al., 2016), por ello se revisa algunos de estas arquitecturas. En un estudio conducido por Lloret et al.(2016), se propuso una arquitectura integrada IoT de tres capas (ver Figura 12) e incluye una revisión de las principales características que tienen los medidores inteligentes y los existentes protocolos de comunicación para la medición inteligente de electricidad, agua y gas entre diferentes sistemas para una ciudad inteligente. En un artículo de Horsburgh et al.(2017) se propuso una arquitectura IoT de código abierto llamada piggy-back system, que incluye procesamiento local y hardware de bajo costo, para medir y registrar el consumo de agua en una universidad. Del mismo modo, Zafar et al.(2018) presentó una arquitectura IoT de bajo costo que, además de su simplicidad, permitía el monitoreo en tiempo real de la temperatura y humedad del medio ambiente. A diferencia de los anteriores trabajos que considera un dispositivo IoT, Stewart et al.(2018) presentó una arquitectura que contempla medidores de agua, electricidad y gas, con el cual a través de un sistema de información muestra el comportamiento multi-dimensional del usuario. En relación de protocolos de comunicación existentes, Alvisi et al.(2019) presentó una arquitectura

(SmartWater Metering Middleware - SWaMM) que incluye una capa adicional (Edge Gateway) que permitía la interacción entre ellos, con lo cual el usuario podía seleccionar entre la variedad de medidores inteligentes sin tener que sentirse amarrado a un proveedor (vendor lock-in), esto se puede ver en la Figura 13.

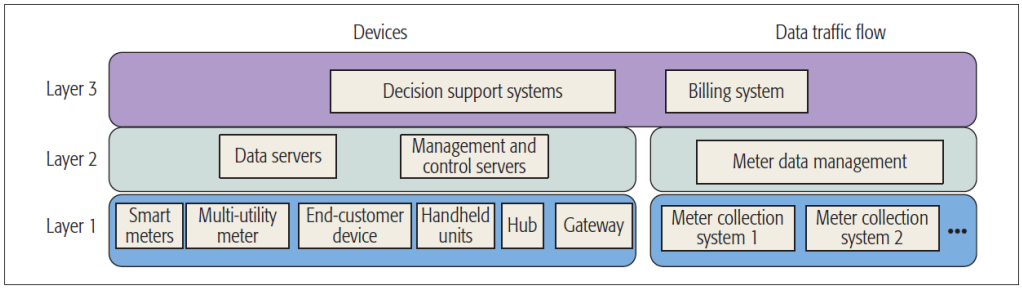


Figura 12. **Arquitectura Integrada IoT**

Fuente. (Lloret et al., 2016)

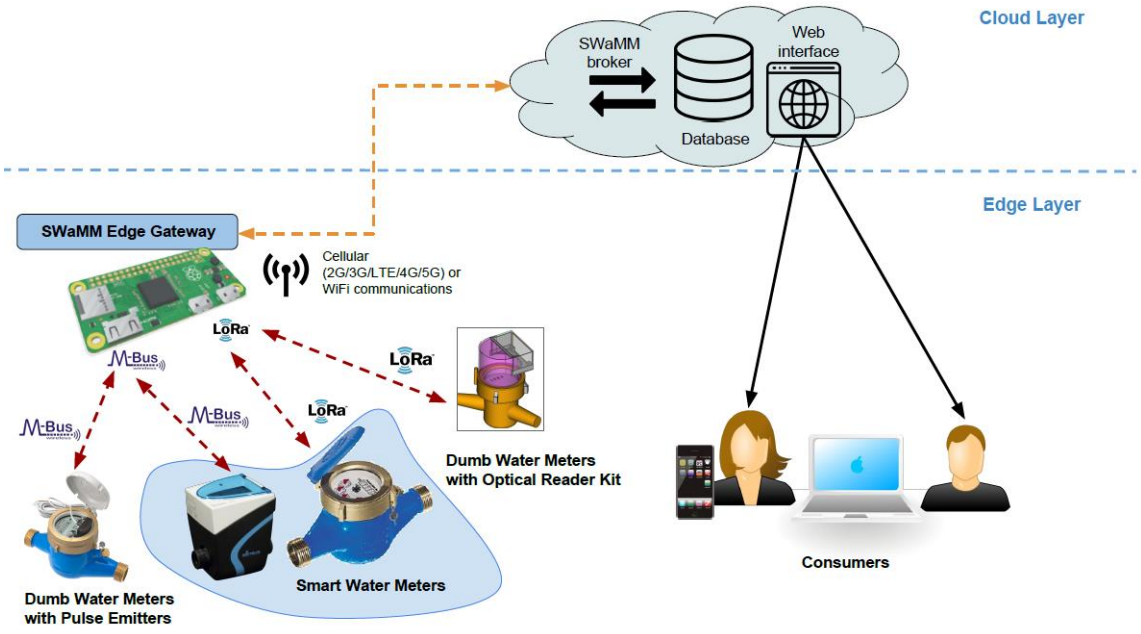


Figura 13. **Arquitectura SWaMM**

Fuente. (Alvisi et al., 2019)

Algunos trabajos sobre arquitecturas de IoT para la medición inteligente del agua se ha orientado a la calidad. Dong et al.(2015) exploró tres principales subsistemas para los sistemas inteligentes de monitoreo de la calidad del agua, las cuales son nombradas data collection subsystem, data transmission subsystem, y data management subsystem, esto se puede ver en la Figura 14. Saravanan et al.(2018) propuso un sistema SCADA que usa IoT para realizar un monitoreo en tiempo real, donde se mide la temperatura, color, flujo, PH y presión. Finalmente, Chen y Han (2018) mostró la factibilidad de recolectar datos en tiempo real con frecuencias altas de envíos y mostrarlos al instante en línea dentro de una ciudad inteligente, para ello construyen un sistema de monitoreo del agua basándose en la plataforma “Bristol Is Open”, la cual se puede ver en la Figura 15, y concluyen que su arquitectura es fácilmente escalable para una red mayor de sensores.

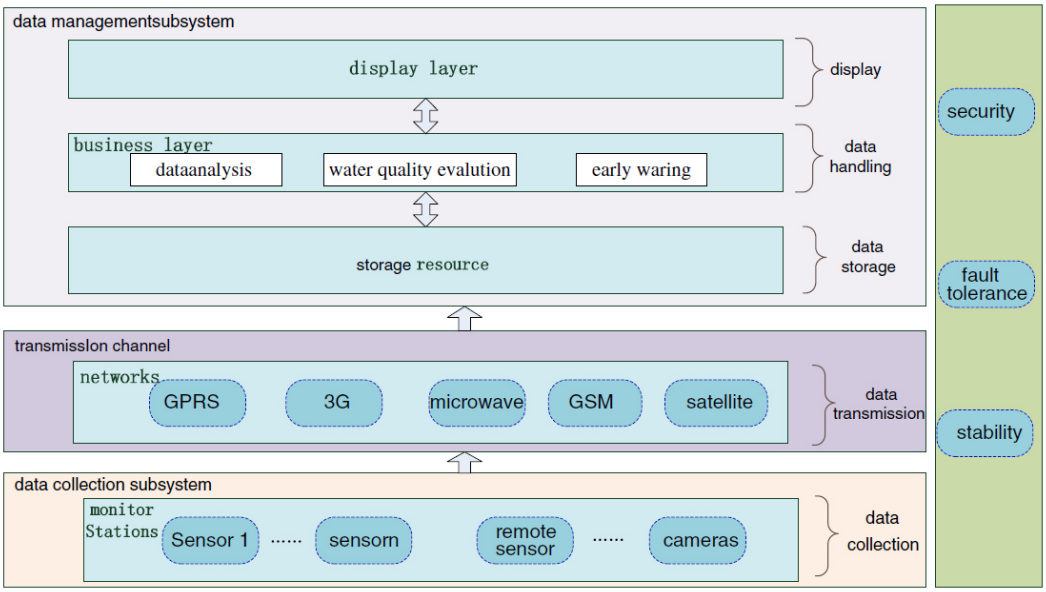


Figura 14. **Arquitectura Smart Water Quality Monitoring System (SWQMS)**
Fuente. (Dong et al., 2015)

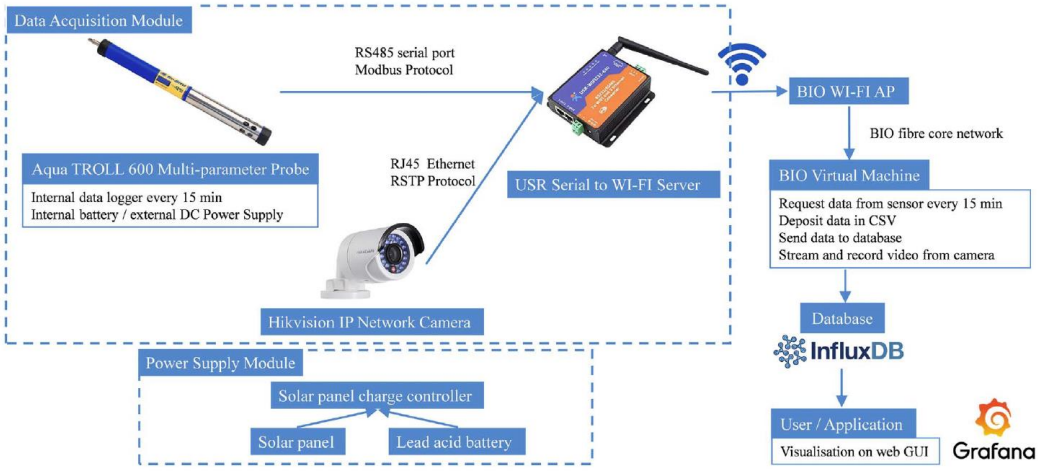


Figura 15. **Arquitectura Wireless Water Quality Monitoring System (WWQMS)**

Fuente. (Chen y Han, 2018)

En la *Tabla 3*, se muestra un resumen de las arquitecturas existentes para un sistema de medición inteligente.

Tabla 3. **Arquitecturas para un sistema de medición inteligente**

Arquitectura	Descripción	Referencia
Integrada IoT	Tiene 3 capas que involucran protocolos de comunicación, formato de la data y su obtención, procesamiento de esta y un sistema de decisiones basado en big data	(Lloret et al., 2016)
Piggy-back system	Tiene 3 capas: captura de datos usando medidores analógicos integrados con sensores de efecto hall y sensores de movimiento. Canal de comunicación usando el protocolo Wi-Fi. Procesamiento y almacenamiento de	(Horsburgh et al., 2017)

	datos usando la computadora RaspberryPi	
IoT de bajo costo	Tiene 3 capas: para adquisición de datos se usó una placa Arduino UNO, el sensor DHT11, un módulo de Wi-Fi ESP8266, para el procesamiento y almacenamientos se usó servicio cloud de IoT ThingSpeak, para visualización se usó una aplicación para Android	(Zafar et al., 2018)
Digital multi-utility	Tiene 3 capas: medidores inteligentes, red de comunicación 5G, servidores de BD donde se realiza el análisis y procesamiento de datasets combinados (agua-luz, agua-gas, etc) que muestran, al final, reportes amigables e interfaz de usuario	(Stewart et al., 2018)
SWaMM	Presenta una capa llamada “Edge Gateway”, que se encuentra cerca a los dispositivos de medición, y es capaz de crear una interfaz tal que pueda comunicarse con diferentes tipos de medidores, protocolos inalámbricos de comunicación y formato de datos	(Alvisi et al., 2019)
SWQMS	Tiene 3 capas: “Data collection”, “Data transmission subsystem”, y “Data management subsystem”	(Dong et al., 2015)
SCADA IoT	Tiene 3 capas: sensores, módulo de comunicación, y un servidor centralizado	(Saravanan et al., 2018)

WWQMS	Tiene 5 módulos: módulo de adquisición de datos (sensores y cámaras), módulo de transmisión de datos (Wi-Fi), módulo de suministro de energía (panel solar), módulo de almacenamiento de los datos (máquinas virtuales en la nube) y módulo de redistribución (interfaz gráfica) de datos	(Chen y Han, 2018)

Fuente. Elaboración propia

3.5.2. Q2: ¿Qué funcionalidades tiene un sistema medición inteligente?

Los autores Liu y Mukheibir (2018) hacen una revisión de varias literaturas con el objetivo de conocer qué funcionalidades de un sistema inteligente de medición del agua influyen en el usuario para lograr un ahorro de esta. Ellos mencionan que varios autores concuerdan en agregar al sistema la funcionalidad de alerta de fugas, información histórica del consumo de agua para ser comparada, un juego simple donde los usuarios compitan en quién ahorra más, tips de cómo una persona puede ahorrar agua, y recibir una retroalimentación de manera frecuente. En su estudio, muestra que en algunas implementaciones se ha logrado ahorrar entre el 2.8 % y 10 % del agua.

Por otro lado, Pan et al. (2015) se dan cuenta que los edificios ecológicos no tienen políticas eficientes para automatizar el ahorro de energía, para mejorar esta situación, se propone un framework basado en la geolocalización del usuario, la cual se muestra en la Figura 16. Con la información de la ubicación de la persona es posible realizar de manera automática, y en tiempo real, el apagado o encendido de los dispositivos dentro de una casa y oficina con el objetivo de ahorrar la energía. Como resultado del experimento, se muestra un

ahorro en casi 50 % de energía sin la intervención de la persona, quien debe tener un sistema de medición inteligente.

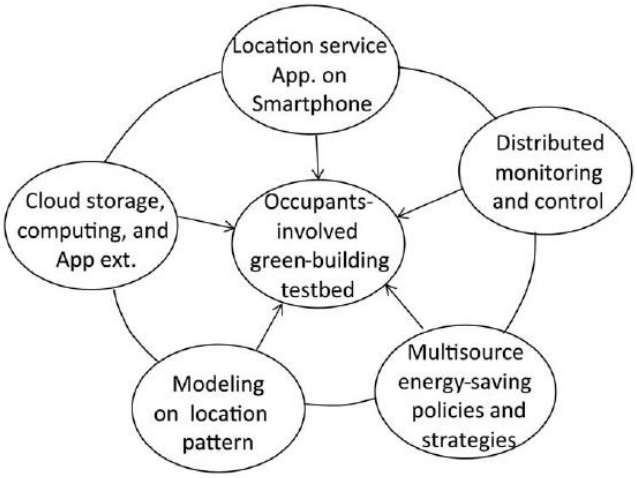


Figura 16. **Marco de trabajo basado en la geo-localización inteligente**
Fuente. (Pan et al., 2015)

En la *Tabla 4* se muestra un resumen de las funcionalidades que debe tener un sistema de medición inteligente.

Tabla 4. **Funcionalidades en sistemas de medición inteligente**

Funcionalidades	Descripción	Referencia
Alerta de fugas	Es muy importante enviar las alertas vía email, sms y llamadas	(Liu y Mukheibir, 2018)
Comparación de uso de agua	Comparar consumo de agua con consumos anteriores, promedios de todos los residentes o vecinos similares	(Liu y Mukheibir, 2018)

Gamification	Juegos donde usuarios compiten en quién consume menos agua	(Liu y Mukheibir, 2018)
Tips de ahorro de agua	Mostrar publicidad de cómo podría ahorrar al máximo el agua	(Liu y Mukheibir, 2018)
Retroalimentación frecuente	Mostrar cada cierto tiempo el consumo realizado	(Liu y Mukheibir, 2018)
Geo-localización inteligente	Realizar de manera automática el encendido o apagado de los dispositivos basado en la geolocalización de la persona	(Pan et al., 2015)

Fuente. Elaboración propia

3.5.3. Q3: ¿Qué tecnologías inalámbricas usa un dispositivo inteligente?

Las soluciones tecnológicas basadas en IoT requieren de tecnologías de comunicación inalámbricas, a través del cual los dispositivos puedan recibir y enviar datos de manera efectiva (Marais et al., 2016), por ello es necesario su revisión. En la investigación de Joo et al.(2015), se realizaron varias pruebas de intercomunicación entre dispositivos IoT, donde se determinó que usando UHF e internet (TCP/IP) la señal es más estable a comparación con UHF, DCU y Wibro, además, sus resultados demuestran que el promedio de la recepción de los paquetes fue de 94.1 %. Por otro lado, para monitorear y recolectar información (pH, temperatura, etc.) del caudal de un río en un área extensa, en un estudio dirigido por Chung y Yoo (2015), se propone usar una red de sensores inalámbricos (WSN) de bajo costo cuyos resultados muestran que la pérdida de datos está por debajo del 1 % y el tráfico de la red se reduce a 1/5. Además, Marais et al.(2016) propuso usar la tecnología ZigBee con la topología en malla para construir una red extensa de dispositivos intercomunicados que permitan

una máxima efectividad al recibir y enviar datos y, con base en sus simulaciones, la pérdida de los datos resultó menor al 0.14 %. En el trabajo de Chen y Han (2018), se construyó en una ciudad un sistema de monitoreo del agua basándose en la plataforma “Bristol Is Open”, así mismo, se hizo uso de Wi-Fi debido a su largo alcance (hasta 100m) y trasmisión de paquete de hasta 7Gbps.

En *Tabla 5*, se muestra un resumen de las tecnologías inalámbricas que usan los dispositivos inteligentes.

Tabla 5. **Tecnologías inalámbricas**

Tecnologías inalámbricas	**Descripción**	**Referencia**
Radio frecuencia UHF	424 MHz, Mejor rendimiento cuando la señal tiene que atravesar pared (en comparación a Zigbee y Bluetooth)	(Joo et al., 2015)
Radio frecuencia en una red WSN	Red de sensores a bajo costo, fácil de implantar, bajo consumo de energía. Se recomienda usar el microcontrolador MSP430F1611. Para reducir la pérdida de datos, se recomienda usar el algoritmo PINCO de compresión, algoritmo de promedio de datos y algoritmo “flooding” de ruteo de datos	(Chung y Yoo, 2015)
ZigBee	Bajo costo y consume poca batería. Alcance corto (10-20m). Ancho de banda bajo (20kbps-250kbps)	(Marais et al., 2016)

		(Chen y Han, 2018)
Wi-Fi	Alcance Medio (30-100m). Ancho banda alto (100-300Mbps, hasta 7Gbps). Consumo de energía alto	(Chen y Han, 2018)
LoRaWAN	Alcance alto (10-15km). Ancho de banda muy bajo (0.3kbps-50kps). Bajo consumo de energía	(Chen y Han, 2018)

Fuente. Elaboración propia

3.5.4. Q4: ¿Qué medidas de seguridad tiene un sistema medición inteligente?

Según Abreu et al.(2018), existen pocos estudios sobre seguridad física y lógica en dispositivos inteligentes, por ello proponen un sistema de detección de anti - manipulación (anti - tampering) de dispositivos basado en el uso de sensores, y de un mecanismo multinivel de privilegios para acceder a estos. Por otro lado, Zhu et al.(2018) afirma que una comunicación inalámbrica siempre está propensa a recibir ciber ataques, motivo por el cual proponen ciertas tácticas de seguridad en la trasmisión, por ejemplo, usar encriptación, de este modo, concluyen que existe un gran reto en buscar un balance entre la seguridad y el rendimiento del sistema.

En la *Tabla 6*, se muestra un resumen de las medidas de seguridad que debe tener un sistema de medición inteligente.

Tabla 6. **Medidas de seguridad en sistemas de medición inteligente**

Seguridad	Descripción	Referencia
Anti-Tampering	Proveer una protección física a los medidores inteligentes con base en sensores y acelerómetros que, al detectar alguna manipulación, generan las alertas	(Abreu et al., 2018)
Encriptación de datos	Usar esquemas de encriptación de los datos	(Zhu et al., 2018)

Fuente. Elaboración propia

3.5.5. Q5: ¿Qué algoritmos o reglas existen para detectar una fuga de agua?

Farah y Shahrour (2017) dirigieron un estudio donde se implementó un sistema de medición inteligente para detectar posibles fugas en un campus universitario, y se propuso combinar las reglas de Water Balance con Minimun Nigh Flow (MNF), que dio como resultado una precisión de 97 %, así, se consiguió reducir el desperdicio de agua por fuga. En un artículo de Schultz et al.(2018), se implementó, en una ciudad de California, un portal donde los residentes podían monitorear su consumo de agua, de este modo, para la detección de fugas, se propuso establecer, con base en un promedio (AVG) un límite (7.5 gph) de agua continua consumida durante un periodo de 24 h, así mismo, sus resultados mostraron que los usuarios que usaron el sistema llegaron a reducir en 50 % las probabilidades de que vuelvan a tener una fuga, pero resalta que su metodología no detecta fugas menores al límite establecido. Por otro lado, Farah y Shahrour (2018) demostró que usando un sistema de medición automática (AMR) para el monitoreo del agua consumida se puede detectar rápidamente alguna fuga de agua, para ello usaron una función de densidad de la probabilidad con el fin de identificar regiones de mayores y menores probabilidades de fuga con base en

los datos históricos de consumo de agua durante días de trabajo, fines de semana y feriados, de esta manera, como resultado pudieron detectar 3 fugas en the Scientific Campus of the University of Lilledurante durante el 2015. Además, Patabendige et al.(2018) observó que la mayoría de sistemas comerciales de consumo de agua solo proveían estadísticas básicas, sin embargo, no detectaban patrones complejos de comportamiento de un consumo anormal del agua, por ello propuso el uso del algoritmo K - Nearest Neighbors (K-NN) para calcular la puntuación de anomalía por cada día, y los resultados mostraron que durante un año detectaron 31 días de fugas, donde se logró alcanzar una precisión del 74 %. Finalmente, con el objetivo de que la gente evite desperdiciar el agua debido a fugas, Luciani et al.(2019) propuso un algoritmo que detecta fugas haciendo uso de las reglas MNF y CNZ, con el cual alcanzó 98 % de precisión.

En *Tabla 7*, se muestra un resumen de las reglas que existen para una predicción de fuga de agua.

Tabla 7. **Reglas de predicción de fuga de agua**

Reglas	Descripción	Referencia
Minimun Night Flow (MNF)	Medición del flujo de agua durante las 2:00 a.m. y 4:00 a.m., donde se determina un límite promedio usando los modelos estadísticos promedio móvil y la desviación estándar móvil, además, hace uso de la matriz de confusión para calibrar su modelo y aumentar la precisión	(Farah y Shahrour, 2017) (Farah y Shahrour, 2018) (Luciani et al., 2019)
Water Balance	Determina el volumen de agua perdida en un área determinada durante un periodo de tiempo, teniendo en cuenta el	(Farah y Shahrour, 2017)

	volumen de agua que marca un medidor general y las medidas de los medidores secundarios	
Definir límite con base en un promedio (AVG)	Establecer un límite de agua continúa consumida por hora durante un periodo de 24h con posible fuga. Este límite es hallado como un promedio de todas las casas que tuvieron alguna fuga durante un mes. El método no es bueno para detección de fugas menores	(Schultz et al., 2018)
Función de densidad de la probabilidad (FDP)	Probabilidad de que una medición aleatoria alcance determinado valor dentro de un intervalo	(Farah y Shahrour, 2018)
Algoritmo de detección de anomalías (machine learning)	Este algoritmo calcula una puntuación de anomalía por cada día usando el algoritmo K-NN. El puntaje de cada día se basa en 10 características de la demanda diaria y su contexto histórico.	(Patabendige et al., 2018)
Continuous Non-Zero Water Consumption (CNZ)	Regla que dice que, si no existe algún consumo cero dentro de un rango de 24 horas, es considerado como una posible fuga	(Luciani et al., 2019)

Fuente. Elaboración propia

CAPÍTULO 4: SISTEMA PROPUESTO DE MEDICIÓN INTELIGENTE DEL CONSUMO DE AGUA

En el presente capítulo, se propone una arquitectura para un sistema de medición inteligente con un algoritmo de detección de fugas. En primer lugar, se hace una descripción general del aporte, sus componentes juntamente con su funcionamiento y cómo resuelve el problema existente. Finalmente, se detalla las entradas, proceso y salidas de cada componente del aporte.

4.1. Diagrama en alto nivel

El sistema de medición inteligente se basa en el desarrollo de una arquitectura para IoT que cubra 5 aspectos importantes. En primer lugar, la captura del consumo de agua, el cual por seguridad debe contar con un mecanismo de encriptación de los datos (Zhu et al., 2018), después, el preprocesamiento local de los consumos recibidos, en tercer lugar, la seguridad física de los dispositivos electrónicos, luego, el almacenamiento y visualización los consumos de agua obtenidos, por último, el análisis de los consumos a través del algoritmo de detección de fugas.

En la Figura 17, se muestra los cinco componentes principales del sistema, los cuales permiten la recolección, almacenamiento, análisis y visualización del consumo de agua. En el componente “Adquisición de datos”, en cada periodo de tiempo t_1 (puede ser 2 minutos), se obtiene el valor del consumo de agua a través de un medidor inteligente, el cual es enviado al componente “Edge Gateway” para su almacenamiento. Dentro de este componente, existe un mecanismo de seguridad “Anti - manipulación” instalado que alerta al usuario y/o administrador en caso de que exista una manipulación del dispositivo. Luego, en cada periodo de tiempo t_2 (t_2>t_1, puede ser 1 hora), se envía el consumo acumulado al servidor “Cloud” para que este valor sea almacenado juntamente con la ubicación del usuario, la cual es obtenida a través del GPS del celular, y ambos sean analizados por el “Algoritmo de detección de fuga”, el cual alerta al usuario en

caso de existir una posible fuga de agua. También, dentro de la “Cloud” se encuentra un portal web que permite al usuario visualizar, en tiempo real, el histórico de sus consumos de agua.

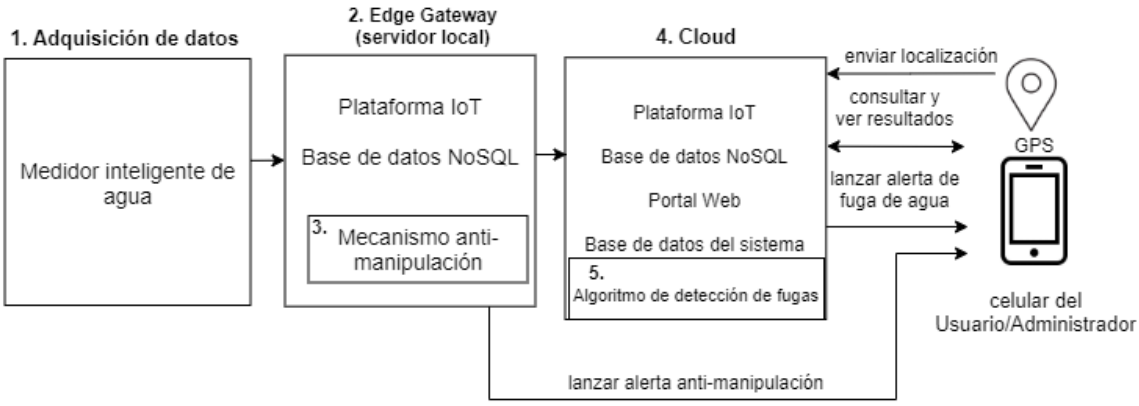


Figura 17. **Diagrama a alto nivel del sistema de medición inteligente del consumo de agua propuesto**

Fuente. Elaboración propia

4.2. Componentes

4.2.1. Adquisición de datos

A través de este componente, en cada tiempo t_1 se hace la captura del consumo de agua, el cual es enviado al servidor local (Edge Gateway) de manera digital para su almacenamiento y procesamiento.

4.2.2. Edge Gateway

Este componente recibe los datos obtenidos del componente “Adquisición de datos”, los cuales son almacenados y procesados para, posteriormente, ser enviados en una sola trama el acumulado en el tiempo t_2 al componente “Cloud”.

4.2.3. Mecanismo de anti – manipulación (anti-tampering)

En este componente, se busca garantizar que, ante cualquier manipulación física del componente “Edge Gateway”, se emita una alerta al administrador con el fin de garantizar el correcto funcionamiento de este. Abreu et al. (2018) considera

que la protección física es un requisito de casi todos los dispositivos IoT a los que cualquier persona puede acceder físicamente.

4.2.4. Cloud

Este componente recibe los datos obtenidos del componente “Edge Gateway” y la ubicación de usuario a través del GPS del celular, que es enviado cada tiempo t_3 ($t_1 < t_3 < t_2$), para que conjuntamente sean almacenados, analizados a través de un algoritmo de detección de fugas, y visualizados en una plataforma Cloud.

4.2.5. Algoritmo de detección de fugas

El algoritmo que se muestra en la Figura 18 detecta la posible existencia de una fuga de agua considerando cuatro escenarios, para ello toma los parámetros de entrada: deviceId, userId, tiempo t_2, el consumo en t_2 , y la ubicación, este último fue usado en el trabajo de Pan et al. (2015) para reducir el consumo en casas inteligente. Cada iteración del algoritmo se ejecuta al finalizar t_2, y en ella se verifica si el consumo de agua encaja con alguno de los cuatro escenarios de fuga: “tendencia negativa” (A), “consumo de las últimas 24 horas” (B), “consumos similares” (C), y “altos consumos anómalos” (D). Esta verificación de escenarios se realiza de forma secuencial A, B, C, D, y, de verificarse al menos una de ellas, se envía una alerta de detección de fuga al usuario y/o al administrador, quienes podrán confirmar o rechazar la alerta a través del aplicativo móvil, y con ello mejora la precisión del algoritmo en una siguiente iteración. A continuación, se explica cada escenario.

- **El escenario A** verifica si el consumo recibido tiene un valor negativo o el consumo total acumulado en las últimas 24 horas tiene una tendencia negativa, esto podría darse debido a fallas en los medidores inteligentes al momento de capturar el consumo (Alvisi et al., 2019).
- **El escenario B** verifica si existe un flujo continuo de consumo de agua en las últimas 24 horas, dado que en ninguna hora hubo un consumo cero, lo cual es muy improbable para un consumo normal, a esta regla se le conoce como CNZ.

- **El escenario C** verifica si el consumo recibido coincide con los dos últimos consumos registrados, ya que es muy poco probable que de manera consecutiva existan consumos muy similares, a esta regla se le denomina C3S (tres consumos similares seguidos).
- **El escenario D** verifica si hay un consumo alto fuera de su comportamiento histórico. Para ello, primero, se obtiene todos los consumos históricos que se asemejen a este, tanto en el trimestre como en el día de la semana, y posteriormente, se descarta todos los que hayan sido marcados como anómalos. Después, estos consumos son indexados por días, y de cada día se obtiene las siguientes cuatro características: consumo medio, consumo mínimo, consumo máximo, y la media del rango de hora a la cual pertenece el consumo que está siendo evaluado, este rango puede estar entre las 0 a 6 h, 6 h a 12 h, 12 h a 18 h y de 18 h a 24 h. A continuación, por cada una de las cuatro características, se aplica el algoritmo K-NN con el objetivo de obtener una lista de los consumos que más se acercan (K=4) al consumo de entrada. Seguidamente, se usa el teorema de Tchebysheff (Barnes, 1994) para construir un intervalo de confianza, la cual garantiza que al menos el 75 % de la lista de consumos obtenidos anteriormente se encuentran dentro de 2 desviaciones estándar de la media, si el valor del consumo recibido se encuentra fuera de este rango, se considera una “posible fuga” de agua. Finalmente, si hay “posible fuga” y la persona no se encuentra en su casa, el consumo se considera alto, ya que está fuera de su comportamiento histórico. A toda esta secuencia de pasos se ha denotado como la regla CHA (consumo anómalo histórico).

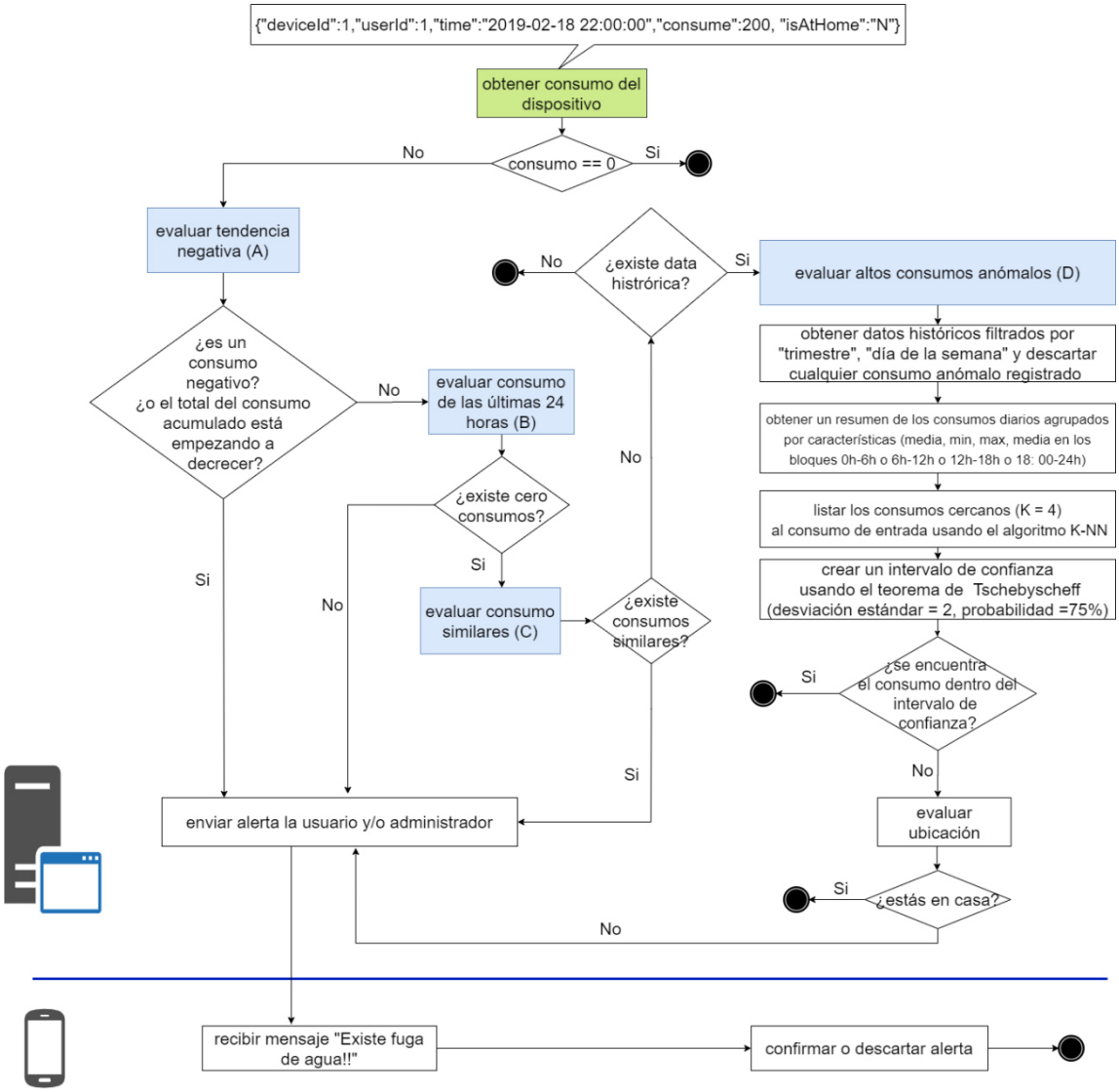


Figura 18. **Algoritmo de detección de fugas**

Fuente. Elaboración propia

CAPÍTULO 5: IMPLEMENTACIÓN DEL SOFTWARE

En este capítulo, se presenta la arquitectura e implementación del software realizado para el sistema de medición inteligente de consumo de agua propuesto. En primer lugar, se hace una descripción de la arquitectura del sistema. Luego, se detalla los requisitos funcionales y no funcionales del software. Después, se menciona las iteraciones de la metodología SCRUM que fueron hechas durante el desarrollo. A continuación, se habla de las limitaciones de hardware y software que tiene el sistema. Seguidamente, se muestra las pruebas a la cual fue sometida el software. Finalmente, se concluye con una descripción general de su funcionamiento.

5.1. Arquitectura del software

En Figura 19 y Figura 20, se aprecia la vista física y tecnológica, respectivamente, de la arquitectura propuesta que conecta los cinco componentes ya mencionados. En la vista física, se muestra en alto nivel los dispositivos físicos que se usan en cada componente juntamente con los servicios principales que se encuentran instalados en ellos. Y en la vista tecnológica, se detalla el nombre del software, lenguaje de programación, base de datos, plataforma, protocolos y sistema operativo utilizado en cada componente.

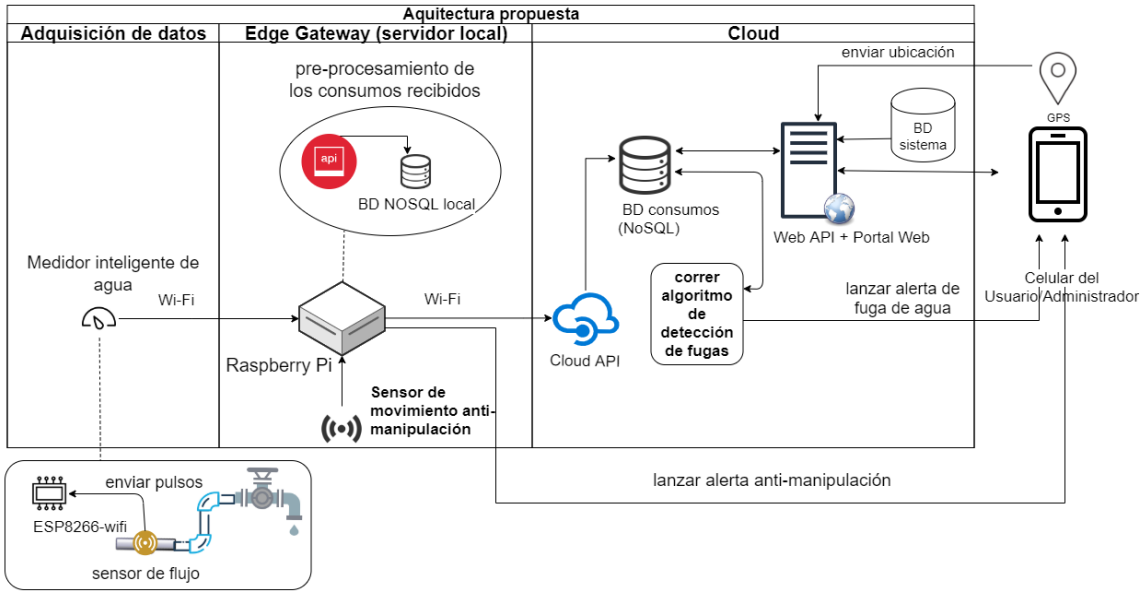


Figura 19. **Vista física de la medición inteligente del consumo de agua**

Fuente. Elaboración propia

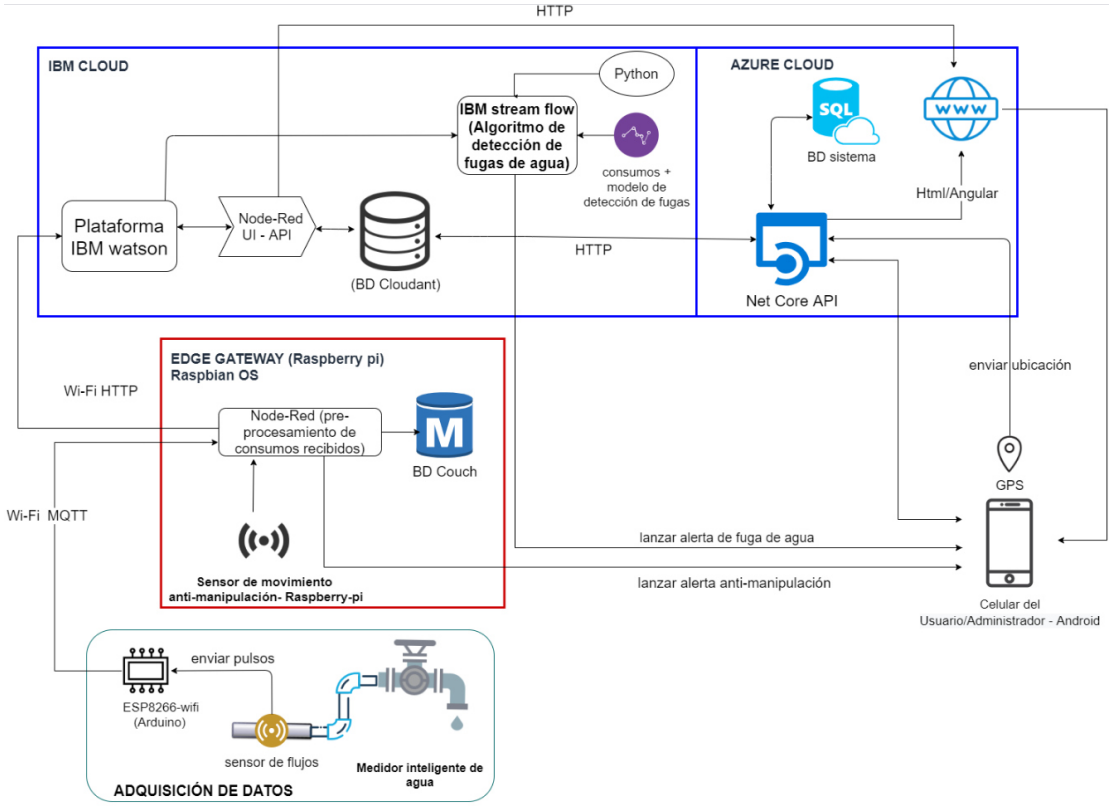


Figura 20. **Vista tecnológica de la medición inteligente del consumo de agua**

Fuente. Elaboración propia

En el componente Adquisición de datos, el consumo se obtiene a través de un sensor de pulsos de agua (seedstudio, s. f.), donde aproximadamente por cada 367 pulsos equivalen al pase de 1 litro de agua. Luego, el módulo NodeMCU ESP8266 (Handson Technology, s. f.) se encarga de transformar estos pulsos a valores digitales con formato JSON, los cuales son enviados al Gateway vía Wi-Fi gracias al uso de un protocolo ligero llamado MQTT. La Figura 21 muestra el medidor inteligente de agua usado.

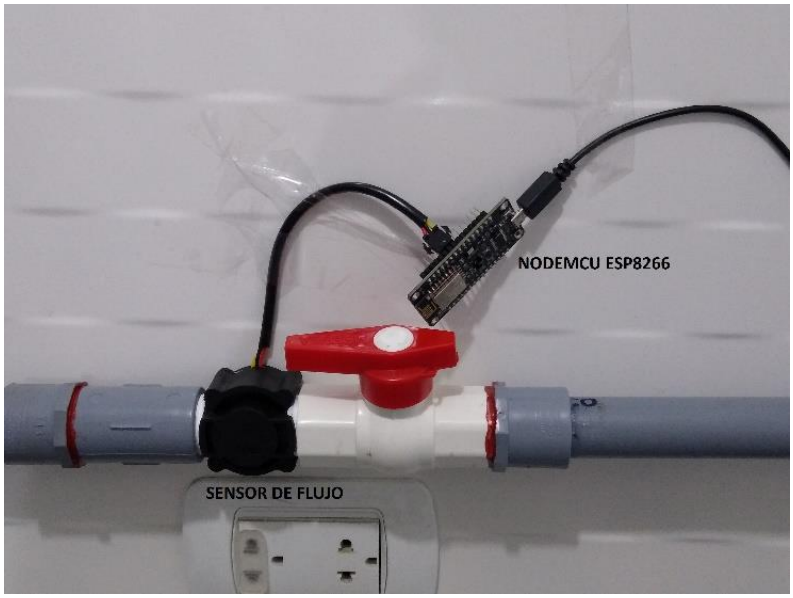


Figura 21. **Medidor inteligente de agua**
Fuente. Elaboración propia

Luego, el *Edge Gateway* o servidor local está montado sobre un pequeño computador con conexión a Wi-Fi de bajo costo llamado “Raspberry Pi” (RaspberryPi, s. f.), el cual es apreciado en la Figura 22. Por otro lado, el procesamiento se hace usando el software “Node-RED”, el cual, por medio de una interfaz de diagrama de flujos, va agregando lógica que permite la transformación y almacenamiento de la data en una base de datos NoSQL llamada “Couch”, tal como se muestra en la Figura 23, la cual contiene un proceso que es ejecutado cada tiempo t_2, y que se encarga de obtener el consumo

acumulado dentro de ese periodo y enviarlo al componente “Cloud” para su posterior almacenamiento y análisis (ver *Figura 24* , cuando $t_2 = 1\ hora$).

Figura 22. **Raspberry Pi (Edge Gateway)**

Fuente. Elaboración propia

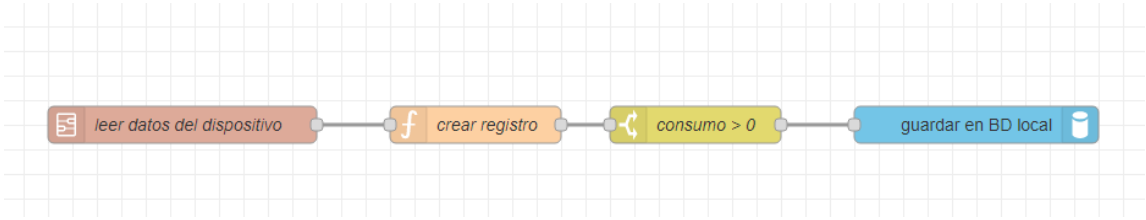


Figura 23. **Node-RED – Flujograma del proceso de transformación y almacenamiento de datos**

Fuente. Elaboración propia

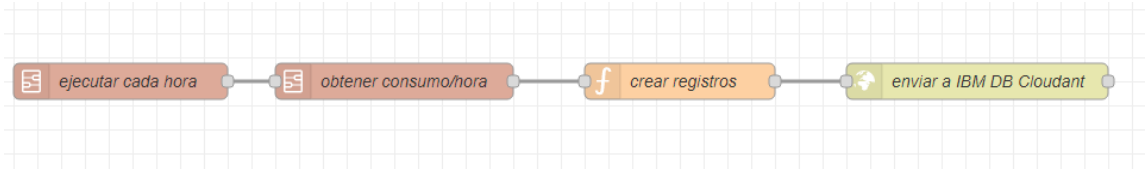


Figura 24. **Node-RED – Flujograma del proceso de envío de datos desde el componente “Edge Gateway”**

Fuente. Elaboración propia

Finalmente, en el componente Cloud, el almacenamiento se realiza en una base de datos NoSQL llamado “Cloudant” de IBM Cloud (IBM, s. f.-c). Luego, el análisis se realiza sobre la plataforma IBM Stream Flow Analysis (IBM, s. f.-a), la cual permite analizar y actuar, en tiempo real, sobre cantidades masivas de datos (estructurados o no estructurados) que pueden venir de diferentes fuentes y que son enviados constantemente, tal como se muestra en la Figura 25, y bajo un algoritmo realizado en Pyhon evalúa al instante cada registro que llega para ver si existe una posible fuga. Además, en este componente “Cloud” existe un portal web, el cual se encuentra desplegado en Azure Cloud (Microsoft, s. f.-a), para que los usuarios puedan ver, en tiempo real, la ubicación de sus medidores inteligentes, que es obtenida por GPS del celular de los usuarios al momento de su instalación, y sus consumos de agua (en tiempo real e histórico), tal como se aprecia en la Figura 26 y Figura 27, respectivamente, por medio de una interfaz hecha en Angular, la cual obtiene los datos a través de un API programado en .NetCore que se conecta con la base de datos “Cloudant” a través del protocolo http.

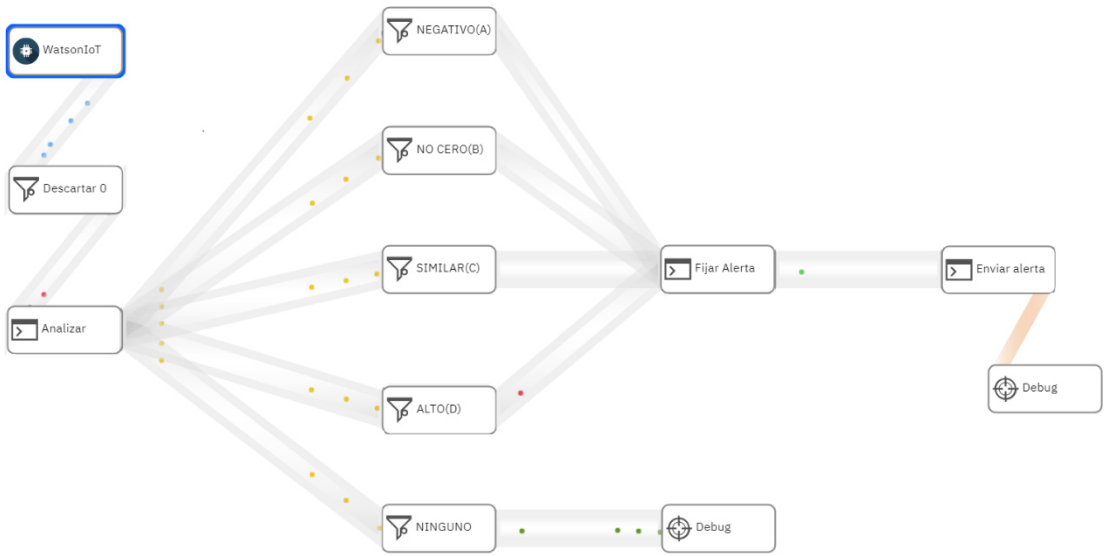


Figura 25. **IBM stream flow analysis**

Fuente. Elaboración propia

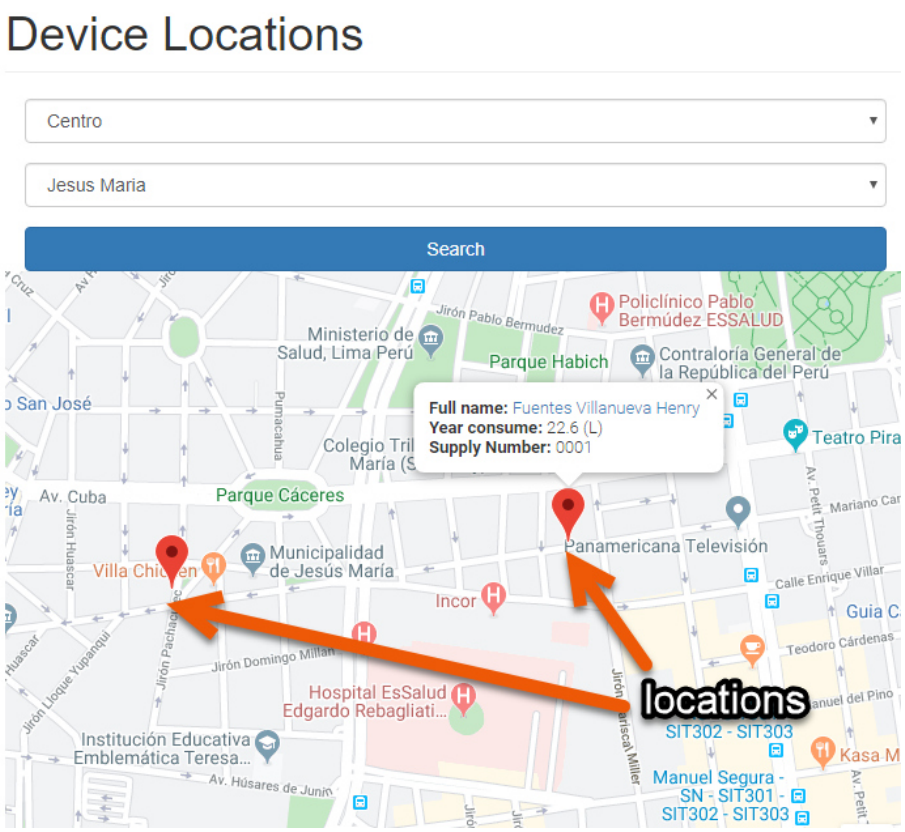
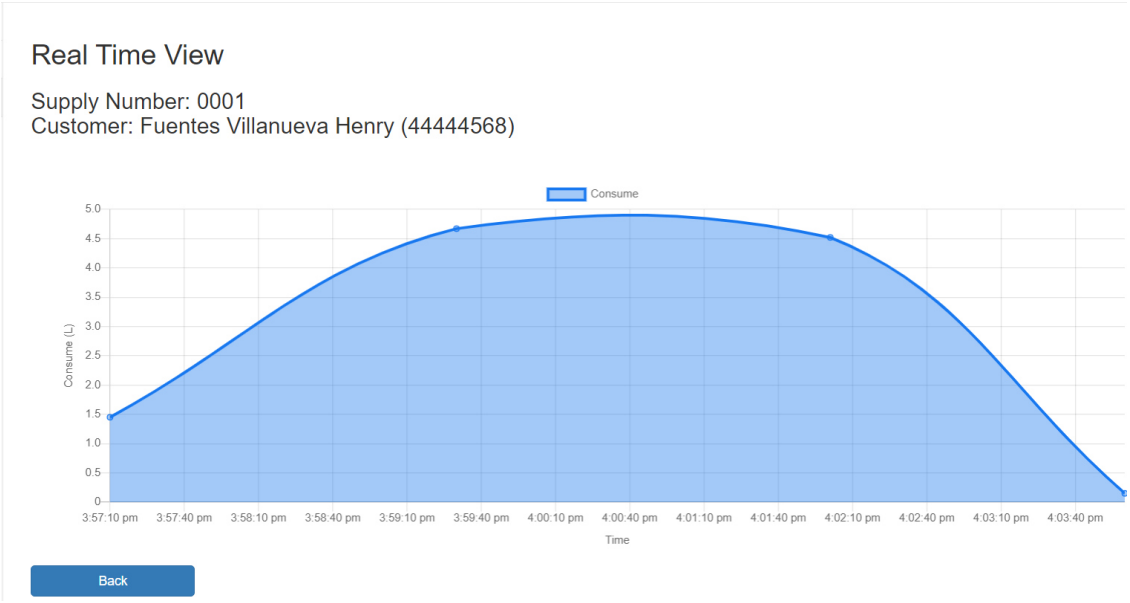
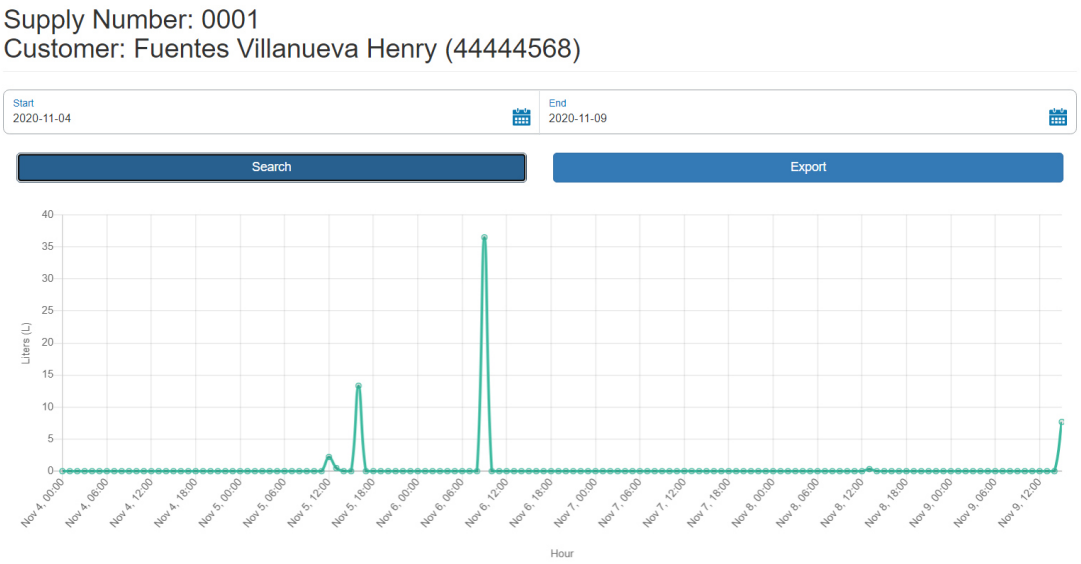


Figura 26. **Vista de ubicación de dispositivos**

Fuente. Elaboración propia



(a)



(b)

Figura 27. **Vista consumo: (a) Tiempo Real; (b) histórico**

Fuente. Elaboración propia

5.2. Requisitos

Los prototipos del sistema son detallados en el Anexo A. A continuación, se describe los requisitos:

- **Roles**
  - Administrador: Tiene acceso total al sistema y ve el consumo y facturación de cualquier usuario miembro.
  - Miembro: Es el usuario que tiene instalado el medidor inteligente y que solo puede visualizar sus consumos realizados.
- **Funcionales**
  - El usuario administrador y miembro debe tener una interfaz web de inicio de sesión.
  - El usuario miembro debe poder ver, en tiempo real, su consumo de agua en litros con un retraso de 2 min dentro de un cuadro de serie temporal.

- El usuario administrador debe poder ver, en tiempo real, el consumo de agua de cualquier usuario miembro en litros con un retraso de 2 min dentro de un cuadro de serie temporal.
- El usuario miembro debe poder ver la facturación mensual de su consumo.
- El usuario administrador debe poder ver la facturación mensual del consumo de cualquier usuario miembro.
- El usuario miembro debe poder hacer consultas de su consumo histórico hasta 1 año atrás.
- El usuario administrador debe poder hacer consultas del consumo histórico de cualquier usuario miembro hasta 1 año atrás.
- El usuario debe recibir alertas en su celular en caso de que hubiese una posible fuga de agua.
- El administrador debe poder ver en un mapa las ubicaciones de sus medidores inteligentes y poder filtrar por zona y distrito.
- El administrador debe ser capaz de visualizar el consumo de agua por zona, distrito y por usuario (DNI).

➢ **No Funcionales**

- El tiempo de carga de la página debe ser como máximo 3 segundos.
- El tiempo de respuesta para visualizar los reportes debe ser como máximo 5 segundos.
- La página web debe adaptarse fácilmente con los dispositivos móviles Android (4.0 en adelante) y IOS (5.0 en adelante), incluyendo sus versiones Tablet.
- La página debe adaptarse fácilmente con los navegadores Chrome, Firefox y Safari con las 5 últimas versiones hasta la fecha de hoy.
- La plataforma para utilizar es IBM Cloud IoT plan Lite.
- La página front end debe estar hecho en Angular 6.
- La base de datos para registrar las mediciones debe ser una NoSQL.
- La base de datos del sistema para el registro de usuarios, dispositivos y ubigeo debe estar en SQL Server 2016.
- El código back-end debe estar hecho en NetCore.

- La aplicación móvil para recibir las alertas debe estar hecha en Flutter 1.12.

5.3. Metodología de desarrollo

Para el desarrollo del software, se hizo uso de la metodología ágil SCRUM, la cual permite tener iteraciones incrementales cortas de 2 semanas con un entregable al final de esta. A continuación, se hace una descripción breve de la primera y última iteración juntamente con el cronograma de cada iteración.

- **Primera Iteración**

Durante la primera iteración, se realizó el análisis de la arquitectura a implementar juntamente con la captura de los requisitos e implementación de los prototipos.

- **Lista de Iteraciones**

A lo largo del resto de iteraciones, se hace la implementación e integración de los componentes más importantes que forman parte del sistema. En la *Tabla 8*, se hace un resumen de cada iteración.

Tabla 8. **Lista de iteraciones del desarrollo del software**

Iteración	Actividad	Semana		Entregable
		Inicio	Fin	
1	Arquitectura + tecnología	1	1	Arquitectura + tecnología
	Requisitos funcionales y no funcionales, prototipos	2	2	Lista de requisitos y prototipos
2	Dominio Tecnología – IBM IoT Watson	3	3	Demostración
	Datos Prueba	4	4	Dataset

3	Interfaz	5	5	Vistas histograma
	Desarrollo componente Cloud	6	6	Histograma de consumo y alertas con datos prueba en tiempo real
	Desarrollo componente Smart Meter + GPS	6	6	Envío de datos a la nube
4	Integración Componentes	7	7	Flujo completo
	Pruebas	7	7	Prueba de software, validación, pruebas unitarias
	Redacción Capítulo V	8	8	Documento capítulo V
	Manuales	8	8	Documento Manuales

Fuente. Elaboración propia

- **Última iteración**

En la última iteración, se hace la integración de los componentes, las pruebas del sistema y documentación de esta.

5.4. Diagrama de base de datos

- **Base de datos mediciones**

En la *Tabla 9*, se muestra los campos usados para registrar la información del consumo y ubicación del usuario (isAtHome) obtenida de los dispositivos.

Tabla 9. **Campos de la tabla de mediciones**

Nombre del Campo	Tipo
_id	Guid
deviceId	Number
userId	Number
consume	Number
time	Date
isAtHome	Boolean
isAnomalous	Boolean
totalConsume	Number
createDate	Date
updateDate	Date

Fuente. Elaboración propia

En la *Tabla 10*, se muestra los campos usados para registrar la información del consumo y ubicación del usuario (isAtHome) obtenida de los dispositivos.

Tabla 10. **Campos de la tabla de alertas**

Nombre del campo	Tipo
_id	Guid
deviceId	Number
userId	Number
consume	Number
time	Date
anomalousType	String
createDate	Date
updateDate	Date

Fuente. Elaboración propia

- **Base de datos sistema**

En la *Figura 28*, se aprecia el modelado de la base de datos donde se hace el registro de los usuarios, roles, dispositivos y localización (zona, distritos) de estos.

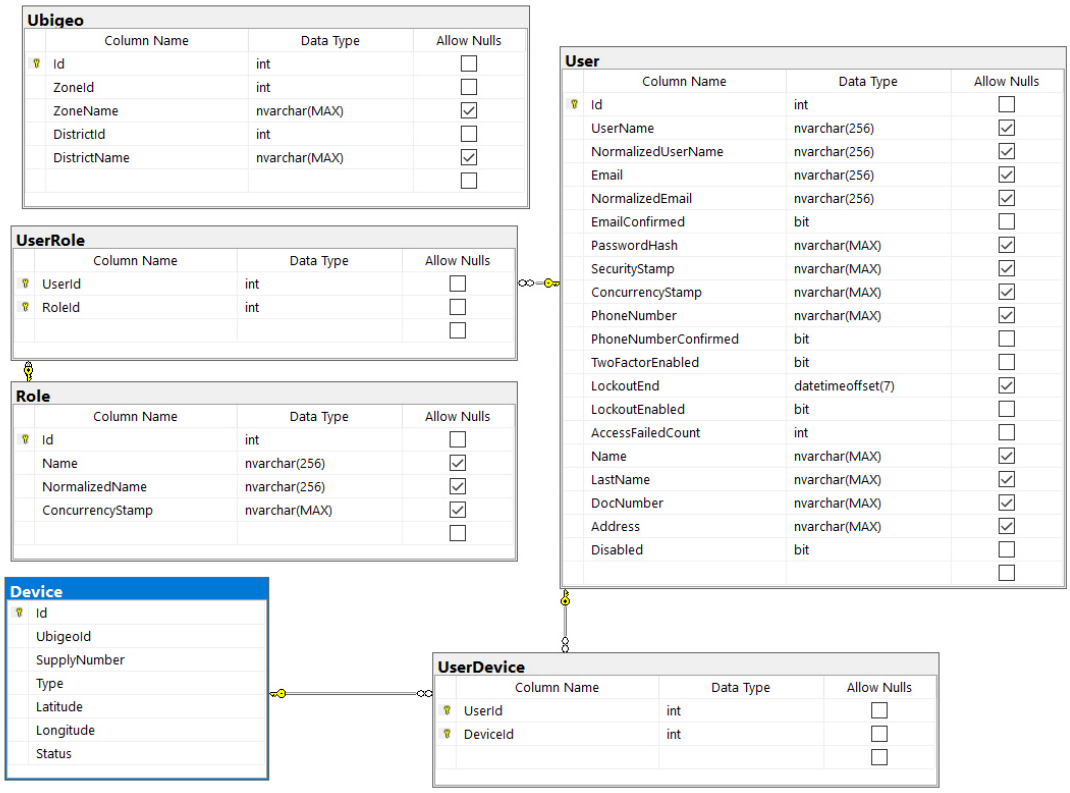


Figura 28. ***Diagrama de Base de datos del sistema***

Fuente. Elaboración propia

5.5. Limitación

➢ Hardware

• Debe existir conexión a Internet.

• El medidor debe estar conectado constantemente a la corriente eléctrica, ya que con el uso de baterías tendría una duración de vida corta de energía.

➢ Software

• IBM Cloud plan Lite solo permite registrar como máximo 500 dispositivos, 200 MB de intercambio de datos al mes, 1 GB de almacenamiento de datos en Cloudant NoSQL y solo 50 horas de análisis de datos.

5.6. Pruebas de software

➢ Funcionales

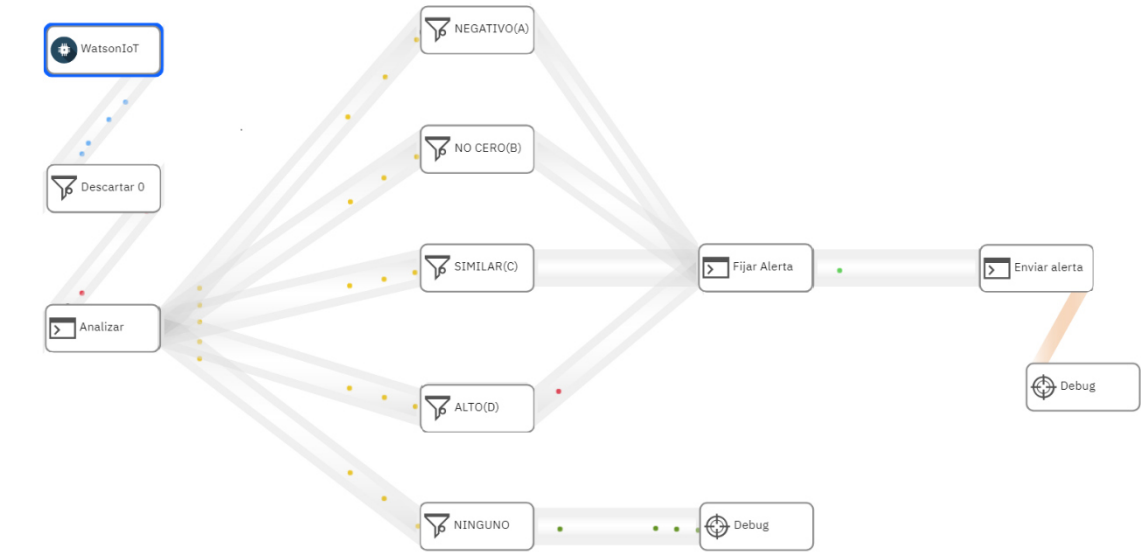
Basados en la ISO 9126 para calidad del producto software, se ha elaborado una métrica que evalúa la característica de la “funcionalidad” del software y su sub - característica “Adecuación”. La métrica mide la cantidad de alertas en el celular que fueron recibidas cuando hubo consumos fuera del rango máximo aceptado, y es detallada, a continuación, en la *Tabla 11*, cabe resaltar que se ejecutó el caso de Consumo anormal alto durante la semana (AHCW) que es explicado en el Capítulo 6 de la validación:

Tabla 11. **Métrica número de alertas recibidas**

Nombre de la Métrica	**Número de alertas recibidas**
Código:	MT-001
Característica:	Funcionalidad
Sub - característica:	Adecuación
Propósito:	¿Cuántas alertas en el celular fueron recibidas cuando hubo consumos fuera del rango máximo aceptado?
Método de aplicación:	Contar la cantidad de alertas recibidas contra el número total de alertas generadas por consumos altos anormales
Medición, fórmula y elementos medibles:	X= (A/B) *100 A = n° de alertas recibidas B = Total alertas generadas por consumos altos anormales
Tipo de medida:	A = Contador B = Contador
Interpretación del valor medido:	0<= X <= 100 Lo más cercano a 100 es lo mejor

Fuente. Elaboración propia

Los resultados en la *Figura 29* muestran que, de todos los casos simulados se registraron 6 alertas de consumo alto, y en la *Figura 30* se ve que se recibieron estas 6 alertas en el celular, de esta manera, el 100% de las pruebas fueron satisfactorias para este escenario.



Flow of Events Enviar alerta → Debug

Last Updated: Nov 8, 2020, 12:45:57 PM

Table View | Json View

user_id	device_id	consume	time	is_at_home	alert_message	alert_type	anomalous_type	id	max_consume	min_consume
1	1	350	2019-02-21 17:00:00	U	Exists a water leak!!	WATER_LEAK	HIGH_CONSUME	4775	341	-90
5	5	410	2019-02-27 18:00:00	U	Exists a water leak!!	WATER_LEAK	HIGH_CONSUME	38108	340	-120
5	5	346	2019-02-28 10:00:00	U	Exists a water leak!!	WATER_LEAK	HIGH_CONSUME	38124	340	-105
5	5	430	2019-02-28 11:00:00	U	Exists a water leak!!	WATER_LEAK	HIGH_CONSUME	38125	340	-105
6	6	210	2019-01-15 12:00:00	U	Exists a water leak!!	WATER_LEAK	HIGH_CONSUME	45269	113	-41
6	6	200	2019-02-18 22:00:00	N	Exists a water leak!!	WATER_LEAK	HIGH_CONSUME	46026	142	-51

Figura 29. **Alertas generadas**

Fuente. Elaboración propia

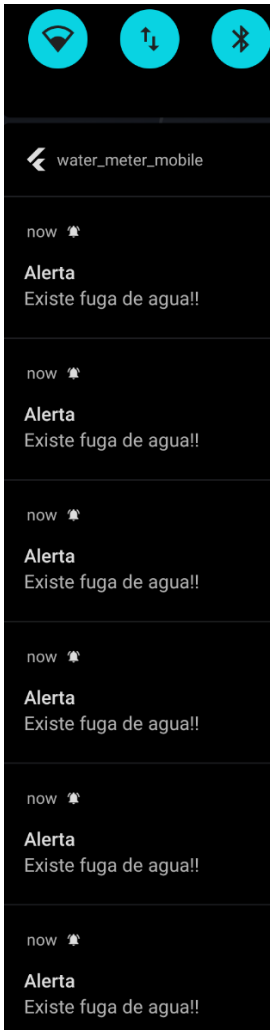


Figura 30: **Número de alertas recibidos**

Fuente. Elaboración propia

5.7. Funcionamiento

- **Caso de ejemplo**

Se va a simular un escenario donde un usuario administrador pueda ver la siguiente información:

• Ubicación de los medidores de agua por distritos

• El consumo histórico de un usuario miembro dentro de un rango de fecha

• El consumo en tiempo real de un usuario miembro

Además, en caso de una posible fuga de agua, el usuario miembro recibirá una notificación vía alerta en el móvil.

En el Anexo B, se puede ver un mayor detalle del funcionamiento del sistema.

- **Ejecución**

En primer lugar, el usuario administrador debe iniciar sesión como se ve en la *Figura 31*.

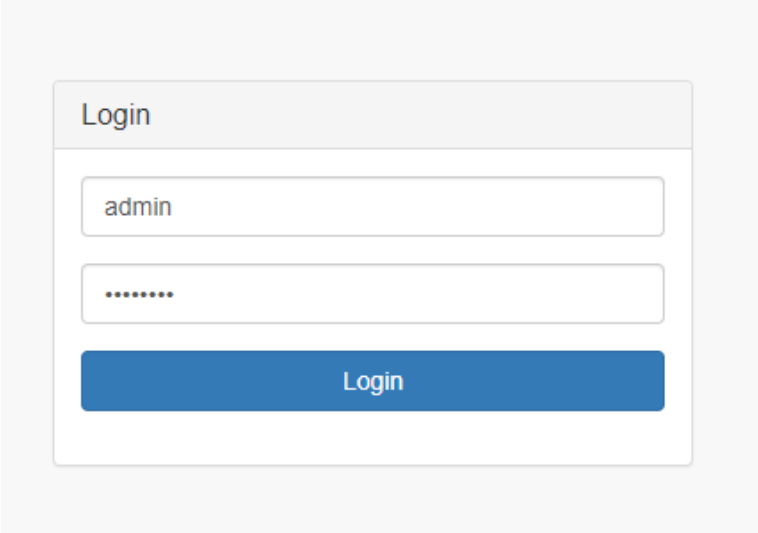


Figura 31: **Ejecución – Vista inicio de sesión**

Fuente. Elaboración propia

Luego debe seleccionar el distrito del cual desea obtener información. Los datos solicitados serán mostrados en un mapa, tal como se ve en la *Figura 32*.

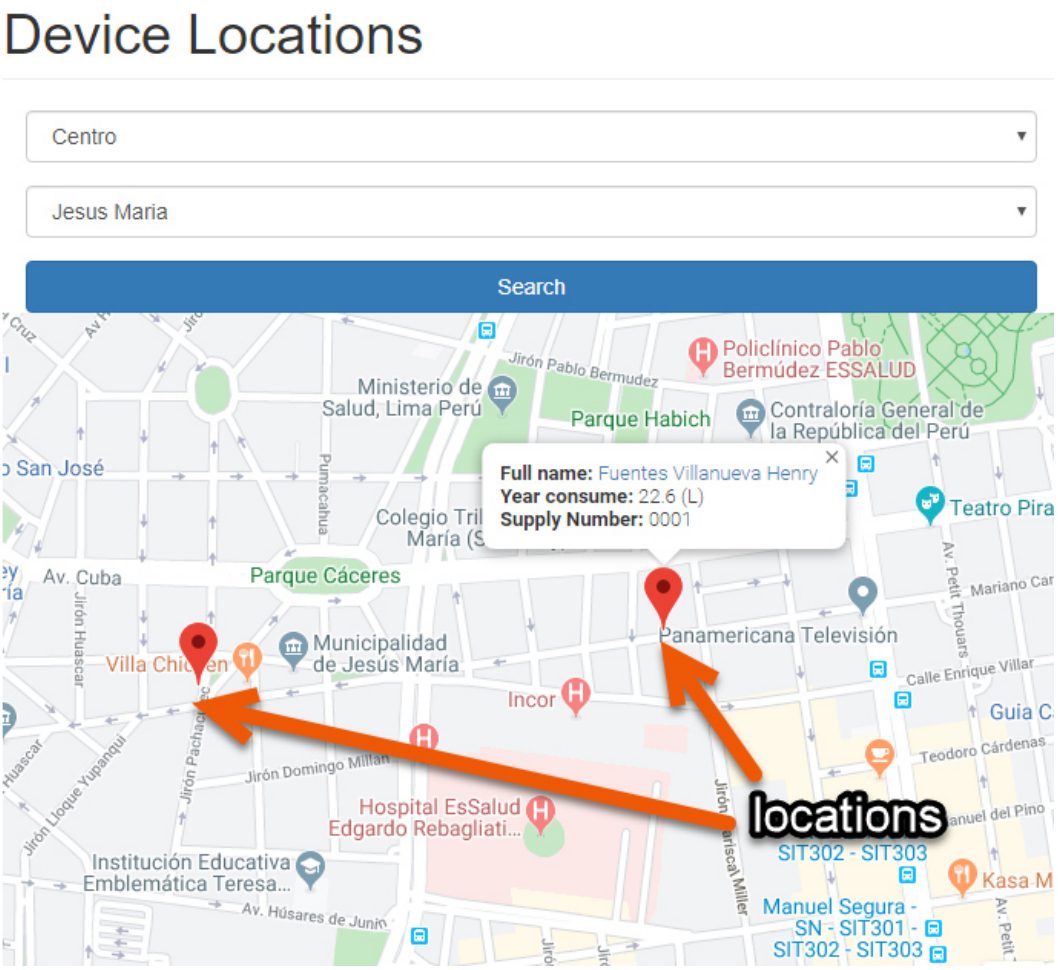


Figura 32: **Ejecución – Vista localización de dispositivos**

Fuente. Elaboración propia

Después, se debe seleccionar el usuario al cual se le desee ver su consumo histórico detallado aplicando los filtros de fecha correspondiente (ver *Figura 33*).

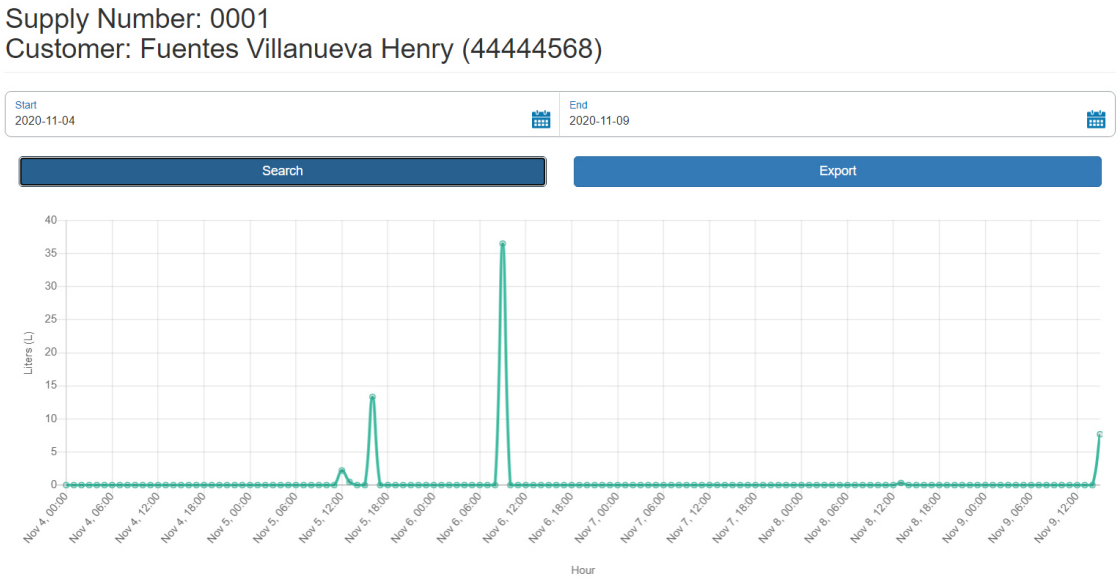


Figura 33: **Ejecución – Vista consumo histórico**

Fuente. Elaboración propia

Seguidamente, de haber abierto el caño que tiene acoplado el medidor inteligente (ver Figura 38), y después de transcurrido 2 minutos, se puede apreciar, en la *Figura 34*, que el Gateway recibe un consumo de 4.21 Litros, el mismo que es visto en la página web de consumo en tiempo real (ver *Figura 35*).

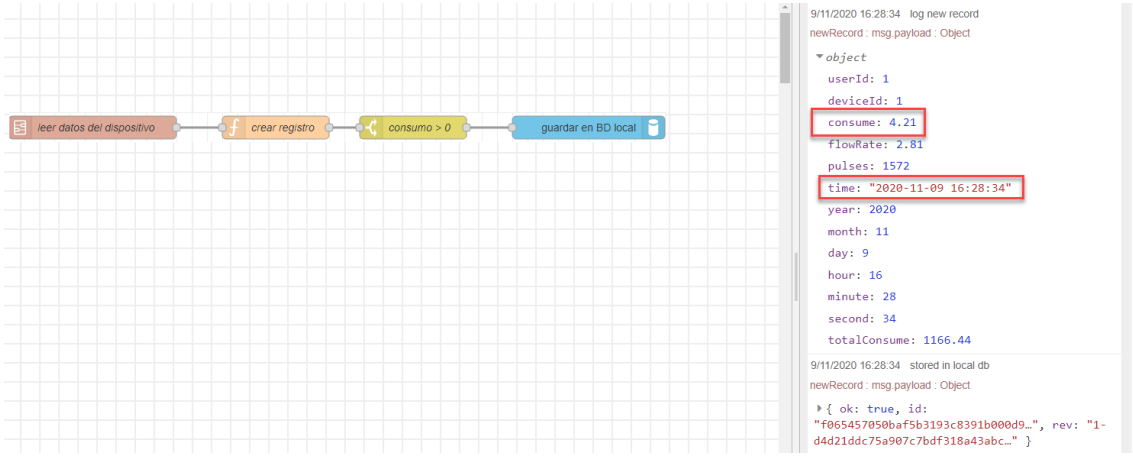


Figura 34: **Ejecución – Vista Node-RED – Gateway recibe consumo en tiempo real**

Fuente. Elaboración propia

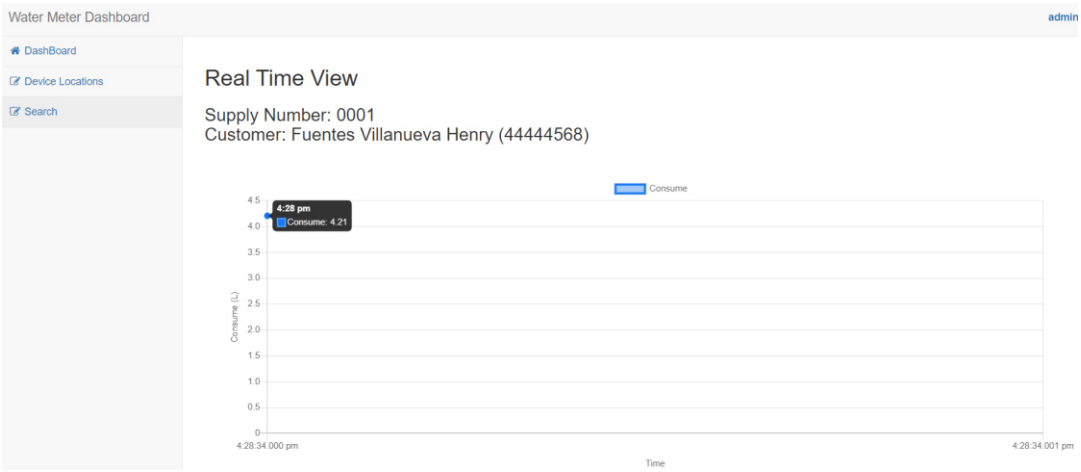
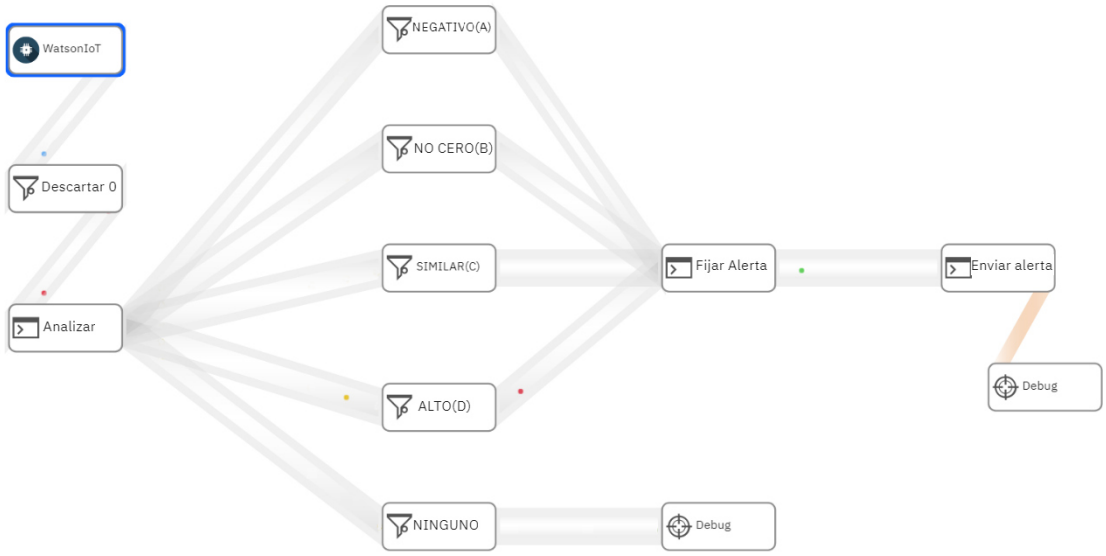


Figura 35: **Ejecución – Vista consumo en tiempo real**

Fuente. Elaboración propia

Para el análisis del consumo de agua y detección de fuga por hora, se hace uso de la plataforma “IBM Stream Flow” mostrado en la Figura 36, en el cual se ha simulado un consumo de agua muy elevado (350L), y, después de ser analizado, se envía una alerta en la aplicación móvil.



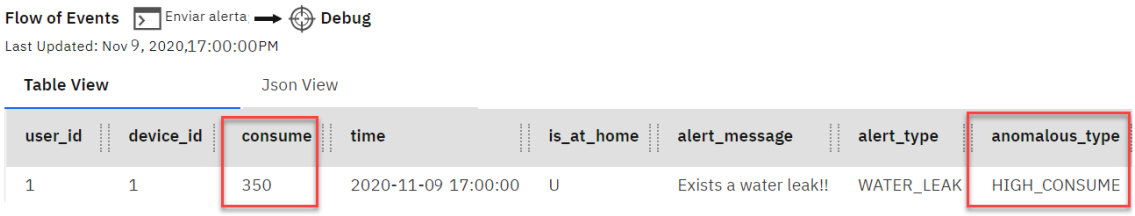


Figura 36: **Ejecución – Vista "IBM Flow Stream"**

Fuente. Elaboración propia

Al existir una probabilidad alta de fuga, el usuario es notificado a su celular como se ve en la Figura 37

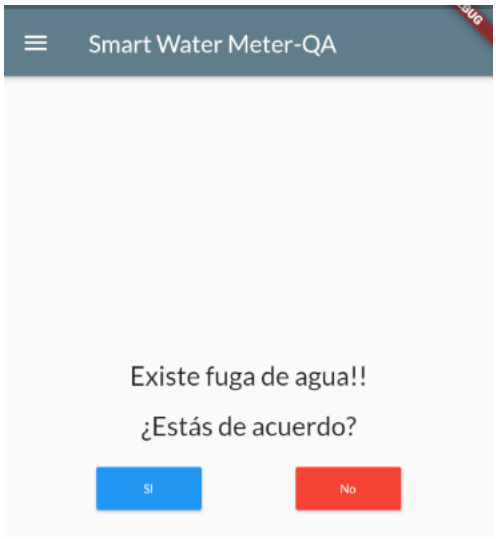


Figura 37: **Ejecución – Alerta celular**

Fuente. Elaboración propia

CAPÍTULO 6: VALIDACIÓN

En el presente capítulo, se realiza la validación de la arquitectura y algoritmo propuesto para un sistema de medición inteligente del consumo de agua. Para ello se elabora un maquete para registrar el consumo de agua y compararlo con lo que muestra el sistema. Por otro lado, se ejecutan diez diferentes escenarios con 275 test para escenarios de consumos normales y 329 para consumos anómalos, los cuales son comparados con otros algoritmos con el objetivo de medir el desempeño del algoritmo propuesto frente a los otros ya existentes.

6.1. Caso de estudio

La solución propuesta fue instalada en un departamento de la ciudad de Lima, y se evaluó en dos aspectos distintos. En primer lugar, se evaluó el porcentaje de error del registro del consumo de agua reportado por el sistema y, en segundo lugar, se midió la precisión del algoritmo de detección de fuga donde se hizo uso de un data set y se simuló los consumos para ver si la aplicación detectaba o no una posible fuga.

6.2. Aplicación

Para evaluar el margen de error del medidor inteligente sobre consumo de agua en litros, se tuvo que armar una maqueta e ir calibrando poco a poco el algoritmo de medición del caudal del agua. En la Figura 38, se aprecia el diseño de la maqueta, donde el caudal de agua es medido por medio de un sensor de flujo que registra las pulsaciones generadas por el paso de agua. Luego, a través de un balde con marcas (0.5L, 1L, 1.5L, 2L, 2.5L, 3L, 3.5L, 4L, 4.5L y 5L), se corroboró los litros registrados por el sistema contra los litros reales que han pasado por la tubería. Cabe mencionar que la calibración empezó con el factor recomendado por la documentación del sensor (seedstudio, s. f.), donde se detalla que 330 pulsos/min equivalen a 1L, pero, debido a que el margen de error era muy alto con ese factor, se siguió calibrando hasta llegar a 372 pulsos/min.

El margen de error es calculado con la siguiente métrica:

Error =| (valor real -valor) / valor real*100)| (1)

Figura 38. **Maqueta de medidor inteligente**
Fuente. Elaboración propia

Por otro lado, para medir la precisión del algoritmo de detección de fuga, se simularon 10 diferentes escenarios, donde este fue comparado con otros algoritmos ya existentes. En primer lugar, se obtuvo la data de prueba (DAIAD, 2019) que sirve para obtener un historial de consumo, esta consta de 674020 registros de 92 consumidores, en 1 año de consumo, con registros de consumo por hora, y en muchos casos menos de 24 registros por día; debido a que se desea que la data sea la más reciente, se actualizó las fechas de las mediciones a los años 2018-2019 y se seleccionó aleatoriamente solo 9 consumidores, de esta manera, se generó un dataset de 69194 registros. Posteriormente, se definieron los siguientes escenarios:

Consumo normal durante la semana (NCW): Son los consumos por hora entre los días lunes y viernes, donde existe un consumo normal de agua sin presencia de fuga.

Consumo normal durante fin de semana (NCWD): Son los consumos por hora entre los días sábado y domingo, donde existe un consumo normal de agua sin presencia de fuga.

Consumo normal en trabajo nocturno (NCNW): Son los consumos por hora en los días donde una persona suele hacer trabajos durante la madrugada y su consumo de agua se considera normal.

Consumo normal en el primer día (NCFD): Se refiere a los consumos por hora en el primer día de uso del sistema, en donde debe existir un consumo normal.

Consumo normal alto cuando se está en casa (NHCIAH): Son los consumos por hora en los días donde existe un incremento alto del consumo de agua, pero el usuario está en casa y no es considerado una anomalía o fuga de agua.

Consumo anormal alto durante la semana (AHCW): Son los consumos por hora entre los días lunes y viernes, donde existe la presencia de fuga debido a un consumo anómalo elevado.

Consumo anormal alto durante fin de semana (AHCWD): Son los consumos por hora entre los días sábado y domingo, donde existe la presencia de fuga debido a un consumo anómalo elevado.

Consumo anormal no cero (ACNZ): Son los consumos por hora en el que durante las últimas 24 horas seguidas no ha parado de registrarse consumo de agua y no existe al menos una hora donde el consumo sea cero.

Consumo anormal similar (ACS): Son los consumos por hora en dónde existe tres consumos seguidos con valores muy similares (+-1Litro), lo cual es considerado anómalo.

Consumo anormal negativo (ACN): Son los consumos por hora en el que durante las últimas 24 horas se ha registrado una tendencia negativa en el consumo acumulado de agua o se ha registrado un consumo negativo.

Los algoritmos aplicados son el Minimum Night Flow (MNF), Continuous Non-Zero Flow (CNZ), y Average per Hour (AVG). El MNF asume que cualquier consumo de agua existente entre las 2:00 a. m. y las 4:00 a. m. son indicios de

una posible fuga. Luego, el CNZ dice que, si no existe algún consumo cero dentro de un rango de 24 horas, es considerado como una posible fuga. Finalmente, el AVG es un promedio de consumos por hora realizados y, si se pasa ese promedio, es un indicio de fuga.

Para simular los consumos anómalos, se tuvo que actualizar los consumos en determinadas horas posteriores al 2019, pero los datos correspondientes al año 2018 no fueron alterados con el objetivo de tener un patrón de comportamiento histórico que ayudaría a detectar algún comportamiento anómalo en el 2019. Las alteraciones fueron realizadas para generar registros para los escenarios anómalos y el escenario NHCIAH, por ejemplo, para el escenario ACNZ, se agregó un valor al azar a los consumos que tenían cero y, para NHCIAH, se estableció que el usuario se encontraba dentro de su casa teniendo un consumo alto. El data set y los escenarios de prueba se encuentran disponibles en https://github.com/henrygustavo/data_set, con la distribución de test por escenario de NCFD con 48, NCNW con 32, NCW con 120, NCWD con 46, NHCIAH con 29, ACN con 68, ACNZ con 96, ACS con 32, AHCW con 85, AHCWD con 48, haciendo un total de 275 test para escenarios de consumos normales y 329 para consumos anómalos. Además, cada test tiene un campo llamado “isAnomalous” con valor 1 o 0, que indica si se debe emitir o no una alerta de fuga para un consumo determinado.

La medición del algoritmo se realiza a través de la matriz de confusión que permite medir el desempeño de los algoritmos contra los consumos de referencia o esperados (Benchmark), la cual es apreciada en la *Tabla 12*, donde sus valores principales son los siguientes:

- **Verdadero Positivo (VP):** Fuga de agua identificada por el algoritmo
- **Falso Positivo (FP):** Fuga de agua inexistente identificada erróneamente por el algoritmo (falsa alarma)
- **Falso Negativo (FN):** Fuga de agua no identificada por el algoritmo

- **Verdadero Negativo (VN):** Ausencia real de fuga de agua (la mayoría de los casos)

Tabla 12. **Matriz de confusión**

		Benchmark	
		Presencia de pérdida de agua	**Ausencia de pérdida de agua**
Algoritmo	**Presencia de pérdida de agua**	VP	FP
	Ausencia de pérdida de agua	FN	VN

Fuente. Elaboración propia

Las siguientes métricas fueron calculadas con base en los resultados provistos por la matriz de confusión:

$$Exactitud = \frac{VP + VN}{\mathrm{VP} + \mathrm{VN} + \mathrm{FP} + \mathrm{FN}} \quad (2)$$

$$Exhaustividad = \frac{VP}{\mathrm{VP} + \mathrm{FN}} \quad (3)$$

$$Precisión = \frac{VP}{\mathrm{VP} + \mathrm{FP}} \quad (4)$$

$$Valor\ F1 = \frac{2\ x\ Precisiónx\ Exhaustividad}{Precisión + Exhaustividad} \quad (5)$$

Exactitud indica el porcentaje de escenarios fugas y no fugas identificados correctamente por el algoritmo. *Exhaustividad* cuantifica la capacidad del

algoritmo para identificar alarmas, medido por el ratio entre alarmas correctamente identificadas y el total numérico de alarmas verdaderas. *Precisión* mide la capacidad del algoritmo de evitar falsas alarmas, basado en el ratio entre el número de alarmas verdaderas identificadas y el total de alarmas identificadas por este. Finalmente, *Valor F1* permite evaluar la capacidad del algoritmo, en una sola métrica, de distinguir entre las horas con y sin pérdida de agua y es calculado como la media harmónica de *Exhaustividad y Precisión*.

En la *Tabla 13* y *Tabla 14*, se muestra los resultados obtenidos de las pruebas realizadas en los distintos escenarios de consumo normal y anómalo, respectivamente. La columna "#Tests" es la cantidad de consumos por hora que han sido sometidos a prueba por los diferentes algoritmos en un determinado día. Luego, la columna "Fugas/día" son la cantidad de fugas que deben detectarse en un determinado día. Posteriormente, las columnas "MNF", "CNZ", "AVG" y "Algoritmo Propuesto" muestran la cantidad de fugas que se han detectado en el día.

Tabla 13. **Escenarios de consumo normal de agua**

Escenario	**Id Usuario**	**Fecha**	**# Tests**	**Fugas/ Día**	**MNF**	**CNZ**	**AVG**	**Algoritmo Propuesto**
NCFD	2	2018-03-01	24	0	0	0	0	0
NCFD	3	2018-03-01	24	0	2	0	0	0
NCNW	1	2019-02-25	16	0	3	0	1	0

NCNW	1	2019-02-27	16	0	3	0	1	0
NCW	2	2019-01-07	24	0	0	0	5	0
NCW	2	2019-01-22	24	0	1	0	4	0
NCW	2	2019-01-23	24	0	0	0	5	0
NCW	3	2019-01-24	24	0	2	0	3	0
NCW	3	2019-02-08	24	0	4	0	1	0
NCWD	4	2019-02-02	24	0	1	0	1	0
NCWD	4	2019-02-24	22	0	1	0	2	0
NHCIAH	1	2019-02-01	9	0	0	0	6	0
NHCIAH	4	2019-01-07	20	0	0	0	1	0
TOTAL			275	0	17	0	30	0

Fuente. Elaboración propia

Tabla 14. **Escenarios de consumo anómalo de agua**

Escenario	Id Usuario	Fecha	# Tests	Fugas/ Día	MNF	CNZ	AVG	Algoritmo propuesto
ACN	7	2019-02-28	22	1	0	0	5	1
ACN	8	2019-02-28	23	1	1	0	1	1
ACN	9	2019-02-28	23	2	1	0	2	2
ACNZ	3	2019-01-02	24	1	5	1	4	1
ACNZ	3	2019-01-18	24	1	5	1	5	1
ACNZ	7	2019-01-17	24	1	5	1	6	1
ACNZ	9	2019-01-15	24	1	5	1	6	1
ACS	1	2019-01-04	4	1	3	0	0	1
ACS	1	2019-02-02	20	1	0	0	5	1
ACS	8	2019-01-03	8	1	0	0	3	1
AHCW	1	2019-02-21	8	1	0	0	1	1

AHCW	5	2019-02-27	16	1	3	0	7	1
AHCW	5	2019-02-28	22	2	3	0	12	2
AHCW	6	2019-01-15	20	1	0	0	3	1
AHCW	6	2019-02-18	19	1	0	0	6	1
AHCWD	8	2019-02-23	24	1	0	0	1	1
AHCWD	8	2019-02-24	24	1	0	0	1	1
TOTAL			329	19	31	4	68	19

Fuente. Elaboración propia

6.3. Discusión de resultados

En la *Tabla 15*, se muestra las diez pruebas realizadas con el último factor mencionado, donde se logró llegar a un porcentaje de 4.63 % de margen de error, para 10 valores aleatorios desde 0.5L a 5L.

Tabla 15. **Comparación de consumo real de agua vs. consumo registrado por el sistema**

#	Pulsos	Valor real(litros)	Valor(litros)	Error
1	367	1.00	0.99	1.00%
2	546	1.50	1.50	0.00%
3	794	2.00	2.14	7.00%
4	1153	3.00	3.10	3.33%
5	1197	3.00	3.22	7.33%
7	1504	4.00	4.04	1.00%
8	1606	4.00	4.32	8.00%
9	1812	4.50	4.87	8.22%
10	1967	5.00	5.29	5.80%
Avg.				4.63%

Fuente. Elaboración propia

En la *Tabla 16*, se muestra el resultado de la matriz de confusión, donde se aprecia que el algoritmo propuesto tiene un *Exactitud*, *Exhaustividad*, *Precisión* y *Valor F1* de 100 %, lo cual demuestra que son superiores a los otros algoritmos.

Tabla 16. **Resultados de la matriz de confusión por algoritmo**

Nombre de la métrica	**MNF**	**CNZ**	**AVG**	**Algoritmo propuesto**
Población	604	604	604	604
Condición positiva	19	19	19	19
Condición negativa	585	585	585	585
Verdadero positivo	1	4	10	19
Verdadero negativo	538	585	497	585
Falso positivo	47	0	88	0
Falso negativo	18	15	9	0
Exactitud	0.89	0.98	0.84	1.00
Exhaustividad	0.05	0.21	0.53	1.00
Precisión	0.02	1.00	0.10	1.00
Valor F1	0.03	0.98	0.17	1.00

Fuente. Elaboración propia

CAPÍTULO 7: CONCLUSIONES

7.1. Conclusiones

En este trabajo, se ha realizado una revisión sistemática de varios estudios, los cuales han aportado un profundo conocimiento sobre arquitecturas IoT, cloud computing y algoritmos de detección de fugas , lo cual ha permitido proponer una arquitectura para un sistema de medición del consumo de agua que cubra cinco aspectos importantes: la captura del consumo de agua, el procesamiento local de los consumos registrados, la seguridad física del dispositivo electrónico, el almacenamiento y la visualización de los consumos obtenidos, y la detección de fugas.

También, se logró proponer una arquitectura funcional e integral, puesto que cubre varios aspectos de la medición y la detección de fuga, y es mantenible por su alto nivel de desacoplamiento. Además, se llegó a implementar un sistema basado en la arquitectura propuesta que, muestra que la arquitectura permite integrar diversas tecnologías y lenguajes programación, la comunicación de diferentes servicios Cloud, el uso de hardware de bajo costo, el uso de software libre, además cubrir diversos aspectos del problema de medición del consumo de agua en tiempo real y detección de fuga.

Cabe resaltar que, las pruebas numéricas sobre 10 registros de consumo de agua mostraron que el sistema presenta un margen de error mínimo de 4.63 % con un equipo de bajo costo.

Para la detección de fuga, se introdujo un algoritmo que considera la ubicación del usuario, la data histórica, y las reglas MNF y CNZ de la literatura, así como C3S y CHA de las reglas propuestas, con lo cual se consiguió cubrir 10 posibles escenarios de consumo de agua entre consumo normal y anómalos. Este algoritmo de detección de fugas para 604 casos de prueba entre normales y anómalos detecta todas las situaciones de fugas, y presenta Exactitud, Exhaustividad, Precisión y Valor F1 de 100 %, lo cual lo hace superior al resto de algoritmos de detección de fuga.

Finalmente, los experimentos numéricos hechos con un dataset de 69194 registros del consumo de solo 1 año de 9 personas permitieron validar la fiabilidad y precisión del algoritmo propuesto frente a otros ya existentes.

7.2. Recomendaciones

Se recomienda extender los estudios expuestos en esta tesis a la integración con otros ecosistemas de medición inteligente, ya sea de electricidad o de gas.

También, se debe considerar el poner a prueba la arquitectura propuesta en un ambiente donde haya bastante concurrencia de usuarios, y realizar las modificaciones pertinentes a fin de que el performance y la escalabilidad no se vean afectados negativamente.

Por otro lado, se recomienda realizar pruebas del registro del consumo de agua con otros medidores de pulso de agua existentes en el mercado, con el objetivo de reducir el margen de error.

Otra recomendación sería, que, basados en los 10 escenarios propuestos, buscar más escenarios, sobre todo anómalos, en los cuales se pueda poner a prueba el algoritmo, y ajustarlo para que no pierda su efectividad.

Por último, sería recomendable, realizar experimentos numéricos con el algoritmo propuesto usando un dataset de consumo de agua, pero segmentadas por zonas, ya que como es bien sabido, el consumo de agua varía por ubicación. Esto podría ayudar a hallar patrones de comportamiento de consumo de agua por zonas, y así mejora la precisión del algoritmo frente a otros escenarios no contemplado en este estudio.

7.3. Trabajos futuros

Futuros trabajos estarán enfocados en dos áreas importantes como son: la seguridad de los datos enviados sin perder el rendimiento del sistema, y el escalamiento de la arquitectura para microservicios, lo cual se considera viable dado el alto desacoplamiento de las componentes.

REFERENCIAS BIBLIOGRÁFICAS

Abreu, V., Santin, A., Xavier, A., Lando, A., Witkovski, A., Ribeiro, R., Stihler, M., Zambenedetti, V., y Chueiri, I. (2018). A Smart Meter and Smart House Integrated to an IdM and Key-based Scheme for Providing Integral Security for a Smart Grid ICT. *Springer*, *23*(4), 967-981. https://doi.org/10.1007/s11036-017-0960-4

Alvisi, S., Casellato, F., Franchini, M., Govoni, M., Luciani, C., Poltronieri, F., Riberto, G., Stefanelli, C., y Tortonesi, M. (2019). Wireless Middleware Solutions for Smart Water Metering. *Sensors*, *19*(8), 1853. https://doi.org/10.3390/s19081853

Barnes, J. W. (1994). *Statistical Analysis for Engineers and Scientists: A Computer-Based Approach*. McGraw-Hill College.

Chen, Y., y Han, D. (2018). Water quality monitoring in smart city: A pilot project. *ELSEVIER SCIENCE BV*, *89*, 307-316. https://doi.org/10.1016/j.autcon.2018.02.008

Chung, W.-Y., y Yoo, J.-H. (2015). Remote water quality monitoring in wide area. *ELSEVIER SCIENCE SA*, *217*, 51-57. https://doi.org/10.1016/j.snb.2015.01.072

DAIAD. (2019, enero 9). *DAIAD Trial Data* [DAIAD Trial Data]. DAIAD Trial Data. https://github.com/DAIAD/data/blob/master/swm_trialA.zip

DeOreo, W. B., Mayer, P., Dziegielewski, B., y Kiefer, J. (2016). *Residential End Uses of Water, Version 2 Executive Report*.

https://www.circleofblue.org/wp-content/uploads/2016/04/WRF_REU2016.pdf

Dong, J., Wang, G., Yan, H., Xu, J., y Zhang, X. (2015). A survey of smart water quality monitoring system. *Springer*, *22*(7), 4893-4906. https://doi.org/10.1007/s11356-014-4026-x

Farah, E., y Shahrour, I. (2017). Leakage Detection Using Smart Water System: Combination of Water Balance and Automated Minimum Night Flow. *SPRINGER*, *31*(15), 4821-4833. https://doi.org/10.1007/s11269-017-1780-9

Farah, E., y Shahrour, I. (2018). Smart water technology for leakage detection: Feedback of large-scale experimentation. *SPRINGER*, *96*(2), 235-242. https://doi.org/10.1007/s10470-018-1137-1

GrowingBlue. (2011). *GrowingBlue. Water. Economics. Life*. https://climate-adapt.eea.europa.eu/metadata/publications/growingblue-water-economics-life/11271003

Handson Technology. (s. f.). *ESP8266 NodeMCU WiFi Devkit*. Recuperado 15 de octubre de 2019, de https://www.handsontec.com/pdf_learn/esp8266-V10.pdf

Hauber-Davidson, G., y Idris, E. (2006). *Smart water metering*.

Horsburgh, J. S., Leonardo, M. E., Abdallah, A. M., y Rosenberg, D. E. (2017). Measuring water use, conservation, and differences by gender using an

inexpensive, high frequency metering system. *ELSEVIER SCI LTD*, *96*, 83-94. https://doi.org/10.1016/j.envsoft.2017.06.035

IBM. (s. f.-a). *Overview of a streams flow in IBM Watson Studio*. Overview of a Streams Flow in IBM Watson Studio. Recuperado 30 de junio de 2020, de https://dataplatform.cloud.ibm.com/docs/content/wsj/streaming-pipelines/overview-streaming-pipelines.html

IBM. (s. f.-b). *What is edge computing?* What Is Edge Computing? Recuperado 10 de septiembre de 2020, de https://www.ibm.com/cloud/what-is-edge-computing

IBM. (s. f.-c). *What is IBM Cloudant?* What Is IBM Cloudant? Recuperado 30 de junio de 2020, de https://www.ibm.com/cloud/cloudant

Joo, J. C., Oh, H. J., Ahn, H., Ahn, C. H., Lee, S., y Ko, K.-R. (2015). Field application of waterworks automated meter reading systems and analysis of household water consumption. *TAYLOR & FRANCIS INC*, *54*(4-5), 1401-1409. https://doi.org/10.1080/19443994.2014.889609

Kitchenham, B., y Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering version 2.3*. https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

Liu, A., y Mukheibir, P. (2018). Digital metering feedback and changes in water consumption – A review. *ELSEVIER SCIENCE BV*, *134*, 136-148. https://doi.org/10.1016/j.resconrec.2018.03.010

Lloret, J., Tomas, J., Canovas, A., y Parra, L. (2016). An Integrated IoT Architecture for Smart Metering. *IEEE COMMUNICATION MAGAZINE*, *54*(12), 50-57. https://doi.org/10.1109/MCOM.2016.1600647CM

Luciani, C., Casellato, F., Alvisi, S., y Franchini, M. (2019). Green Smart Technology for Water (GST4Water): Water Loss Identification at User Level by Using Smart Metering Systems. *Water*, *11*(3), 405. https://doi.org/10.3390/w11030405

Marais, J., Malekian, R., Ye, N., y Wang, R. (2016). A Review of the Topologies Used in Smart Water Meter Networks: A Wireless Sensor Network Application. *HINDAWI LTD*, *2016*, 1-12. https://doi.org/10.1155/2016/9857568

Microsoft. (s. f.-a). *Get to know Azure*. Get to Know Azure. Recuperado 30 de junio de 2020, de https://azure.microsoft.com/en-us/overview/

Microsoft. (s. f.-b). *What is cloud computing?* What Is Cloud Computing? Recuperado 13 de septiembre de 2020, de https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/

Microsoft. (s. f.-c). *What Is IoT?* What Is IoT? Recuperado 4 de octubre de 2020, de https://azure.microsoft.com/en-us/overview/internet-of-things-iot/what-is-the-internet-of-things/

Node-RED. (s. f.). *Node-RED*. Node-RED. Recuperado 10 de noviembre de 2020, de https://nodered.org/

Oracle. (s. f.). *What Is IoT?* What Is IoT? Recuperado 4 de octubre de 2020, de https://www.oracle.com/internet-of-things/what-is-iot.html

Pan, J., Jain, R., Paul, S., Vu, T., Saifullah, A., y Sha, M. (2015). An Internet of Things Framework for Smart Energy in Buildings: Designs, Prototype, and Experiments. *IEEE INTER OF THINGS JOURNAL*, *2*(6), 527-537. https://doi.org/10.1109/JIOT.2015.2413397

Patabendige, S., Cardell-Oliver, R., Wang, R., y Liu, W. (2018). Detection and interpretation of anomalous water use for non-residential customers. *ELSEVIER SCI LTD*, *100*, 291-301. https://doi.org/10.1016/j.envsoft.2017.11.028

Perumal, T., Sulaiman, M. N., y Leong, C. Y. (2015). Internet of Things (IoT) enabled water monitoring system. *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, 86-87. https://doi.org/10.1109/GCCE.2015.7398710

RaspberryPi. (s. f.). *Raspberry Pi 3 Model B+*. Raspberry Pi 3 Model B+. Recuperado 26 de junio de 2020, de https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

Saravanan, K., Anusuya, E., Kumar, R., y Son, L. H. (2018). Real-time water quality monitoring using Internet of Things in SCADA. *SPRINGER*, *190*(9). https://doi.org/10.1007/s10661-018-6914-x

Schultz, W., Javey, S., y Sorokina, A. (2018). Smart Water Meters and Data Analytics Decrease Wasted Water Due to Leaks. *WILEY*, *110*(11), E24-E30. https://doi.org/10.1002/awwa.1124

seedstudio. (s. f.). *G3/4 inch Water Flow sensor*. Recuperado 15 de octubre de 2019, de http://wiki.seeedstudio.com/G3-4_Water_Flow_sensor/

Sethi, P., y Sarangi, S. R. (2017). Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, *2017*, 1-25. https://doi.org/10.1155/2017/9324035

Sharma, S. K., y Vairavamoorthy, K. (2009). Urban water demand management: Prospects and challenges for the developing countries. *Water and Environment Journal*, *23*(3), 210-218. https://doi.org/10.1111/j.1747-6593.2008.00134.x

Stewart, R. A., Nguyen, K., Beal, C., Zhang, H., Sahin, O., Bertone, E., Vieira, A. S., Castelletti, A., Cominola, A., Giuliani, M., Giurco, D., Blumenstein, M., Turner, A., Liu, A., Kenway, S., Savić, D. A., Makropoulos, C., y Kossieris, P. (2018). Integrated intelligent water-energy metering systems and informatics: Visioning a digital multi-utility service provider. *ELSEVIER SCI LTD*, *105*, 94-117. https://doi.org/10.1016/j.envsoft.2018.03.006

SUNASS. (2017). *Yo Cuido el Agua*. https://twitter.com/sunassperu/status/900029102652104704

SUNASS. (2018, noviembre 29). *SUNASS INICIA CAMPAÑA DE AHORRO DEL AGUA PARA EVITAR SU DERROCHE DURANTE EL VERANO*. http://www.sunass.gob.pe/doc/NotasPrensa/2018/Enero/np004_2018.pdf

United Nations. (2014). *Human right to water*. Human right to water. https://www.un.org/waterforlifedecade/human_right_to_water.shtml

Zafar, S., Miraj, G., y Baloch, R. (2018). An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service. *ENGINEERING TECHNOLOGY & APPLIED SCIENCE RESEARCH*, *8*(4), 5.

Zhu, L., Li, M., Zhang, Z., Du, X., y Guizani, M. (2018). Big Data Mining of Users' Energy Consumption Patterns in the Wireless Smart Grid. *IEEE WIRELESS COMMUNICATIONS*, *25*(1), 84-89. https://doi.org/10.1109/MWC.2018.1700157

ANEXOS

ANEXO A. Prototipos

En la Figura 39, se muestra la vista de inicio de sesión.

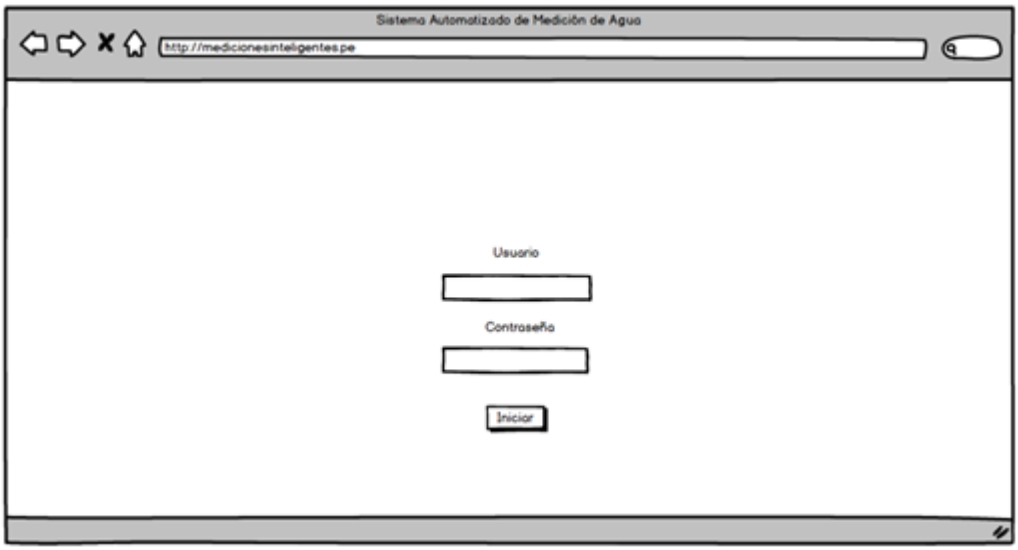


Figura 39. **Prototipo – Vista inicio de sesión**

Fuente. Elaboración propia

En la Figura 40 se muestra la vista de pantalla inicial.

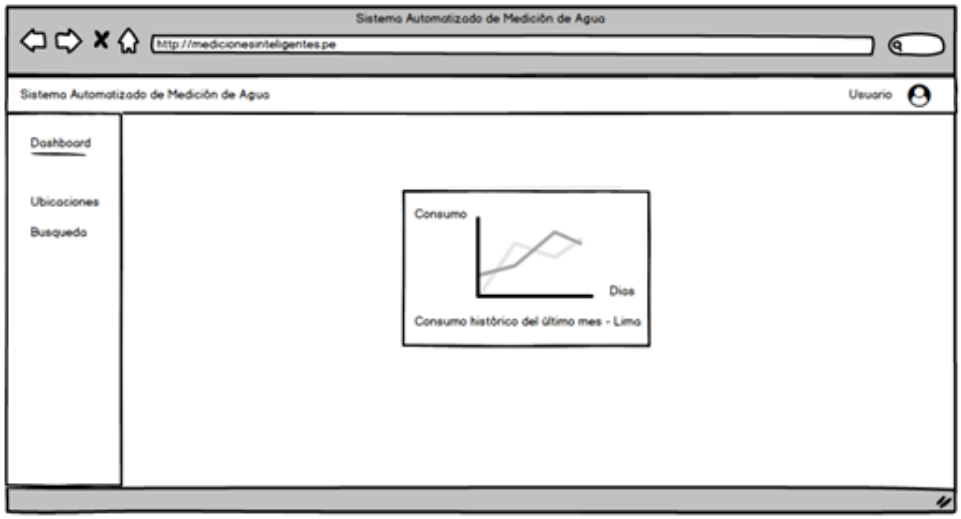


Figura 40. **Prototipo – Vista Pantalla inicial**

Fuente. Elaboración propia

En la Figura 41 se muestra la vista de filtro de dispositivos por ubicación.

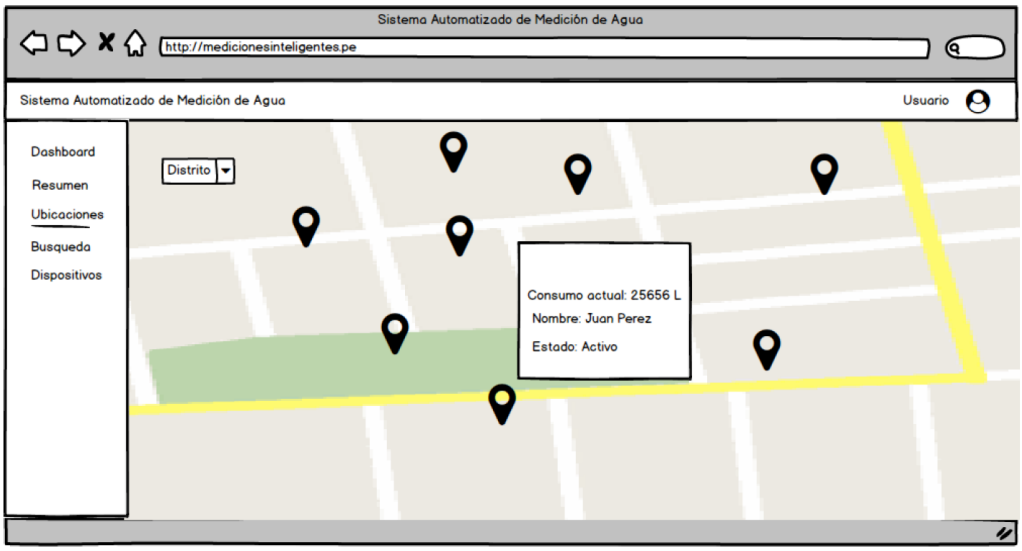


Figura 41. **Prototipo – Vista filtro dispositivos**

Fuente. Elaboración propia

En la Figura 42 se muestra la vista de búsqueda de usuarios con su consumo de agua.

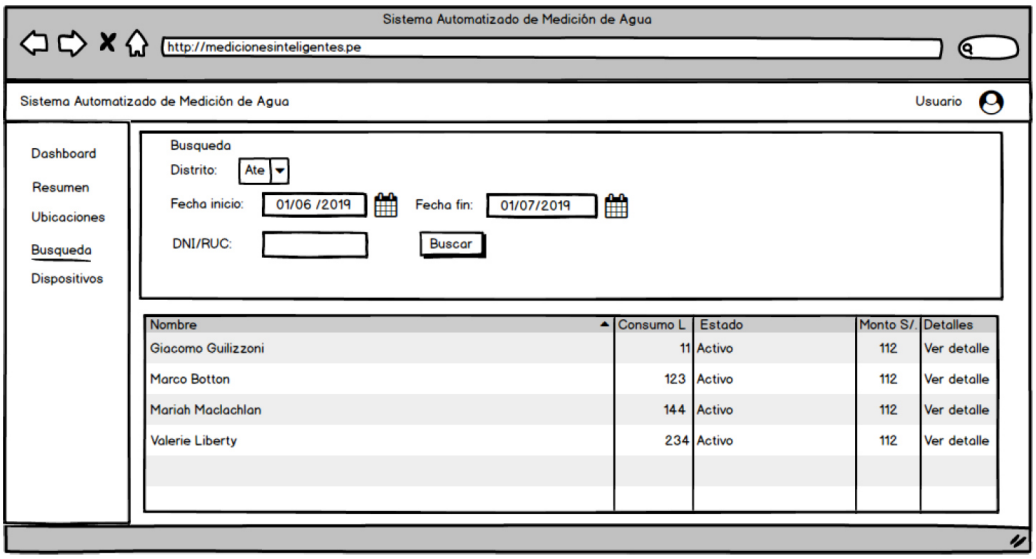


Figura 42. **Prototipo – Vista búsqueda usuarios consumo**

Fuente. Elaboración propia

En la Figura 43, se muestra la vista detallada de los dispositivos asignados a un usuario miembro.

Figura 43. **Prototipo – Vista detalle usuario – dispositivo**
Fuente. Elaboración propia

En la Figura 44, se muestra la vista histórica del consumo de agua por dispositivo y, en la Figura 45, la vista del consumo en tiempo real.

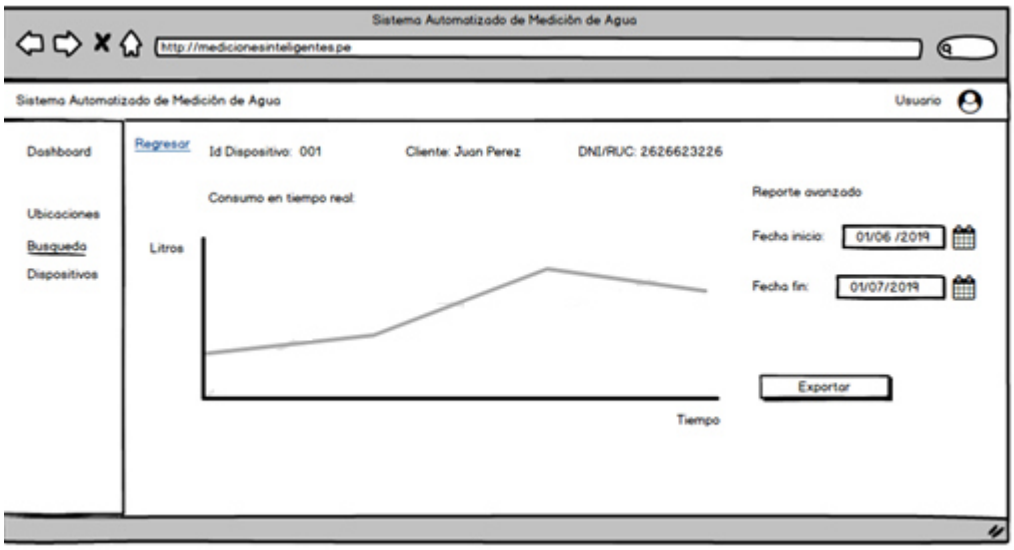


Figura 44. **Prototipo – Vista histograma consumo histórico**

Fuente. Elaboración propia

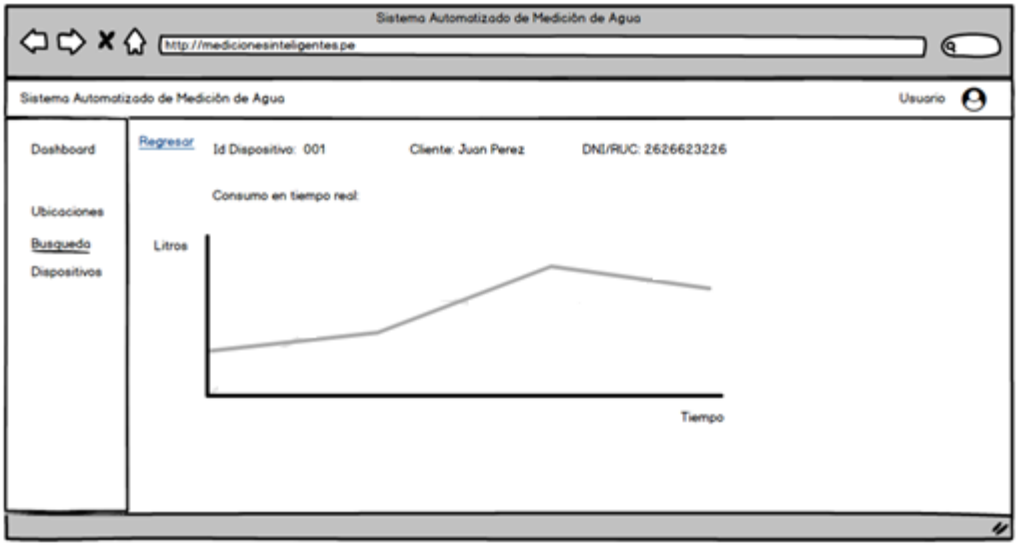


Figura 45. **Prototipo – Vista tiempo real consumo**

Fuente. Elaboración propia

ANEXO B. Manual de Usuario

El usuario administrador debe iniciar sesión en esta Url https://smart-water-meter-qa.azurewebsites.net/#/account/login, con su nombre de usuario y contraseña, tal como se muestra en la Figura 46.

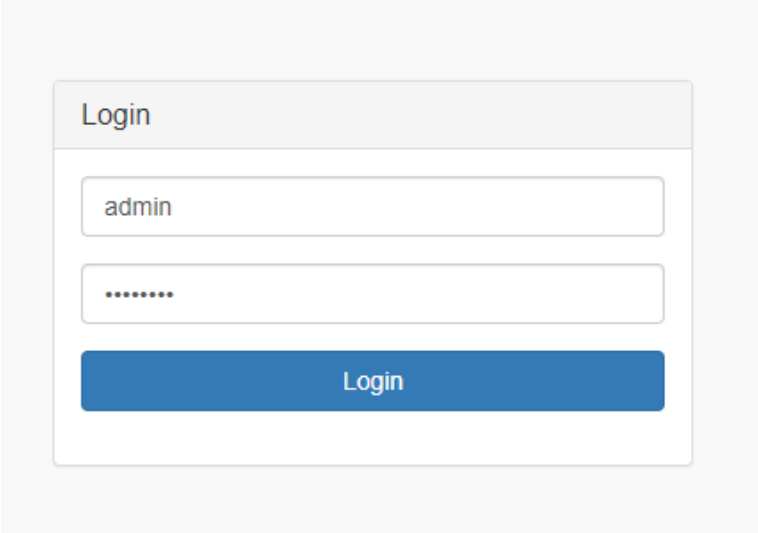


Figura 46. **Manual de usuario – Vista inicio de sesión**
Fuente. Elaboración propia

Luego, será redirigido a la pantalla principal de bienvenida, tal como se muestra en la Figura 47.

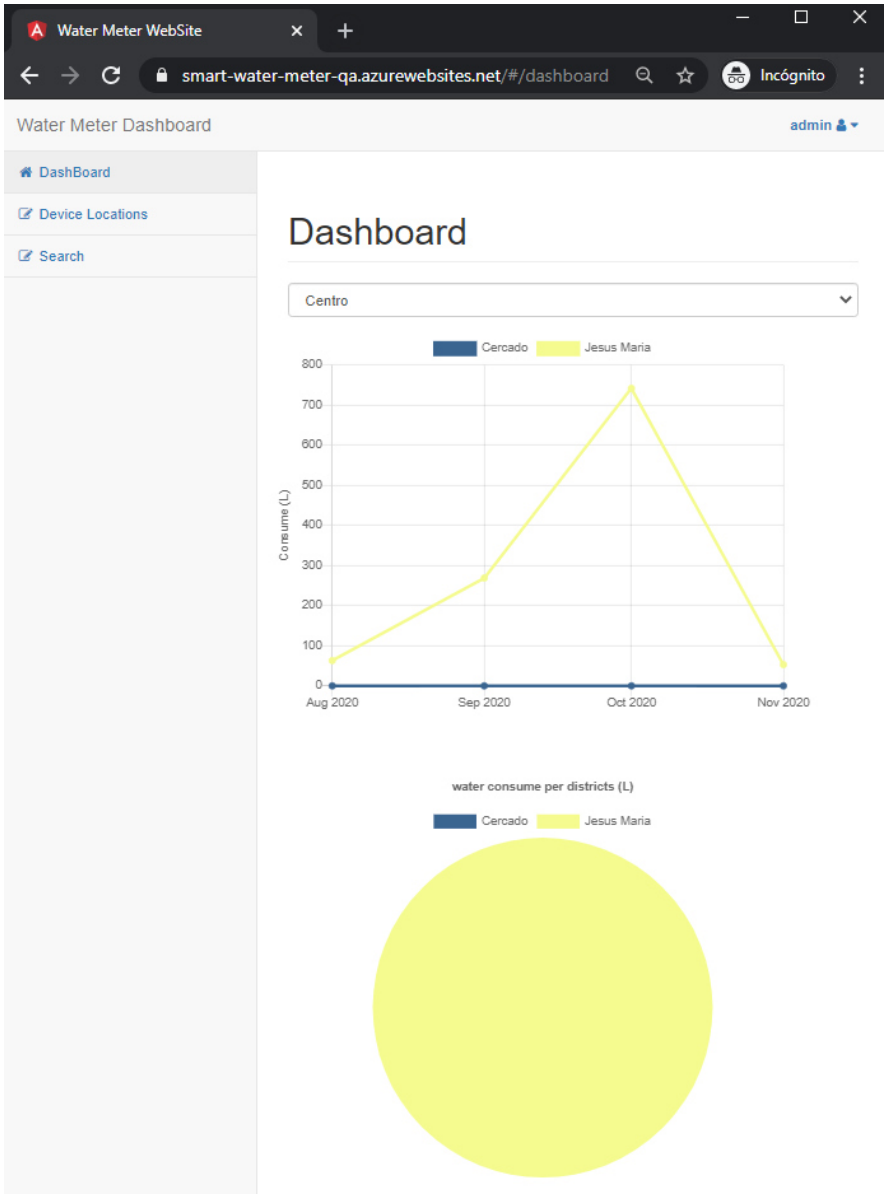


Figura 47. **Manual de usuario – Vista bienvenida**

Fuente. Elaboración propia

Después, si se desea buscar los dispositivos por distrito, se debe ir a la sección “Device Locations” y seleccionar el distrito del cual desea obtener información, tal como se enseña en la Figura 48.

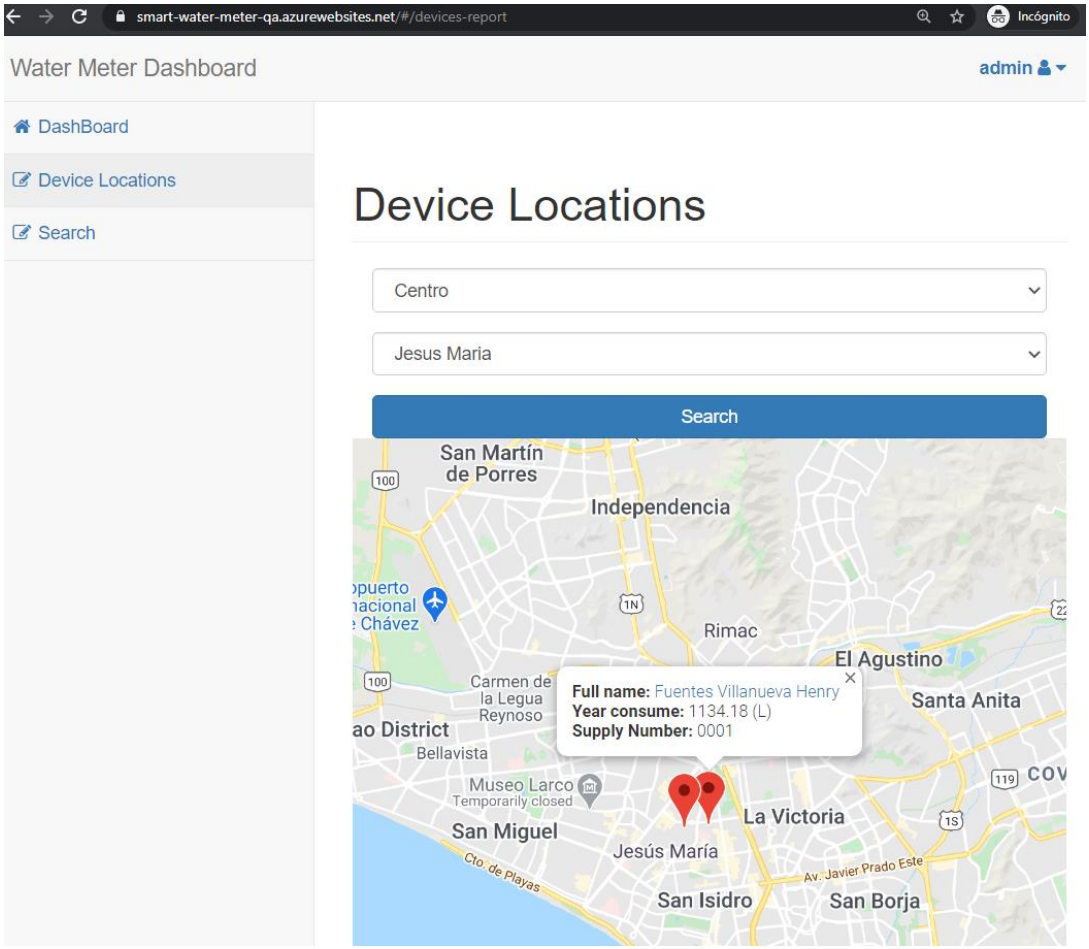


Figura 48. **Manual de usuario – Vista localización de dispositivos**
Fuente. Elaboración propia

En caso de que se desea ver el consumo histórico de los usuarios pertenecientes a una zona o distrito o DNI/RUC dentro de un lapso, se debe ir a la sección “Search” y aplicar los filtros, la cual es detalla en la Figura 49.

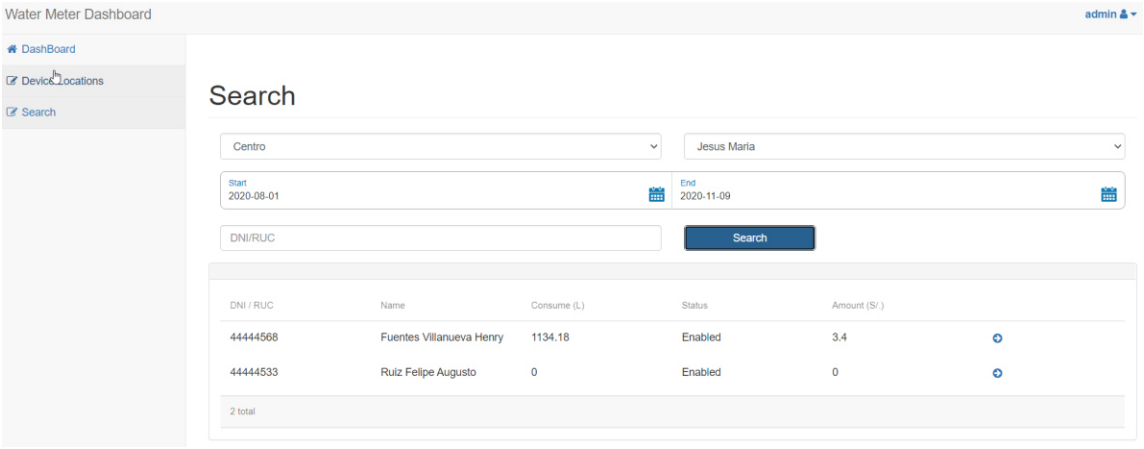


Figura 49. **Manual de usuario – Vista búsqueda de usuarios consumo**

Fuente. Elaboración propia

Para ver información detallada del usuario y sus respectivos medidores, se debe hacer clic en la fecha azul y será redireccionado a la página del usuario, tal como se ve en la Figura 50.

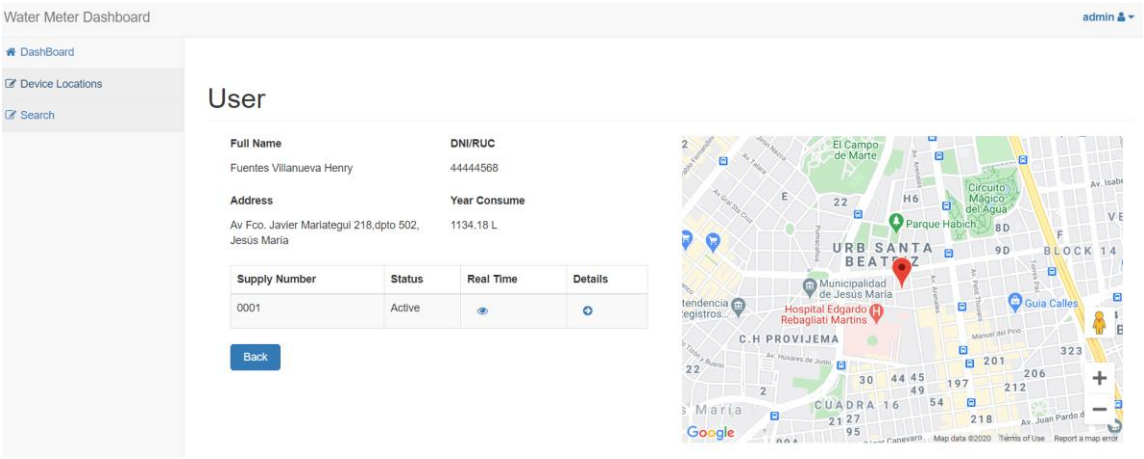


Figura 50: **Manual de usuario – Vista detalle usuario consumo**

Fuente. Elaboración propia

En caso de que se desea ver el consumo histórico del dispositivo de un usuario miembro, se hace clic en la flecha azul, la cual te llevará a una nueva página. Ahí se aplicará los filtros de fechas que se desea y se verá en un histograma la información solicitada al igual que se ve en la Figura 51. Cabe mencionar que, al

hacer clic en el botón “Export”, todos los consumos por hora son descargados en un archivo Excel(.xls).

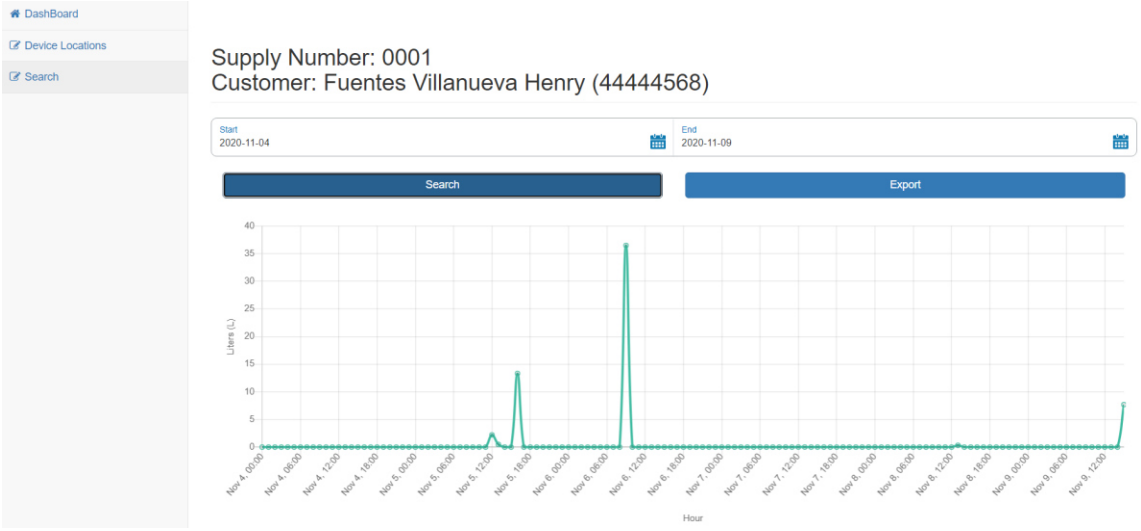


Figura 51: **Manual de usuario – Vista consumo histórico**

Fuente. Elaboración propia

Para ver en tiempo real el consumo de un usuario miembro, se debe ir a la sección “Real time” y se mostrará una línea de consumo(L) vs. tiempo, tal como se aprecia en la Figura 52.

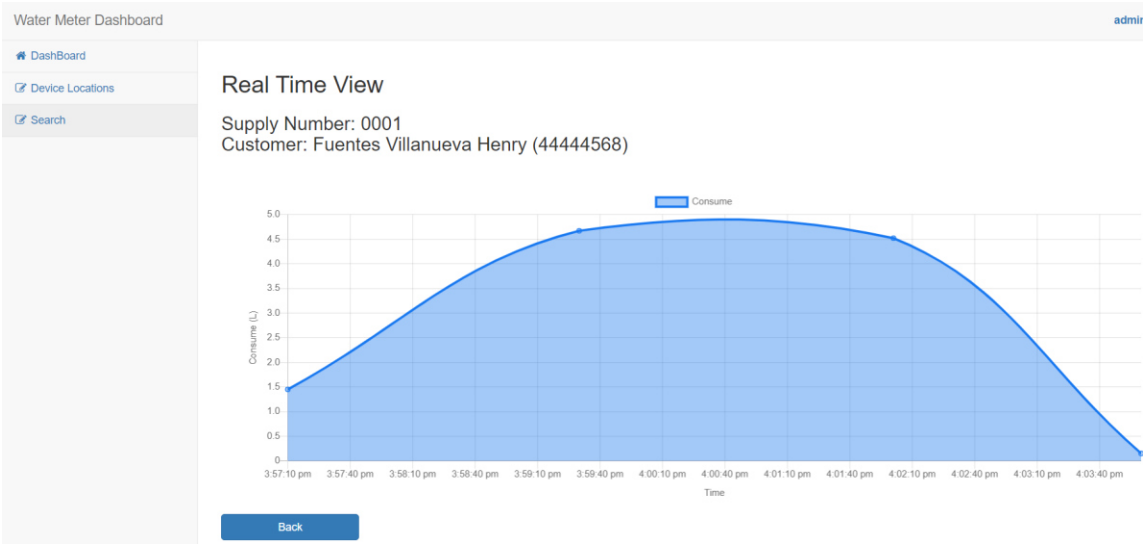


Figura 52: **Manual de usuario – Vista consumo tiempo real**

Fuente. Elaboración propia

En caso de detectarse una fuga, recibirá una notificación, tal como se aprecia en la Figura 53.

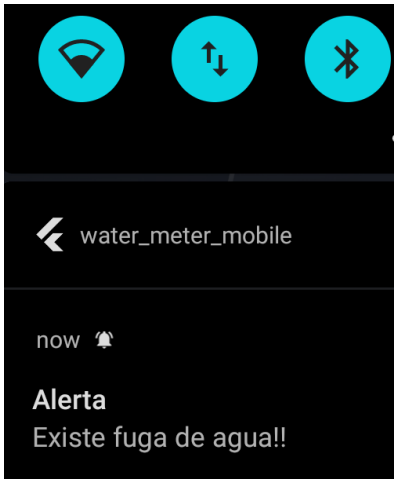


Figura 53: **Manual de usuario – Alerta en celular**
Fuente. Elaboración propia

Al presionar la notificación, el usuario será redireccionado a la aplicación móvil en donde se le consultará si está de acuerdo o no en que existe una fuga (ver Figura 54), esto se realiza con el objetivo de ir retroalimentando el algoritmo.

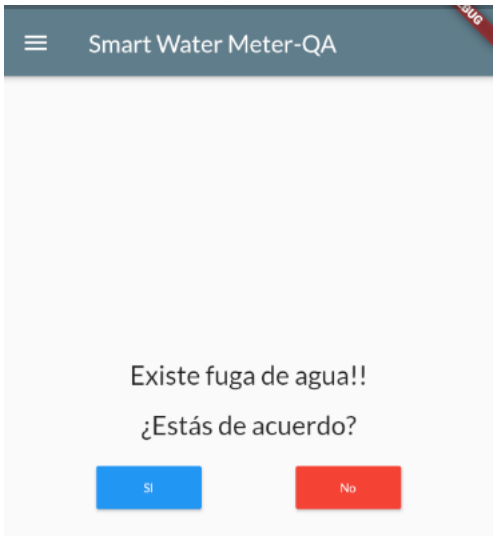


Figura 54: **Manual de usuario – Vista completa de alerta en celular**
Fuente. Elaboración propia

ANEXO C. Producción Científica sobre la investigación

ANEXO C.1. Carta de Aceptación de artículo

8/9/2020 Fwd: Your Submission EMAS-D-20-00644R1 - henry.fuentes@unmsm.edu.pe - Correo de UNMSM

De: **EMAS <** <em@editorialmanager.com>
Date: jue., 30 jul. 2020 a las 8:45
Subject: Your Submission EMAS-D-20-00644R1
To: David Mauricio <dmauricios@unmsm.edu.pe>

Dear Dr. Mauricio,

We are pleased to inform you that your manuscript, "SMART WATER CONSUMPTION MEASUREMENT SYSTEM FOR HOUSES USING IOT AND CLOUD COMPUTING", has been accepted for publication in Environmental Monitoring and Assessment.

You will receive an email from Springer in due course with regards to the following items:

1. Offprints

2. Colour figures

3. Open choice

4. Transfer of Copyright

Please remember to quote the manuscript number, EMAS-D-20-00644R1, whenever inquiring about your manuscript.

Thank you.

With best regards,
Journals Editorial Office
Springer

ANEXO C.2. Resumen del artículo

https://doi.org/10.1007/s10661-020-08535-4

Environmental Monitoring and Assessment (2020) 192:602
https://doi.org/10.1007/s10661-020-08535-4

Smart water consumption measurement system for houses using IoT and cloud computing

Henry Fuentes · David Mauricio

Received: 27 February 2020 / Accepted: 30 July 2020
© The Author(s) 2020

Abstract Presently, in several parts of the world, water consumption is not measured or visualized in real time, in addition, water leaks are not detected in time and with high precision, generating unnecessary waste of water. That is why this article presents the implementation of a smart water measurement consumption system under an architecture design, with high decoupling and integration of various technologies, which allows real-time visualizing the consumptions, in addition, a leak detection algorithm is proposed based on rules, historical context, and user location that manages to cover 10 possible water consumption scenarios between normal and anomalous consumption. The system allows data to be collected by a smart meter, which is preprocessed by a local server (Gateway) and sent to the Cloud from time to time to be analyzed by the leak detection algorithm and, simultaneously, be viewed on a web interface. The results show that the algorithm has 100% *Accuracy*, *Recall*, *Precision*, and *F1 score* to detect leaks, far better than other procedures, and a margin of error of 4.63% recorded by the amount of water consumed.

D. Mauricio (✉) · H. Fuentes
Faculty of System Engineering and Informatic, Universidad Nacional Mayor de San Marcos, Calle Germán Amézaga N° 375, Lima, Perú
e-mail: dmauricios@unmsm.edu.pe

H. Fuentes
e-mail: henry.fuentes@unmsm.edu.pe

Keywords Smart metering · Water consumption · IoT · Cloud computing

Introduction

Water is the most important natural resource for humans, so the World Health Organization (WHO) recommends that a person should consume an average of 100 L per day to meet all their needs (United Nations 2014), but in the main capitals of South America it exceeds what is recommended, for example, La Paz consumes 120 L, followed by Bogotá with 168 L, then Santiago with 200 L, Quito with 220 L and, finally, Lima with 250 L, whose surplus is equivalent to more than 77 thousand Olympic water pools per year (SUNASS 2017). On the other hand, in the USA inside a house, the daily consumption of water is approximately 138 gallons (522 L), being the flush of the toilet where water is used the most (24%), followed by the faucets (20%), showers (20%), clothes washer (16%), leaks (13%), bathtubs (3%), dishwasher (2%), and others (3%) (DeOreo et al. 2016). Due to this excessive water consumption, water treatment supply companies are aiming to raise public awareness about the responsible use of water.

One of the functions of companies that supply water is to identify how and where waste is generated, which, in general, can be due to people's neglect or leakage. Studies reveal that the amount of

Published online: 28 August 2020

ANEXO C.3. Ranking scimagojr