



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES**  
**Y REDES**

**“DISEÑO DE UNA RED DE COMUNICACIONES PARA LA MEDICIÓN Y**  
**CONTROL DEL CONSUMO DE AGUA POTABLE EN TIEMPO REAL**  
**PARA LA PARROQUIA YARUQUÍES PERTENECIENTE A LA CIUDAD**  
**DE RIOBAMBA”**

**TRABAJO DE TITULACIÓN: PROPUESTA TECNOLÓGICA**

**Para optar al Grado Académico de:**

**INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y REDES**

**AUTORES:**

**ISABEL JURITSA CASTILLO SOSA**

**ABEL SANTIAGO MURILLO PAGUAY**

**Riobamba-Ecuador**

**2017**



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES**  
**Y REDES**

**“DISEÑO DE UNA RED DE COMUNICACIONES PARA LA MEDICIÓN Y**  
**CONTROL DEL CONSUMO DE AGUA POTABLE EN TIEMPO REAL**  
**PARA LA PARROQUIA YARUQUÍES PERTENECIENTE A LA CIUDAD**  
**DE RIOBAMBA”**

**TRABAJO DE TITULACIÓN: PROPUESTA TECNOLÓGICA**

**Para optar al Grado Académico de:**

**INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y REDES**

**AUTORES: ISABEL JURITSA CASTILLO SOSA**

**ABEL SANTIAGO MURILLO PAGUAY**

**TUTOR: ING. JOSÉ ENRIQUE GUERRA SALAZAR**

**Riobamba-Ecuador**

**2017**

**@2017, Isabel Juritsa Castillo Sosa, Abel Santiago Murillo Paguay.**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de los Autores.

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES**

El Tribunal del Trabajo de Titulación certifica: “DISEÑO DE UNA RED DE COMUNICACIONES PARA LA MEDICIÓN Y CONTROL DEL CONSUMO DE AGUA POTABLE EN TIEMPO REAL PARA LA PARROQUIA YARUQUÍES PERTENECIENTE A LA CIUDAD DE RIOBAMBA”, de responsabilidad de Isabel Juritsa Castillo Sosa y Abel Santiago Murillo Paguay, ha sido minuciosamente revisado por los miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
Ing. Washington Luna		
<b>DECANO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA</b>	_____	_____
Ing. Franklin Moreno		
<b>DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES</b>	_____	_____
Ing. José Guerra		
<b>DIRECTOR DEL TRABAJO DE TITULACIÓN</b>	_____	_____
Ing. Wilson Zuñiga		
<b>MIEMBRO DEL TRIBUNAL</b>	_____	_____

Nosotros, Isabel Juritsa Castillo Sosa y Abel Santiago Murillo Paguay, somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

ISABEL JURITSA CASTILLO SOSA

ABEL SANTIAGO MURILLO PAGUAY

## **DEDICATORIA**

A Dios por ser la magia de toda esta aventura, a toda mi familia, a mis amigos, a mis tíos, a mis primos que en cualquier momento las pocas palabras que me transmitieron me sirvieron en esta etapa profesional y crecer como persona.

**Isabel**

A Dios por darme la fortaleza para cumplir mis metas, a mis padres Sergio y Rosa por el apoyo incondicional, por sus consejos y por todo ese amor y cariño que fueron fuente de inspiración para cumplir este objetivo. A mis hermanos Javier y Elizabeth que con sus palabras de aliento y compañía me enseñaron a no rendirme y mejorar cada día más como ser humano.

**Santiago**

## **AGRADECIMIENTO**

Agradeciendo a Dios por darme fortaleza, decisiones, perseverancia para seguir adelante, superar obstáculos durante toda esta etapa profesional, a mis padres Gary Castillo y Verónica Sosa por el apoyo moral y económico, y a mis hermanos Yandri, Kiabeth y Yurisbeth, por acompañarme y su apoyo incondicional.

**Isabel**

Agradezco a Dios por el darme la vida y conocimientos para poder seguir escalando y adquiriendo nuevas experiencias.

Un agradecimiento a toda mi familia y en especial a mis padres por ser mi ejemplo a seguir, llenarme de orgullo con sus acciones y ser el motor que me impulsa a llegar cada día más lejos. Agradezco a mis hermanos por su apoyo moral y económico que fueron fundamentales para culminar este trabajo de titulación.

Al Ing. Andrés Yépez funcionario de la EMAPAR por la apertura brindada y al Ing. José Guerra director del trabajo de titulación por el aporte de sus conocimientos, paciencia y consejos que permitieron la culminación del presente trabajo.

**Santiago**

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE GRÁFICOS.....	xi
RESUMEN.....	xiii
SUMMARY .....	xiv
INTRODUCCIÓN .....	1
<b>CAPÍTULO I</b>	
<b>1. MARCO TEÓRICO .....</b>	<b>7</b>
<b>1.1 Procedimiento de la lectura de consumo de agua potable .....</b>	<b>7</b>
<b>1.2 Red de comunicaciones .....</b>	<b>8</b>
<i>1.2.1 Redes inteligentes .....</i>	<i>9</i>
<i>1.2.2 Sistemas AMR .....</i>	<i>10</i>
<i>1.2.2.1 Tecnología existente.....</i>	<i>11</i>
<b>1.3 Factores que influyen en el diseño .....</b>	<b>14</b>
<b>1.4 Tecnologías de redes en telecomunicaciones.....</b>	<b>18</b>
<b>1.5 Dispositivos de medición de consumo de agua potable.....</b>	<b>19</b>
<i>1.5.1 Medidores inteligentes de agua potable.....</i>	<i>19</i>
<i>1.5.1.1 Tarjetas de desarrollo .....</i>	<i>20</i>
<i>1.5.1.2 Sensores de flujo de agua.....</i>	<i>21</i>
<i>1.5.1.3 Medidores convencionales para lectura remota .....</i>	<i>22</i>
<b>CAPÍTULO II</b>	
<b>2. MARCO METODOLÓGICO.....</b>	<b>23</b>
<b>2.1 Levantamiento de la información .....</b>	<b>23</b>
<b>2.2 Requerimientos de hardware en la infraestructura de comunicaciones .....</b>	<b>24</b>
<b>2.3 Selección de la tecnología de comunicación .....</b>	<b>24</b>
<b>2.4 Diseño de la infraestructura física de la red .....</b>	<b>26</b>
<i>2.4.1 Arquitectura general de la red.....</i>	<i>26</i>
<i>2.4.1.1 Medición del consumo de agua potable.....</i>	<i>27</i>
<i>2.4.1.2 Transmisión de datos .....</i>	<i>27</i>
<i>2.4.1.3 Control y monitorización de datos en tiempo real.....</i>	<i>28</i>
<i>2.4.2 Simulación de la red.....</i>	<i>28</i>
<i>2.4.2.1 Parámetros georreferenciales.....</i>	<i>29</i>
<i>2.4.2.2 Parámetros de la red.....</i>	<i>29</i>



2.4.2.3	<i>Simulación</i> .....	33
<b>2.5</b>	<b>Implementación del prototipo</b> .....	<b>36</b>
2.5.1	<i>Selección de los equipos que conforman la RCMAP</i> .....	36
2.5.1.1	<i>Equipo de medición para el consumo de agua potable</i> .....	36
2.5.1.2	<i>Transmisión de datos</i> .....	39
2.5.1.3	<i>Monitorización y control del consumo de agua potable</i> .....	39
2.5.2	<i>Esquema de conexión de los equipos para la medición y corte de agua potable</i> .....	40
2.5.3	<i>Requerimientos de software para el prototipo de la RCMAP</i> .....	41
2.5.4	<i>Descripción del software utilizado</i> .....	42
2.5.4.1	<i>Arduino IDE</i> .....	42
2.5.4.2	<i>Servidor Mosquitto Broker</i> .....	42
2.5.4.3	<i>Delphi 10.2 Tokyo Starter (Promo Edition)</i> .....	43
2.5.4.4	<i>Gestor de base de datos MySQL Community Edition</i> .....	43
2.5.5	<i>Diagrama de flujo de la RCMAP</i> .....	43
<b>CAPÍTULO III</b>		
<b>3.</b>	<b>ANÁLISIS Y RESULTADOS</b> .....	<b>50</b>
3.1	<b>Caracterización de la RCMAP</b> .....	<b>50</b>
3.1.1	<i>Caracterización de la etapa de medición de consumo de agua potable</i> .....	51
3.1.1.1	<i>Pruebas de medición</i> .....	51
3.1.2	<i>Caracterización de la etapa de transmisión de datos</i> .....	53
3.1.2.1	<i>Pruebas de conectividad</i> .....	53
3.1.2.2	<i>Pruebas de retardo en la transmisión de consumo del medidor inteligente</i> .....	58
3.1.2.3	<i>Pruebas de retardo medidor inteligente - servidor Broker MQTT</i> .....	59
3.1.2.4	<i>Pruebas de retardo servidor Broker MQTT - GUI</i> .....	60
3.1.2.5	<i>Pruebas de almacenamiento de datos en MySQL</i> .....	61
3.1.3	<i>Caracterización de la etapa de monitorización y control de agua potable</i> .....	61
3.1.3.1	<i>Pruebas de tiempo de respuesta</i> .....	62
3.2	<b>Pruebas con una plataforma en la nube para IoT</b> .....	<b>63</b>
3.3	<b>Costos de un medidor inteligente</b> .....	<b>65</b>
<b>CONCLUSIONES</b> .....		<b>68</b>
<b>RECOMENDACIONES</b> .....		<b>69</b>
<b>BIBLIOGRAFÍA</b>		
<b>ANEXOS</b>		

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b> Modelos de sensor de flujo de agua.....	22
<b>Tabla 1-2:</b> Comparación entre tecnología inalámbrica y cableada .....	25
<b>Tabla 2-2:</b> Comparación entre tecnologías inalámbricas .....	25
<b>Tabla 3-2:</b> Parámetros del modem .....	31
<b>Tabla 1-3:</b> Pruebas de medición del sensor de flujo de agua YF-S201.....	51
<b>Tabla 2-3:</b> Mediciones del sensor de flujo de agua calibrado .....	51
<b>Tabla 3-3:</b> Mediciones entre el medidor convencional y el sensor de flujo de agua .....	52
<b>Tabla 4-3:</b> Nivel de cobertura de router .....	54
<b>Tabla 5-3:</b> Promedio de valores de la red .....	56
<b>Tabla 6-3:</b> Capacidad, tiempo y ancho de banda .....	57
<b>Tabla 7-3:</b> Tiempo en que el medidor inteligente publica de los datos.....	58
<b>Tabla 8-3:</b> Tiempo de conexión /reconexión de agua potable .....	63
<b>Tabla 9-3:</b> Tiempo de retardo en que los valores de consumo de agua potable llegan a la plataforma Adafruit IO.....	64
<b>Tabla 10-3:</b> Tiempo de retardo en corte/reconexión de agua potable .....	65
<b>Tabla 11-3:</b> Precio de un medidor inteligente junto a herramientas de hardware y software que cumplen con la transmisión de datos en tiempo real.....	66

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b> Red de comunicaciones.....	8
<b>Figura 2-1:</b> Modelo simplificado para las comunicaciones (Diagrama de bloques).....	9
<b>Figura 3-1:</b> Red inteligente .....	9
<b>Figura 4-1:</b> Estructura de un AMR .....	11
<b>Figura 5-1:</b> Tecnología hand held (Walk-by) .....	12
<b>Figura 6-1:</b> Tecnología touch.....	12
<b>Figura 7-1:</b> Tecnología drive-by .....	13
<b>Figura 8-1:</b> Red de lectura fija .....	14
<b>Figura 9-1:</b> Tipos de redes de transmisión de datos según el enlace geográfico .....	15
<b>Figura 10-1:</b> Topología de red árbol.....	16
<b>Figura 11-1:</b> Topología de red estrella.....	16
<b>Figura 12-1:</b> Topología de red malla .....	17
<b>Figura 13-1:</b> Modelos de Arduino que trabajan con IoT .....	20
<b>Figura 14-1:</b> Modelo de Wi-Fi ESP8266.....	21
<b>Figura 1-2:</b> Arquitectura general de la red de comunicaciones .....	26
<b>Figura 2-2:</b> Diagrama de bloques de la etapa de medición del consumo de agua potable.....	27
<b>Figura 3-2:</b> Diagrama de bloques de la etapa de transmisión de datos .....	28
<b>Figura 4-2:</b> Diagrama de bloques de la etapa de control y monitorización de datos .....	28
<b>Figura 5-2:</b> Zona de estudio .....	29
<b>Figura 6-2:</b> Parámetros de tráfico .....	30
<b>Figura 7-2:</b> Transmisores existentes por cada usuario .....	31
<b>Figura 8-2:</b> Estudio de predicciones de cobertura (a) Por nivel de señal (b) Por rendimiento. .	32
<b>Figura 9-2:</b> Suscriptores que conforman la red.....	33
<b>Figura 10-2:</b> Simulación de conexión al modem por usuario .....	34
<b>Figura 11-2:</b> Diseño de la RCMAP en packet tracer 7.0 .....	35
<b>Figura 12-2:</b> Transmisión de datos en tiempo real.....	35
<b>Figura 13-2:</b> Sensor de flujo de agua YF-S201 .....	36
<b>Figura 14-2:</b> Medidor convencional de agua potable .....	37
<b>Figura 15-2:</b> Incorporación del sensor de flujo en el medidor convencional.....	37
<b>Figura 16-2:</b> Módulo ESP8266 NodeMCU v3 .....	38
<b>Figura 17-2:</b> Funcionamiento básico de un Access Point.....	39
<b>Figura 18-2:</b> Electroválvula o válvula de solenoide .....	40
<b>Figura 19-2:</b> Esquema de conexión para la medición del consumo de agua .....	40
<b>Figura 20-2:</b> Diagrama de flujo del monitoreo y control de consumo de agua potable.....	44

<b>Figura 21-2:</b> Conexión/Esperando conexión del servidor Broker MQTT .....	46
<b>Figura 22-2:</b> Funcionamiento de la función ISRCountPulse() .....	47
<b>Figura 23-2:</b> Conexión del módulo al AP, dirección IP, MAC y nivel de señal.....	48
<b>Figura 1-3:</b> Elementos que componen la (RCMAP).....	50
<b>Figura 2-3:</b> Medidor inteligente .....	53
<b>Figura 3-3:</b> Medición de jitter y pérdida de datos (a) Intervalo de 10s (b) Intervalo de 1s .....	55
<b>Figura 4-3:</b> Pruebas de ancho de banda .....	57
<b>Figura 5-3:</b> Conectividad del medidor inteligente al AP .....	58
<b>Figura 6-3:</b> Transmisión en tiempo real (a) Medidor inteligente (b) Interfaz gráfica del servidor Broker MQTT .....	59
<b>Figura 7-3:</b> Transmisión en tiempo real (a) Interfaz gráfica de servidor Broker MQTT (b) Interfaz Gráfica de Usuario (GUI) .....	60
<b>Figura 8-3:</b> Almacenamiento de valores de consumo de agua potable en MySQL .....	61
<b>Figura 9-3:</b> Interfaz de simulación (a) Corte (b) Reconexión de agua potable .....	62
<b>Figura 10-3:</b> Transmisión en tiempo real (a) Medidor inteligente (b) API de la plataforma IoT de Adafruit IO .....	64

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-2:</b> Usuario que disponen de servicio del líquido vital e Internet en la parroquia de Yaruquíes .....	24
----------------------------------------------------------------------------------------------------------------------	----

## ÍNDICE DE ANEXOS

**Anexo A.** Sensor de Flujo YF-S201

**Anexo B.** Medidor convencional para lectura remota

**Anexo C.** Sensor digital de efecto hall A3144

**Anexo D.** Modulo ESP8266x

**Anexo E.** Router Huawei HG532 – ADSL

**Anexo F.** Modem Huawei echo life HG8245 – Fibra óptica

**Anexo G.** Modem Linksys

**Anexo H.** Programación de Modulo ESP8266 NodeMCU

**Anexo I.** Código para diseñar la GUI en Delphi Starter 10.2

**Anexo J.** Cotización de medidor inteligente inalámbrico de agua potable

**Anexo K.** Costo de licencia Delphi Tokio Professional

## RESUMEN

El objetivo del trabajo de titulación fue diseñar una red de comunicaciones para la medición y control del consumo de agua potable en tiempo real para la parroquia Yaruquíes perteneciente a la ciudad de Riobamba. Para el diseño de la red se investigó sobre la tecnología existente, factores y dispositivos que se ajusten a los requerimientos de la red, realizando una simulación de la misma, que permitió verificar la factibilidad de implementar un prototipo de la red. Consta de tres etapas: medición de consumo de agua potable, transmisión de datos, control y monitorización en tiempo real, para lo cual se implementó un medidor inteligente partiendo del medidor tradicional incorporándole un sensor de efecto hall y un módulo Wi-Fi ESP8266, que por medio de comunicación inalámbrica a través del punto de acceso (AP) publica los datos hacia el servidor *Broker* MQTT que es el intermediario de la comunicación. Se trabajó con el paquete de desarrollo Delphi Starter para crear una Interfaz Gráfica de Usuario donde se visualiza el consumo en tiempo real y el gestor de base de datos MySQL que almacena la información. La eficiencia de la red se determinó mediante simulaciones y pruebas reales con respecto a la conectividad, donde el dispositivo de medición presenta errores de lectura de 0.06 litros, cobertura máxima de -80 dBm y ancho banda mayor a 900 Kbps para una conexión estable con el AP, el tiempo de transmisión y control de datos en toda la red no debe ser mayor a 2s. Se concluye que la red de comunicación implementada junto con el medidor inteligente, mejora tiempos y procesos de operación, con relación a la lectura convencional. Se recomienda que para que los medidores inteligentes no pierdan conexión a la red se implementen enrutadores inalámbricos adicionales, en especial en zonas donde la intensidad de señal sea más baja.

**Palabras Claves:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <RED DE COMPUTADORES>, <COMUNICACIONES INALÁMBRICAS>, <WIFI (TECNOLOGÍA)>, <MEDICIÓN INTELIGENTE>, <BROKER MQTT (HARDWARE-SOFTWARE)>, <MEDICIÓN AGUA POTABLE>, <CONTROL DE AGUA POTABLE>.

## SUMMARY

The aim of the degree work was to design a communications network for the measurement and control of drinking water consumption in real time for the parish of Yaruquies belonging to the city of Riobamba. For the design of the network, the existing technology, factors, and devices that adjust to the requirements of the network were investigated, performing a simulation of the same, which allowed verifying the feasibility of implementing a prototype of the network. It consists of three stages: drinking water consumption measurement, data transmission, real-time monitoring control, for which an intelligent meter was implemented starting from the traditional meter incorporating a hall effect sensor and a Wi-Fi module ESP8266, which through wireless communication through the access point (AP) it publishes the data to the Broker MQTT server that is the intermediary of the communication. We worked with the Delphi Starter development package to create a User Graph interface where the consumption in real time is visualized and the MySQL database manager that stores the information. The efficiency of the network was determined by simulations and real tests with respect to connectivity, where the measurement device has reading errors of 0.06 liters, maximum coverage of 80 dBm and a bandwidth greater than 900 Kbps for a stable connection with the AP, the time of transmission and control of data throughout the network should not be greater than 2s. It is concluded that the communication network implemented together with the smart meter improves times and operation processes, in relation to conventional reading. It is recommended that for the smart meters, no connection to the network be lost, additional wireless routers should be implemented, especially in areas where the signal strength is lower.

Keywords: < TECHNOLOGY AND SCIENCE OF ENGINEERING >, <COMPUTER NETWORK>, <WIRELESS COMMUNICATIONS>, <WI FI (TECHNOLOGY)>, <INTELLIGENT MEASUREMENT>, <BROKER MQTT> <HARDWARE-SOFTWARE> < MEASUREMENT OF DRINKING WATER > <DRINKING WATER CONTROL>



## INTRODUCCIÓN

Actualmente en la ciudad de Riobamba existe una gran demanda del líquido vital debido al crecimiento poblacional, en las zonas urbanas y rurales, uno de los desafíos que tiene la EMPRESA MUNICIPAL DE AGUA POTABLE Y ALCANTARILLADO DE RIOBAMBA es realizar la lectura, digitalización y control del consumo de la red de medidores. Estos procesos se realizan de forma manual con personal técnico que recoge la información de cada medidor una vez al mes, esto no permite tener la información actualizada y precisa, ocasionando problemas tanto para el usuario como para la empresa.

Para lograrlo se procedió a investigar sobre el avance de las tecnologías de medición remota, además se realizó un análisis de la infraestructura actual para con ello diseñar una red de comunicaciones y un prototipo de medidor inteligente que permita la integración del medidor convencional y sensores para la medición en tiempo real; por lo que, se utilizó un módulo que admita la comunicación inalámbrica Wi-Fi para utilizar la infraestructura actual del proveedor de servicios de internet pública y se evaluó la factibilidad de diseñar un software de gestión.

Para el desarrollo de la red, se realizan respectivas investigaciones y análisis sobre la arquitectura de redes de comunicaciones inalámbricas existentes, modos y protocolos de comunicación entre los dispositivos finales y dispositivos detectores de flujo de agua e intermediadores para la transmisión y almacenamiento de la información en tiempo real, para posteriormente realizar la respectiva simulación, implementación del prototipo, evaluación de resultados y documentación.

Con este proyecto se aporta a la comunidad estudiantil y empresas proveedoras del servicio de agua potable que se interesen en utilizar nuevas tecnologías para el monitoreo en tiempo real de dispositivos a través de internet, integrando la infraestructura que poseen actualmente, para lograr un ahorro importante del líquido vital, ahorrar recursos y mantenerse a la vanguardia del avance tecnológico a nivel mundial.

## ANTECEDENTES

El rápido avance de las tecnologías de información ha hecho que en los últimos años el uso de internet aumente de manera exponencial, pasando de un modelo de sociedad estático a uno de información. El IoT (Internet of Things, por sus siglas en inglés), define que todos los objetos o parámetros pueden ser conectados a la red de internet y generar información útil en tiempo real, como el consumo de energía eléctrica o agua potable, calidad del aire, humedad, etc.

Uno de los factores más importantes en esta sociedad es el ahorro, calidad, cantidad de consumo de agua potable; el desarrollo y globalización de la tecnología permite nuevos y mejores sistemas de medición, control y supervisión del mismo, ya que ha ido sufriendo muchas modificaciones debido a que su demanda es creciente permitiendo a las empresas de agua potable participar de manera activa en la administración eficiente del líquido vital.

A nivel mundial existen varias empresas dedicadas al desarrollo de un sistema de red de lectura automática de primera calidad, mediante una red de comunicaciones con diferentes tecnologías, haciendo uso del internet para que sus clientes puedan obtener datos en tiempo real; entre las más importantes están:

“Arad technologies Ltd. (Yokneam, Israel), desarrolla sistemas de lectura automáticas de áreas táctiles, portátiles, montadas en vehículos y fijas.”

“Arad Metering Technologies Wuhan Co Ltd. (Wuhan, China), es una filial del Grupo Arad, cuyo objetivo es prestar servicios al grupo en el territorio chino.”

“Master Meter Inc. (Texas, Estados Unidos), ensambla, comercializa y mantiene sistemas para cubrir las necesidades de medición de agua a lo largo y ancho de Estados Unidos y Canadá, ofreciendo una gran variedad de tamaños, configuraciones, accesorios y medidores para aplicaciones especiales.”

“Cicasa, 1953 (México), se especializa en el desarrollo, la fabricación y la distribución de equipos de medición de agua.”

“Contazara se fundó 1984 y ha desarrollado exclusivas soluciones electrónicas de medición de agua, durante años de actividad, logrando asegurarse una cuota significativa en el mercado español desarrollando a la par una red propia para la medición automática.”

En el Ecuador existen varios estudios relacionados con la implementación de prototipos de medición digital de agua potable, que transmiten solo a la vivienda del usuario, también existen estudios acerca de una infraestructura de medición avanzada de energía eléctrica.

- “Marcelo Quispe (2008, p. 1) realiza un estudio de telemetría o tele-lectura para medidores de agua potable en el centro del DMQ.”
- “Jácome Omayra y León José, estudiantes de la Universidad de las Fuerzas Armadas del Ecuador (2014, pp. xvi) diseñaron y construyeron un sistema de medición inteligente para la gestión de la energía eléctrica y medición del consumo de agua potable en los hogares, que incorpora tecnología inalámbrica y capacidad de visualización de la energía y agua potable que se consume.”
- “Wilmer Urbina estudiante de la Pontificia Universidad Católica del Ecuador (2014) realiza un estudio y diseño de un sistema de medición remota utilizando sensores inalámbricos bajo el estándar 802.15.4, para los servicios de agua potable y energía eléctrica, implementa un prototipo con una red de sensores realizando el estudio con las diferentes tecnologías existentes.

Sin embargo, se desconoce que exista un diseño de una infraestructura de red de comunicaciones para alguna empresa públicas proveedora de servicios de energía eléctrica, agua potable o gas. En la Empresa Municipal de Agua Potable y Alcantarillado de la ciudad de Riobamba (EMAPAR), los registros de los valores de consumo de agua potable son obtenidos de forma manual por personal que recorre toda la ciudad de manera sectorizada, los datos obtenidos son digitalizados de la misma manera y colocados en el sistema online donde los ciudadanos consultan el valor mensual de su planilla de agua potable. “Los usuarios tienen un mes para cancelar sus planillas sin recargo, pasado los 30 días se aplicarán intereses y si no cancelan por dos meses se suspende el servicio y se aplican valores adicionales, como dos dólares por reconexión y un porcentaje por gastos administrativos.

Actualmente en muchos países existen medidores inteligentes de agua potable que han ido opacando a los medidores de agua convencional, generando datos más precisos, detecta fugas, robo de agua, reduce el tiempo de toma de estado de datos, disminuyendo los gastos operacionales y a la par se ha creado infraestructuras de redes, resultando fundamental el diseño de una red de comunicaciones de medición y control del consumo de agua potable para la ciudad de Riobamba, al momento la EP-EMAPAR impulsa la generación de proyectos de digitalización y transmisión de la información de su red de medidores de agua potable.

## **FORMULACIÓN DEL PROBLEMA**

¿Es posible diseñar una red de comunicaciones para la medición y control del consumo de agua potable en tiempo real para la parroquia Yaruquíes perteneciente a la ciudad de Riobamba?

## **SISTEMATIZACIÓN DEL PROBLEMA**

¿Por qué se debe reestructurar la forma actual de medición y lectura de los medidores convencionales de agua potable?

¿Cómo se puede realizar el diseño de una red de comunicaciones para la medición y control del consumo de agua potable en tiempo real para la parroquia Yaruquíes?

¿Se podrá simular el diseño de la red?

¿Qué software nos ayudaría a simular una red de comunicaciones?

¿Es posible la implementación de un prototipo que permita la medición y control de los medidores de agua potable en tiempo real?

¿Qué características técnicas y normas deben cumplir la red de comunicaciones, los componentes y equipos necesarios para el diseño?

## **JUSTIFICACIÓN TEÓRICA**

Existen un problema común, complejo y costoso al que se enfrentan las empresas proveedoras de servicios básicos, (agua y energía eléctrica) desde el punto de vista computacional y económico, que es la de tomar mediciones manuales del consumo generado por los clientes tomando en cuenta su dispersión geográfica. (Aldas, 2015, pp. 2)

La Empresa Municipal de Agua Potable y Alcantarillado de la Ciudad de Riobamba (EP EMAPAR), para realizar la lectura de aproximadamente 35500 medidores de agua potable distribuidos por toda la ciudad, se utiliza alrededor de 3000 hojas impresas mensuales donde un total de 6 trabajadores toman las medidas de forma manual, implicando mayores gastos.

Según entrevistas realizadas al personal de la empresa se sabe que el sistema de lectura y control de consumo se realiza de manera mensual, en ocasiones existe pérdida o errores en los datos adquiridos y no existe una red que permita enviar de manera digital y en tiempo real esta información hacia la empresa.

La presente investigación se enfoca en el estudio y diseño de una red de comunicaciones para la medición y control del consumo de agua potable en tiempo real para la Ciudad de Riobamba, parroquia Yaruquíes con la respectiva implementación de un prototipo, permitiendo a la EP EMAPAR y usuarios obtener datos en tiempo real y poder realizar un control de consumo mediante plataformas Online. Para su diseño, seleccionaremos los elementos más adecuados, que se acople a las necesidades del proyecto y la infraestructura actual de la empresa.

Finalmente se contribuirá con soluciones tecnológicas e innovadoras tanto a la EP EMAPAR, como a empresas públicas o privadas que provean el servicio de agua potable o servicios básicos.

## **JUSTIFICACIÓN APLICATIVA**

Con el avance tecnológico y el apogeo del internet de las cosas, las personas exigen un nuevo estilo de vida donde el tema de las ciudades inteligentes, escenario donde se administra, gestiona, controla los recursos públicos como luz, agua, teléfono es de gran importancia para una mejora significativa en su estilo de vida.

El sistema de medición inteligente, junto con una red centralizada de comunicaciones tienen como objetivo la automatización del ciclo comercial de lectura, suspensión y reconexión del suministro de energía eléctrica e inclusive de otros servicios (como gas y agua) a través de un sistema de comunicación bidireccional que asocia diversos elementos hardware (contadores inteligentes, integradores, concentradores de datos, entre otros) y software para el manejo de los datos y sistemas de gestión de usuarios. Para implementar los sistemas de comunicación es posible utilizar diversas tecnologías: RF, PLC, GPRS, banda ancha (Ethernet, Wi-Fi, WiMax, ADSL), Satelital y Fibra Óptica, e inclusive combinaciones de ellas.

## **OBJETIVOS**

### ***OBJETIVO GENERAL***

Diseñar una red de comunicaciones para la medición y control del consumo de agua potable en tiempo real para la parroquia Yaruquíes perteneciente a la ciudad de Riobamba.

### ***OBJETIVOS ESPECÍFICOS***

- Analizar la infraestructura actual de medición y control de consumo de agua potable en la parroquia Yaruquíes perteneciente a la ciudad de Riobamba.
- Investigar y seleccionar los equipos y dispositivos que cumplan con los requerimientos de la red, realizando un análisis comparativo.
- Diseñar una red de comunicaciones que permita la medición y control de consumo de medidores de agua en tiempo real para la parroquia Yaruquíes.
- Simular los principales nodos que conforman la red de comunicaciones.
- Implementar un prototipo de la red de comunicaciones para la medición y control de consumo en tiempo real.

El presente documento, resume el diseño de una red de comunicaciones para la medición y control del consumo de agua potable en tiempo real para la parroquia Yaruquíes perteneciente a la ciudad de Riobamba, el cual presenta 3 capítulos, donde el primer capítulo menciona la parte teórica, el segundo la metodológica y en el tercero se reflejan los resultados.

## CAPÍTULO I

### 1. MARCO TEÓRICO

El presente capítulo aborda la parte teórica donde se desarrolla el proceso de lectura de los medidores convencionales que actualmente se encuentran instalados en la parroquia Yaruquíes, como se forma una red inteligente de comunicaciones, tecnologías y equipos que existen para su construcción.

#### 1.1 Procedimiento de la lectura de consumo de agua potable

La Empresa Municipal de Agua Potable y Alcantarillado de la ciudad de Riobamba (EP EMAPAR), tiene la responsabilidad de brindar un buen servicio a todos los usuarios que habitan en su área de atención. Actualmente el proceso de toma de lectura de consumo de agua potable se realiza manualmente, con la intervención del hombre. Para lograr una buena obtención de la información sobre el consumo de agua potable de cada cliente, se tiene la información personal del mismo (código único o referencia) y ubicación geográfica (código geográfico de ubicación o ruta, parroquia, sector y número de casa), para posteriormente localizarlos rápida y ordenadamente.

##### - **Recorrido del lector de consumo de agua potable**

Para realizar la lectura cada mes de aproximadamente 35500 medidores de agua potable distribuidos en la ciudad de Riobamba, se necesitan de 6 lectores y alrededor de 3000 hojas impresas para la anotación del consumo generado por el usuario. En la adquisición de una correcta lectura de consumo de agua potable, el lector, verifica primero que el número del medidor impreso sea el mismo de la vivienda y concluido su trabajo, inmediatamente regresa a la empresa a dejar todo el registro obtenido durante el recorrido de los lugares que se le han establecido.

##### - **Digitalización y entrega de datos**

Una vez que los datos son corroborados por el departamento encargado, éstos son digitalizados manualmente en el computador por el personal establecido.

## - Facturación

Al terminar la digitalización de datos, se procede a la facturación, que consiste en la verificación de los datos digitalizados.

Ya que éste proceso es completamente manual, van a existir errores humanos, tales como:

- Interpretación errónea de la lectura del medidor
- Consumo de agua potable mal anotados
- La persona que digitaliza interpreta incorrectamente los datos generados por el lector o escribe incorrectamente los datos

En conclusión, por los errores humanos, se presentan reclamos de los clientes que se sienten perjudicados por altos consumos. Finalmente, cada usuario se presentará en ventanilla para el pago del servicio básico consumido.

## 1.2 Red de comunicaciones

Diseñadas para diferentes servicios con diferentes tecnologías conectando numerosos ordenadores para intercambiar o consumir información. La principal de ella fue ARPANET que estuvo en disposición de interconectar los primeros ordenadores en 1972 y que incorporó inmediatamente la primera aplicación de “gran uso”, el correo electrónico. Posteriormente, se convirtió en Internet con la normalización de protocolos y la incorporación de numerosas herramientas de *software*, permitiendo conectar millones de hogares. (Caldevilla, 2013, p.56). Según Stallings William (2004, p.597), la red de comunicación, es un sistema que proporciona un servicio de transferencia de datos entre estaciones conectadas a la red. En la Figura 1-1 se muestra una red de comunicaciones y se puede apreciar sus tres elementos principales que la intervienen, como transmisor, receptor y el medio de comunicación.

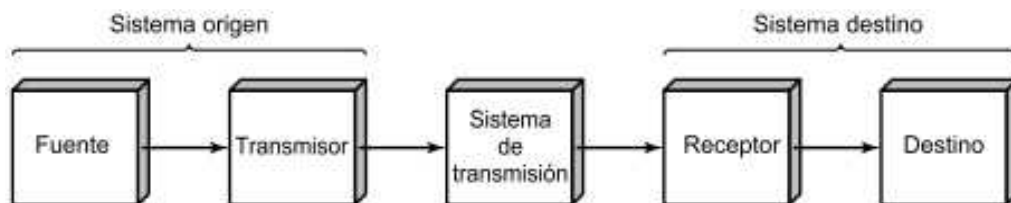


**Figura 1-1:** Red de comunicaciones

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017



Para el intercambio de información, es necesario que el sistema de comunicaciones sea el correcto, tomando en cuenta las interferencias o fallos que pueden mostrarse desde el origen hacia el destino (detección y corrección de errores), que los datos transmitidos no saturen el destino (control de flujo) y gestionar la red de manera que se pueda planificar crecimientos a futuro. (Stallings, 2004, p.12). En la Figura 2-1, se puede apreciar un diagrama de bloques, indicando como se comunica cada elemento de un sistema de comunicaciones.



**Figura 2-1:** Modelo simplificado para las comunicaciones (Diagrama de bloques)

Fuente: Stallings, 2004, p.10.

### 1.2.1 Redes inteligentes

Actualmente, una de las redes de comunicaciones, son las redes inteligentes o *Smart Grid*, como se muestra en la Figura 3-1, transportando información de forma bidireccional por medio de sistemas digitales y aplicaciones inteligentes, a través del internet, siendo ventajoso o desventajoso para las industrias de servicios públicos, proveedores de tecnología y automatización. (Barroso, 2011, p.30)



**Figura 3-1:** Red inteligente

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Surgió a partir de una infraestructura de medición avanzada o AMI (*Advanced Metering Infrastructure, por sus siglas en inglés*) y una lectura de medición automática o AMR (*Automatic Meter Reading, por sus siglas en inglés*), construyendo una protección confiable, segura y acertada para la medición neta de consumo de los servicios básicos, donde monitores y sensores serán omnipresentes en la red comunicación inteligente.

Para conseguir la comunicación entre los equipos, además de una arquitectura basada en redes de área amplia y local, se requiere algo más que solo cables, necesariamente se establecen tecnologías avanzadas de control y gestión, teniendo resultados flexibles, fiables y con la seguridad de satisfacer las necesidades de la economía global. (Barroso, 2011, pp. 75–76). Por lo tanto, a nivel funcional, *Smart Grid* se compone de diferentes niveles como: medición, distribución y consumo inteligente.

### **1.2.2 Sistemas AMR**

Se centra en la recopilación automática de consumo, diagnóstico y estado de datos adquiridos de dispositivos de medición de agua o energía que se transfieren hacia una base de datos central para análisis, solución de problemas y facturación. En la Figura 4-1, se exponen los componentes básicos de un sistema AMR independiente de la tecnología que se utilice, a continuación, se detallan los mismos.

**Contadores o Medidores:** Dispositivos que se encargan de transmitir las medidas de consumo, se clasifican en:

- Medidor convencional: Lee y los datos adquiridos los convierte en una secuencia digital de bits, que estarán disponibles en un microcontrolador (sensor infrarrojo, sensor radio frecuencia y sensor de pulso) para transmisión.
- Medidor inteligente: Es un sistema donde el microcontrolador viene programado para transmitir o recibir dependiendo el caso.

**Concentradores de datos:** Equipo que almacena los datos adquiridos de los contadores y los envía a la estación de control.

**Sistemas o infraestructura de comunicación:** Se componen por diferentes componentes como, los enlaces (cables o inalámbrico) que permiten la conexión de todos los equipos y la tecnología que usan los equipos para transmitir. Capaces de soportar tráfico bidireccional, de ser escalables, flexibles y sobre todo fiables.

**Centro de control:** Es el lugar donde se reservará, controlará y gestionará la información del consumo medido por los contadores.



**Figura 4-1:** Estructura de un AMR

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los sistemas AMR pueden ser implementados en redes de comunicación a través de la red de telefonía (cable o inalámbrico), radio frecuencia (RF, Radio Frequency) o línea de potencia de transmisión (PLC, Power Line Transmission). (Huang Qi et al., 2015: pp.148-149)

#### 1.2.2.1 Tecnología existente

Para realizar la lectura de medidores a distancia existen diferentes métodos de comunicación y medios automáticos, presentados a continuación.

##### - Métodos de comunicación

La comunicación es categorizada en 2 tipos: por cables y de forma inalámbrica, donde las empresas pueden establecer sus propias redes o redes ya establecidas para transferir los datos que necesitan. Gracias a tecnologías como GPRS (*General Packet Radio Service, por sus siglas en inglés*) muchas empresas de servicios públicos están llevando a cabo pruebas pilotos para la implementación de sistemas AMR. (Montesdeoca, 2012, p. 4)

##### - Medios automáticos

En algunos casos la demanda, se recupera a través de medios automáticos tales como: drive-by vehicle o walk-by que fueron los primeros sistemas AMR para los clientes residenciales, comerciales o industriales, a continuación, se presentan alguno de ellos.

- Sistemas walk-by

Basados en la transmisión de señal por radio, es decir, una vez captada la información de los medidores, ésta se envía por medio de una señal de radio y llega a un colector de información o también llamado hand held como se muestra en la Figura 5-1, para analizar los datos adquiridos posteriormente. Los equipos que se adicionan al contador mecánico van equipados con baterías que duran de 10 a 15 años. Puede ser utilizado sin el uso de tecnología AMR. (Montesdeoca, 2012, p. 6)



**Figura 5-1:** Tecnología hand held (Walk-by)

**Realizado por:** CASTILLO Isabel & MURILLO Santiago. 2017

- Tecnología touch

Existen contadores eléctricos y ópticos en donde el empleado se acerca a los contadores y se conecta a una interfaz del medidor, mediante un dispositivo de recolección de datos, como se puede observar en la Figura 6-1, los mismos son descargados y de manera manual los almacena en la base de datos. (Montesdeoca, 2012, p. 5)



**Figura 6-1:** Tecnología touch

**Fuente:** (Montesdeoca, 2012)

Es una manera similar a la lectura actual de los medidores convencionales, con la única diferencia que utilizan un dispositivo electrónico para leer los datos, por lo cual también es conocida como “in situ”, que en español significa “en el sitio”.

- Sistemas a radio drive-by

En la Figura 7-1, se observa el funcionamiento del sistema, donde un dispositivo de lectura instalado en un vehículo adquiere los datos del medidor automáticamente, cuando se hace el recorrido por el área de servicio sin ir por un orden en particular. A menudo este sistema incluye funciones de navegación y la cartografía proporcionada por un Sistema de Posicionamiento Global o GPS (*Global Positioning System por sus siglas en inglés*) y el software de geolocalización. (Montesdeoca, 2012, pp. 6–7)



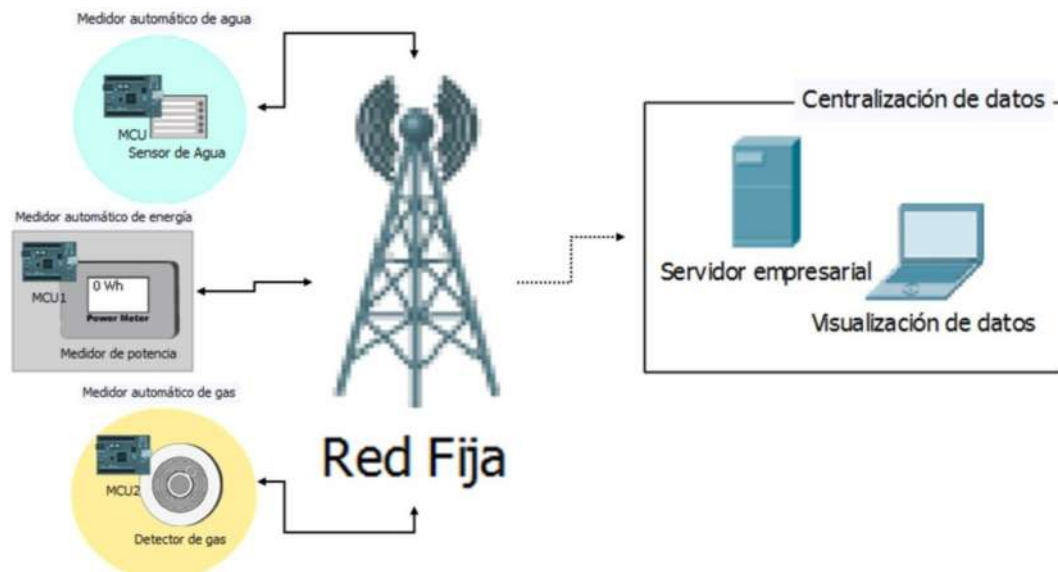
**Figura 7-1:** Tecnología drive-by

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

- Sistema con red fija

Se consideran como una evolución del sistema “walk-by”, en el que el lector ha sido sustituido por un equipo fijo en el edificio (o en la zona) que recoge las lecturas y las transmite mediante tecnología GPRS o por la red telefónica al servidor informático del gestor. La mayoría de los fabricantes establecen un paso intermedio entre los contadores y el equipo que se comunica con el servidor. Estos equipos intermedios (concentradores, repetidores, colectores) suelen ser equipos de coste relativamente bajo encargados de recopilar los datos de un número determinado de contadores. (Montesdeoca, 2012, pp. 7–8)

En la Figura 8-1, se muestra la red de lectura fija, que es un esquema flexible para el control y monitorización de agua potable, porque no requiere de personal para la toma de lectura, además si se incorpora un sistema inalámbrico en la adquisición de datos de los contadores, sería menos costosa la inversión, utilizando la red ya existente para subir los datos a la nube haciendo referencia al Internet de las cosas o visualizarlos en la empresa con la utilización de su propio servidor. Es ideal para prototipos que trabajan con sensores ya que no requieren ancho de banda muy grande para enviar datos por internet.



**Figura 8-1:** Red de lectura fija

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Algunos de estos sistemas son proporcionados por diferentes empresas, que desarrollan su propio kit que incorpora medidor inteligente, concentrador de datos y software de gestión, generando procesos de lectura eficientes y una calidad de datos óptima. Varios han implementado sistemas AMR, como: EEUU como país promotor con cerca de 193 proyectos enfocados a esta área, Italia como primer país con medición inteligente, Colombia, Suecia, entre otros

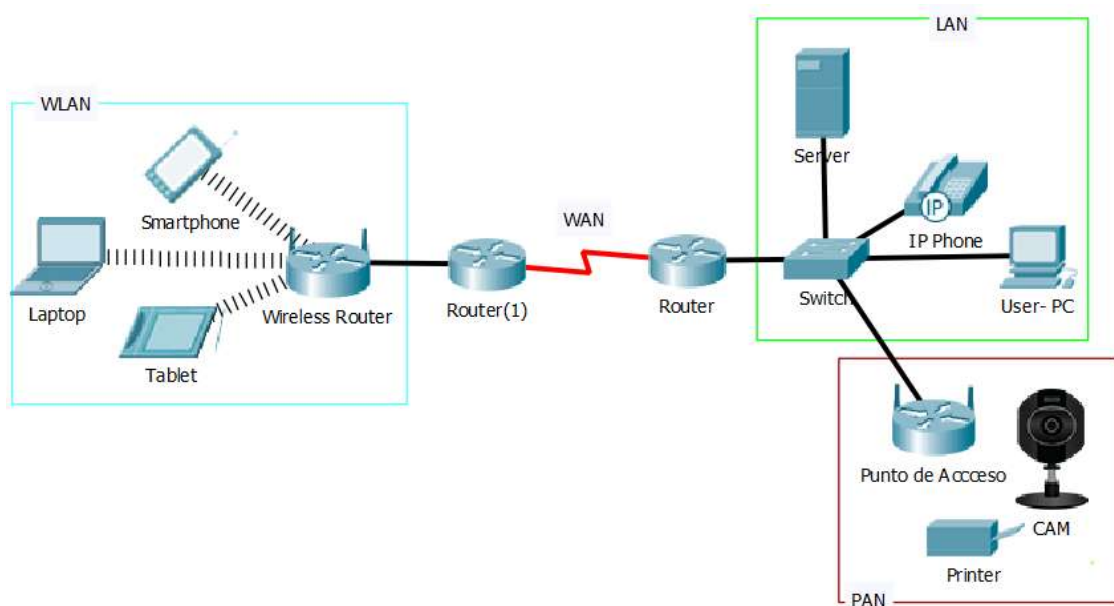
### 1.3 Factores que influyen en el diseño

Permiten garantizar un funcionamiento estable y correcto de la red, generando satisfacciones en los usuarios finales y la empresa que provee el servicio. Los factores que más se deben considerar son los siguientes:

**Rango de red:** Para resolver el problema de que los dispositivos desean conectarse entre ellos y transferir información, va a depender de las propiedades o la implementación a realizar. A continuación, las redes más utilizadas según su área de cobertura.

- Red de área amplia o WAN (*Wide Area Network, por sus siglas en inglés*), son redes que cubren un área extensa, mediante conmutación de circuitos o conmutación de paquetes, de tal manera que los dispositivos se interconecten y así alcanzar su destino. (Stallings, 2004, p.15)
- Red de área local o LAN (*Local Area Network, por sus siglas en inglés*), pueden ser cableadas, inalámbricas (WLAN, Wireless LAN) o una combinación de ambas, que permiten la comunicación de dispositivos a una distancia amplia. Cubren áreas pequeñas (0.01 – 1 Km) y transmiten datos hasta 54 Mbps. (Stallings, 2004, p. 17)
- Red de área local o PAN (*Personal Area Network, por sus siglas en inglés*), generalmente son inalámbricas que cubren áreas de 10 m, pero su tasa de datos varía dependiendo de los sensores que se utilicen.

En la Figura 9-1, se puede observar como una red se enlaza con cualquiera de otro tipo según su enlace geográfico, garantizando una transmisión de datos entre varias ciudades o regiones, según sea el caso.

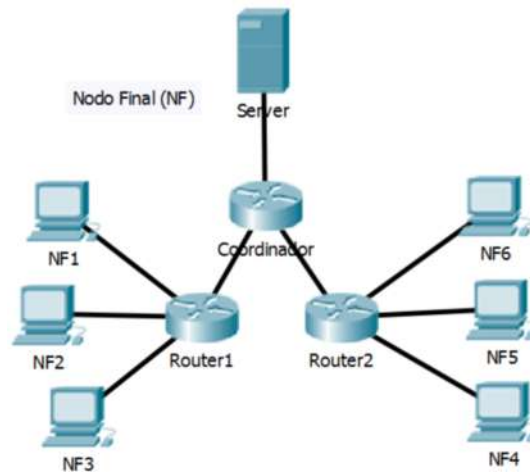


**Figura 9-1:** Tipos de redes de transmisión de datos según el enlace geográfico

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

**Topologías de red:** Las principales son: estrella, árbol y malla, también existen la híbrida que es una combinación entre éstas.

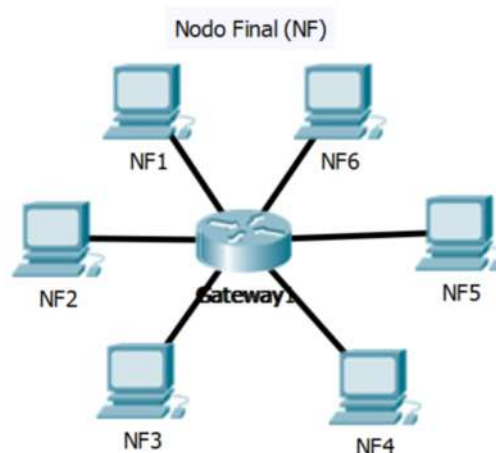
- Árbol, en la Figura 10-1, se observa la composición de la topología, donde existe un coordinador que controla la red permitiendo que los datos de los nodos finales lleguen a su destino, por lo que necesita enlazarse con los enrutadores y éstos con los nodos finales.



**Figura 10-1:** Topología de red árbol

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

- Estrella, es un sistema donde los nodos finales tienen comunicación directa con el Gateway que hace de puente con la red cableada, es decir, que la información enviada solo da un salto, como se muestra en la Figura 11-1. Esta topología es la que menor consumo energético tiene, pero en la información solo puede transmitirse por un solo camino y en caso de que el nodo se caiga los datos se pierden.

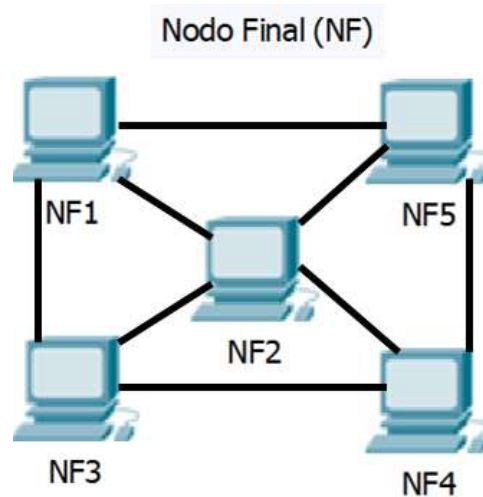


**Figura 11-1:** Topología de red estrella

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017



- Malla, en la Figura 12-1, se muestra cómo opera la topología malla, donde los nodos finales pueden comportarse como nodos enrutadores, por lo cual existe comunicación entre ellos y con la puerta de enlace. Sin embargo, estas redes son complejas de diseñar y tienen mayor retraso enrutando la información desde un nodo remoto.



**Figura 12-1:** Topología de red malla

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Esta topología es altamente tolerante ante fallos, ya que los nodos tienen diversos caminos para enrutar sus paquetes, por lo tanto, si un nodo enrutador se cae, la red seguirá disponible para el paso de información.

- Híbrida, Puede ser alguna combinación de las anteriores topologías: árbol, malla y estrella. Al realizar la combinación entre estas topologías la red es más robusta, ya que presenta diferentes maneras de enrutar los paquetes y se aprovecha cada característica de ellas.

**Tolerancia a fallos:** Cuando los nodos se averían o se bloquean por falta de energía o condiciones ambientales, no debe interferir en el funcionamiento global de la red.

**Escalabilidad:** El diseño futuro de la red deberá operar de la misma manera sin importar el número de nodos con los que trabaje y así no sature un futuro inmediato.

**Costos de producción:** La red comunicaciones desde el punto de vista económico, deberá ser más rentable que una red tradicional, de tal manera que se pueda justificar.

**Entorno:** Para la adaptación del diseño de una red inalámbrica se debe contar con las condiciones climáticas o físicas del lugar a implementarla, de tal manera que los nodos trabajen y transmitan los más preciso posible sus datos.

**Consumo y obtención de energía:** El consumo de energía de los nodos de una red va a depender del uso que se les dé, pero en especial de la comunicación que estos realizan, ya sea transmitiendo o recibiendo señales y de esta manera se deduce el número de bytes a transmitir.

#### 1.4 Tecnologías de redes en telecomunicaciones

Existen diversas tecnologías inalámbricas y cableadas para la transmisión de datos y según la necesidad requerida se selecciona la ideal para el diseño de la red de Comunicaciones

- Línea de potencia de comunicación o PLC

Es un método donde los datos son transmitidos sobre líneas eléctricas, ofreciendo y garantizando servicios de telecomunicaciones, gracias a su infraestructura y cobertura extensa. Para utilizar esta tecnología como acceso a Internet, se necesita instalar un modem de cabecera, los que se comunican con el proveedor de servicio de Internet mediante conexiones fibra óptica. (García et al., 2014, p. 329)

- ADSL (*Asymmetric Digital Subscriber Line, por sus siglas en inglés*)

Los datos son transmitidos por cobre, permitiendo velocidades de transferencia asimétricas. Además, requiere de la instalación de un modem para la inclusión del Internet. Presenta ventajas competitivas frente a la tecnología PLC, debido a la provisión de servicio con banda ancha en mercados masivos y pequeñas empresas. (García et al., 2014, p. 330)

- Fibra óptica

Es una tecnología que tuvo el mayor avance en las telecomunicaciones, debido a su velocidad de transmisión en comparación con las redes de cobre, además de que no garantizan una transmisión segura por su impedancia que en cierta longitud atenúan la señal. (www.anixter.com/en\_us/resources/literature/techbriefs/innovations-in-optical-fiber.html, 2017)

- Zigbee

Es un conjunto de protocolos de alto nivel de comunicación de corto alcance, bajo consumo de energía, baja velocidad de datos, bajo costos de implementación. Opera en la banda libre de 2,4 GHz, 858MHz en Europa y 915MHz en Estados Unidos para la comunicación con el resto de dispositivos, basada en el estándar de la IEEE, el IEEE 802.15.4. (Akshay Kanwar, 2012, pp. 67-68)

- Wi-Fi

Es la tecnología más utilizada en redes inalámbricas basado en el estándar IEEE 802.11, que trabaja en frecuencia de 2,4 GHz-5GHz, generando interoperabilidad entre los dispositivos, facilidad de instalación y movilidad. (Moreno Martín, 2015, p. 7)

- Sistema celular

Son redes celulares de grandes extensiones geográficas haciendo reutilización de canales de radio. Se usa especialmente para contadores aislados, donde los datos de los equipos finales son recolectados por un único contador, los almacena, elabora y transmite a un servidor gestor mediante GSM/GPRS, limitándose a la recogida de pulsos por tiempo (normalmente una vez al día). (Echenique y Rivero, 2012)

## **1.5 Dispositivos de medición de consumo de agua potable**

Las variables que provienen de sensores son adquiridas a través de tarjetas de desarrollo que procesan la información y generan registros de los mismos en tiempo real.

### ***1.5.1 Medidores inteligentes de agua potable***

Fueron diseñados para evitar los errores humanos de lectura de consumo de agua, hurto de agua, detección de fugas y así evitar gastos operacionales. Todo medidor inteligente que mide el consumo de agua de un usuario, envía la información adquirida instantáneamente a un receptor y ésta es ingresada a un software de lectura para ser analizada, gestionada y facturada. Se conforman por una tarjeta inteligente bidireccional y un software que administra el proceso. Por lo tanto, actualmente en el mercado existen un sin número de medidores convencionales o sensores de

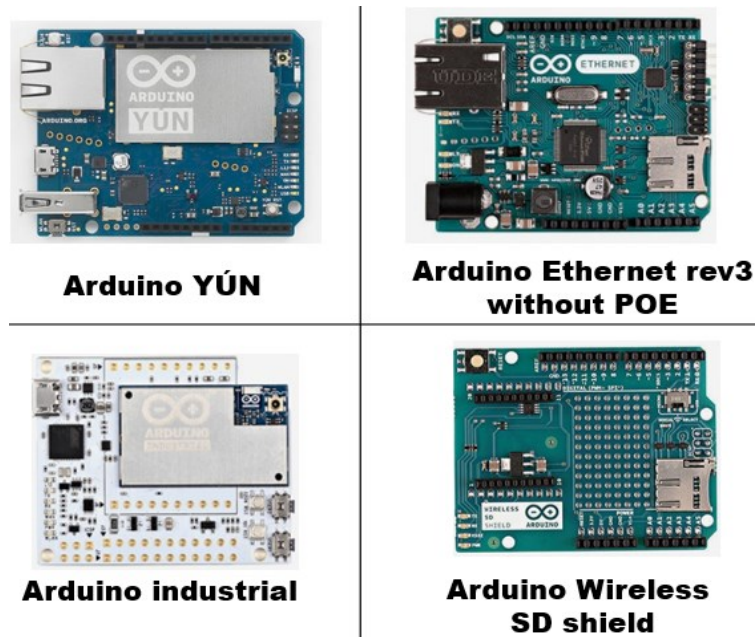
flujo que, junto a tarjetas de desarrollo, trabajan inteligentemente. A continuación, se presentan los elementos que pueden formar un medidor inteligente.

### 1.5.1.1 Tarjetas de desarrollo

Existen diversas tarjetas que se basan en AMR, como Arduino, Raspberry Pi, módulo ESP8266, entre otras, garantizando una lectura y transmisión remota de datos, pero por razones de tamaño la Raspberry Pi, se descartó. A continuación, se describen dos tarjetas de desarrollo que actualmente son las más conocidas en el mercado y se acoplan al proyecto.

#### - Arduino

Es una plataforma electrónica de código abierto que puede leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida (activar un motor, encender un LED y publicar datos en línea). Es el cerebro de miles de proyectos, desde objetos cotidianos hasta complejos instrumentos científicos. Las placas Arduino tienen un alto precio, debido a sus componentes y diferentes aplicaciones que se pueden construir. Es una tarjeta multiplataforma (se ejecuta en Windows, Macintosh OS X, y Linux), con entorno de programación simple y claro, y software de código abierto y extensible. En la Figura 13-1, se aprecian las diferentes versiones de Arduino que trabajan con el Internet de las cosas.

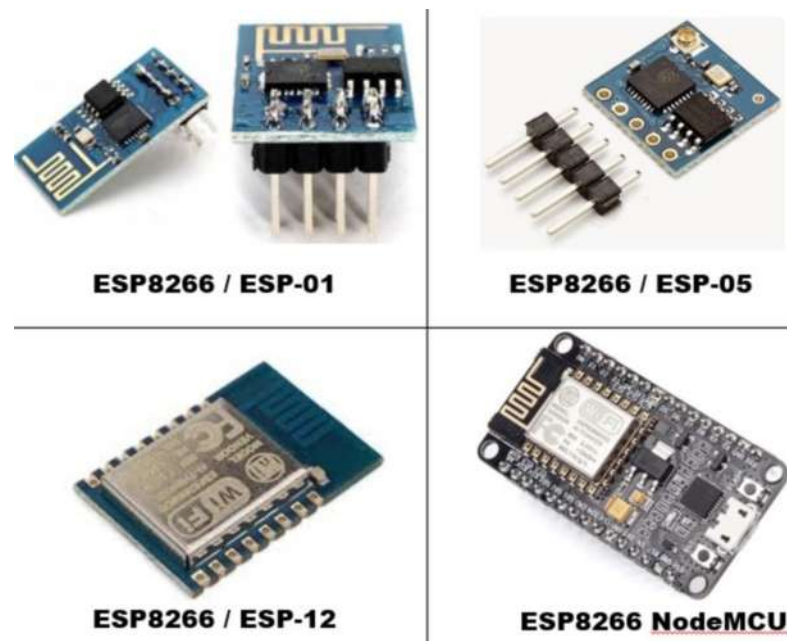


**Figura 13-1:** Modelos de Arduino que trabajan con IoT

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

- Módulo ESP8266

Esta tarjeta es ideal para proyectos de IoT (*Internet de las cosas, por sus siglas en inglés*), que tiene integrado un chip Wi-Fi de bajo costo con pila TCP / IP completa y capacidad MCU (*Micro Controller Unit, por sus siglas en inglés*). Tiene un firmware de Arduino que permitir realizar diferentes eventos. Existen diferentes versiones presentadas en la Figura 14-1, las cuales son utilizadas dependiendo las necesidades. Muchas de estas han sido descartadas por los programadores debido a que la actualización de sus versiones posee mejores requerimientos.



**Figura 14-1:** Modelo de Wi-Fi ESP8266

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

### 1.5.1.2 Sensores de flujo de agua

A nivel doméstico existen sensores de flujo que se usan para medir el consumo de agua, estableciendo diferentes tipos de caudales y conexiones según la medida de la rosca de la tubería de agua, como se muestran en la Tabla 1-1.

El sensor tiene un rotor y una de sus paletas un imán, que al girar causa un campo magnético en conjunto con un sensor de efecto hall, emitiendo un pulso y enviándolo por uno de los cables del sensor. Posteriormente los pulsos son convertidos a flujo mediante tarjetas de desarrollo ya sea por las placas de Arduino u otro controlador que se desee usar.

**Tabla 1-1:** Modelos de sensor de flujo de agua

<b>Modelo</b>	<b>Conexión (plug)</b>	<b>Caudal (L/min)</b>	<b>Sensor de flujo</b>
<b>YF-S201</b>	$\frac{1}{2}$	1-30	
<b>FS300A</b>	$\frac{3}{4}$	1-60	
<b>FS400A</b>	1	1-60	

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

### *1.5.1.3 Medidores convencionales para lectura remota*

En la actualidad existen diversas empresas que fabrican medidores convencionales capaces de adaptarse a una infraestructura de medición inteligente, incorporando componentes adicionales para su proceso de lectura automatizada. Una de las empresas líderes a nivel mundial innovadora en tecnología de medición es ZENNER, que ofrece una cartera de productos completa de contadores de agua convencionales, de calefacción y sistemas AMR, con una construcción ampliable con los accesorios adecuados y tecnología de sistemas más vanguardista.

## CAPÍTULO II

### 2. MARCO METODOLÓGICO

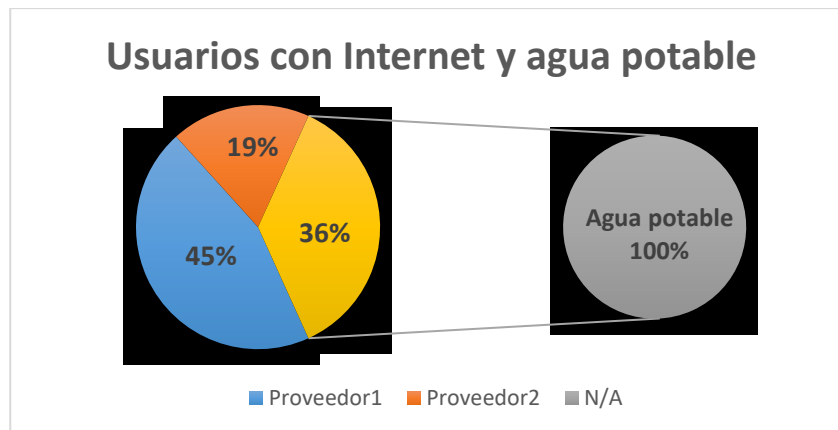
En el presente capítulo se detalla el diseño de hardware del prototipo de una red de comunicaciones para la medición y control del consumo de agua potable en tiempo real (RMCAP) para la parroquia Yaruquíes perteneciente a la ciudad de Riobamba, tomando en cuenta las características primordiales y factores para ejecutar la misma. Durante el proceso se inició con una etapa de levantamiento de la información (número de usuarios y servicios disponibles) para determinar los requerimientos técnicos que debe cumplir la red, selección de la tecnología a utilizar, diseño físico de la red, arquitectura general del sistema y simulación de la red, justificando de esta manera la selección de dispositivos que lo integran.

#### 2.1 Levantamiento de la información

En la actualidad existen diferentes maneras inteligentes de tomar los datos del consumo de agua potable registrados por los medidores, pero la más empleada para contadores aislados es la utilización de la red GPRS, aprovechando el acceso y la infraestructura de la red ya existente, se tiene una inversión mínima con respecto a la creación de su propia red de transmisión de datos. Debido a esta forma de implementar y a la ubicación de los medidores convencionales de agua potable, se ha decidido trabajar con la infraestructura de red de los proveedores de servicio de Internet ya existente en la parroquia Yaruquíes para la transmisión de datos en tiempo real.

La técnica de recopilación de información necesaria para el trabajo, fue de entrevista directa, que mediante el diálogo entre el investigador y el entrevistado se obtuvo información completamente actualizada. La técnica realizada permitió conocer el número de usuarios que constan con servicio de agua potable y servicio de Internet.

La parroquia de Yaruquíes, es la más completa en servicios básicos e internet, por lo tanto, el estudio se establecerá en todas sus zonas. Existe alrededor de 1136 usuarios que tienen el servicio de agua potable que representa el 100% de usuarios que tiene la parroquia, pero 722 abonados tienen un contrato de servicio de Internet que representa el 64%, como se muestra en el Gráfico 1-2, los cuales se encuentran ubicado en distintos puntos geográficos.



**Gráfico 1-2:** Usuario que disponen de servicio del líquido vital e Internet en la parroquia de Yaruquíes

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

## 2.2 Requerimientos de *hardware* en la infraestructura de comunicaciones

Por medio del levantamiento de información se establecieron parámetros que tiene que cumplir la red de comunicación para el monitoreo y control de agua potable en la parroquia de Yaruquíes, los que son presentados a continuación:

- Dispositivos con tecnología de bajo costo, facilidad de instalación y escalable
- Conectividad inmediata entre todos los dispositivos
- Sensores que minimicen errores en las lecturas del consumo de agua.
- Comunicación bidireccional para controles potenciales (cierre/apertura de la electroválvula)
- Tecnología para aplicaciones IoT
- Proporcionar información en tiempo real

## 2.3 Selección de la tecnología de comunicación

La selección de la tecnología para transmitir datos, además de estar en función del levantamiento de información y las características de las tecnologías planteadas en el Capítulo I, está regida a la relación calidad/precio. La mayoría de redes inteligentes están diseñadas a base de redes híbridas, porque no solamente usan un medio de comunicación.



En la Tabla 1-2, se presentan las principales diferencias entre la tecnología inalámbrica y cableada, donde la interferencia de señales afecta bastante en la tecnología inalámbrica, por lo tanto, si se implementa una red de este tipo, esta parte debe ser muy considerada, para evitar pérdida de datos.

**Tabla 1-2:** Comparación entre tecnología inalámbrica y cableada

<b>Aspecto/Tecnología</b>	<b>Cableada</b>	<b>Inalámbrica</b>
<b>Tasa de envío</b>	Gbps	Mbps
<b>Movilidad</b>	No	Si
<b>Acceso al medio</b>	No	Si
<b>Interferencia</b>	Poca	Muy susceptible
<b>Complejidad y costo de implementación</b>	Alta	Baja
<b>Facilidad de reconfiguración</b>	Baja	alta

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Debido a que se va a trabajar con la red proveedora de servicio de Internet, cabe mencionar que actualmente estas empresas están migrando de cobre a fibra óptica, obteniendo mayor velocidad de datos y eficiencia de los mismos. La transmisión de datos desde los medidores hasta los concentradores será de manera inalámbrica. A fin de escoger la mejor tecnología, se hizo una comparación entre ellas, la cual se muestra en la Tabla 2-2, en cuanto a costo de instalación, alcance, frecuencia de trabajo, velocidad, duración de batería y seguridad.

**Tabla 2-2:** Comparación entre tecnologías inalámbricas

<b>Aspecto /Tecnología</b>	<b>WI-FI (IEEE 802.11x)</b>	<b>Zigbee (IEEE 802.15.4)</b>
<b>Frec. de trabajo</b>	2.4GHz, 5GHz	868 MHz (EU), 900-928 MHz, 2.4GHz
<b>Alcance máximo (m)</b>	30-100	10-30
<b>Duración de batería</b>	12 – 48 horas	100-1000 días
<b>Costo / Módulo (\$)</b>	11	9
<b>Velocidad de datos</b>	11Mbps/54Mbps	250Kbps
<b>Seguridad</b>	WPA / WPA2 Personal, TLS, SSL	AES (128 bits)
<b>Aplicación</b>	WLAN	Control/Monitoreo (WPAN)

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

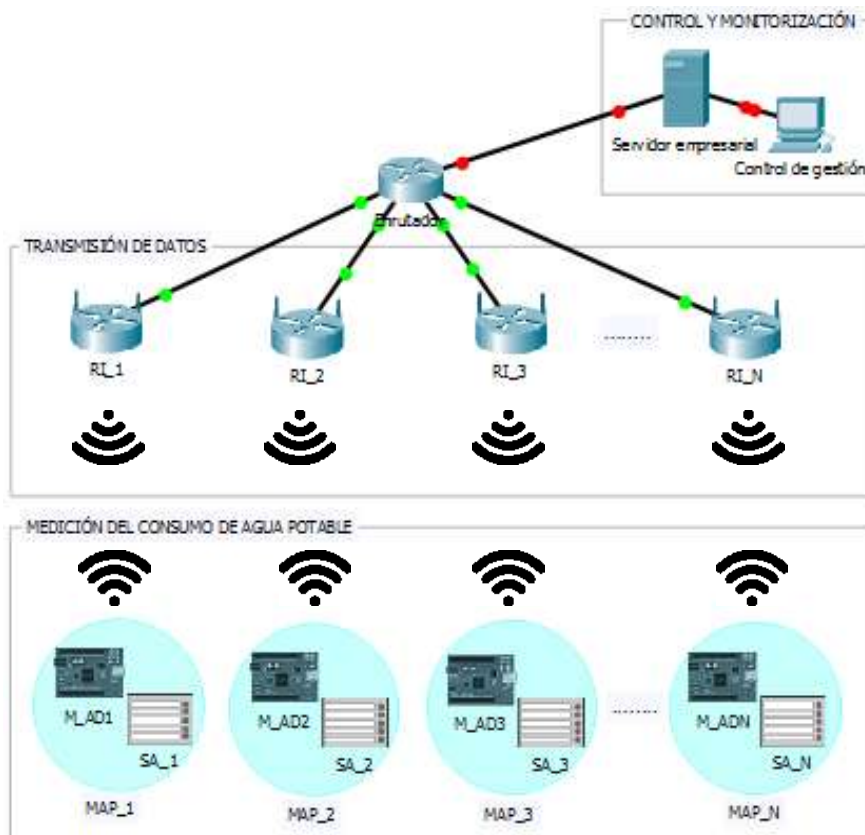
Gracias a las prestaciones técnicas de la tecnología inalámbrica Wi-Fi (IEEE 802.11), su relación calidad/precio y alcance máximo es justa para los requerimientos planteados.

## 2.4 Diseño de la infraestructura física de la red

Para la transmisión de los datos desde el modem hacia el servidor empresarial a través de la red mundial más conocido como Internet, no existe problema alguno, ya que la empresa prestadora de servicio de internet o empresa indirecta tiene su propia red ya establecida, y solamente se genera una tarifa mensual por el costo de transferencia de datos. Sin embargo, el transporte de datos desde el medidor o sensor hasta el modem es por el medio inalámbrico. La topología para conectar los dispositivos finales es estrella, ya que, a un AP pueden estar conectados varios nodos finales.

### 2.4.1 Arquitectura general de la red

En la Figura 1-2, se muestra la arquitectura de la RCMAP, que consta de tres etapas fundamentales: medición del consumo de agua potable, transmisión de datos y control y monitorización en tiempo real, donde cada etapa su presenta un diagrama de bloques, que indica las funciones que llevan a cabo cada uno de sus componentes.



**Figura 1-2:** Arquitectura general de la red de comunicaciones

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

#### 2.4.1.1 Medición del consumo de agua potable

Consiste en la detección y adquisición de datos para determinar el consumo de agua potable de un usuario específico. El diagrama de bloques presentado en la Figura 2-2, muestra el bloque de alimentación de energía para los componentes que se utilizan en el bloque de dispositivos detectores de agua potable y tarjeta de desarrollo.



**Figura 2-2:** Diagrama de bloques de la etapa de medición del consumo de agua potable

**Realizado por:** CASTILLO Isabel & MURILLO Santiago. 2017

Los dispositivos detectores realizan el conteo de la variable física (agua potable) y la tarjeta de desarrollo procede a transformarlos en valores de metros cúbicos ( $m^3$ ) acorde a como registra actualmente el personal del agua potable de Riobamba, informando la cantidad de agua que pasa por el detector.

#### 2.4.1.2 Transmisión de datos

Esta etapa considera el envío de la información desde el medidor de agua potable hasta la visualización de los mismos en tiempo real. En la Figura 3-2, se muestra el diagrama de bloques correspondiente a la etapa de transmisión de datos, indicando que se empieza a transmitir una vez que se tiene un identificador único, permitiendo atravesar por el sistema de comunicación sin que su información se dirija hacia otro usuario.

Primero la información del consumo de agua potable por cada usuario, pasa de forma inalámbrica (Wi-Fi) por el modem con servicio a Internet, la empresa proveedora del servicio guiará los datos mediante cobre o fibra pasando por un enrutador, hasta una base de datos de la empresa de agua potable, que posteriormente los visualizará.



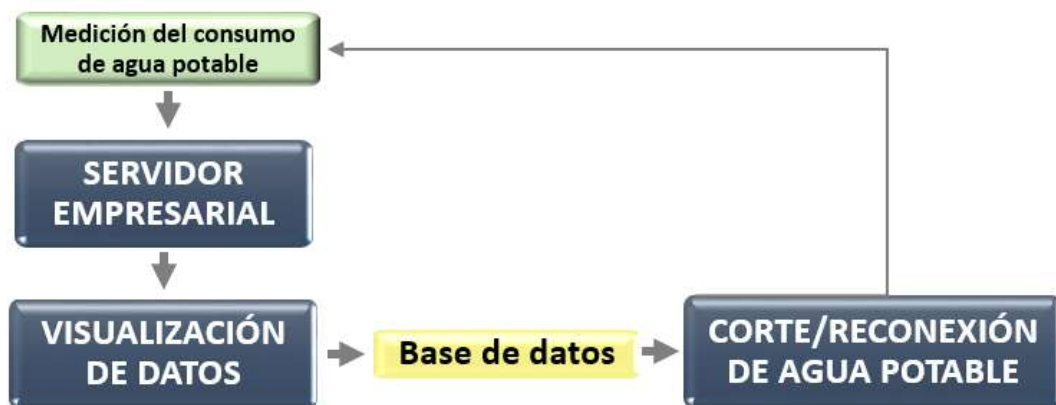
**Figura 3-2:** Diagrama de bloques de la etapa de transmisión de datos

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

#### 2.4.1.3 Control y monitorización de datos en tiempo real

Los datos una vez recibidos, son visualizados en tiempo real por la empresa proveedora de servicio de agua potable, donde el personal encargado tendrá el deber de controlar y monitorear la información del consumo de agua potable de cada usuario.

En la Figura 4-2, se presenta su diagrama de bloques, que mediante el bloque de visualización de datos se observa el consumo de agua potable de cada usuario, que se transmiten por un servidor, y se almacenan en una base de datos, dependiendo si su identificador único, ya ha sido registrado en la misma. El corte y la reconexión remoto del servicio se realizan dependiendo las condiciones de cada empresa con el usuario.



**Figura 4-2:** Diagrama de bloques de la etapa de control y monitorización de datos

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

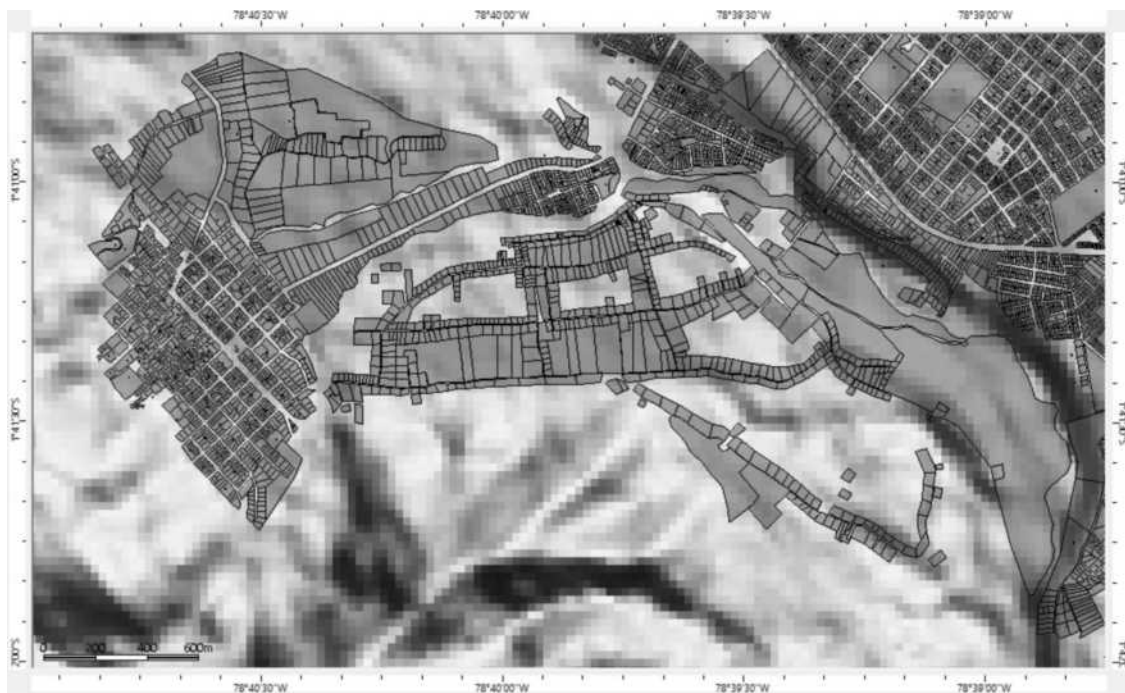
#### 2.4.2 Simulación de la red

La propagación de la señal es un factor clave para la simulación de la red, debido a que se puede presentar condiciones atmosféricas desfavorables o superficies que causan reflexión, difracción, etc., por lo que hoy en día los simuladores emplean técnicas que modelan el comportamiento de

la señal según el modelo de propagación. Para la simulación de la RCMAP, se utiliza el software de planificación Atoll en su versión portable, que es una herramienta de diseño, planificación y optimización para redes de multiplataforma con usuarios fijos y móviles, donde se analiza la cobertura del modem y acceso de usuarios y el simulador de red Packet Tracer 7.0, donde se estableció la estructura de física de la RCMAP. A continuación, se detalla el procedimiento para el diseño de la red, la incorporación de mapas y parámetros necesarios para modelar la misma.

#### 2.4.2.1 Parámetros georreferenciales

Se utiliza el sistema de coordenadas WGS 84 / UTM zone 17S que se posiciona en Ecuador, para cargar un modelo de terreno digital o DTM (*Digital Terrain Model por sus siglas en inglés*) donde se encuentran las elevaciones de la parroquia de Yaruquíes (zona de estudio), de la ciudad de Riobamba, mostrada en la Figura 5-2, en la que se puede observar un mapa que indica los predios de cada usuario.



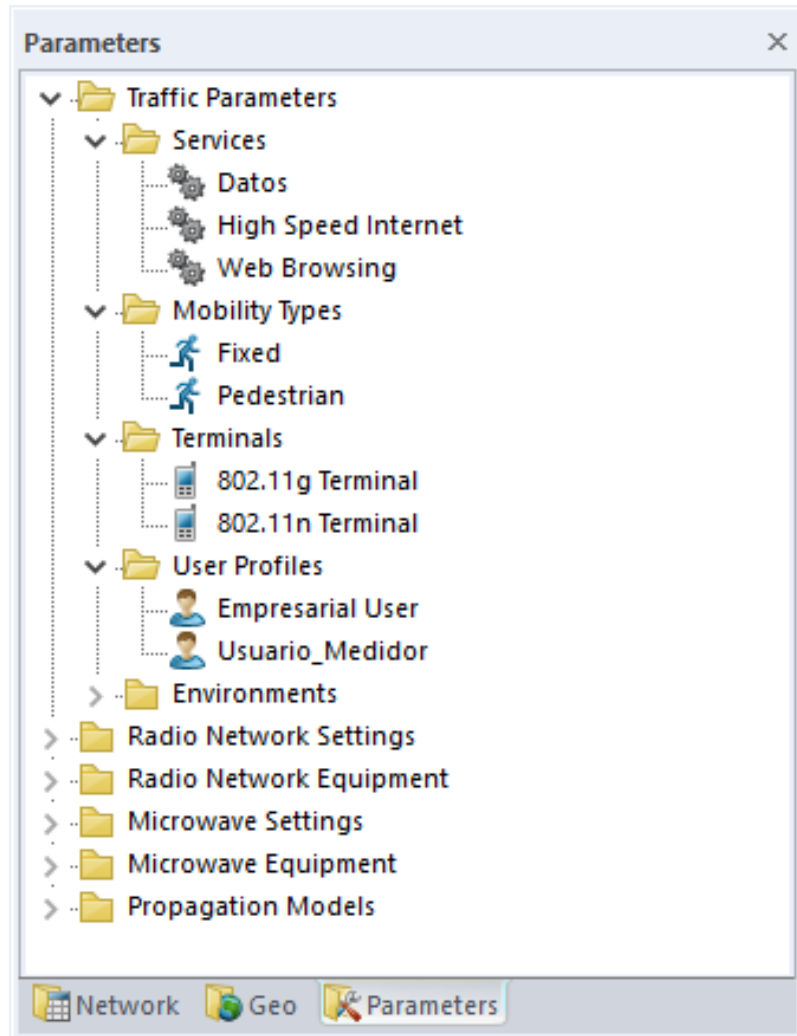
**Figura 5-2:** Zona de estudio

**Realizado por:** CASTILLO Isabel & MURILLO Santiago. 2017

#### 2.4.2.2 Parámetros de la red

Se configuran los eventos y componentes principales para la planificación de la RCMAP, como emplazamientos o puntos geográficos donde está situado el modem, predicciones de cobertura, número de usuarios (terminales) disponibles por modem, potencia de transmisión por modem,

modelo de propagación, entre otros. En la Figura 6-2, se observa la configuración de los parámetros de tráfico de la RCMAP como el tipo de servicio a usar (datos y velocidad alta de internet), terminales (802.11g/n), tipo de terminales (fijos), perfiles de usuarios (empresarial y usuario domiciliario) y ambientes (urbanos y suburbanos) donde se realiza la simulación.



**Figura 6-2:** Parámetros de tráfico

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

- Parámetros de transmisor

Las características de los módems, se encuentran detallados en la Tabla 3-2, que se determinaron a partir del modem (Anexo E/Anexo F) instalado en el domicilio de cada usuario.

**Tabla 3-2:** Parámetros del modem

Aspectos	Transmisor
Tipo de transmisor	Intra-network (Server and Interferer)
Altura desde la superficie	1m, 2m, 3m, 4m
Tipo de antena	2400MHz Omni 5.7 dBi
<b>Celda</b>	
Banda de frecuencia	2.4GHz – 20MHz
Número de canal	1, 5, 9, 13
Potencia (dBm)	20, 18, 15
Configuración de trama	802.11n – 20MHz
Equipo de recepción	802.11 n
Número de usuarios	10, 5
<b>Propagación</b>	
Modelo de propagación	Erceg-Greenstein (SUI)
Radio	50

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

La Figura 7-2, indica la ubicación de los usuarios que disponen de modem, debido a su cercanía se ha hecho reutilización de frecuencia, asignándole un canal distinto a cada uno, como se refleja en los parámetros de la celda de la Tabla 3-2, de igual manera la potencia transmitida ha sido controlada garantizando mínimos niveles de señales y mínimas interferencias. La selección del modelo de propagación es debido a las características del proyecto (condiciones, frecuencias y tecnologías), para este caso es Erceg-Greenstein (SUI) que es un modelo que está basado en medidas experimentales, que incluye difracciones y pérdidas por espacio libre.



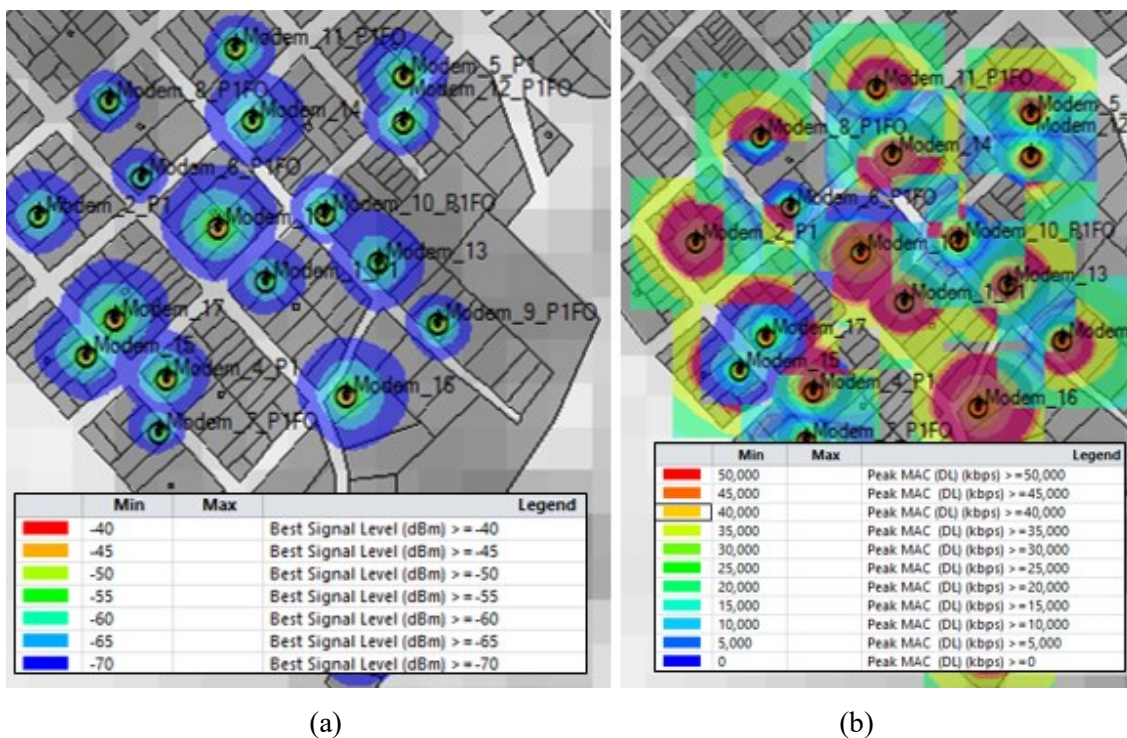
**Figura 7-2:** Transmisores existentes por cada usuario

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

- Parámetros de predicciones por cobertura

Los estudios de cobertura muestran los resultados para determinadas condiciones de la red, basándose en el modelo de propagación. Para las condiciones de cobertura por nivel de señal, se considerará una sensibilidad del receptor de -70 dBm, lo que teóricamente es para el estándar Wi-Fi, garantizando que la mejor recepción sea por encima de este valor y que el terminal sea capaz de conectarse a la red previniendo cortes de conexión por desvanecimiento de la señal.

En la Figura 8-2(a), se muestra el resultado del estudio de cobertura por nivel de señal, en un rango de -40 dBm a -70 dBm, consiguiendo niveles por encima de la sensibilidad del receptor, donde cada terminal tiene mayor probabilidad de conexión mientras más cerca se encuentre del transmisor. En la Figura 8-2(b) se observa el intervalo de transferencia de datos que va desde 0 hasta un máximo valor de 50000 Kbps, donde, mientras la tasa de transferencia sea mayor, mayor será la velocidad de transmisión de datos.



**Figura 8-2:** Estudio de predicciones de cobertura (a) Por nivel de señal (b) Por rendimiento.

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017



- Suscriptores

Son todos los usuarios que conforman la RCMAP. En la Figura 9-2, se muestra las ubicaciones de cada usuario y de los módems, donde los usuarios que se encuentren dentro de la cobertura de un modem en particular, podrán conectarse a la red a través de él, siempre y cuando cuente con el ID y contraseña del mismo.



**Figura 9-2:** Suscriptores que conforman la red  
Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Una vez creada la lista de usuarios se puede configurar los mismos, como nombres, altura de la antena y perfil de usuario. El resto de propiedades las generará el programa una vez que se realice la parte de cálculos.

#### 2.4.2.3 Simulación

Se realiza el estudio de optimización de nuestra red, con distribuciones reales de los usuarios y demanda de servicio generada. El funcionamiento de las simulaciones dependerá del servicio por usuario, perfiles de usuario y estado del modem según la probabilidad de conexión. Cabe mencionar la simulación se realiza con 19 módems y alrededor de 200 usuarios para análisis de conexión de los mismos.

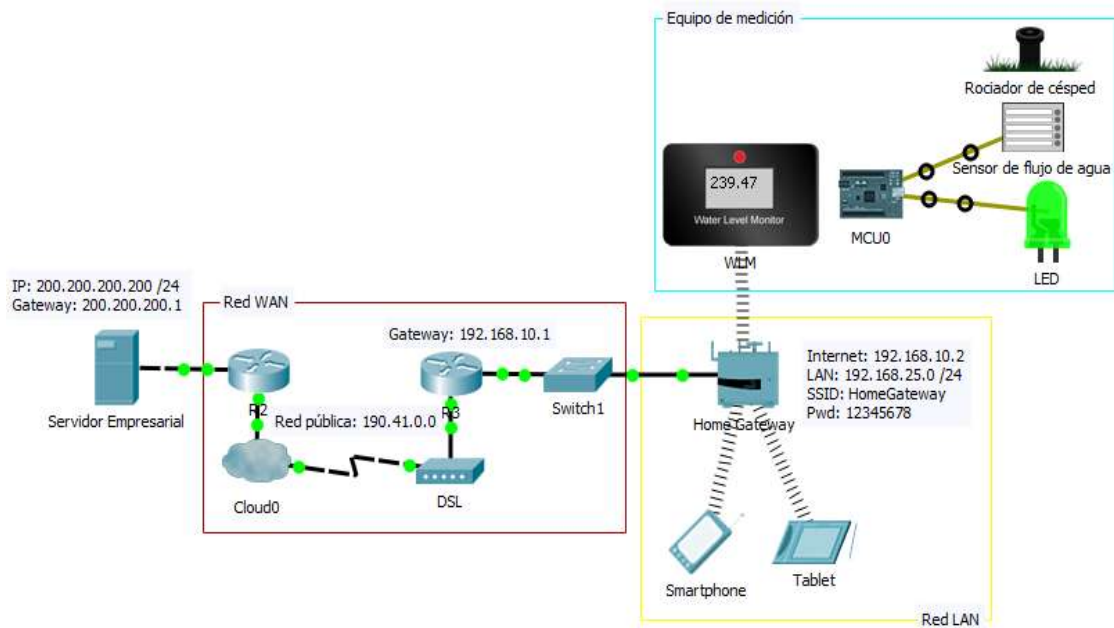
En la Figura 10-2, se puede observar los usuarios que disponen de cobertura de color verde y los que están sin servicio de color rojo. En este caso, el 82.4% de un total de 200 usuarios logran conectarse a un modem disponible, mientras que el 17.6% de usuarios que no alcanzan cobertura no se pueden conectar al servicio. Por lo tanto, cada modem puede dar servicio a varios usuarios, dependiendo su alcance, sin importar el tamaño del paquete que pase por él, ya que el tamaño de paquete de consumo de agua es pequeño y no afectaría en el rendimiento de la red.



**Figura 10-2:** Simulación de conexión al modem por usuario

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

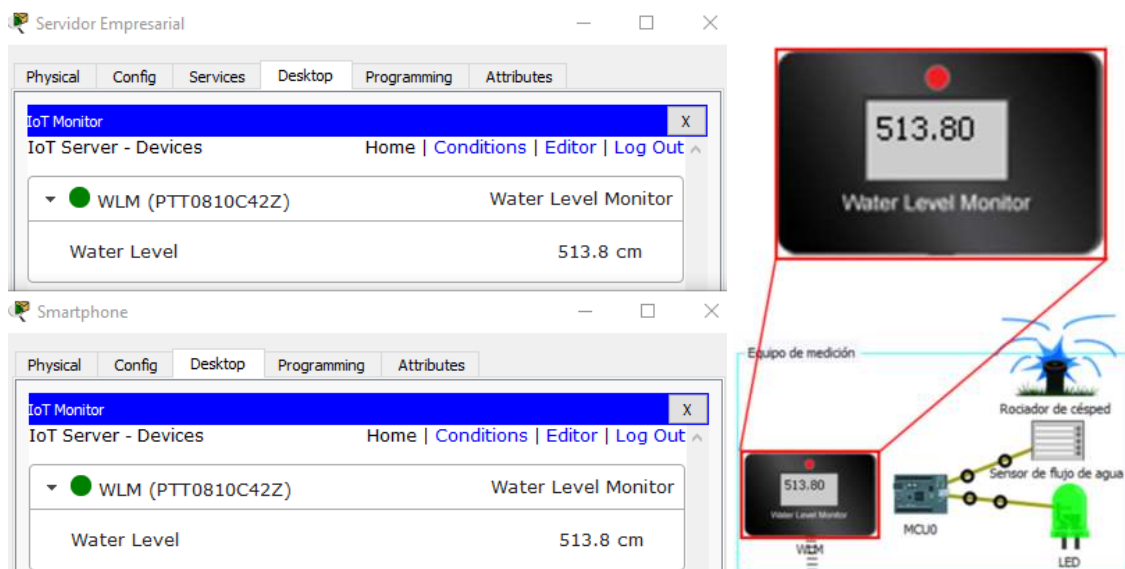
Como se mencionó anteriormente, para determinar la transmisión de datos en tiempo real, se utiliza el simulador de red Packet Tracer 7.0, donde se creó la RCMAP, como se puede observar en la Figura 11-2, con todos los dispositivos, equipos y topología que posteriormente la pueden conformar. Para la comunicación pública se colocó un modem DSL, estableciendo una conexión hacia el Internet (Cloud) que termina en un enrutador genérico y finalmente llega a un servidor que atiende los servicios.



**Figura 11-2:** Diseño de la RCMAP en packet tracer 7.0

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Mediante esta configuración se comprueba que los datos pueden ser enviados en tiempo real y visualizados en una interfaz gráfica de la empresa o mediante aplicaciones móviles, como se muestra en la Figura 12-2, demostrando que es factible la implementación de la red.



**Figura 12-2:** Transmisión de datos en tiempo real

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

## 2.5 Implementación del prototipo

Se eligieron los componentes y dispositivos a partir de la información indicada en el capítulo I con respecto al entorno, facilidad de instalación y obtención de los mismos, de tal manera que cumplan con la monitorización y control del consumo de agua potable en tiempo real.

### 2.5.1 Selección de los equipos que conforman la RCMAP

Establecidos los requerimientos de *hardware*, la tecnología de comunicación y la arquitectura de la red, a continuación, se presenta los equipos y componentes para el control y monitorización de agua potable en tiempo real.

#### 2.5.1.1 Equipo de medición para el consumo de agua potable

En el mercado existe un sin número de equipos que permiten medir el consumo de agua potable y que sus datos sean visualizados en tiempo real, por lo general, se usan medidores inteligentes con su propia interfaz gráfica, pero debido a su alto costo, se ha optado por dispositivos que junto a tarjetas de desarrollo realizan la misma función, son de fácil adquisición y bajo precio.

- Sensor de flujo de agua yf-s201

Su función se basa en el efecto hall, cada vez que el rotor da una vuelta, es decir, que cuando se genera un campo magnético emite un pulso eléctrico, indicando la cantidad de agua que pasa por él. Tiene tres cables: rojo (5-24V), negro (tierra) y amarillo (salida del pulso), como se puede observar en la Figura 13-2 y sus características técnicas se presenta en el Anexo A.



**Figura 13-2:** Sensor de flujo de agua YF-S201

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Estos sensores no son de precisión, por lo que varía su medida dependiendo de la orientación del sensor, velocidad y presión del flujo de agua. Para una precisión de más del 10% se debe calibrar. (<https://www.vistronica.com/sensores/caudal/sensor-medidor-de-flujo-yf-s201-detail.html>, 2016)

- Medidor de agua convencional

Son utilizados en viviendas e inmuebles industriales, permiten diferentes volúmenes de lectura de datos y realizan lectura remota mediante un emisor de pulso eléctrico que se activa cuando un campo magnético se acerca a él. En la Figura 14-2, se puede apreciar el medidor convencional avanzado que actualmente se encuentra instalado en los domicilios de la parroquia de Yaruquíes, sus características técnicas se presentan en el Anexo B.



**Figura 14-2:** Medidor convencional de agua potable

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Para generar la cantidad de agua que pasa por el medidor, existen emisores de impulsos (reed) diseñados para estos que se activan por campo magnético emitiendo una señal y dependiendo del tipo de medidor, varía su forma y colocación, pero el principio de funcionamiento es el mismo en todos. Por inexistencia en el país y costo elevado de obtención no se usó el emisor, pero para realizar pruebas de su funcionamiento se usa un sensor digital de efecto hall A3144, donde su tarea es igual que la del emisor, sus características técnicas se pueden ver en el Anexo C. En la Figura 15-2 se puede apreciar el mismo, y su conexión adaptada con el medidor convencional que trae incorporado un imán cilíndrico que produce el campo magnético. Dependiendo de la ubicación de éste en el medidor, se podrá tener el número de litros por cada pulso.



**Figura 15-2:** Incorporación del sensor de flujo en el medidor convencional

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

- Tarjeta de desarrollo

Para elegir la tarjeta de procesamiento, basado en Wi-Fi, se exploraron las diferentes características técnicas, siendo el costo el factor determinante para la elección. Entre los modelos de Arduino, existe uno que tiene un shield de Wi-Fi, por lo que, se realizó la comparación con el módulo Wi-Fi ESP8266x.

Al evaluar el tamaño físico se considera que el Arduino es más grande que el módulo ESP8266x, además estos módulos son de bajo costo, tienen acceso a las extensiones y bibliotecas del entorno de Arduino, ganando popularidad y cantidad de implementaciones, razones por las cuales se ha elegido como el módulo de procesamiento y comunicación para el sensor de flujo y medidor de agua convencional. Existen varias versiones de módulos ESP8266, algunas de ellas necesitan de un adaptador para su programación, pero otras como el NodeMCU v3 ya trae uno incorporado, facilitando su configuración. Por lo tanto, para este trabajo va a ser utilizado el Módulo ESP8266 NodeMCU v3.

- Módulos ESP8266 NodeMCU v3

Es el microcontrolador más utilizado para conectar las cosas al Internet, tiene potentes capacidades a bordo de procesamiento y almacenamiento que le permiten integrarse con sensores y dispositivos específicos de aplicación a través de sus terminales GPIO con un desarrollo mínimo y carga mínima durante el tiempo de ejecución. En la Figura 16-2, se puede observar el componente y sus características técnicas en el Anexo D.



**Figura 16-2:** Módulo ESP8266 NodeMCU v3

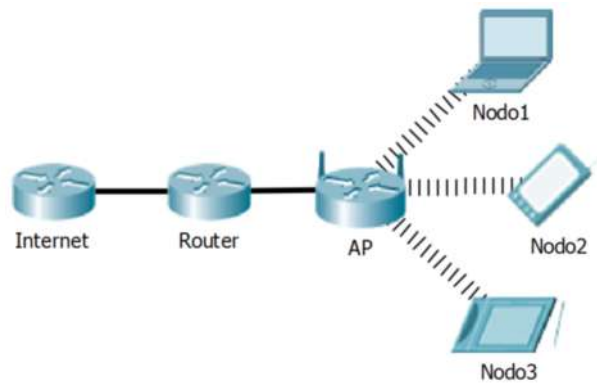
Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Estos módulos son imbatibles en cuanto a precio, además de un completo sistema Wi-Fi, un potente procesador SOC (*System on Chip, por sus siglas en inglés*) e incluye un plugin en la

Interfaz de desarrollo de Entorno de Arduino para poderlo programar.  
(<https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>, 2016)

### 2.5.1.2 Transmisión de datos

La transmisión inalámbrica Wi-Fi será desde el equipo de medición de agua potable hasta el punto de acceso o AP (*Access Point, por sus siglas en inglés*), que sirve de intermediario para que la información llegue al servidor empresarial, siempre y cuando el nodo esté dentro de su cobertura, como se muestra en la Figura 17-2. Además, sirve para que los datos puedan almacenarse o visualizarse en un servidor en la nube.



**Figura 17-2:** Funcionamiento básico de un Access Point

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Las características técnicas de los AP (ver Anexo E, Anexo F y Anexo G) seleccionados, se basaron en los actualmente se encuentran instalados en los domicilios de los usuarios, los cuales pertenecen a las empresas proveedoras del servicio de Internet con mayor cobertura. Su medio de transmisión es mediante cobre o fibra óptica.

### 2.5.1.3 Monitorización y control del consumo de agua potable

Se necesita una computadora con características buenas de procesador y memoria que actúe como servidor para almacenar y direccionar todos los datos para monitorizarlos. Para el control de corte y reconexión de agua potable se utiliza una electroválvula solenoide y dependerá si el usuario ha cancelado o no, en el tiempo establecido de pago.

- Electroválvula o válvula solenoide

Es aquella que permite o cierra el paso de un fluido de agua, normalmente están cerradas impidiendo el paso de fluido a través del cuerpo de la válvula, pero cuando están alimentadas por

corriente eléctrica se abren permitiendo el paso de flujo. Se alimenta a DC12V, 1A (Amperaje máximo) y soporta presión de 0.02 a 0.8 Mpa (Megapascal). En la Figura 18-2, se puede apreciar físicamente la electroválvula.

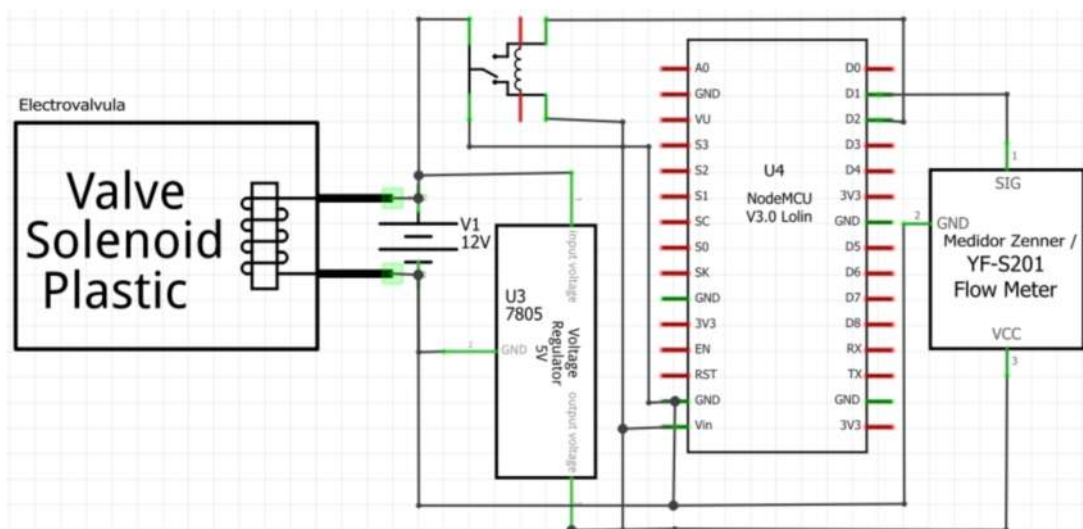


**Figura 18-2:** Electroválvula o válvula de solenoide

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

### 2.5.2 Esquema de conexión de los equipos para la medición y corte de agua potable

Para el diagrama de conexión del sensor de flujo de agua o medidor ZENNER, módulo ESP8266 NodeMCU v3 y electroválvula para el corte de agua, se basó en los manuales técnicos de cada uno. En la Figura 19-2, se muestra el esquema de conexión, donde el módulo ESP8266 NodeMCU v3, es el componente principal, para la transmisión en tiempo real de valores de consumo y corte de agua potable.



**Figura 19-2:** Esquema de conexión para la medición del consumo de agua

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017



- La electroválvula une su salida al terminal D2 del módulo ESP8266 NodeMCU v3, se mantiene cerrada cuando se envía un pulso desde el módulo, caso contrario se abre, esta acción se activa por la intermediación de un relé. Para la alimentación de la válvula se transforma los 110VAC a 12V DC mediante el uso de un adaptador y se dispone de una fuente reguladora de voltaje que entrega 5V DC, para alimentar el sensor de flujo de agua o medidor convencional y el módulo ESP8266 NodeMCU v3.
- El sensor de flujo YF-S201, une su salida de pulso (SIG) al terminal digital D1 del Módulo ESP8266 NodeMCU v3. En caso del medidor convencional, a partir del sensor de efecto hall A3144 se realiza la misma conexión al módulo, adicionando una resistencia de 10K $\Omega$  de ¼ Watt [W], entre VCC y la salida de pulso. El sensor de flujo y el medidor convencional se alimentan con 5V DC.
- Finalmente, todos los valores que ingresan por el terminal digital D1 del módulo ESP8266 NodeMCU v3, son convertidos de L/min a m<sup>3</sup>, y transmitidos por RF (2,4GHz) al AP que llega al servidor y se muestra en la GUI en tiempo real.
- El consumo aproximado de corriente en todo el esquema es de 345mA, según las características de los dispositivos siguientes:
  - Válvula Solenoide = 250 mA
  - Medidor/sensor = 15mA
  - Módulo ESP8266 = 80mA

### ***2.5.3 Requerimientos de software para el prototipo de la RCMAP***

Existen diversas maneras para lograr un ambiente de desarrollo entre todos los componentes utilizados, satisfaciendo las necesidades de cada uno y que se puedan comunicar entre sí. A continuación, se describen los requerimientos que cada etapa debería cumplir:

- Registrar el valor de consumo de agua potable en un periodo de tiempo variable
- Conversión de unidades de medida de litros a metros cúbicos (m<sup>3</sup>)
- Conexión de dispositivos con tecnología Wi-Fi
- Mostrar el valor medido en tiempo real
- Almacenar los valores medidos para posteriores reclamos
- Modificar la interfaz gráfica de usuario de acuerdo a los requerimientos del sistema

#### 2.5.4 Descripción del software utilizado

Para la comunicación entre todos los dispositivos que existen en la red, se usó una programación estructurada y varias herramientas de software para el desarrollo de la misma, a continuación, su detalle.

##### 2.5.4.1 Arduino IDE

El Entorno de Desarrollo Integrado o IDE (*Integrated Development Environment, por sus siglas en inglés*), se utilizó para la programación del módulo ESP8266 NodeMCU mediante la inclusión de librerías y el plugin ESP8266. Por lo tanto, el IDE de Arduino sirve para cargar códigos para programar cualquier placa física compatible con su plataforma.

Su sketch o lugar donde se escribe el código, obligatoriamente debe tener dos funciones, *setup ()*, la cual se ejecuta una sola vez y realiza todas las iniciativas que el sistema necesita y *loop ()*, donde se establecen todos los procesos que se deben repetir.

##### 2.5.4.2 Servidor Mosquitto Broker

Mosquitto es un servidor intermediario MQTT (*Message Queue Telemetry Transport por sus siglas en inglés*) que tiene una gran compatibilidad con la comunidad y es fácil de instalar y configurar.

Usa el protocolo MQTT para mensajes de máquina a máquina, especialmente diseñado para comunicación ligera de transferencia de pequeños paquetes de datos enviados a través de enlaces de alta latencia y bajo ancho de banda entre dispositivos que requieren de conexión a Internet como, sensores.

Este protocolo trabaja con mensajes de código abierto, usa puertos TCP tales como, 1883 (no encriptado), 8883 (encriptado) y 8884 (encriptado, cliente certificado requerido) y puertos WebSockets como, 8080 (no encriptado) y 8081 (encriptado). TCP como es un protocolo orientado a la conexión, permite la corrección de errores y garantiza que los paquetes se reciban en orden, pero no significa que los mensajes no puedan perderse.

Hoy en día es el protocolo más usado para realizar pruebas de control y monitoreo desde internet mediante el modelo PUBLICACIÓN / SUSCRIPCIÓN, del cual se basa su funcionamiento, el que está compuesto por:

- Cliente: Es un dispositivo que publica información, se suscribe para recibir la información o puede hacer ambas acciones.
- Servidor Broker: Acepta la información publicada por el cliente, haciendo una difusión de esa información entre los clientes
- Publicar: Cuando un cliente envía información al bróker
- Tópico: La información debe estar con un tópico (identificador único). Una vez que los clientes se suscriben con tópicos específicos, estos reciben la información publicada por los tópicos. (<https://ricveal.com/blog/primeros-pasos-mqtt/>, 2016)

Este tipo de servidor está dirigido a dos grupos de usuarios según Baczkowicz (2016):

- Innovadores que necesitan una herramienta para crear prototipos de IoT o proyectos de integración
- Usuarios avanzados que necesitan una utilidad avanzada para sus entornos de trabajo

#### 2.5.4.3 *Delphi 10.2 Tokyo Starter (Promo Edition)*

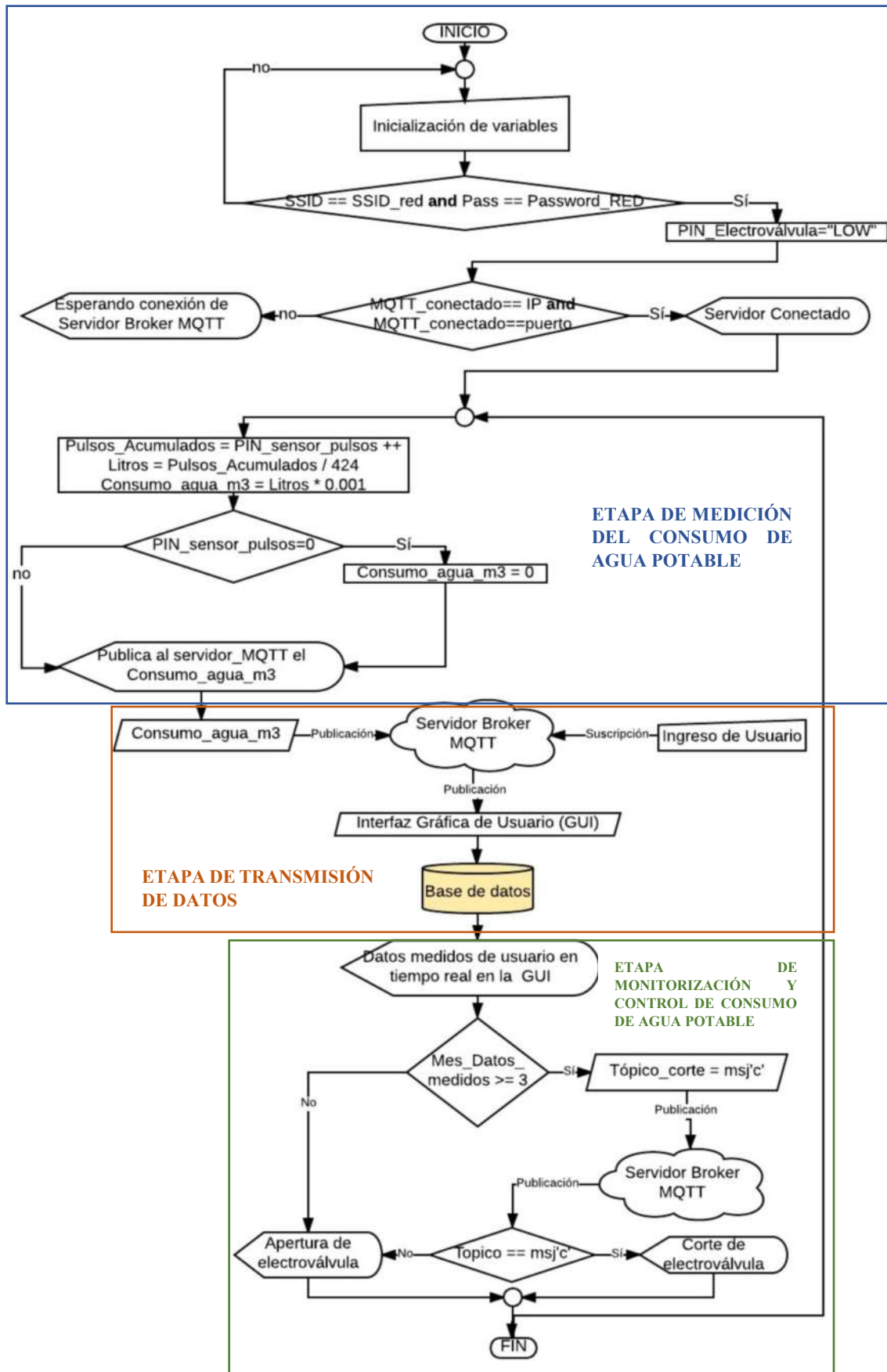
Es un paquete de desarrollo de aplicaciones para usuario final que trabaja hasta 5 veces más rápido en múltiples plataformas de Windows y bases de datos. Es una versión gratuita con limitaciones en la elección de sus componentes. En esta plataforma se configurará la Interfaz Gráfica de Usuario que está relacionada con el gestor base de datos MySQL para agregar y adquirir datos personales y de consumo de los usuarios. (<https://www.embarcadero.com/products/delphi/starter/info>, 2017)

#### 2.5.4.4 *Gestor de base de datos MySQL Community Edition*

Para el almacenamiento de datos se usará el gestor de base de datos *MySQL* como sistema de administración, para una colección estructurada de datos mediante tablas separadas optimizando la velocidad de los datos. El modelo lógico, con objetos como bases de datos, tablas, vistas, filas y columnas, ofrece un entorno de programación flexible. Es el lenguaje de consulta estructurado más común para acceder a las bases de datos. Para el software *MySQL* se usa la GPL (Licencia pública general de Linux), para fines no comerciales. (<https://www.mysql.com/why-mysql/>, 2017)

#### 2.5.5 *Diagrama de flujo de la RCMAP*

A continuación, se describe que herramienta de software se ha utilizado para desarrollar el prototipo y un diagrama de flujo de la RCMAP que se puede visualizar en la Figura 20-2, indicando todo el proceso que cumple el prototipo conforme a sus etapas.



**Figura 20-2:** Diagrama de flujo del monitoreo y control de consumo de agua potable

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

- Etapa de medición de consumo de agua potable

Se usa el entorno de desarrollo de Arduino, donde cumplirá las funciones de adquisición y transformación de datos detectados por el sensor de flujo de agua. En el sketch se reflejan dos funciones (*setup* y *loop*) que son obligatorias y permitirán el desarrollo de la programación (Anexo H).

#### Inicialización

- Inclusión de librerías para la comunicación entre el módulo ESP8266 NodeMCU v3 y el servidor Broker MQTT
- Declaración del SSID y password de la red Wi-Fi para conectarse al AP del usuario
- Declaración de la IP y puerto de comunicación (1883) del servidor Broker MQTT
- Declaración de variables globales para determinar el consumo de agua potable y corte/reconexión de agua
- Declaración del módulo ESP8266 NodeMCU v3 en modo cliente
- Definir tópico o identificador único, para que el servidor Broker MQTT pueda publicar los datos del sensor, adquiridos por el módulo ESP8266 NodeMCU v3

#### Configuración de arranque - Función *setup()*

- Condiciones para la conexión a la red Wi-Fi. Si el SSID y password de la red Wi-Fi, es igual que las del módulo ESP8266 NodeMCU v3, inmediatamente se conectará al AP del usuario, caso contrario, tendrá que esperar para retomar conexión.
- Se procede con la activación de los terminales del módulo ESP8266 NodeMCU v3, para adquirir los valores del sensor y corte/reconexión de agua potable.

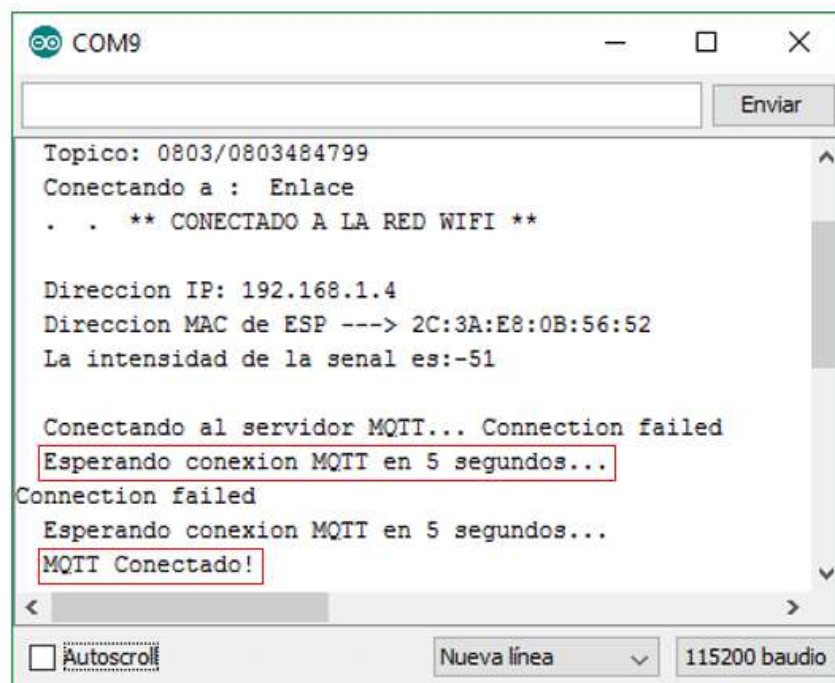
#### Lazo – Función *loop ()*

- Mediante la función `MQTT_connect ()`, el servidor Broker se activará y si cumple la condición de detección, se imprimirá en pantalla “Servidor conectado”, caso contrario “Esperando conexión de Servidor Broker MQTT”.
- El proceso de cálculo para transformación de litro/pulsos [L/pul] a metros cúbicos [m3]. Para la acumulación de pulsos fue necesaria la utilización de la función `ISRCountPulse ()`, que realiza el respectivo cálculo de los litros que pasan en 424 pulsos y posteriormente en m3.
- Publicación de valores censados al servidor Broker MQTT, por medio de transmisión inalámbrica (Wi-Fi).

- Para corte/reconexión de agua potable, la condición se activa cuando de la etapa de monitorización y control envía un carácter que genera la activación/desactivación de la electroválvula.

Las funciones que integran la codificación de la etapa de medición del consumo de agua, se describen a continuación:

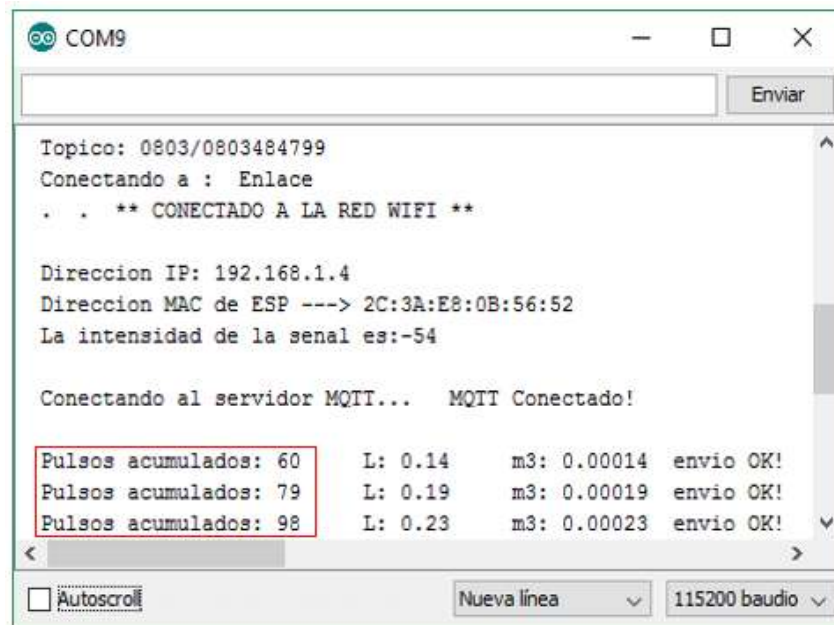
*MQTT\_connect()*: Se activa dentro de la función *loop()*, pero es una función que se aísla, para realizar el condicionamiento de conexión del servidor Broker MQTT. En caso de que establezca conexión imprimirá “Conectando al servidor MQTT...”, caso contrario “Esperando conexión MQTT en 5 segundos...”, como se puede observar en la Figura 21-2.



**Figura 21-2:** Conexión/Esperando conexión del servidor Broker MQTT

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

*ISRCountPulse()*: Realiza en conteo de los pulsos eléctricos emitidos por el sensor de efecto hall, y la respectiva acumulación de los mismos, que ayudan con el cálculo de consumo de agua potable. En la Figura 22-2, se puede observar su funcionamiento.



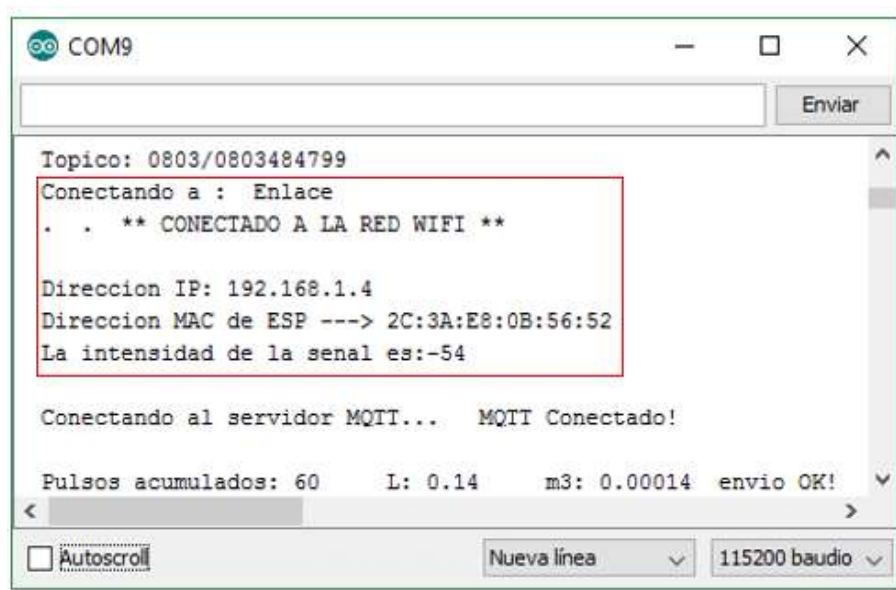
**Figura 22-2:** Funcionamiento de la función *ISRCountPulse()*

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Las librerías que integran el programa de esta etapa, son 3 que están asociadas al módulo ESP8266 NodeMCU v3 y servidor Broker MQTT.

*ESP8266WiFi.h*: Diseñada para cumplir con la funcionalidad de Wi-Fi. Posee diferentes clases como, clase Wi-Fi, clase de dirección IP, clase del servidor, clase de cliente y clase UDP.

- Clase Wi-Fi, permite la conexión con un AP, Para este caso se utiliza `WiFi.begin()`, para determinar el SSID y password al que se desea conectar, retornando un valor de cambio de estado (conexión exitosa con el AP) mediante la función `WiFi.status()`. En la Figura 23-2, se puede apreciar su ejecución y mediante las funciones que de describen a continuación, adquiere una dirección IP, MAC y nivel de señal.
  - *WiFi.localIP()*, devuelve la dirección IP, obtenida por DHCP por el AP.
  - *WiFi.macAddress()*, Devuelve la dirección MAC del módulo ESP8266 NodeMCU v3.
  - *WiFi.RSSI()*, Devuelve la intensidad actual de la señal en dB. (<https://alselectro.wordpress.com/2016/11/29/esp8266-wifi-library-on-arduino-ide/>, 2016)



**Figura 23-2:** Conexión del módulo al AP, dirección IP, MAC y nivel de señal

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

*Adafruit\_MQTT.h* y *Adafruit\_MQTT\_Client.h*: Arduino tiene acuerdos con Adafruit, que permiten conectar los proyectos al Internet. (<https://learn.adafruit.com/adafruit-io-basics-esp8266-arduino/overview>, 2016)

- Etapa de transmisión de datos

El servidor Broker MQTT debe tener una dirección IP fija (192.168.1.3) e iniciar sesión como sistema local automático, la base de datos debe contener las tablas (usuario, tópico, consumo de agua y cortes) y sus respectivos campos, y en Delphi Starter se debe crear el diseño de la GUI (

Anexo I) y definir la función para enlazar al servidor Broker MQTT, base de datos MySQL y la GUI.

- Ingreso de usuarios (Nombre, Apellido, CI, Numero de medidor, Dirección, Celular)
- El usuario se suscribe al servidor Broker MQTT y sus datos se guardan en las respectivas tablas creadas en la base de datos MySQL.
- Se realizan las condiciones para que automáticamente se cree el tópico o identificador único, el mismo que estará registrado en el módulo ESP8266 NodeMCU v3.
- El módulo ESP8266 NodeMCU v3, publica los valores de consumo de agua potable por Wi-Fi dependiendo su tópico, reconoce al servidor Broker MQTT por su dirección IP (192.168.1.3). Estos datos son almacenados en los respectivos campos creados según sus tablas.



- El servidor Broker MQTT empareja el t3pico publicado con el suscrito, imprimiendo el valor de consumo de agua potable en la GUI en tiempo real.
- Etapa de monitorizaci3n y control del consumo de agua potable
  - Los valores medidos se reflejan en la GUI en tiempo real, adem3s del nombre y apellido de usuario, numero de medidor, t3pico o identificador 3nico y direcci3n. Se puede hacer consultas de consumo y pago del mismo.
  - Mediante el monitoreo, se controla si un usuario ha pasado el l3mite (3 meses en adelante) de no pagar el consumo de agua potable, si es as3, se env3a un caracter al servidor Broker MQTT y este publica en mensaje de informaci3n al m3dulo ESP8266 NodeMCU v3, el cual emite un pulso el3ctrico al terminal digital de la electrov3lvula, realizando el respectivo corte del l3quido vital. Caso contrario la electrov3lvula permanece abierta

## CAPÍTULO III

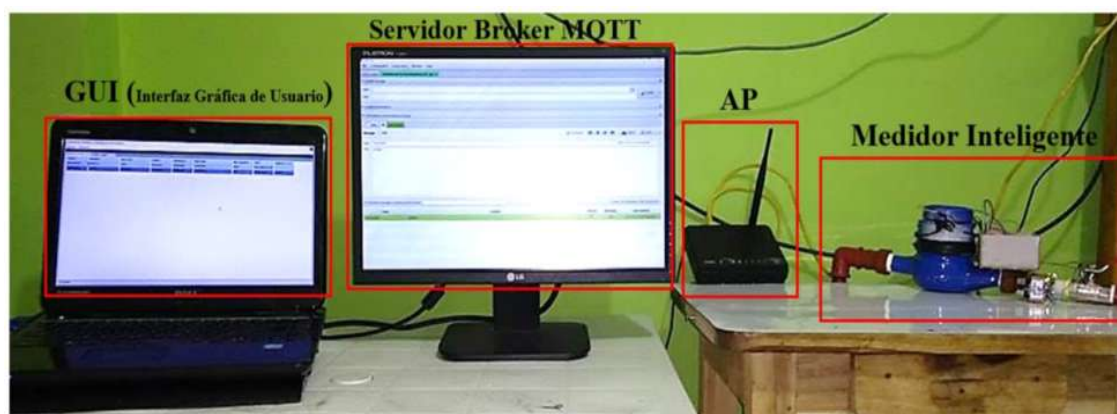
### 3. ANALISIS Y RESULTADOS

Se presenta los resultados alcanzados en las diferentes pruebas realizadas, así como el análisis en cada una. A partir del diseño de hardware y software se evaluó la RCMAP en función de los requerimientos mencionados anteriormente, corroborando valores medidos y errores en la comunicación, mediante pruebas de medición de consumo, calidad de enlace, cobertura y conectividad. También se establece una tabla comparativa de los costos de medidores inteligentes junto al sistema de comunicación que existen en la actualidad.

La RCMAP trabaja en una red de área local, debido a que el contrato de una dirección IP pública fija con la cual trabajaría remotamente el servidor *Broker* MQTT, tiene un costo elevado para realizar pruebas. Sin embargo, para examinar el tiempo de retardo fuera de la red LAN, se utilizó de un servidor en la nube.

#### 3.1 Caracterización de la RCMAP

En la Figura 1-3, se puede observar la RCMAP con sus diferentes elementos que la conforman, desde el equipo de medición que debe ser colocado en cada domicilio, AP por donde pasan los valores de consumo de agua potable, servidor intermediario Broker MQTT y GUI.



**Figura 1-3:** Elementos que componen la (RCMAP)

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

### 3.1.1 Caracterización de la etapa de medición de consumo de agua potable

Las pruebas de esta etapa consisten en determinar, si los dispositivos (sensor de flujo o medidor convencional de agua), realizan con exactitud o un error mínimo de  $\pm 0.1$  litro en la detección del líquido vital, la cobertura máxima en que los dispositivos se pueden ubicar y calidad de enlace.

#### 3.1.1.1 Pruebas de medición

- Mediciones realizadas por el medidor convencional y el sensor de flujo de agua

Para este caso, se tomó como referencia un medidor convencional, para mostrar los errores que tienen el sensor de flujo de agua YF-S201 y cuanta precisión tiene al medir. El sensor de flujo de agua por características técnicas, emite 450 Pulsos/Litro, y al realizar las respectivas pruebas se observó que presenta errores no mayores a 0.07 litros con 1 litro como volumen de referencia con repetitividad de 5, como se muestra en la Tabla 1-3. Al calibrar el sensor, se obtuvo un promedio de 424 pulsos/litro. En el Anexo H, se presenta la programación de calibración en el módulo ESP8266 NodeMCU v3.

**Tabla 1-3:** Pruebas de medición del sensor de flujo de agua YF-S201

Volumen de agua referencia (L)	Pulsos	Error Absoluto (%)
1	428	0.048
1	422	0.062
1	420	0.066
1	417	0.073
1	431	0.042
<b>Pulso Promedio</b>	<b>424</b>	-

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Según la Tabla 2-3, realizando las mediciones con la nueva referencia de pulsos, se obtuvo un porcentaje de error no más a 0.01 litros, dándonos como resultado, mediciones más precisas que las reflejadas en la Tabla 1-3.

**Tabla 2-3:** Mediciones del sensor de flujo de agua calibrado

Volumen de agua de ref. (L)	Pulsos	Error Absoluto (%)
1	423	0.0023
1	424	0
1	421	0.0070
1	420	0.0094
1	420	0.0094

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

En la Tabla 3-3, se presenta los valores adquiridos por el medidor convencional y el sensor de flujo de agua, con un volumen de referencia establecido y así tomar el porcentaje de error absoluto, para evaluar cuál de los dos es el más preciso al entregar las medidas. Se realizaron 5 muestras de litros aleatorios, considerando que el medidor convencional entrega 1 litro por pulso. De esta manera se comprueba si la etapa de medición de consumo de agua, no incorpora más errores de lo indicado en la Tabla 2-3.

**Tabla 3-3:** Mediciones entre el medidor convencional y el sensor de flujo de agua

N° Pruebas	Cantidad Medida (L)	Medidor convencional		Sensor de flujo	
		Medición	Error absoluto	Medición	Error absoluto
1	2	2.15	-0.15	1.98	0.02
2	3	3.15	-0.15	2.95	0.05
3	4	4	0	3.95	0.05
4	5	5.02	-0.02	4.95	0.05
5	6	6.1	-0.1	5.96	0.04
6	8	8.03	-0.03	7.9	0.1
7	10	10.05	-0.05	9.9	0.1

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Según los resultados en la Tabla 3-3, se realizaron medidas continuas a partir del tercer litro, y se puede observar que la medición del sensor de flujo de agua disminuye o aumenta conforme al incremento de la cantidad de agua, mientras que el medidor convencional arroja valores menos robustos que el sensor, llegando a la conclusión de que la precisión del medidor convencional es más exacta, con errores de lectura de media de 0.06 litros por cada 10 litros.

Al trabajar con los medidores convencionales, se tiene la dificultad para abrir la tapa y colocar el imán del medidor en la aguja que cuenta los vuelas por litro, además de adicionarle el sensor de efecto hall A3144 para medir los litros/pulso, considerando que se puede integrar los emisores diseñados para este tipo de medidores. En la Figura 2-3, se muestra el medidor inteligente con todos los componentes que lo conforman: medidor convencional, módulo ESP8266 NodeMCU v3, electroválvula, resistencias y fuente regulable.



**Figura 2-3:** Medidor inteligente

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

### **3.1.2 Caracterización de la etapa de transmisión de datos**

Las pruebas que se realizan en esta etapa, consisten en verificar la fiabilidad de conectividad entre el medidor inteligente y el AP, también si los valores de consumo del medidor inteligente al ser publicados al servidor *Broker* MQTT no mantienen ningún tipo de retardo, de igual manera desde el servidor a la GUI y en el almacenamiento de datos.

#### **3.1.2.1 Pruebas de conectividad**

Se basa en el alcance o cobertura del modem, pérdida de paquetes, ruta de comunicación de los paquetes, etc. Para este tipo de pruebas se utilizaron diferentes herramientas de software, que serán mencionadas a continuación.

- Determinación de alcance del router

Las mediciones de cobertura se realizaron mediante una aplicación móvil que permite analizar la potencia de la señal de un router específico considerando los obstáculos, como se pueden observar en la Tabla 4-3.

**Tabla 4-3:** Nivel de cobertura de router

	<b>Proveedor1</b>	<b>Proveedor2</b>
<b>distancia (m)</b>	Potencia (dBm)	Potencia (dBm)
<b>1</b>	-42	-38.01
<b>5</b>	-56	-42
<b>7</b>	-60	-53
<b>9</b>	-69	-61
<b>12</b>	-73	-62
<b>15</b>	-75	-74
<b>18</b>	-83	-78
<b>24</b>	-91	-81

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Se determinó la potencia de la señal a una distancia establecida y se pudo analizar que el modem del Proveedor2 tiene mayor alcance debido a las características presentadas en el Anexo G. Por lo tanto, la distancia máxima a la que pueden estar los modem es de 18m, con una potencia de señal entre -78 dBm y -83 dBm, para que los módulos ESP8266 puedan mantener una conexión estable considerando la sensibilidad de recepción detallado en el Anexo D. Se debe considerar que los obstáculos pueden variar, por lo que no es recomendable basarnos en la distancia entre el medidor y el modem, sino cuando se tenga valores aproximados de potencia de -80 dBm o por encima.

- Pruebas de calidad del enlace de red

Para determinar la calidad del enlace de red, se debe cuantificar el ancho de banda por protocolo TCP, jitter (variación de latencia), y pérdidas de paquetes por protocolos UDP, mediante herramientas informáticas que miden el rendimiento de la red. En este trabajo se utiliza la herramienta *Iperf* que funciona bajo comandos, ingresados en el intérprete de comandos del cliente (*smartphone*) y del servidor (*laptop, SO windows*), compatible con sistemas Linux y Windows.

- Pruebas de jitter y pérdidas de paquetes

Se realizan mediante datagramas UDP. Para tener una buena calidad de enlace es necesario que la pérdida de paquetes de datos no supere al 1%, caso contrario genera muchas retransmisiones afectando el ancho de banda.

En la Figura 3-3(a), se puede observar la cantidad de paquetes que se envían y reciben en una petición con intervalo de 10s, desde el cliente con dirección IP 192.168.0.101 que se enlaza al servidor local *windows* por un puerto libre, el cual escucha por el puerto 5001 mediante la dirección IP 192.168.0.103. En la Figura 3-3(b), el número de paquetes que recibe el servidor es de 1470 bytes los cuales son evaluados en 3 periodos de 10s con intervalo 1s cada uno y se trabajó con una potencia de -80 dBm.

```
C:\>iperf.exe -s -u -i 10

Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)

OpenSCManager failed - Acceso denegado. (0x5)
[156] local 192.168.0.103 port 5001 connected with 192.168.0.101 port 44684
[ ID] Interval      Transfer     Bandwidth    Jitter    Lost/Total Datagrams
[156] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  4.167 ms  0/ 89 (0%)
```

(a)

```
C:\>iperf.exe -s -u -i 1

Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 64.0 KByte (default)

OpenSCManager failed - Acceso denegado. (0x5)
[156] local 192.168.0.103 port 5001 connected with 192.168.0.101 port 47286
[ ID] Interval      Transfer     Bandwidth    Jitter    Lost/Total Datagrams
[156] 0.0-1.0 sec   129 KBytes  1.06 Mbits/sec  1.457 ms  0/ 90 (0%)
[156] 1.0-2.0 sec   128 KBytes  1.05 Mbits/sec  5.483 ms  0/ 89 (0%)
[156] 2.0-3.0 sec   129 KBytes  1.06 Mbits/sec  3.448 ms  0/ 90 (0%)
[156] 3.0-4.0 sec   121 KBytes  988 Kbits/sec  9.126 ms  0/ 84 (0%)
[156] 4.0-5.0 sec   132 KBytes  1.08 Mbits/sec  5.211 ms  1/ 93 (1.1%)
[156] 5.0-6.0 sec   125 KBytes  1.02 Mbits/sec  1.949 ms  0/ 87 (0%)
[156] 6.0-7.0 sec   131 KBytes  1.07 Mbits/sec  9.150 ms  0/ 91 (0%)
[156] 7.0-8.0 sec   129 KBytes  1.06 Mbits/sec  7.876 ms  0/ 90 (0%)
[156] 8.0-9.0 sec   129 KBytes  1.06 Mbits/sec  2.220 ms  0/ 90 (0%)
[156] 9.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  7.842 ms  1/ 893 (0.11%)
[156] local 192.168.0.103 port 5001 connected with 192.168.0.101 port 46864
[ ID] Interval      Transfer     Bandwidth    Jitter    Lost/Total Datagrams
[156] 0.0-1.0 sec   129 KBytes  1.06 Mbits/sec  6.494 ms  0/ 90 (0%)
[156] 1.0-2.0 sec   128 KBytes  1.05 Mbits/sec  4.007 ms  0/ 89 (0%)
[156] 2.0-3.0 sec   129 KBytes  1.06 Mbits/sec  4.885 ms  0/ 90 (0%)
[156] 3.0-4.0 sec   128 KBytes  1.05 Mbits/sec  2.482 ms  0/ 89 (0%)
[156] 4.0-5.0 sec   128 KBytes  1.05 Mbits/sec  3.240 ms  0/ 89 (0%)
[156] 5.0-6.0 sec   128 KBytes  1.05 Mbits/sec  7.749 ms  0/ 89 (0%)
[156] 6.0-7.0 sec   122 KBytes  1000 Kbits/sec  13.011 ms  0/ 85 (0%)
[156] 7.0-8.0 sec   134 KBytes  1.09 Mbits/sec  4.132 ms  0/ 93 (0%)
[156] 8.0-9.0 sec   128 KBytes  1.05 Mbits/sec  1.818 ms  0/ 89 (0%)
[156] 9.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec  1.856 ms  0/ 893 (0%)
[156] local 192.168.0.103 port 5001 connected with 192.168.0.101 port 43907
[ ID] Interval      Transfer     Bandwidth    Jitter    Lost/Total Datagrams
[156] 0.0-1.0 sec   118 KBytes  964 Kbits/sec  12.954 ms  0/ 82 (0%)
[156] 1.0-2.0 sec   112 KBytes  917 Kbits/sec  20.323 ms  0/ 78 (0%)
[156] 2.0-3.0 sec   58.9 KBytes  482 Kbits/sec  33.241 ms  0/ 41 (0%)
[156] 3.0-4.0 sec   37.3 KBytes  306 Kbits/sec  29.263 ms  1/ 27 (3.7%)
[156] 4.0-5.0 sec   51.7 KBytes  423 Kbits/sec  30.811 ms  0/ 36 (0%)
[156] 5.0-6.0 sec   181 KBytes  1.48 Mbits/sec  8.841 ms  0/ 126 (0%)
[156] 6.0-7.0 sec   306 KBytes  2.50 Mbits/sec  7.413 ms  0/ 213 (0%)
[156] 7.0-8.0 sec   131 KBytes  1.07 Mbits/sec  3.453 ms  0/ 91 (0%)
[156] 8.0-9.0 sec   118 KBytes  964 Kbits/sec  5.706 ms  0/ 82 (0%)
[156] 9.0-10.0 sec  138 KBytes  1.13 Mbits/sec  5.704 ms  0/ 96 (0%)
[156] 0.0-10.0 sec  1.22 MBytes  1.03 Mbits/sec  5.465 ms  1/ 873 (0.11%)
```

(b)

**Figura 3-3:** Medición de jitter y pérdida de datos (a) Intervalo de 10s (b) Intervalo de 1s

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Según los resultados que se reflejan en la Figura 3-3(a), existe 0% de pérdida de datos, debido a que la transmisión trabajó con una potencia de -50dBm. Mientras que en la Figura 3-3(b), existen varios paquetes perdidos trabajando a una potencia de -80 dBm. En la Tabla 5-3, se detalla el valor promedio y tiempo de respuesta de la red para el envío de paquetes UDP en las 3 peticiones.

Cabe mencionar que la conexión a la red va a depender de la cantidad de dispositivos que estén conectados a ella y el tipo de servicio que utilicen.

**Tabla 5-3:** Promedio de valores de la red

<b>Muestras</b>	<b>N° Peticiones</b>	<b>% Datos perdidos</b>	<b>Jitter (ms)</b>
<b>1</b>	10	0.11	7.842
<b>2</b>	10	0	1.818
<b>3</b>	10	0.11	5.465
<b>Total</b>	30	0.22	15.125
<b>Promedio</b>	-	0.073	5.04

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Se puede observar que el 0.07% representa al total de pérdida de paquetes de 1479 bytes que transmite la red, llegando a la conclusión, que los paquetes perdidos en las 30 peticiones realizadas, es menor al 1%, obteniendo una calidad de enlace buena, completando toda la transmisión de la información, en un promedio de 5.04 ms y 0.168 ms por petición, siendo imperceptible al usuario en distinguir la latencia.

- Pruebas de ancho de banda (AB)

Representa la cantidad de información que se puede enviar y/o recibir en un tiempo determinado, donde su ancho de información dependerá del servicio generado o solicitado por algún dispositivo. La velocidad de conexión de los módulos ESP8266 NodeMCU v3, según el fabricante y el estándar al que trabajan, suelen tener velocidades de 11 Mbps, 54 Mbps y 600Mbps, pero para una conexión estable, el requerimiento mínimo del módulo es de 900 Kbps.

Por lo tanto, como los valores de consumo de agua potable emitido por el medidor inteligente no son tan grandes no van a ocupar mucho ancho de banda y la tasa de transferencia disminuirá considerablemente. En la Figura 4-3, se puede observar cuanto ancho de banda se ocupa según el tamaño del paquete. Para este caso, se hizo la prueba con un paquete TCP de tamaño de 256KByte, con 3 peticiones al servidor *Iperf* que escucha por el puerto 5001, en un intervalo de 10 segundos por petición.



```

Server listening on TCP port 5001
TCP window size: 256 KByte (default)
-----
Client connecting to 192.168.0.3, TCP port 5001
TCP window size: 216 KByte (default)
-----
[ 31] local 192.168.0.101 port 37632 connected with 192.168.0.3 port 5001
Waiting for server threads to complete. Interrupt again to force quit.
[ ID] Interval  Transfer  Bandwidth
[ 31] 0.0-10.1 sec 22.2 MBytes 18.5 Mbits/sec
[ ID] Interval  Transfer  Bandwidth
[ 31] 0.0-10.0 sec 21.0 MBytes 17.6 Mbits/sec
[ ID] Interval  Transfer  Bandwidth
[ 31] 0.0-10.0 sec 20.5 MBytes 17.2 Mbits/sec

```

```

C:\>iperf.exe -s
-----
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
-----
OpenSCManager failed - Acceso denegado. (0x5)
[316] local 192.168.0.3 port 5001 connected with 192.168.0.101 port 37632
[ ID] Interval  Transfer  Bandwidth
[316] 0.0-10.1 sec 22.3 MBytes 18.4 Mbits/sec
connect failed: Connection refused
C:\>iperf.exe -s
-----
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
-----
OpenSCManager failed - Acceso denegado. (0x5)
[228] local 192.168.0.3 port 5001 connected with 192.168.0.101 port 37632
[ ID] Interval  Transfer  Bandwidth
[228] 0.0-10.1 sec 21.0 MBytes 17.4 Mbits/sec
connect failed: Connection refused
C:\>iperf.exe -s
-----
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
-----
OpenSCManager failed - Acceso denegado. (0x5)
[140] local 192.168.0.3 port 5001 connected with 192.168.0.101 port 37632
[ ID] Interval  Transfer  Bandwidth
[140] 0.0-10.1 sec 20.5 MBytes 17.1 Mbits/sec
connect failed: Connection refused

```

**Figura 4-3:** Pruebas de ancho de banda

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los resultados muestran que cuando se envía un paquete TCP de 256 KBytes, en la primera petición el servidor responde transmitiendo 22.3 MBytes/10s, la segunda 21 MBytes/10s y la tercera 20.5 MBytes/10s. En la Tabla 6-3, se observa que al realizar las pruebas de ancho de banda cliente-servidor, se determina que en promedio las 3 peticiones envían un total de 21.23 MB de información transmitido por el cliente y de igual manera el servidor, con un ancho de banda de 17.63 Mbps.

**Tabla 6-3:** Capacidad, tiempo y ancho de banda

Peticiones	Tamaño transferido [MB]	Tiempo de duración del paquete [s]	Ancho de banda Cliente [Mbps]	Ancho de banda Servidor [Mbps]
1	22.2	10	18.5	18.4
2	21	10	17.6	17.4
3	20.5	10	17.2	17.1
<b>Total</b>	63.7	30	53.3	52.9
<b>Promedio</b>	21.23	-	17.76	17.63

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Por lo tanto, como se tiene un ancho de banda que supera los 17Mbps, se cumple con el requerimiento mínimo de 900Kbps para una conexión estable entre el servidor *Broker MQTT* y el medidor inteligente, teniendo una red robusta y sin pérdidas de bits en la comunicación.

En la Figura 5-3, se aprecia la conexión del medidor inteligente a la red, donde el módulo al detectar el AP, gracias al SSID y password de la red, se conecta directamente a él por medio de transmisión inalámbrica, cuando le otorga una dirección IP por DHCP (*Dynamic Host Configuration Protocol, por sus siglas en inglés*).



**Figura 5-3:** Conectividad del medidor inteligente al AP

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

### 3.1.2.2 Pruebas de retardo en la transmisión de consumo del medidor inteligente

La Tabla 7-3, presenta 4 muestras para realizar pruebas de tiempo en que el medidor inteligente se demora en publicar al servidor *Broker*, es decir, una vez que detecta el pulso, en que tiempo se envía la información.

**Tabla 7-3:** Tiempo en que el medidor inteligente publica de los datos

Muestras	Tiempo de detección [ms]	Tiempo de envío [ms]	Tiempo de retardo [ms]
1	56711	57915	1204
2	16912	18096	1184
3	30191	31378	1187
4	3328	4596	1268
<b>Promedio</b>	-	-	<b>1192</b>

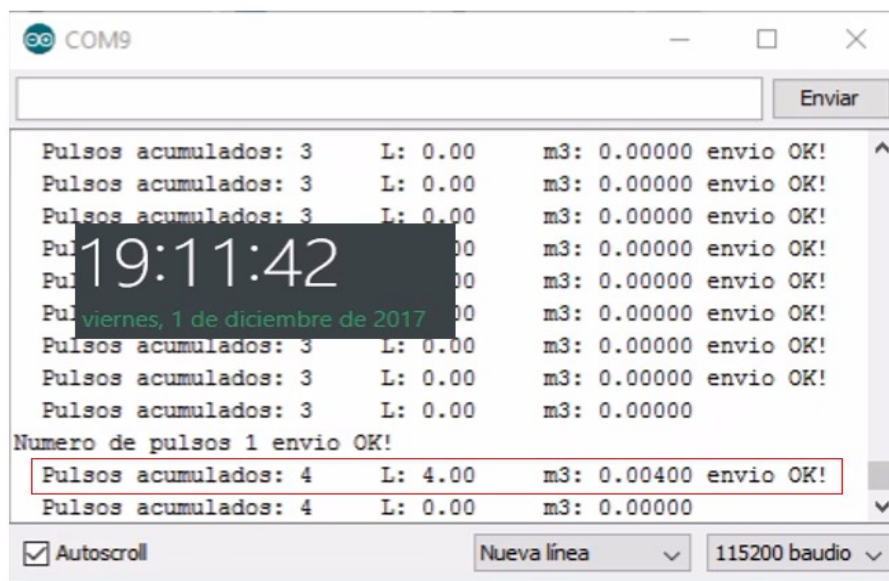
Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los resultados demuestran que cada valor de consumo de agua potable que es detectado por el medidor inteligente, tiene un promedio de publicación de 1192 ms, lo que significa que es un valor adecuado y eficiente para transmitir.

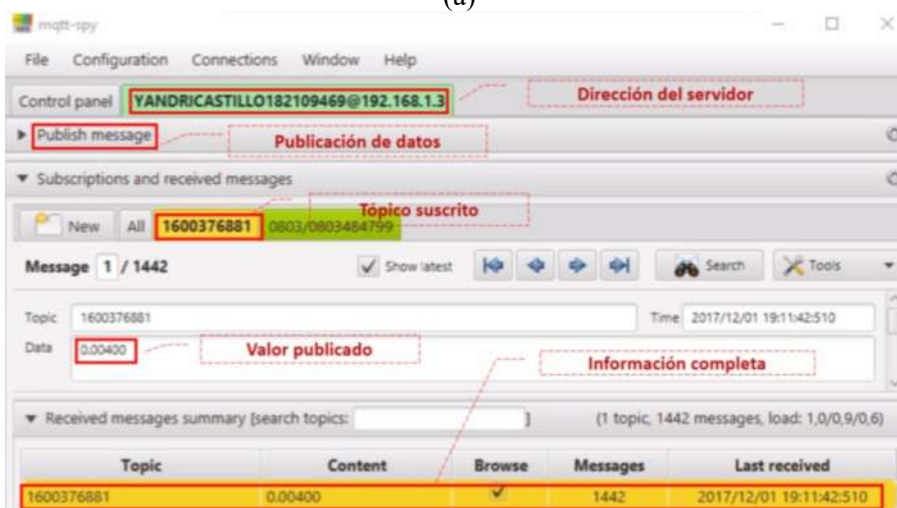
### 3.1.2.3 Pruebas de retardo medidor inteligente - servidor Broker MQTT

El medidor inteligente se conecta al servidor *Broker* MQTT, publicando sus valores de consumo de agua potable. Para verificar que los datos están siendo transmitidos en tiempo real, hay una aplicación otorgada por Servidor, llamada *mqtt-spy*, destinada a ayudar al administrador a supervisar la actividad de los tópicos MQTT.

En la Figura 6-3(b), se puede observar la aplicación y sus paneles donde suscribe y publica valores referentes a un tópico (Ej. 1600376881), mientras que la Figura 6-3(a), muestra los valores que publica el medidor inteligente y la hora en que publica.



(a)



(b)

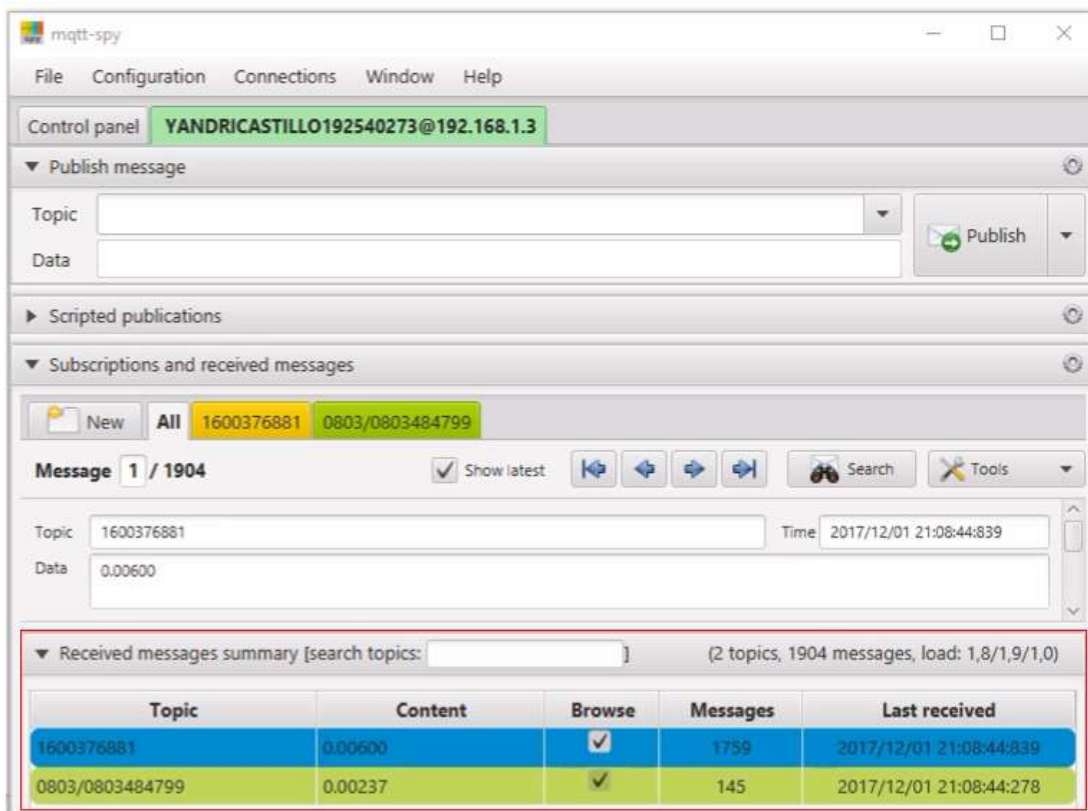
**Figura 6-3:** Transmisión en tiempo real (a) Medidor inteligente (b) Interfaz gráfica del servidor Broker MQTT

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

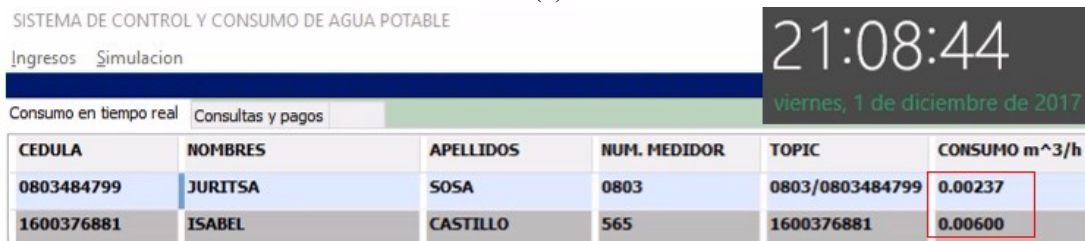
Al visualizar ambas figuras los resultados indican, que los valores publicados del medidor inteligente al servidor Broker MQTT, inmediatamente se reflejan en él sin ningún tipo de retardo ya transmite en el mismo tiempo, tomando en cuenta que se está trabajando en una red local y el envío de los datos de dos dispositivos inteligentes.

### 3.1.2.4 Pruebas de retardo servidor Broker MQTT - GUI

En la Figura 7-3(a), se puede apreciar la interfaz gráfica del servidor *Broker* MQTT, el tiempo en que recibe los datos del medidor inteligente y los valores de consumo de agua y la Figura 7-3(b) de igual manera, solo que los son mostrados en la GUI.



(a)



(b)

**Figura 7-3:** Transmisión en tiempo real (a) Interfaz gráfica de servidor *Broker* MQTT (b) Interfaz Gráfica de Usuario (GUI).

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los resultados muestran que el tiempo de respuesta es inmediato al ser publicados los valores de consumo de agua potable desde el servidor *Broker* MQTT a la GUI de la empresa, lo que significa, que no existe retardo en la transmisión. De igual manera se debe tomar en cuenta que se trabaja en una red local.

### 3.1.2.5 Pruebas de almacenamiento de datos en MySQL

En la Figura 8-3, se puede observar el almacenamiento de datos, una vez que los valores de consumo de agua potable son vinculados a la GUI, como se muestra en la Figura 7-3(b). El tiempo de almacenamiento puede variar dependiendo como se transmiten los datos, en este caso se asigna un *timer* de 1s debido a que el servidor *Broker* MQTT publica los datos en ese tiempo, y también se considera para mayor exactitud en la recolección de datos.

idUsuario	numMedidor	consumo	Pago	mesConsumo	fecha	estado	
0803484799	0803	0803/0803484799	0.00237		diciembre	1/12/2017 21:08:45	NO PAGADO
0803484799	0803	0803/0803484799	0.00198		diciembre	1/12/2017 21:08:19	NO PAGADO
0803484799	0803	0803/0803484799	0.00237		diciembre	1/12/2017 21:08:45	NO PAGADO
0803484799	0803	0803/0803484799	0.00175		noviembre	30/11/2017 11:01:09	NO PAGADO
0803484799	0803	0803/0803484799	0.00180		noviembre	30/11/2017 11:01:11	NO PAGADO
0803484799	0803	0803/0803484799	0.00188		noviembre	30/11/2017 11:01:15	NO PAGADO
0803484799	0803	0803/0803484799	0.00193		noviembre	30/11/2017 11:01:17	NO PAGADO
0803484799	0803	0803/0803484799	0.00198		noviembre	30/11/2017 11:01:19	NO PAGADO
0803484799	0803	0803/0803484799	0.00200		noviembre	30/11/2017 11:01:20	NO PAGADO
1600376881	565	1600376881	0.00600		diciembre	1/12/2017 21:08:45	NO PAGADO
1600376881	565	1600376881	0.00500		diciembre	1/12/2017 21:08:19	NO PAGADO
1600376881	565	1600376881	0.00600		diciembre	1/12/2017 21:08:45	NO PAGADO
1600376881	565	1600376881	0.00800	(full)	noviembre	24/11/2017 10:32:51	NO PAGADO

**Figura 8-3:** Almacenamiento de valores de consumo de agua potable en MySQL

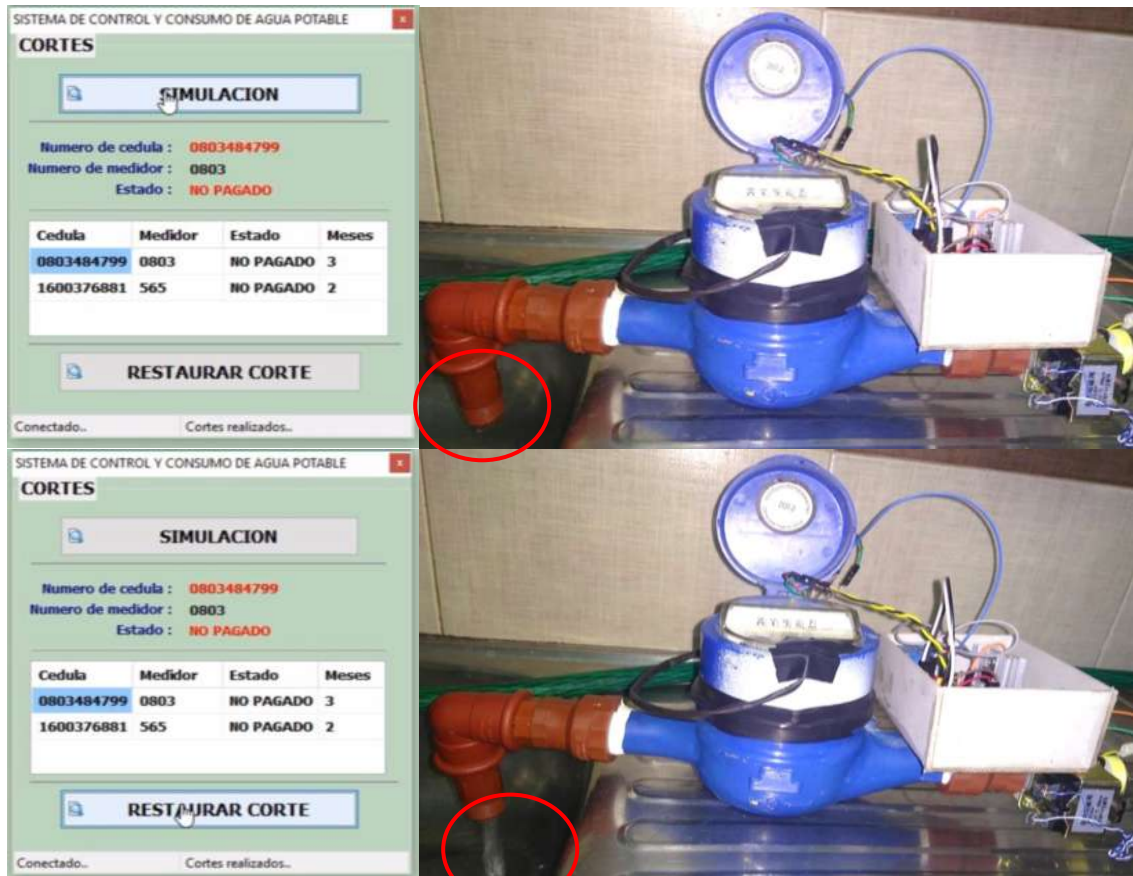
Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los resultados reflejan que los datos se almacenan después de 1 segundo, de acuerdo a la programación realizada en Delphi Starter, garantizando un almacenamiento sin pérdida de los mismos y de manera casi inmediata.

### 3.1.3 Caracterización de la etapa de monitorización y control de agua potable

La monitorización de los datos emitidos por el medidor inteligente, se pueden visualizar en tiempo real en la GUI y se pueden realizar consultas de un consumo acumulado.

Dependiendo de la empresa se pueden analizar los datos para tomar medidas que resuelvan problemas remotamente. Se realiza una pequeña interfaz, para muestras de facturación y una para realizar una simulación de corte y reconexión de agua potable. En la Figura 9-3, se puede apreciar el corte y reactivación del agua desde la interfaz gráfica, que se realiza al mandar un pulso a la electroválvula mediante el módulo ESP8266 NodeMCU v3.



**Figura 9-3:** Interfaz de simulación (a) Corte (b) Reconexión de agua potable

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

### 3.1.3.1 Pruebas de tiempo de respuesta

En la Tabla 8-3, se puede observar 5 pruebas que se realizaron para corte y reconexión de agua potable, determinando el tiempo en que se demora ejecutando la acción.

**Tabla 8-3:** Tiempo de conexión /reconexión de agua potable

<b>Muestras</b>	<b>Tiempo de Corte [s]</b>	<b>Tiempo de reconexión [s]</b>
<b>1</b>	1.18	1.33
<b>2</b>	0.97	0.71
<b>3</b>	0.6	0.3
<b>4</b>	0.55	0.7
<b>5</b>	0.89	0.2
<b>Promedio</b>	<b>1.39</b>	<b>1.08</b>

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

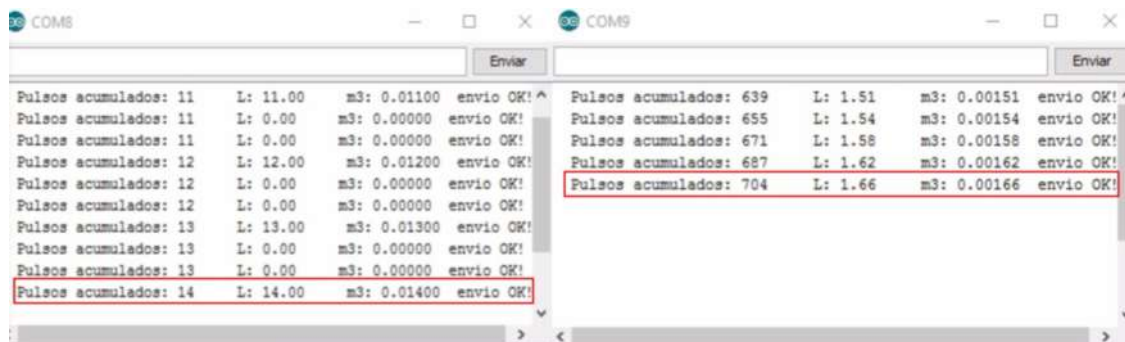
En los resultados refleja que por corte de agua potable se demorará un promedio de 1.39 s en activarse el evento, mientras que para reconexión se tiene un tiempo de 1.08 s en que la válvula permita el paso de agua, determinando que es un tiempo considerable y eficiente para el tiempo en que se demora el personal en activar el medidor convencional para el paso de agua que es dentro de 24 horas.

### 3.2 Pruebas con una plataforma en la nube para IoT

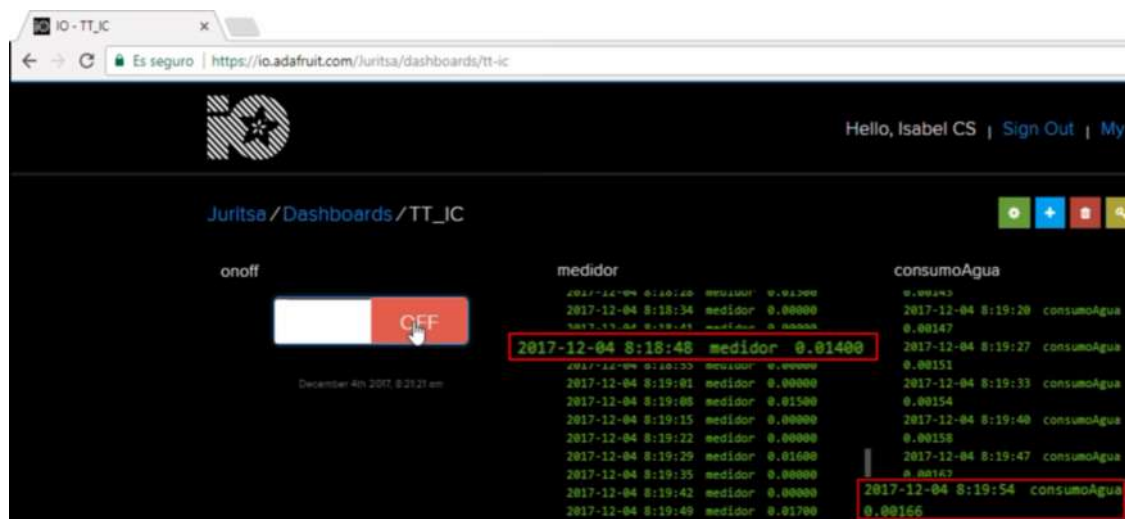
Para demostrar que la RCMAP puede funcionar en fuera de una red local, existen varias plataformas en la nube con licencia gratuita, de código abierto o de pago, que son diseñadas para el Internet de las Cosas y que trabajan bajo el protocolo MQTT en el transporte de datos muy ligeros, como: Adafruit IO, Arduino Cloud, entre otros o también se puede utilizar servidores *Brokers* en la nube como Cloud MQTT, facilitando el manejo y gestión de datos mediante una API (*Application Programming Interface*, por sus siglas en inglés).

Para este caso se ha utilizado la plataforma en la nube Adafruit IO, que además de trabajar con dispositivos que hablan MQTT también funciona con los de HTTP. Tiene algunos planes de pago que dependiendo del elegido trabaja con diferentes números de nodos, en este trabajo se utiliza el plan gratuito que se limita a 5 nodos y un almacenamiento que se renueva cada 24 horas. (<https://io.adafruit.com/blog/>, 2017).

En la Figura 10-3, se puede apreciar las transmisión en tiempo real del consumo de agua, donde la Figura 10-3(a), representa los valores publicados por el medidor inteligente que se muestran en el monitor serie de Arduino y la Figura 10-3(b), la plataforma de Adafruit IO, donde se puede visualizar un bloque de los valores de medición y un bloque para realizar algún tipo de control como corte/reconexión de agua potable. En esta plataforma se puede utilizar varios bloques dependiendo la aplicación que se requiera realizar.



(a)



(b)

**Figura 10-3:** Transmisión en tiempo real (a) Medidor inteligente (b) API de la plataforma IoT de Adafruit IO

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los resultados muestran que todos los valores que son publicados por el medidor inteligente son reflejados en la plataforma de Adafruit IO, sin considerar el tiempo de retardo en que son receptados. En la Tabla 9-3, se presenta 3 muestras para considerar tiempo de retardo promedio en que los valores de consumo de agua potable llegan a la plataforma, considerando un dispositivo de transmisión.

**Tabla 9-3:** Tiempo de retardo en que los valores de consumo de agua potable llegan a la plataforma Adafruit IO

Muestras	Tiempo de envío [ms]	Tiempo de recepción [ms]	Tiempo de retardo [ms]
1	57915	59383	1468
2	18180	19584	1404
3	31295	32980	1685
<b>Promedio</b>	-	-	<b>1519</b>

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017



Los resultados determinan que existe un retardo promedio de 1519 ms en que se reciben los datos en la plataforma, realizando una comparación con el tiempo de retardo que se presenta al trabajar en una red local con el servidor Broker, se verifica que no varía considerablemente y el retardo dependerá de la distancia a la plataforma Adafruit IO o servidor en función y los tiempos de retardo que maneje el proveedor servicio de Internet. En la Tabla 10-3, se establecieron 5 muestras para determinar el tiempo de retardo en que la válvula solenoide se active para el corte/reconexión de agua potable.

**Tabla 10-3:** Tiempo de retardo en corte/reconexión de agua potable

Muestras	Tiempo de Corte [s]	Tiempo de reconexión [s]
1	0.53	0.65
2	0.6	0.4
3	0.61	0.6
4	0.37	0.61
5	0.62	0.63
<b>Promedio</b>	<b>0.55</b>	<b>0.58</b>

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los valores generados demuestran, un tiempo promedio de corte de agua de 0.55s, mientras que para reconectar un tiempo de 0.58s, comparando con los valores utilizando el servidor *Broker*, que se muestran en la Tabla 8-3, el tiempo es menor y más rápida al gestionar la acción, comprobando que los servidores o plataformas en la nube proporcionan mejores tiempos de respuesta agilizando procesos de gestión.

Las plataformas o servidores en la nube, tienen diferentes maneras de crear los eventos o instrucciones que se desea realizar, ya sea diseñando o incorporando los elementos para futuras aplicaciones.

### 3.3 Costos de un medidor inteligente

En la Tabla 11-3, se puede observar el costo del planteo de distintas maneras de un medidor inteligente, donde la utilización de elementos ya existentes como son los medidores convencionales disminuye el gasto económico. También el costo aproximado que conllevaría la implementación de un proyecto piloto de 1000 medidores.

**Tabla 11-3:** Precio de un medidor inteligente junto a herramientas de hardware y software que cumplen con la transmisión de datos en tiempo real.

<b>Costo para realizar pruebas</b>				
<b>Sistema de lectura remota</b>	<b>Aspectos</b>	<b>Cant.</b>	<b>Precio/ unidad (\$)</b>	<b>Precio total</b>
<b>Compra</b>	Medidor inalámbrico AMR de agua potable (Ver Anexo J)	2	60.00	120.00
	Concentrador de datos	1	460.00	460.00
	Software de gestión	-	N/A	N/A
	Costo de envío	-	158.00	158.00
	<b>Costo Total</b>	-	<b>678.00</b>	<b>738.00</b>
<b>Construcción</b>	Sensor de flujo YF-S201	1	10.00	10.00
	Válvula solenoide 1/2"	1	9.00	9.00
	Módulo ESP8266 NodeMCU v3	1	11.00	11.00
	Componentes electrónicos	1	10.00	10.00
	Costo del medidor inteligente	-	40.00	40.00
	Software de gestión (Delphi 10.2 Tokio Professional, ver Anexo K)	1	1405.00	1405.00
	<b>Costo Total</b>	-	<b>1445.00</b>	<b>1445.00</b>
<b>Incorporación</b>	Sensor de efecto hall	1	3.00	3.00
	Válvula solenoide 1/2"	1	9.00	9.00
	Módulo ESP8266 NodeMCU v3	1	11.00	11.00
	Componentes electrónicos	1	10.00	10.00
	<b>Costo Total</b>	-	<b>33.00</b>	<b>33.00</b>
<b>Costo para implementación real</b>				
<b>Compra</b>	Medidor inalámbrico AMR de agua potable (Ver Anexo J)	1000	53.00	53000.00
	Concentrador de datos (100 nodos)	10	460.00	4600.00
	Software de gestión	1	5000	5000
	Costo de envío	-	500.00	500.00
	<b>Costo Total</b>	-	<b>6013.00</b>	<b>63100.00</b>
<b>Construcción</b>	Medidor Inteligente (Sensor de flujo YF-S201, electroválvula, módulo ESP8266)	1000	40.00	40000.00
	Software de gestión (Delphi 10.2 Tokio Professional, ver Anexo K)	1	1405.00	1405.00
	<b>Costo Total</b>	-	<b>1445.00</b>	<b>41405.00</b>
<b>Incorporación</b>	Medidor Inteligente (Sensor de efecto hall, electroválvula, Módulo ESP8266)	1000	33.00	33000.00
	Software de gestión (Delphi 10.2 Tokio Professional, ver Anexo K)	1	1405.00	1405.00
	<b>Costo Total</b>	-	<b>33.00</b>	<b>34405.00</b>

Realizado por: CASTILLO Isabel & MURILLO Santiago. 2017

Los resultados indican que construyendo el medidor inteligente usando el medidor convencional que se encuentra instalado en los domicilios de la parroquia Yaruquies, un software para gestionar los valores de consumo y un servidor de código abierto con su máxima seguridad, otorga un costo considerable para implementar la red de comunicación, sin descartar la posibilidad de realizar la implementación del prototipo de medidor inteligente que es la segunda opción o comprar un sistema de lectura remota ya que puede garantizar mayor seguridad y menor tiempo de retardo en la transmisión de datos, aunque resulte más caro.

## CONCLUSIONES

Se diseñó una red de comunicaciones para la medición y control de agua potable de parroquia de Yaruquíes, en base a la infraestructura que presenta actualmente. Opera en tres etapas: medición, transmisión de datos y monitorización y control del consumo de agua potable mediante comunicación inalámbrica Wi-Fi y la infraestructura de Internet ya existente, agilizando los procesos de transmisión y recepción de la información. Evita la demora y errores en lectura y registro siendo una herramienta de ayuda para la empresa.

Al realizar la simulación de la red con 19 módems y 200 usuarios, mediante las pruebas se obtuvo que el 82.4% del total de usuarios pueden conectarse a un modem disponible y solo el 17.6% no alcanzan cobertura, por lo tanto, se concluye que la mayoría de usuarios podrán disponer del servicio siempre y cuando se encuentre dentro de la cobertura de un AP, incidiendo positivamente en la implementación del prototipo.

Se implementó un medidor inteligente partiendo del medidor convencional instalado en la parroquia de Yaruquíes, al modificar su diseño para medir caudales menores e incorporando un sensor de efecto hall A3144, se determinó que los errores promedio de lectura es de 0.06 litros por cada 10 litros y es congruente con el error del medidor antes de su modificación.

Al realizar las pruebas de conectividad, se determinó que el medidor inteligente desarrollado alcanza una cobertura máxima de 18m con una potencia de la señal de -80 dBm. También que, para una buena calidad de enlace, se requiere un ancho de banda mayor a 900 Kbps y una latencia menor al 1%. Se logró un ancho de banda de 17Mbps determinando que no existe pérdida de paquetes y una latencia de 0.07% con un tiempo de 0.168ms siendo imperceptible al usuario. Por lo tanto, se comprobó que existe una conexión estable entre el medidor inteligente y el servidor Broker MQTT.

Para el tiempo de retardo en la transmisión datos y control del medidor inteligente desarrollado, se determinó que trabajando en una red LAN, se demora 1192ms para publicar el consumo de agua potable en el servidor Broker MQTT; para corte y reconexión de agua se tiende 1.39s y 1.08s respectivamente y para el almacenamiento de la información es 1s después de que se visualicen en la GUI. Por lo tanto, se concluye que los tiempos encontrados son imperceptibles para el usuario final.

Se comprueba que la red de comunicaciones implementada junto al medidor inteligente desarrollado a diferencia de la forma de lectura tradicional, optimiza tiempos y procesos de operación, siendo económica y completamente escalable debido a que permite ampliar el número de medidores sin afectar su estructura.

## **RECOMENDACIONES**

Para que los medidores inteligentes, no pierdan conexión a la red, se puede evaluar y posteriormente implementar enrutadores inalámbricos adicionales, en especial en zonas donde la intensidad de señal sea más baja.

Estudiar la posibilidad de usar servidores en la nube que trabajen junto con el protocolo MQTT, ya que resulta una manera económica y factible para los diferentes eventos que se pueden ejecutar y así aprovechar las ventajas que ofrece el Internet de las cosas.

Este tipo de diseño de red según su funcionamiento, puede ser aplicado en otras ciudades, donde se disponga de servicio de Internet, y la cuente con una red de medidores similares a los mencionados en este documento.

La alimentación del medidor inteligente debe ser de acuerdo al consumo promedio de corriente y voltaje, para su funcionamiento óptimo y así evitar sobrecalentamientos y daños en sus componentes.

Para proyectos futuros se debe incorporar otro tipo de comunicación adicional que permita la redundancia de red e instrucciones tales como: enviar una alarma sobre fuga de agua, pérdida de energía eléctrica del medidor, fallos en la transmisión de datos y otras. También implementar proyectos piloto en colaboración con la empresa proveedora del servicio de agua potable.

## BIBLIOGRAFÍA

Adafruit IO Basics: ESP8266 + Arduino [en línea]. Todd Treece, 2016. [Consulta: 23 octubre 2017]. Disponible en: <https://learn.adafruit.com/adafruit-io-basics-esp8266-arduino/overview>

**ALDAS COLLAGUAZO, Alejandro David.** *Interoperabilidad entre medidores inteligentes de energía eléctrica residencial para el DMQ bajo las normas ANSI*. [en línea] (Tesis de pregrado). Universidad Politécnica Salesiana SEDE Quito, Ingeniería Eléctrica. Quito-Ecuador. 2015. p.2 [Consulta: 2016-06-15]. Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/8889/1/UPS-KT01086.pdf>

**AKSHAY KANWAR, Aditi.** “ZigBee: The New Bluetooth Technology”. *International Journal of Engineering and Computer Science* [en línea], 2012, (India) 1, pp. 67-68. [Consulta: 15 octubre 2017]. ISSN 2319-7242. Disponible en: <http://www.ijecs.in/ijecsissue/wp-content/uploads/2012/12/67-74.pdf>

**ARADGROUP Integrated Metering Technologies.** “Soluciones AMR/AMI” [en línea]. 2014. [Consulta: 14 junio 2016]. Disponible en: <http://arad.co.il/assets/ARAD-Dialog3G-ESP-December-2014.pdf>

**BACZKOWICZ, Kamil.** *MQTT Toolbox – mqtt-spy* [en línea]. 2016. [Consulta: 23 octubre 2017]. Disponible en: <https://www.hivemq.com/blog/mqtt-toolbox-mqtt-spy>

**CAICEDO EDUARDO, Rodriguez Cristian.** *Infraestructura de Medición Avanzada* [en línea]. Colombia: Universidad del Valle, enero 2014. [Consulta: 15 junio 2016]. Disponible en: <https://campusvirtual.univalle.edu.co/moodle/course/info.php?id=19617>

**CALDEVILLA DOMÍNGUEZ, David.** *LA COMUNICACIÓN A TRAVÉS DE LA HISTORIA* [En línea]. España – Madrid: ACCI, 2013. [Consulta: 16 abril 2017]. Disponible en: [https://books.google.com.ec/books?id=N8HOBAAAQBAJ&pg=PT50&lpg=PT50&dq=historia+de+la+red+de+comunicacion&source=bl&ots=zftzu3jafA&sig=\\_yuko8cT18rrjJMuEvLboY0nsak&hl=en&sa=X&ved=0ahUKEwig2or03bbTAhUJxCYKHRZYBAc4ChDoAQhRMAY#v=onepage&q&f=false](https://books.google.com.ec/books?id=N8HOBAAAQBAJ&pg=PT50&lpg=PT50&dq=historia+de+la+red+de+comunicacion&source=bl&ots=zftzu3jafA&sig=_yuko8cT18rrjJMuEvLboY0nsak&hl=en&sa=X&ved=0ahUKEwig2or03bbTAhUJxCYKHRZYBAc4ChDoAQhRMAY#v=onepage&q&f=false)

**CHIRINOS GARCÍA, E, PEROZO SIERRALTA, B, PELAYO LÁZARO, R.** “Fundamentos teóricos para la comunicación a través de una red eléctrica como alternativa de acceso a la Internet”. *Multiciencias* [en línea], 2014, (Venezuela) 14(3), pp. 329-330. [Consulta:

15 octubre 2017]. ISSN 1317-2255. Disponible en: <http://www.redalyc.org/articulo.oa?id=90432809012>

*Delphi Starter Info - Embarcadero Website* [en línea]. Embarcadero Technologies, 2017. [Consulta: 23 octubre 2017]. Disponible en: <https://www.embarcadero.com/products/delphi/starter/info>

**ECHENIQUE, Yesirée.; & RIVERO, Rafael.** “Diseño y desarrollo de un módulo de recolección, almacenamiento y transmisión de datos a través de la red celular para contadores electrónicos monofásicos de energía eléctrica”. *Facultad de Ingeniería Universidad Central de Venezuela*, n° 4 (2012), (Venezuela) pp. 65-74.

*ESP8266–WiFi library on Arduino IDE* [en línea]. alselectro, 2016. [Consulta: 23 octubre 2017]. Disponible en: <https://alselectro.wordpress.com/2016/11/29/esp8266-wifi-library-on-arduino-ide/>

*Groups, Feeds, and Proper MQTT Topics* [blog]. 2017. [Consulta: 31 octubre 2017]. Disponible en: <https://io.adafruit.com/blog/>

**HUANG, Qi, et al:** *Innovative testing and measurement solutions for smart grid*. Sichuan Electric Power Research Institute, China: John Wiley & Sons Singapore Pte, 2015, pp.148-149.

“*Innovations in Optical Fiber Technology*” [en línea]. USA: 2017. [Consulta: 15 octubre 2017]. Disponible en: [www.anixter.com/en\\_us/resources/literature/techbriefs/innovations-in-optical-fiber.html](http://www.anixter.com/en_us/resources/literature/techbriefs/innovations-in-optical-fiber.html)

**JÁCOME RIERA, Omayra Jaqueline., & LEÓN AMORES, José Gabriel.** *Diseño y construcción de un sistema de medición inteligente para la gestión de la energía eléctrica y medición de agua potable en los hogares* [en línea] (Proyecto de titulación). Universidad de la Fuerzas Armadas ESPE Extensión Latacunga, Carrera de Ingeniería Electrónica e Instrumentación. Latacunga – Ecuador. 2014. p. xvi [Consulta: 2016-06-14]. Disponible en: <https://repositorio.espe.edu.ec/bitstream/21000/8293/1/T-ESPEL-ENI-0321.pdf>

**MONTESDEOCA ESPÍN, Sandro Zenón.** *DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE LECTURA AUTOMÁTICA DE PARÁMETROS ELÉCTRICOS (AMR AUTOMATIC METER READING) PARA MEDIDORES DE ENERGÍA ELÉCTRICA DE TIPO INDUSTRIAL A TRAVÉS DE GPRS* [en línea] (Trabajo de titulación). Escuela Politécnica

Nacional, Facultad de ingeniería eléctrica y electrónica0, Escuela de Ingeniería en Electrónica y Telecomunicaciones. Quito - Ecuador. 2012. pp. 4-8, p. 10. [Consulta: 2017-04-18]. Disponible en: [bibdigital.epn.edu.ec/bitstream/15000/4450/1/CD-4059.pdf](http://bibdigital.epn.edu.ec/bitstream/15000/4450/1/CD-4059.pdf)

**MORALES BARROSO, José.** *Guía de Redes Inteligentes de energía y comunicación, La <<Red Inteligente>> de energía y comunicaciones. Generalidades y visión de futuro* [en línea]. Madrid. [Consulta: 17 de abril del 2017]. Disponible en: <https://www.fenercom.com/pdf/publicaciones/Guia-de-redes-inteligentes-de-energia-y-comunicacion-fenercom-2011.pdf>

**MORENO MARTÍN, Marta.** *Análisis, diseño y despliegue de una red WiFi en Santillana del Mar* [en línea] (Tesis). Universidad Autónoma de Madrid, Departamento de Tecnología Electrónica y de las Comunicaciones, España, Madrid. 2015. p.7. [Consulta: 2017-10-15]. Disponible en: <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20150123MartaMorenoMartin.pdf>

*NodeMCU y el IoT tutorial paso a paso desde cero* [en línea]. PROGRAMAR FÁCIL, 2016. [Consulta: 23 octubre 2017]. Disponible en: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

*PRIMEROS PASOS CON MQTT* [en línea]. Ricardo Vega, 2016. [Consulta: 23 octubre 2017]. Disponible en: <https://ricveal.com/blog/primeros-pasos-mqtt/>

**QUISHPE, Marcelo.** *TELEMETRIA O TELELECTURA PARA MEDIDORES DE AGUA POTABLE EN EL CENTRO DEL DMQ* [en línea] (Tesis de maestría). Instituto de Altos Estudios Nacionales, Facultad de Gerencia Empresarial. Centro del DMQ (Quito-Ecuador). 2008. p.1 [Consulta: 2016-06-14]. Disponible en: <http://repositorio.iaen.edu.ec/bitstream/24000/215/1/IAEN-013-2008.pdf>

*Sensor Medidor de Flujo YF-S201* [en línea]. 2016. [Consulta: 23 octubre 2017]. Disponible en: <https://www.vistronica.com/sensores/caudal/sensor-medidor-de-flujo-yf-s201-detail.html>

**STALLINGS, William.** *Comunicaciones y redes de computadoras.7.* Madrid - España: PEARSON EDUCACIÓN, 2004, pp. 10, 12, 14 -15, 597.



**URBINA GAMBOA, Wilmer Marcelo.** *Estudio y diseño de un sistema de medición remota utilizando sensores inalámbricos bajo el estándar 802.15.4 para los servicios de energía eléctrica y agua potable* [en línea] (Tesis de Maestría). PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR, FACULTAD DE INGENIERÍA MAESTRÍA EN REDES DE COMUNICACIONES. Quito-Ecuador. 2014. [Consulta: 2016-06-14]. Disponible en: <http://repositorio.puce.edu.ec/bitstream/handle/22000/10917/6.16.001606.pdf;sequence=4>

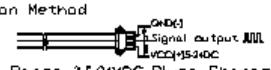
*Why MySQL?* [en línea]. Oracle Corporation, 2017. [Consulta: 23 octubre 2017]. Disponible en: <https://www.mysql.com/why-mysql/>

## ANEXOS

### Anexo A. Sensor de Flujo YF-S201

**YIFA the plastics Ltd**  
Product Introduction

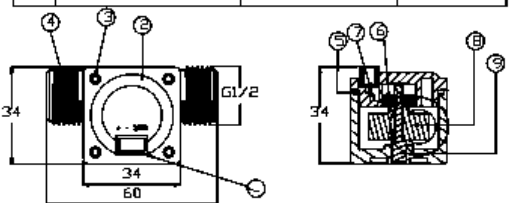
1. Model: YF-21  
2. Product Name: Hall sensor  
3. Flow Range: 1-30L/MIN  
4. (1) Connection Method



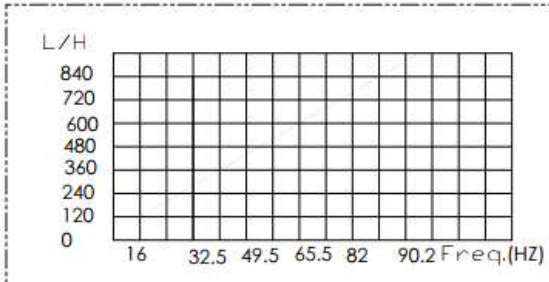
(2) Voltage Range 3.5-24VDC, Pulse Characteristic: F=7Q(L/MIN)  
(3) Extent of error: ±5%  
(4) Flow-Pulse  
2L/MIN=16HZ 4L/MIN=32.5HZ 6L/MIN=49.3HZ  
8L/MIN=65.5HZ 10L/MIN=82HZ

5. BOM

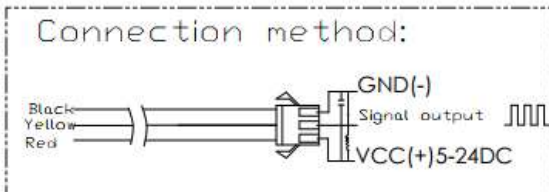
No.	Item	Material	Qty.
1	Connection wire		1
2	Bonnet	PA	1
3	Screw		4
4	Valve body	PA	1
5	Leak press valve		1
6	Magnet		1
7	Hall		1
8	Impeller	POM	1
9	Rustless steel axis	SUS304	1
10			
11			



Flow-Pulse	Flow-Curve	Connection method
Flow Range: 100L/H-1800H-L/H		
Flow (L/H)	Freqz.(HZ)	Error range  ±10
120L/H	16	
240L/H	32.5	
360L/H	49.3	
480L/H	65.5	
600L/H	82	
720L/H	90.2	

Connection method:



### Anexo B. Medidor convencional para lectura remota

#### MEDIDOR DE AGUA MTK 45° CON RELOJERÍA INCLINADA

Variante especial de MTK para situaciones de montaje difíciles



Medidor de agua MTK 45° con relojería inclinada Q3 = 2,5



El MTK 45° es un medidor de chorro múltiple de esfera húmeda para agua fría con relojería inclinada 45°.

Nuestro medidor de agua MTK es un medidor de turbina de chorro múltiple para el registro del consumo de agua en casas y viviendas e inmuebles industriales. La relojería inclinada del acabado especial MTK 45° permite la lectura de los datos del medidor desde diferentes perspectivas.

Este medidor se utiliza siempre que las situaciones de montaje horizontales estrechas impiden la vista horizontal directa de la relojería. Con la relojería inclinada 45° también es posible en este caso realizar una lectura sin problemas.

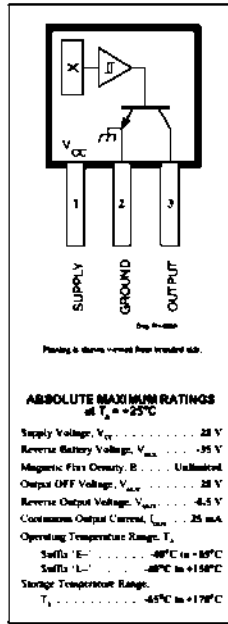
#### Características

- Relojería de esfera seca con 5 rodillos
- Relojería inclinada 45°
- Fase de presión PN 16
- Carcasa también disponible en modelo Copper Can IP 68
- Homologado según MID

#### Características del producto

Longitud contador sin racores	165 mm 190 mm
Material de cuerpo	Latón
Caudal Permanente Q3 (= Qn 1,5) Q3=4 (= Qn 2,5)	Q3=2,5 (= Qn 1,5) Q3=4 (= Qn 2,5)
Diámetro nominal DN (mm)	DN 15 DN 20
Registro	Esfera seca
Diámetro nominal (pulgada)	3/4" 1"
Principio de medición	Medidores de chorro múltiple
Temperatura máxima	Agua fría 30° C Agua fría 50° C
Indicadores	Indicador de metros cúbicos (m³)
Opciones adicionales	Registro de 45° Anillo-Anti-Imanes Nano-cubrimiento (Antifog) Soldadura ultrasónica (IP 67)
Opciones para la lectura remota	Emisor de impulsos (reed)
Clase Metroológica horizontal (max.)	B 80
Presión de servicio (max.)	MAP 16

## Anexo C. Sensor digital de efecto hall A3144



### ELECTRICAL CHARACTERISTICS at $V_{CC} = 8\text{ V}$ over operating temperature range.

Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Units
Supply Voltage	$V_{CC}$	Operating	4.5	—	24	V
Output Saturation Voltage	$V_{OUT(SAT)}$	$I_{OUT} = 20\text{ mA}$ , $B > B_{OP}$	—	175	400	mV
Output Leakage Current	$I_{OFF}$	$V_{OUT} = 24\text{ V}$ , $B < B_{RP}$	—	<1.0	10	$\mu\text{A}$
Supply Current	$I_{CC}$	$B < B_{RP}$ (Output OFF)	—	4.4	9.0	mA
Output Rise Time	$t_r$	$R_L = 820\ \Omega$ , $C_L = 20\ \text{pF}$	—	0.04	2.0	$\mu\text{s}$
Output Fall Time	$t_f$	$R_L = 820\ \Omega$ , $C_L = 20\ \text{pF}$	—	0.18	2.0	$\mu\text{s}$

### MAGNETIC CHARACTERISTICS in gauss over operating supply voltage range.

Characteristic	Part Numbers*											
	A3141-			A3142-			A3143-			A3144-		
	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
$B_{OP}$ at $T_A = 25^\circ\text{C}$	50	100	160	130	180	230	220	280	340	70	—	330
over operating temp. range	30	100	175	115	180	245	205	280	355	35	—	450
$B_{RP}$ at $T_A = 25^\circ\text{C}$	10	45	130	75	125	175	165	225	285	50	—	330
over operating temp. range	10	45	145	80	125	190	150	225	300	25	—	430
$B_{HY}$ at $T_A = 25^\circ\text{C}$	20	55	80	30	55	80	30	55	80	20	55	—
over operating temp. range	20	55	80	30	55	80	30	55	80	20	55	—

NOTES: Typical values are at  $T_A = +25^\circ\text{C}$  and  $V_{CC} = 8\text{ V}$ .

$B_{OP}$  = operate point (output turns ON),  $B_{RP}$  = release point (output turns OFF),  $B_{HY}$  = hysteresis ( $B_{OP} - B_{RP}$ )

\* Complete part number includes a suffix to identify operating temperature range (E- or L-) and package type (-LT, -U, or -UA)

## Anexo D. Módulo ESP8266x

### 1.2. Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power < 10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range  $-40^\circ\text{C} \sim 125^\circ\text{C}$
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

### 1.3. Parameters

Categories	Items	Values
WiFi Parameters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
	802.11 n: +14 dBm	
	Rx Sensitivity	802.11 b: -91 dBm (11 Mbps) 802.11 g: -75 dBm (54 Mbps) 802.11 n: -72 dBm (MCS7)
Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip	
Hardware Parameters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
	GPIO/PWM	
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	$-40^\circ\text{C} \sim 125^\circ\text{C}$
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
External Interface	N/A	
Software Parameters	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
Network Protocols	IPv4, TCP/UDP/HTTP/FTP	

## Anexo E. Router Huawei HG532 – ADSL

### 3.1 Interface Features

#### 3.1.1 DSL Interface

##### Multiple DSL Standards

- ADSL2+
  - Supports G.992.5 (G.dmt.bitplus) Annex A
  - Supports G.992.5 (G.dmt.bitplus) Annex M
- ADSL2
  - Supports G.992.3 (G.dmt.bis) Annex A
  - Supports G.992.3 (G.dmt.bis) Annex L
  - Supports G.992.3 (G.dmt.bis) Annex M
- ADSL
  - Supports G.992.1 (G.dmt) Annex A
  - Supports G.994.1 (G.hs)
  - Supports ANSI T1.413 Issue 2

##### Other Features

- Supports multiple permanent virtual channels (most 8 PVCs)
- Supports manual configuration of PVC parameters

#### 3.1.2 Ethernet Interface

- Provision of four 10/100 M adaptive Ethernet interfaces
- Supports IEEE 802.3 and IEEE 802.3u standard
- Supports line Auto MDI and MDIX Auto-sensing

#### 3.1.3 WLAN Interface

- Supports 802.11g, 802.11n (2.4 GHz), 802.11b
- Supports WPS2.0 (PBC mode and PIN mode)

- Supports DQPSK, DSSSS, CCK, OFDM, BPSK, QPSK, 16-QAM and 64-QAM wireless modulation method
- Supports SSID hiding
- Supports multiple SSIDs (4 SSIDs)
- Supports Open System, WPA-PSK and WPA2-PSK security
- Supports 64/128 bits WEP encryption
- Supports TKIP encryption
- Supports AES encryption
- Supports WMM (Wireless Fidelity Multimedia)
- Supports enable or disable the WLAN function by press WLAN button or config the Web-based utility
- WLAN Rates:
  - 802.11b: 1 Mbit/s, 2 Mbit/s, 5.5 Mbit/s, 11 Mbit/s
  - 802.11g: 1 Mbit/s, 2 Mbit/s, 5.5 Mbit/s, 6 Mbit/s, 9 Mbit/s, 11 Mbit/s, 12 Mbit/s, 18 Mbit/s, 24 Mbit/s, 36 Mbit/s, 48 Mbit/s, 54 Mbit/s
  - 802.11n (with 2T2R antenna used): 7.2 Mbit/s, 14.4 Mbit/s, 15.0 Mbit/s, 21.7 Mbit/s, 28.8 Mbit/s, 30.0 Mbit/s, 41.3 Mbit/s, 45.0 Mbit/s, 57.6 Mbit/s, 60.0 Mbit/s, 65.0 Mbit/s, 72.2 Mbit/s, 86.7 Mbit/s, 90.0 Mbit/s, 115.6 Mbit/s, 120.0 Mbit/s, 130.0 Mbit/s, 135.0 Mbit/s, 144.4 Mbit/s, 150.0 Mbit/s, 180.0 Mbit/s, 200.0 Mbit/s, 270.0 Mbit/s, 300.0 Mbit/s

#### 3.2 Security Features

- Supports powerful wireless network security
- Supports IP/MAC address filtering
- Supports port/URL filtering
- Supports ACL
- Supports DMZ
- Supports the PAP (RFC 1334), CHAP (RFC 1094) and the default PAP/CHAP adaptive
- Supports SPI (Stateful Packet Inspection)
- Prevents DoS attacks such as the SYN flooding, port scanning, ICMP Redirection, ping of death, spoofing, and LAND

#### 3.3 Routing & Bridged Features

- Supports NAT, NATT (RFC 1631, RFC 2661, RFC 2766, RFC 3023) and ALG extension
- Supports RIP v1 (RFC 1058), RIPv2 (RFC 1389, RFC 1723, RFC 2453)
- Supports multiple PVC working mode:
  - L2CA (RFC 2664 Bridged) mode
  - L2CE (RFC 2664 Bridged Static IP, RFC 2664 Bridged DMZ PC mode)
  - PPPoE (RFC 1661, RFC 2516)

- PPPoA (RFC 1661, RFC 2664)
- Supports DHCP server, DHCP client, DHCPv6 Server, DHCPv6 Client and DHCP relay
- Supports DNS client and DNS proxy
- Supports IGMP proxy and IGMP snooping
- Supports IPv6
  - Supports IPv4 and IPv6 dual-stack
  - Supports DSLite
- Supports port mapping
- Supports UPnP

#### 3.4 QoS Features

- Support multiple methods of traffic classification based on:
  - LAN interface and WLAN SSID
  - IP address (source and destination address)
  - Ports (source ports and destination ports) at the fourth layer
  - Mac address (source and destination address)
  - 802.1p
  - VLAN and 802.1q
  - Differentiated Services Code Point (DSCP)
  - Protocol (TCP, UDP, ICMP)
- Support for re-marking the result of traffic classification based on:
  - 802.1p
  - DSCP
  - ToS/IPP
- Support for queuing methods based on priorities (up to four queues):
  - Priority queuing (PQ)
  - Weighted Fair Queuing (WFQ)

#### 3.5 ATM Features

- Supports LLC-SNAP and VC-MUX
- Supports OAM F5
- Supports ATM OAM F5 loop
- Supports multiple ATM QoS service levels (CBR, nVBR, unVBR, UBR with PCR, UBR without PCR)

### 3.6 Network Management

- Supports TR-069 and upgrading through TR-069
- Supports Views system logs
- Supports remote and local web configuration and management

### 3.7 Power Supply Specifications

- Entire-device power supply: 12 V DC, 0.5 A
- Entire-device power consumption: < 6 W

### 3.8 Physical Specifications

- Dimensions (W x L x H): 31 mm ± 0.5 mm 145 mm ± 1.0 mm
- Weight: about 360 g

### 3.9 Environmental Specifications

- Ambient temperature for operation: 0°C to 40°C (32°F to 104°F)
- Relative humidity for operation: 5% to 95%, non-condensing

## Anexo F. Modem Huawei echo life HG8245 – Fibra óptica

	Current value	Referenced value
Optical Status:	auto	auto
Tx Optical Power:	2.36dBm	0.5 — 5dBm
Rx Optical Power:	-13.66dBm	-27 — -8dBm
Transceiver Voltage:	3366mV	3100 — 3500mV
Bias Current:	9mA	0 — 90mA
Transceiver Temperature:	47°C	-10 — 85°C

Device type:	HG8245
Description:	EchoLife HG8245 GPON Terminal (CLASS B+/PRODUCT ID:2102310GLK6TD2024771)
SN	485754436340A115 (HWTC6340A115)
Hardware version:	130D4600
Software version:	V1R006C00S211
ONT registration status:	O5 (Operation state)
ONT ID:	4
CPU usage:	1%
Memory usage:	94%

## Anexo G. Modem Linksys

Specifications		
Model Name	Linksys E4200	Receive Sensitivity
Description	Maximum Performance Dual-Band N Router	2.4 GHz 802.11b: -87 dBm @ 11 Mbps (Typical) 802.11g: -77 dBm @ 54 Mbps (Typical) 802.11n: 20 MHz: -71 dBm @ MCS15 (Typical) 802.11n: 40 MHz: -68 dBm @ MCS15 (Typical)
Model Number	E4200	5 GHz 802.11a: -71 dBm @ 54 Mbps (Typical) 802.11n: 20 MHz: -70 dBm @ MCS23 (Typical) 802.11n: 40 MHz: -68 dBm @ MCS23 (Typical)
Standards	802.11n, 802.11a, 802.11g, 802.11b, 802.3, 802.3u, 802.3ab	Antenna Gain in dBi
Radio Frequency	2.4 and 5 GHz	2.4 GHz (3 internal PIFA antennas) PIFA 1 <= 3.6 dBi (Right) PIFA 2 <= 3.8 dBi (Left) PIFA 3 <= 3.8 dBi (Front)
Switch Port Speed	10/100/1000 Mbps (Gigabit Ethernet)	5 GHz (3 internal PIFA antennas) PIFA 1 <= 4.8 dBi (Right) PIFA 2 <= 5.3 dBi (Left) PIFA 3 <= 5.2 dBi (Front)
Ports	Power, USB, Internet, Ethernet (1-4)	Supported File Systems for Storage Device
Buttons	Reset, Wi-Fi Protected Setup	FAT32, NTFS, and HFS+
LEDs	Top Panel: Power, Back Panel: Internet, Ethernet (1-4)	UPnP
Number of Antennas	6 Total, 3 Internal Antennas per Each 2.4 GHz and 5 GHz Radio Band	Supported
Detachable (Yes/No)	No	Security Features
Modulation	802.11b: CCK, QPSK, BPSK 802.11g: OFDM 802.11a: OFDM 802.11n: BPSK, QPSK, 16-QAM, 64-QAM	Security Key Bits
		Up to 128-Bit Encryption
		<b>Environmental</b>
		Dimensions
		8.86" x 0.98" x 6.30" (225 x 25 x 160 mm)
		Weight
		12.52 oz (355 g)
		Power
		12V, 2A
		Certification
		FCC, IC, CE, Wi-Fi a/b/g/n, Windows 7
		Operating Temperature
		32 to 104°F (0 to 40°C)
		Storage Temperature
		-4 to 140°F (-20 to 60°C)
		Operating Humidity
		10 to 80% Relative Humidity and Noncondensing
		Storage Humidity
		5 to 90% Noncondensing

## Anexo H. Programación de Modulo ESP8266 NodeMCU

```

#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

/***** Acceso aal Access Point *****/
#define WLAN_SSID "Enlace" //Nombre de la red Wi-Fi
#define WLAN_PASS "Isabel2017" //Password de la red Wi-Fi

/***** Iniciar servidor Broker MQTT *****/
#define AIO_SERVER "192.168.1.3"
#define AIO_SERVERPORT 1883

/*****Asignación y declaración de variables*****/
const int sensorPin = 5;
const int measureInterval = 2500;
volatile int pulseConter;

const float factorK = 7.06;
float volumen = 0;
long t0 = 0;
float litrosHora, metrosCubicos, litros;
String msj;
const int corteONOFF = 4;

unsigned long pulsos_Acumulados=0;

/*****Declaración global*****/

//Crear un Modulo ESP8266 Cliente para conectar al servidor
WiFiClient client;

//Inicia la declaración MQTT Cliente
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT);

/*****Publicaciones/Suscripciones*****/
//PUBLICADOR
Adafruit_MQTT_Publish consumoAgua =
Adafruit_MQTT_Publish(&mqtt, "0803/0803484799");

//SUBSCRITOR
Adafruit_MQTT_Subscribe corte =
Adafruit_MQTT_Subscribe(&mqtt, "0803484799/medidor/corte");

void MQTT_connect();

void setup()
{
  Serial.begin(115200); //Inicia la comunicación serial a enviar
  delay(10);

  Serial.println(); Serial.println();
  Serial.println(" Topico: 0803/0803484799");
  Serial.print(" Conectando a : ");
  Serial.println(WLAN_SSID);
  //WiFi.SID ();

  WiFi.begin(WLAN_SSID, WLAN_PASS); //Conectar a la red Wi-Fi
  while (WiFi.status() != WL_CONNECTED) // Espera para conectar a la red Wi-Fi
  {
    delay(500);
    Serial.print(" .");
  }
  Serial.println(" ** CONECTADO A LA RED WIFI **");
  Serial.println();

  Serial.print(" Direccion IP: ");
  Serial.println(WiFi.localIP());
  Serial.print(" Direccion MAC de ESP ---> ");
  Serial.println (WiFi.macAddress ());
  Serial.print(" La intensidad de la senal es:");
  Serial.println (WiFi.RSSI ());
  Serial.println();
  mqtt.subscribe(&corte);

  pinMode(sensorPin, INPUT);
  attachInterrupt(digitalPinToInterrupt(5), ISRCountPulse, RISING);
  t0 = millis();

  pinMode(corteONOFF, OUTPUT);
  delay(500);
  digitalWrite(corteONOFF , LOW);
}

void loop() {
  MQTT_connect();
  Adafruit_MQTT_Subscribe *subscription;

  while ((subscription = mqtt.readSubscription(0)))
  {
    if (subscription == &corte)
    {
      msj = ((char *)corte.lastread);
      if (msj == "c")
      {
        Serial.print(" valvula OFF ");
        digitalWrite(corteONOFF , HIGH);
      }
      if (msj == "a")
      {
        Serial.print(" valvula ON ");
        digitalWrite(corteONOFF , LOW);
      }
    }
  }

  ////Cálculos

  pulsos_Acumulados+= pulseConter; // Número de pulsos acumulados
  litrosHora = (pulseConter * 60 / factorK); // Q = frecuencia * 60/ 7.5 (L/Hora)
  litros = pulsos_Acumulados * 1.0/424; // Cada 450 pulsos son un litro
  //Serial.print(" Numero de pulsos ");
  //Serial.print(pulseConter);
  pulseConter = 0; // Pongo nuevamente el número de pulsos a cero
  if (litrosHora == 0)
  {
    litros = 0;
  }
  Serial.println();
  Serial.print(" Pulsos acumulados: ");
  Serial.print(pulsos_Acumulados);
  Serial.print(" L: ");

```

```

Serial.print(litros);
metrosCubicos = (litros * 0.001);
Serial.print(" m3: ");
Serial.print(metrosCubicos,5);
delay(1000);

if (! consumoAgua.publish(metrosCubicos,5))
  Serial.println(F(" envio fallido"));
else
  Serial.println(F(" envio OK!"));
}

///// CONENEXION MQTT-SERVER

void MQTT_connect() {
  int8_t ret;

  if (mqtt.connected()) {
    return;
  }
}

```

```

Serial.print(" Conectando al servidor MQTT... ");

uint8_t retries = 3;
while ((ret = mqtt.connect()) != 0) {
  Serial.println(mqtt.connectErrorString(ret));
  Serial.println(" Esperando conexion MQTT en 5
segundos...");
  mqtt.disconnect();
  delay(5000);
  retries--;
  if (retries == 0) {
    while (1);
  }
}
Serial.println(" MQTT Conectado!");
}

///// CONTADOR DE PULSOS

void ISRCOUNTpulse()
{
  pulseCounter++;
}

```

## Anexo I. Código para diseñar la GUI en Delphi Starter 10.2

### ENTRADA A LA GUI

```

unit uTesis;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
  System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, sSkinManager,
  acMagn,
  Vcl.ExtCtrls, Vcl.ComCtrls, Vcl.ToolWin, Vcl.StdCtrls,
  sLabel, sEdit, sButton,
  System.ImageList, Vcl.ImgList;

type
  TfrmIngreso = class(TForm)
    sSkinManager1: TsSkinManager;
    lbUsuario: TsLabel;
    lbContraseña: TsLabel;
    edUsuario: TsEdit;
    edContraseña: TsEdit;
    sButton1: TsButton;
    sButton2: TsButton;
    ImageList1: TImageList;
    procedure sButton1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmIngreso: TfrmIngreso;

implementation
{$R *.dfm}

uses uPrincipal;
procedure TfrmIngreso.sButton1Click(Sender: TObject);
var
  F:TfrmPrincipal;
begin
  F:=TfrmPrincipal.Create(Application);

  F.ShowModal;

end;

end.

```

### PRINCIPAL GUI

```

unit uPrincipal;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
  System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, sSkinManager,
  System.ImageList,
  Vcl.ImgList, Vcl.Menus, Vcl.StdCtrls, sButton,
  sComboBox, sEdit, Vcl.ExtCtrls,
  sLabel, sGroupBox, sPanel, Data.DB, MemDS, DBAccess,
  Uni, UniProvider,
  MySQLUniProvider, acPNG, Vcl.Grids, Vcl.DBGrids,
  Vcl.ComCtrls, sTabControl,
  sPageControl, Vcl.Buttons, sSpeedButton,
  Data.Bind.EngExt, Vcl.Bind.DBEngExt,
  Data.Bind.Components, Data.Bind.DBScope,
  TMS.MQTT.Client, TMS.MQTT.Global,
  Vcl.Bind.Grid, System.Rtti, System.Bindings.Outputs,
  Vcl.Bind.Editors,
  Data.Bind.Grid, sStatusBar, sBevel,
  acImage;

type
  TfrmPrincipal = class(TForm)
    sSkinManager1: TsSkinManager;
    Image1: TImage;
    TMSMQTTClient1: TTMSMQTTClient;
    Timer1: TTimer;
    pgcConsumoTiempoReal: TsPageControl;
    tsConsumosVista: TsTabSheet;
    dgcConsultasPagos: TsTabSheet;
    MainMenu1: TMainMenu;
    ImageList1: TImageList;
    I1: TMenuItem;
    I2: TMenuItem;
    qTopicosUsuario: TUniQuery;
    UniConnection1: TUniConnection;
    MySQLUniProvider1: TMySQLUniProvider;
    BindingsList1: TBindingsList;
    qTopicosUsuarioidUsuario: TStringField;

```

```

qTopicosUsuarioNombres: TStringField;
qTopicosUsuarioApellidos: TStringField;
qTopicosUsuarioCiudad: TStringField;
qTopicosUsuarioParroquia: TStringField;
qTopicosUsuarioDireccion: TStringField;
qTopicosUsuarionumMedidor: TStringField;
sgTiempoReal: TStringGrid;
sStatusBar1: TsStatusBar;
qTopicos: TUniQuery;
qTopicosnumMedidor: TStringField;
qTopicosidUsuario: TStringField;
qTopicostopico: TStringField;
qTopicosUsuariotopico: TStringField;
qConsumoagua: TUniQuery;
qConsumoaguaidUsuario: TStringField;
qConsumoaguanumMedidor: TStringField;
qConsumoaguatopico: TStringField;
qConsumoaguaconsumo: TStringField;
qConsumoaguaPago: TStringField;
qConsumoaguamesConsumo: TStringField;
qConsumoaguafecha: TStringField;
gbConsultas: TsGroupBox;
edCedula: TsEdit;
sLabel1: TsLabel;
edBuscarMedidor: TsEdit;
sLabel2: TsLabel;
sLabel3: TsLabel;
cbMesConsumo: TsComboBox;
btBuscarUsuario: TsSpeedButton;
btBuscarMedidor: TsSpeedButton;
btMesConsumo: TsSpeedButton;
qConsultaPagos: TUniQuery;
sgConsultaPagosConsumo: TStringGrid;
BindSourceDB1: TBindSourceDB;
LinkGridToDataSourceBindSourceDB1:
TLinkGridToDataSource;
gbPagos: TsGroupBox;
sLabel4: TsLabel;
lbPagoldUsuarios: TsLabel;
LinkPropertyToFieldCaption: TLinkPropertyToField;
sLabel5: TsLabel;
sLabel6: TsLabel;
lbNumMedidor: TsLabel;
sLabel8: TsLabel;
sLabel9: TsLabel;
lbNombres: TsLabel;
lbParroquia: TsLabel;
lbApellidos: TsLabel;
lbDireccion: TsLabel;
LinkPropertyToFieldCaption3: TLinkPropertyToField;
LinkPropertyToFieldCaption4: TLinkPropertyToField;
qConsultaPagosidUsuario: TStringField;
qConsultaPagosnumMedidor: TStringField;
qConsultaPagosmesConsumo: TStringField;
qConsultaPagosconsumo: TStringField;
qConsultaPagosfecha: TStringField;
qConsultaPagosestado: TStringField;
qConsultaPagosNombres: TStringField;
qConsultaPagosApellidos: TStringField;
qConsultaPagosCiudad: TStringField;
qConsultaPagosParroquia: TStringField;
qConsultaPagosDireccion: TStringField;
qConsultaPagosCTotal: TFloatField;
qConsultaPagosPago: TFloatField;
lbConsumoTotal: TsLabel;
sLabel11: TsLabel;
sLabel12: TsLabel;
lbDeudaTotal: TsLabel;
lbDeudaMedidores: TsLabel;
sLabel10: TsLabel;
lbConsumoMedidor: TsLabel;
sLabel14: TsLabel;
lbDeudaMesConsumo: TsLabel;
sLabel16: TsLabel;
lbMesConsumo2: TsLabel;
sLabel18: TsLabel;

sBevel2: TsBevel;
sBevel3: TsBevel;
sLabel7: TsLabel;
lbMesConsumo: TsLabel;
btPagarDeudaTotal: TButton;
sImage1: TsImage;
sImage2: TsImage;
btDeudaMedidores: TButton;
sImage3: TsImage;
sImage4: TsImage;
sImage5: TsImage;
brDeudaMensual: TButton;
BindSourceDB2: TBindSourceDB;
qConsumoaguaestado: TStringField;
s1: TMenuItem;
S2: TMenuItem;
tsBlanco: TsTabSheet;
Shape1: TShape;
PopupMenu1: TPopupMenu;
r1: TMenuItem;
PopupMenu2: TPopupMenu;
A1: TMenuItem;
procedure I2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure sgTiempoRealDrawCell(Sender: TObject);
ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
  procedure
TMSMQTTClient1.ConnectedStatusChanged(ASender:
TObject);
  const AConnected: Boolean; AStatus:
TTMSMQTTConnectionStatus);
  procedure FormCreate(Sender: TObject);
  procedure TMSMQTTClient1.PublishReceived(ASender:
TObject; APacketID: Word;
  ATopic: string; APayload: TArray<System.Byte>);
  procedure insertarConsumo;
  procedure Timer1.Timer(Sender: TObject);
  procedure btBuscarUsuarioClick(Sender: TObject);
  procedure btMesConsumoClick(Sender: TObject);
  procedure btBuscarMedidorClick(Sender: TObject);
  procedure sgConsultaPagosConsumoDrawCell(Sender:
TObject);
  ACol, ARow: Integer; Rect: TRect; State:
TGridDrawState);
  procedure brDeudaMensualClick(Sender: TObject);
  procedure btDeudaMedidoresClick(Sender: TObject);
  procedure btPagarDeudaTotalClick(Sender: TObject);
  procedure S2Click(Sender: TObject);
  procedure r1Click(Sender: TObject);
  procedure pgcConsumoTiempoRealChange(Sender:
TObject);
private
  { Private declarations }

public
  { Public declarations }
  procedure borrarCosumosDeudas;
  procedure verGridDatos;
end;

var
  frmPrincipal: TfrmPrincipal;

implementation

{$R *.dfm}

uses uIngresoUsuarios, uCortes;

const
  reset = '1600376881/medidor/consumo/reset';

var
  contador, contMeses: Integer;

```



```

procedure TfrmPrincipal.I2Click(Sender: TObject);
var
  F: TfrmIngresoUsuarios;
begin
  pgcConsumoTiempoReal.ActivePage := tsBlanco;
  tsBlanco.Refresh;
  F := TfrmIngresoUsuarios.Create(Application);
  F.ShowModal;
end;

procedure TfrmPrincipal.insertarConsumo;
var
  numRecords, i: Integer;
  valor, formattedDateTime: string;
begin
  qConsumoagua.Open;
  qConsumoagua.First;
  qConsumoagua.Insert;
  numRecords := qTopicosUsuario.RecordCount;
  for i := 0 to numRecords - 1 do
  begin
    qConsumoagua.Insert;
    qConsumoaguanumMedidor.AsString :=
sgTiempoReal.Cells[6, i + 1];
    qConsumoaguaidUsuario.AsString :=
sgTiempoReal.Cells[0, i + 1];
    qConsumoaguatopico.AsString := sgTiempoReal.Cells[7,
i + 1];
    qConsumoaguaconsumo.AsString :=
sgTiempoReal.Cells[8, i + 1];
    qConsumoaguaestado.AsString := 'NO PAGADO';
    qConsumoaguafecha.AsString := DateTimeToStr(Now);
    DateTimeToString(formattedDateTime, 'mmmm', Now);
    qConsumoaguamesConsumo.AsString :=
(formattedDateTime);
    qConsumoagua.Post;
  end;
end;

procedure
TfrmPrincipal.pgcConsumoTiempoRealChange(Sender:
TObject);
var
  numRecords, i, j: Integer;
begin
  sgTiempoReal.Refresh;
  verGridDatos;
  qTopicosUsuario.Close;
  qTopicosUsuario.Open;
end;

procedure TfrmPrincipal.r1Click(Sender: TObject);
begin
  TMSMQTTClient1.Disconnect;
  TMSMQTTClient1.Connect(True);
  sgTiempoReal.Refresh;
  verGridDatos;
end;

procedure TfrmPrincipal.S2Click(Sender: TObject);
var
  F: TfrmCortes;
begin
  pgcConsumoTiempoReal.ActivePage := tsBlanco;
  tsBlanco.Refresh;
  F := TfrmCortes.Create(Application);
  F.ShowModal;
end;

procedure
TfrmPrincipal.sgConsultaPagosConsumoDrawCell(Sender:
TObject;
  ACol, ARow: Integer; Rect: TRect; State:
TGridDrawState);
var
  dx: Integer;
begin
  with (Sender as TStringGrid) do
  begin
    with (Sender as TStringGrid) do
    begin
      if (ARow = 0) then
      begin
        Canvas.Brush.Color := clBtnFace;
        Canvas.TextRect(Rect, Rect.Left + 2, Rect.Top + 2,
Cells[ACol, ARow]);
        Canvas.FrameRect(Rect);
      end;
    end;

    with (Sender as TStringGrid) do
    begin
      if (ACol = -1) or (ARow = 0) then
      begin
        Canvas.Brush.Color := clBtnFace
      end;
      if (ARow mod 2 = 0) then
      begin
        Canvas.Brush.Color := clSilver
      end;
      if (ARow mod 2 = 0) then
      begin
        Canvas.Brush.Color := $00FFEBDF;
        Canvas.TextRect(Rect, Rect.Left + 2, Rect.Top + 2,
Cells[ACol, ARow]);
        Canvas.FrameRect(Rect);
      end;
    end;
  end;
end;

procedure TfrmPrincipal.sgTiempoRealDrawCell(Sender:
TObject;
  ACol, ARow: Integer; Rect: TRect; State:
TGridDrawState);
var
  dx: Integer;
begin
  with (Sender as TStringGrid) do
  begin
    with (Sender as TStringGrid) do
    begin
      if (ARow = 0) then
      begin
        Canvas.Brush.Color := clBtnFace;
        Canvas.TextRect(Rect, Rect.Left + 2, Rect.Top + 2,
Cells[ACol, ARow]);
        Canvas.FrameRect(Rect);
      end;
    end;

    with (Sender as TStringGrid) do
    begin
      if (ACol = -1) or (ARow = 0) then
      begin
        Canvas.Brush.Color := clBtnFace
      end;
      if (ARow mod 2 = 0) then
      begin
        Canvas.Brush.Color := clSilver
      end;
      if (ARow mod 2 = 0) then
      begin
        Canvas.Brush.Color := $00FFEBDF;
        Canvas.TextRect(Rect, Rect.Left + 2, Rect.Top + 2,
Cells[ACol, ARow]);
        Canvas.FrameRect(Rect);
      end;
    end;
  end;
end;

procedure TfrmPrincipal.Timer1Timer(Sender: TObject);

```

```

begin
  contador := contador + 1;
  if contador = 2 then
  begin
    sStatusBar1.Panels[1].Text := IntToStr(contMeses);
  end;
  if contador = 2 then
  begin
    contMeses := contMeses + 1;
    insertarConsumo;
    contador := 0;
    sStatusBar1.Panels[1].Text := 'Consumo registrado.';
  end;
  if contMeses = 3 then
  begin
    sStatusBar1.Panels[1].Text := 'Corte de agua.';
  end;
end;

procedure
TfrmPrincipal.TMSMQTTClient1.ConnectedStatusChanged
(ASender: TObject;
const AConnected: Boolean; AStatus:
TTMSMQTTConnectionStatus);
begin
  if AConnected then
  begin
    while not qTemas.Eof do
    begin
      TMSMQTTClient1.Subscribe(qTemas.AsString);
      qTemas.Next;
    end;
    sStatusBar1.Panels[0].Text := 'Conectado.';
  end;
end;

procedure
TfrmPrincipal.TMSMQTTClient1.PublishReceived(ASender:
TObject;
APacketID: Word; ATopic: string; APayload:
TArray<System.Byte>);
var
  numRecords, i: Integer;
  msg: string;
begin
  numRecords := qTemas.RecordCount;
  msg := TEncoding.UTF8.GetString(APayload);
  if (msg > '0.00000') then
  begin
    Timer1.Enabled := True;
  end
  else
  begin
    Timer1.Enabled := false;
  end;

  for i := 1 to numRecords do
  begin
    if sgTiempoReal.Cells[7, i] = ATopic then
    begin
      sgTiempoReal.Cells[8, i] := msg;
    end;
  end;
end;

procedure TfrmPrincipal.verGridDatos;
var
  numRecords, i, j: Integer;
begin
  i := 0;
  qTemasUsuario.Close;
  qTemasUsuario.Open;
  qTemasUsuario.Refresh;
  qTemasUsuario.First;

  sgTiempoReal.ColWidths[0] :=
sgTiempoReal.ColWidths[0];
sgTiempoReal.Cells[0, 0] := 'CEDULA';
sgTiempoReal.ColWidths[1] :=
sgTiempoReal.ColWidths[1];
sgTiempoReal.Cells[1, 0] := 'NOMBRES';
sgTiempoReal.ColWidths[2] :=
sgTiempoReal.ColWidths[2];
sgTiempoReal.Cells[2, 0] := 'APELLIDOS';
sgTiempoReal.ColWidths[3] :=
sgTiempoReal.ColWidths[3];
sgTiempoReal.Cells[3, 0] := 'CIUDAD';
sgTiempoReal.ColWidths[4] :=
sgTiempoReal.ColWidths[4];
sgTiempoReal.Cells[4, 0] := 'PARROQUIA';
sgTiempoReal.ColWidths[5] :=
sgTiempoReal.ColWidths[5];
sgTiempoReal.Cells[5, 0] := 'DIRECCION';
sgTiempoReal.ColWidths[6] :=
sgTiempoReal.ColWidths[6];
sgTiempoReal.Cells[6, 0] := 'NUM. MEDIDOR';
sgTiempoReal.ColWidths[7] :=
sgTiempoReal.ColWidths[7];
sgTiempoReal.Cells[7, 0] := 'TOPIC';
sgTiempoReal.ColWidths[8] :=
sgTiempoReal.ColWidths[8];
sgTiempoReal.Cells[8, 0] := 'CONSUMO m^3/h';
numRecords := qTemasUsuario.RecordCount;
sgTiempoReal.RowCount := numRecords + 1;
while not qTemasUsuario.Eof do
begin
  for i := 0 to qTemasUsuario.FieldCount - 1 do
  begin
    sgTiempoReal.Cells[i, j + 3] :=
qTemasUsuario.Fields[i].AsString;
    sgTiempoReal.Refresh;
    sgTiempoReal.Repaint;
  end;
  qTemasUsuario.Next;
  Inc(j);
end;
qTemasUsuario.Close;
qTemasUsuario.Open;
qTemasUsuario.Refresh;
sgTiempoReal.Repaint;
end;

procedure TfrmPrincipal.borrarCosumosDeudas;
begin
  qConsultaPagos.Close;
  qConsultaPagos.Open;
  /// //////////////////////////////////////
  lbConsumoTotal.Caption := "";
  lbDeudaTotal.Caption := "";
  lbNombres.Caption := "";
  lbApellidos.Caption := "";
  lbParroquia.Caption := "";
  lbDireccion.Caption := "";
  lbPagosUsuarios.Caption := "";
  /// //////////////////////////////////////
  lbConsumoMedidor.Caption := "";
  lbDeudaMedidores.Caption := "";
  lbNumMedidor.Caption := "";
  /// //////////////////////////////////////
  lbMesConsumo2.Caption := "";
  lbDeudaMesConsumo.Caption := "";
  lbMesConsumo.Caption := "";
  /// //////////////////////////////////////
  cbMesConsumo.Text := "";
  edBuscarMedidor.Text := "";

end;

procedure TfrmPrincipal.brDeudaMensualClick(Sender:
TObject);
var

```

```

consulta, idUsuario, numMedidor: string;
begin
qConsumoagua.Close;
qConsumoagua.SQL.Clear;
qConsumoagua.SQL.Add('UPDATE consumoagua SET
estado = :estado');
qConsumoagua.SQL.Add('WHERE idUsuario = ' +
QuotedStr(edCedula.Text) +
'and numMedidor = ' +
QuotedStr(edBuscarMedidor.Text) + 'and mesConsumo = ' +
QuotedStr(cbMesConsumo.Text));
qConsumoagua.ParamByName('estado').AsString :=
'PAGADO';
qConsumoagua.ExecSQL;
borrarCosumosDeudas;
end;

```

```

procedure TfrmPrincipal.btBuscarMedidorClick(Sender:
TObject);
var
consumo, pago: real;
i: Integer;
begin
consumo := 0;
cbMesConsumo.Text := '';
qConsultaPagos.Filtered := false;
qConsultaPagos.FilterSQL := 'mesConsumo like' +
QuotedStr('%' + cbMesConsumo.Text + '%');
qConsultaPagos.Filtered := True;
qConsultaPagos.Close;
qConsultaPagos.Open;

```

```

qConsultaPagos.Filtered := false;
qConsultaPagos.Filter := 'numMedidor like' +
QuotedStr('%' + edBuscarMedidor.Text + '%');
qConsultaPagos.Filtered := True;
qConsultaPagos.Close;
qConsultaPagos.Open;
lbNumMedidor.Caption :=
qConsultaPagos.numMedidor.AsString;
lbParroquia.Caption :=
qConsultaPagos.Parroquia.AsString;
lbDireccion.Caption :=
qConsultaPagos.Direccion.AsString;
for i := 1 to qConsultaPagos.RecordCount do
begin
consumo := 0;
lbConsumoMedidor.Caption := FloatToStr(consumo) +
m^3';
end;
pago := consumo * 0.48;
lbDeudaMedidores.Caption := '$ ' + FloatToStr(pago);

```

end;

```

procedure TfrmPrincipal.btBuscarUsuarioClick(Sender:
TObject);
var
consumo, pago: real;
i: Integer;
begin
consumo := 0;
cbMesConsumo.Text := '';
qConsultaPagos.Filtered := false;
qConsultaPagos.FilterSQL := 'mesConsumo like' +
QuotedStr('%' + cbMesConsumo.Text + '%');
qConsultaPagos.Filtered := True;
qConsultaPagos.Close;
qConsultaPagos.Open;
qConsultaPagos.Filtered := false;
qConsultaPagos.Filter := 'idUsuario like' +
QuotedStr('%' + edCedula.Text + '%');
qConsultaPagos.Filtered := True;
qConsultaPagos.Close;
qConsultaPagos.Open;
for i := 1 to qConsultaPagos.RecordCount do

```

```

begin
consumo
:=StrToFloat(sgConsultaPagosConsumo.Cells[7, i]);
lbConsumoTotal.Caption := FloatToStr(consumo) +
m^3';
end;
pago := consumo * 0.48;
lbDeudaTotal.Caption := '$ ' + FloatToStr(pago);
end;

```

```

procedure TfrmPrincipal.btDeudaMedidoresClick(Sender:
TObject);
begin
qConsumoagua.Close;
qConsumoagua.SQL.Clear;
qConsumoagua.SQL.Add('UPDATE consumoagua SET
estado = :estado');
qConsumoagua.SQL.Add('WHERE idUsuario = ' +
QuotedStr(edCedula.Text) +
'and numMedidor = ' +
QuotedStr(edBuscarMedidor.Text));
qConsumoagua.ParamByName('estado').AsString :=
'PAGADO';
qConsumoagua.ExecSQL;
borrarCosumosDeudas;
end;

```

```

procedure TfrmPrincipal.btMesConsumoClick(Sender:
TObject);

```

```

var
consumo, pago: real;
i: Integer;
begin
consumo := 0;
qConsultaPagos.Filtered := false;
qConsultaPagos.FilterSQL := 'mesConsumo like' +
QuotedStr('%' + cbMesConsumo.Text + '%');
qConsultaPagos.Filtered := True;
qConsultaPagos.Close;
qConsultaPagos.Open;
lbMesConsumo.Caption := cbMesConsumo.Text;
for i := 1 to qConsultaPagos.RecordCount do
begin
consumo :=
StrToFloat(sgConsultaPagosConsumo.Cells[7, i]);
lbMesConsumo2.Caption := FloatToStr(consumo) +
m^3';
end;
pago := consumo * 0.48;
lbDeudaMesConsumo.Caption := '$ ' + FloatToStr(pago);
end;

```

```

procedure TfrmPrincipal.btPagarDeudaTotalClick(Sender:
TObject);

```

```

begin
qConsumoagua.Close;
qConsumoagua.SQL.Clear;
qConsumoagua.SQL.Add('UPDATE consumoagua SET
estado = :estado');
qConsumoagua.SQL.Add('WHERE idUsuario = ' +
QuotedStr(edCedula.Text));
qConsumoagua.ParamByName('estado').AsString :=
'PAGADO';
qConsumoagua.ExecSQL;
borrarCosumosDeudas;
end;

```

```

procedure TfrmPrincipal.FormCreate(Sender: TObject);
begin

```

```

TMSMQTTClient1.BrokerHostName := '192.168.1.3';
TMSMQTTClient1.Connect();
contador := 0;
contMeses := 0;
pgcConsumoTiempoReal.ActivePage :=
dgcConsultasPagos;
dgcConsultasPagos.Refresh;

```

```

Sleep(10);
pgcConsumoTiempoReal.ActivePage := tsConsumosVista;
tsConsumosVista.Realign;
sgTiempoReal.Align := alClient;

```

```
end;
```

```
procedure TfrmPrincipal.FormShow(Sender: TObject);
```

```

var
  numRecords, i, j: Integer;
begin
  i := 0;
  qTopicosUsuario.Close;
  qTopicosUsuario.Open;
  qTopicosUsuario.Refresh;
  qTopicosUsuario.First;
  sgTiempoReal.ColWidths[0] :=
sgTiempoReal.ColWidths[0] + 50;
  sgTiempoReal.Cells[0, 0] := 'CEDULA';
  sgTiempoReal.ColWidths[1] :=
sgTiempoReal.ColWidths[1] + 100;
  sgTiempoReal.Cells[1, 0] := 'NOMBRES';
  sgTiempoReal.ColWidths[2] :=
sgTiempoReal.ColWidths[2] + 100;
  sgTiempoReal.Cells[2, 0] := 'APELLIDOS';
  sgTiempoReal.ColWidths[3] :=
sgTiempoReal.ColWidths[3] + 50;
  sgTiempoReal.Cells[3, 0] := 'CIUDAD';
  sgTiempoReal.ColWidths[4] :=
sgTiempoReal.ColWidths[4] + 50;
  sgTiempoReal.Cells[4, 0] := 'PARROQUIA';
  sgTiempoReal.ColWidths[5] :=
sgTiempoReal.ColWidths[5] + 150;
  sgTiempoReal.Cells[5, 0] := 'DIRECCION';
  sgTiempoReal.ColWidths[6] :=
sgTiempoReal.ColWidths[6] + 50;
  sgTiempoReal.Cells[6, 0] := 'NUM. MEDIDOR';
  sgTiempoReal.ColWidths[7] :=
sgTiempoReal.ColWidths[7] + 50;
  sgTiempoReal.Cells[7, 0] := 'TOPIC';
  sgTiempoReal.ColWidths[8] :=
sgTiempoReal.ColWidths[8] + 50;
  sgTiempoReal.Cells[8, 0] := 'CONSUMO m^3/h';
  numRecords := qTopicosUsuario.RecordCount;
  sgTiempoReal.RowCount := numRecords + 1;
  while not qTopicosUsuario.Eof do
  begin
    for i := 0 to qTopicosUsuario.FieldCount - 1 do
    begin
      sgTiempoReal.Cells[i, j + 3] :=
qTopicosUsuario.Fields[i].AsString;
    end;
    qTopicosUsuario.Next;
    sgTiempoReal.Refresh;
    Inc(j);
  end;
end;
end.

```

## INGRESO DE USUARIOS

```
unit uIngresoUsuarios;
```

```
interface
```

```

uses
  Winapi.Windows, Winapi.Messages, System.SysUtils,
  System.Variants,
  System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,
  sButton, sComboBox, sEdit,
  Vcl.ExtCtrls, sLabel, sGroupBox, Data.DB, DBAccess,
  Uni, UniProvider,

```

```

MySQLUniProvider, MemDS, System.ImageList,
Vcl.ImgList;

```

```
type
```

```

TfmrIngresoUsuarios = class(TForm)
  ImageList1: TImageList;
  qTopicos: TUniQuery;
  qTopicosnumMedidor: TStringField;
  qTopicosidUsuario: TStringField;
  qTopicostopico: TStringField;
  MySQLUniProvider1: TMySQLUniProvider;
  UniConnection1: TUniConnection;
  qUsuarios: TUniQuery;
  qUsuariosidUsuario: TStringField;
  qUsuariosNombres: TStringField;
  qUsuariosApellidos: TStringField;
  qUsuariosTelefono: TStringField;
  qUsuariosCiudad: TStringField;
  qUsuariosParroquia: TStringField;
  qUsuariosDireccion: TStringField;
  qUsuariosnumMedidor: TStringField;
  gbIngresoUsuarios: TsGroupBox;
  sLabel1: TsLabel;
  sLabel2: TsLabel;
  sLabel3: TsLabel;
  sLabel4: TsLabel;
  sLabel5: TsLabel;
  sLabel6: TsLabel;
  sLabel7: TsLabel;
  sLabel8: TsLabel;
  edCedula: TsEdit;
  edNombres: TsEdit;
  edApellidos: TsEdit;
  edCelular: TsEdit;
  edCiudad: TsEdit;
  cbParroquias: TsComboBox;
  btIngresar: TsButton;
  btGuardar: TsButton;
  btCancelar: TsButton;
  edDireccion: TsEdit;
  edNumMedidor: TsEdit;
  qPagos: TUniQuery;
  qPagosidUsuario: TStringField;
  qPagosnumMedidor: TStringField;
  qPagosmesConsumo: TStringField;
  qPagosconsumo: TStringField;
  qPagosTotal: TStringField;
  procedure FormCreate(Sender: TObject);
  procedure controlErrores();
  procedure insertar();
  procedure insertarTopicos();
  procedure insertarPagos();
  procedure limpiarEdits();
  procedure consultaUsuarioMedidor();
  procedure enableEdits;
  procedure disableEdits;
  procedure btIngresarClick(Sender: TObject);
  procedure btGuardarClick(Sender: TObject);
  procedure btCancelarClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action:
TCloseAction);

```

```

private
  { Private declarations }
public
  { Public declarations }
end;

```

```
var
```

```
fmrIngresoUsuarios: TfmrIngresoUsuarios;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TfmrIngresoUsuarios.btCancelarClick(Sender:
TObject);
```

```

begin
  limpiarEdits;
end;

procedure TfmrIngresoUsuarios.btGuardarClick(Sender:
TObject);
begin
  controlErrores;
  consultaUsuarioMedidor();
  insertar;
  btGuardar.Enabled := false;
  btCancelar.Enabled := false;
  btIngresar.Enabled := true;
  insertarTopicos;
  // insertarPagos;
  disableEdits;
end;

procedure TfmrIngresoUsuarios.btIngresarClick(Sender:
TObject);
begin
  btGuardar.Enabled := true;
  btCancelar.Enabled := true;
  limpiarEdits;
  btIngresar.Enabled := false;
  enableEdits;
end;

procedure TfmrIngresoUsuarios.consultaUsuarioMedidor;
var
  consulta, idUsuario, numMedidor: string;
begin
  qUsuarios.SQL.Clear;
  consulta := 'SELECT * FROM usuarios WHERE
idUsuario = ' +
  QuotedStr(edCedula.Text) + 'and numMedidor = ' +
  QuotedStr(edNumMedidor.Text);
  qUsuarios.SQL.Add(consulta);
  qUsuarios.Open;
  idUsuario := qUsuariosidUsuario.AsString;
  numMedidor := qUsuariosnumMedidor.AsString;
  if (idUsuario = edCedula.Text) and (numMedidor =
edNumMedidor.Text) then
  begin
    ShowMessage('El usuario y el numero de medidor ya han
sido registrados ');
    abort;
  end;
end;

procedure TfmrIngresoUsuarios.controlErrores;

begin
  if edCedula.Text = '' then
  begin
    ShowMessage('Ingrese el numero de cedula');
    abort;
  end;
  if edNombres.Text = '' then
  begin
    ShowMessage('Ingrese los nombres');
    abort;
  end;
  if edApellidos.Text = '' then
  begin
    ShowMessage('Ingrese los apellidos');
    abort;
  end;
  if edCelular.Text = '' then
  begin
    ShowMessage('Ingrese el numero de celular');
    abort;
  end;
  if edDireccion.Text = '' then
  begin
    ShowMessage('Ingrese la direccion');
  end;
end;

```

```

  abort;
end;
if edNumMedidor.Text = '' then
begin
  ShowMessage('Ingrese el numero de medidor');
  abort;
end;
if (cbParroquias.ItemIndex = -1) or (cbParroquias.Text = '')
then
begin
  ShowMessage('Escoja una parroquia');
  abort;
end;
end;

procedure TfmrIngresoUsuarios.disableEdits;
begin
  edCedula.Enabled := false;
  edNombres.Enabled := false;
  edApellidos.Enabled := false;
  edCelular.Enabled := false;
  cbParroquias.Enabled := false;
  edDireccion.Enabled := false;
  edNumMedidor.Enabled := false;
  cbParroquias.Enabled := false;
end;

procedure TfmrIngresoUsuarios.enableEdits;
begin
  edCedula.Enabled := true;
  edNombres.Enabled := true;
  edApellidos.Enabled := true;
  edCelular.Enabled := true;
  cbParroquias.Enabled := true;
  edDireccion.Enabled := true;
  edNumMedidor.Enabled := true;
  cbParroquias.Enabled := true;
end;

procedure TfmrIngresoUsuarios.FormClose(Sender:
TObject);
var Action: TCloseAction);
begin
  Action := caFree;
end;

procedure TfmrIngresoUsuarios.FormCreate(Sender:
TObject);
begin
  gbIngresoUsuarios.Visible := true;
  btGuardar.Enabled := false;
  btCancelar.Enabled := false;
  disableEdits;
end;

procedure TfmrIngresoUsuarios.insertar;
begin
  begin
    qUsuarios.SQL.Text := 'insert into usuarios (idUsuario,
Nombres, Apellidos,'
    + ' Telefono, Ciudad, Parroquia, Direccion,
numMedidor)' +
    ' values (:idUsuario, :Nombres, :Apellidos, :Telefono,
:Ciudad,' +
    ' :Parroquia, :Direccion, :numMedidor)';
    qUsuarios.ParamByName('idUsuario').AsString :=
edCedula.Text;
    qUsuarios.ParamByName('Nombres').AsString :=
edNombres.Text;
    qUsuarios.ParamByName('Apellidos').AsString :=
edApellidos.Text;
    qUsuarios.ParamByName('Telefono').AsString :=
edCelular.Text;
    qUsuarios.ParamByName('Ciudad').AsString :=
edCiudad.Text;
  end;
end;

```

```

    qUsuarios.ParamByName('Parroquia').AsString :=
cbParroquias.Text;
    qUsuarios.ParamByName('Direccion').AsString :=
edDireccion.Text;
    qUsuarios.ParamByName('numMedidor').AsString :=
edNumMedidor.Text;
    qUsuarios.Execute;
end;
end;

```

```

procedure TfrmIngresoUsuarios.insertarPagos;
begin
    qPagos.SQL.Text := 'insert into pagos ( idUsuario,
numMedidor)' +
    ' values (:idUsuario, :numMedidor)';
    qPagos.ParamByName('idUsuario').AsString :=
edCedula.Text;
    qPagos.ParamByName('numMedidor').AsString :=
edNumMedidor.Text;
    qPagos.Execute;
end;

```

```

procedure TfrmIngresoUsuarios.insertarTemas;
var
    tematico: string;
begin
    tematico := edNumMedidor.Text + '/' + edCedula.Text;
    qTemas.SQL.Text := 'insert into temas (numMedidor,
idUsuario, tematico)' +
    ' values (:numMedidor, :idUsuario, :tematico)';
    qTemas.ParamByName('numMedidor').AsString :=
edNumMedidor.Text;
    qTemas.ParamByName('idUsuario').AsString :=
edCedula.Text;
    qTemas.ParamByName('tematico').AsString := tematico;
    qTemas.Execute;
end;

```

```

procedure TfrmIngresoUsuarios.limpiarEdits;
begin
    edCedula.Text := '';
    edNombres.Text := '';
    edApellidos.Text := '';
    edCelular.Text := '';
    cbParroquias.Text := '';
    edDireccion.Text := '';
    edNumMedidor.Text := '';
end;

```

end.

## SIMULACIÓN DE CORTE

```

unit uCortes;

```

```

interface

```

```

uses

```

```

    Winapi.Windows, Winapi.Messages, System.SysUtils,
System.Variants,
    System.Classes, Vcl.Graphics,
    Vcl.Controls, Vcl.Forms, Vcl.Dialogs, sSkinManager,
System.ImageList,
    Vcl.ImgList, Vcl.StdCtrls, sButton, sEdit, Vcl.ExtCtrls,
sGroupBox, sLabel,
    Data.DB, UniProvider, MySQLUniProvider, DBAccess,
Uni, MemDS, Vcl.ComCtrls,
    sStatusBar, TMS.MQTT.Global, TMS.MQTT.Client,
Data.Bind.EngExt,
    Vcl.Bind.DBEngExt, Vcl.Bind.Grid, System.Rtti,
System.Bindings.Outputs,
    Vcl.Bind.Editors, Data.Bind.Components, Data.Bind.Grid,
Data.Bind.DBScope,
    Vcl.Grids;

```

```

type

```

```

TfrmCortes = class(TForm)
    sSkinManager1: TSkinManager;
    gbCortes: TGroupBox;
    lbNumeroMedidor: TLabel;
    Label3: TLabel;
    Shape1: TShape;
    sButton1: TButton;
    ImageList1: TImageList;
    Label1: TLabel;
    lbCedula: TLabel;
    lbEstado: TLabel;
    Label5: TLabel;
    qCortes: TUniQuery;
    UniConnection1: TUniConnection;
    MySQLUniProvider1: TMySQLUniProvider;
    qCortescontador: TIntegerField;
    qCortesEstado: TStringField;
    qCortesidUsuario: TStringField;
    qCortesnumMedidor: TStringField;
    sStatusBar1: TStatusBar;
    TMSMQTTClient1: TMSMQTTClient;
    StringGrid1: TStringGrid;
    BindSourceDB1: TBindSourceDB;
    BindingsList1: TBindingsList;
    LinkGridToDataSourceBindSourceDB1:
TLinkGridToDataSource;
    Shape2: TShape;
    Shape3: TShape;
    btRestaurarCorte: TButton;
    procedure sButton1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure
TMSMQTTClient1.ConnectedStatusChanged(ASender:
TObject);
    const AConnected: Boolean; AStatus:
TMSMQTTConnectionStatus;
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
    procedure btRestaurarCorteClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

```

```

var

```

```

    frmCortes: TfrmCortes;

```

```

implementation

```

```

{$R *.dfm}

```

```

const

```

```

    corteMedidor = '1600376881/medidor/corte';

```

```

procedure TfrmCortes.btRestaurarCorteClick(Sender:
TObject);
begin
    TMSMQTTClient1.Publish(corteMedidor,'a')
end;

```

```

procedure TfrmCortes.FormClose(Sender: TObject; var
Action: TCloseAction);
begin
    TMSMQTTClient1.Disconnect;
end;

```

```

procedure TfrmCortes.FormCreate(Sender: TObject);
begin

```

```

    TMSMQTTClient1.BrokerHostName := '192.168.1.3';
    TMSMQTTClient1.Connect();
end;

```

```

procedure TfrmCortes.sButton1Click(Sender: TObject);

```

```

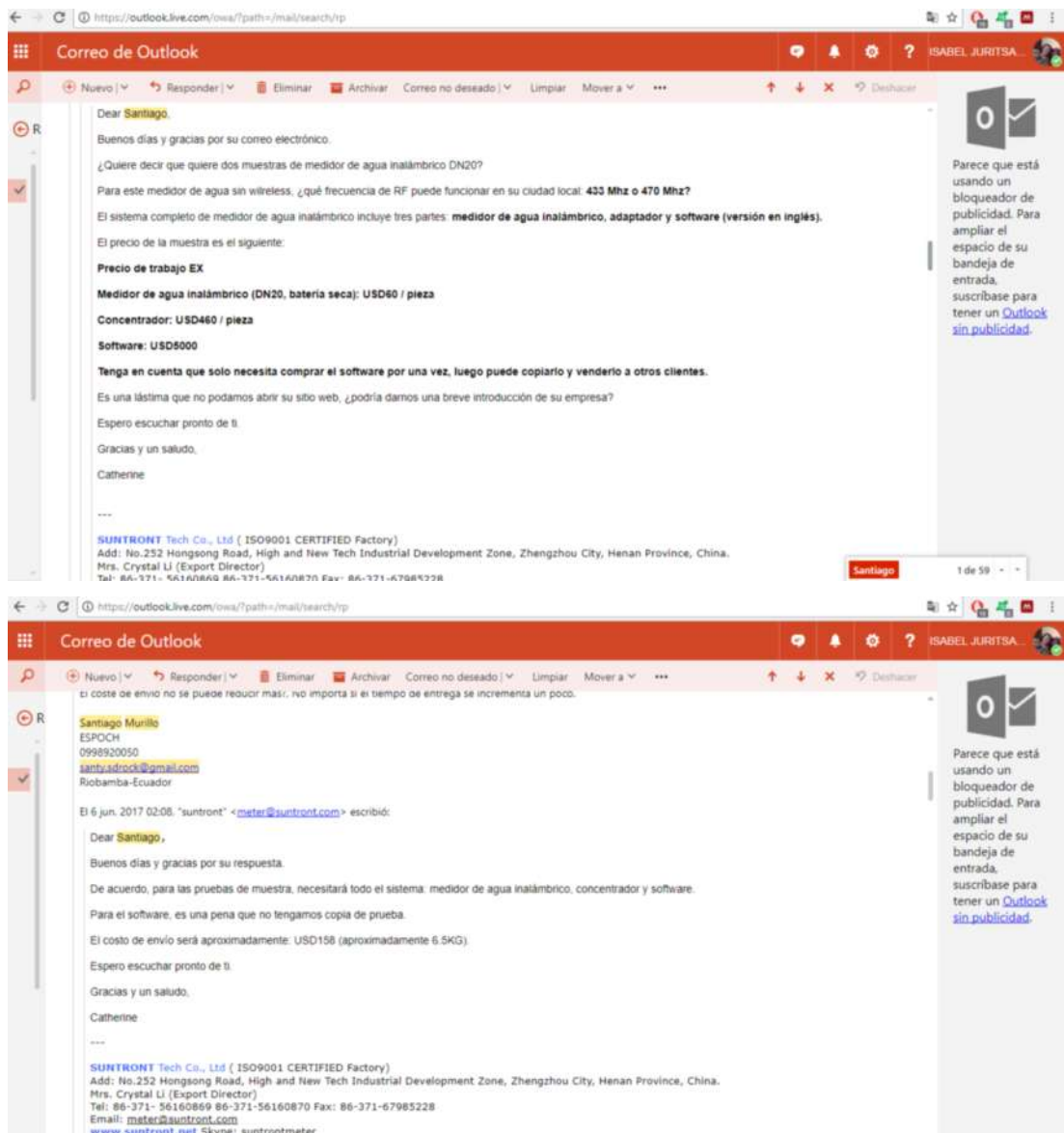
var
  corte: integer;
begin
  qCortes.Close;
  qCortes.Open;
  while not qCortes.Eof do
  begin
    corte := qCortescontador.AsInteger;
    if (corte >= 3) then
    begin
      lbCedula.Caption := qCortesidUsuario.AsString;
      lbNumeroMedidor.Caption :=
qCortesnumMedidor.AsString;
      lbEstado.Caption := qCortesEstado.AsString;
      TMSMQTTClient1.Publish(corteMedidor, 'c');
      sStatusBar1.Panels[1].Text := 'Cortes realizados..';
      abort;
    end;
  end;

  qCortes.Next;
end;
end;

procedure
TfrmCortes.TMSMQTTClient1.ConnectedStatusChanged(A
Sender: TObject;
  const AConnected: Boolean; AStatus:
TTMSMQTTConnectionStatus);
begin
  if AConnected then
  begin
    sStatusBar1.Panels[0].Text := 'Conectado..';
  end;
end;
end.

```

## Anexo J. Cotización de medidor inteligente inalámbrico de agua potable



## Anexo K. Costo de licencia Delphi Tokio Professional



Delphi 10.2 Tokyo

### Ediciones de productos

Características	Professional	Enterprise	Architect
Desarrollo apps para...			
Seto de componentes Integrales VCL y FireMonkey <sup>1</sup>	✓	✓	✓
Código fuente de biblioteca <sup>2</sup>	✓	✓	✓
Crear aplicaciones de bases de datos con conectividad local/integrada <sup>3</sup>	✓	✓	✓
Full Commercial use license	✓	✓	✓
1 <sup>st</sup> YEAR UPDATE SUBSCRIPTION <sup>4</sup>	✓	✓	✓
Crear aplicaciones de bases de datos con conectividad local/integrada <sup>3</sup>	Opcional <sup>5</sup>	✓	✓
Mobile Add-On Pack <sup>6</sup>	Opcional	✓	✓
FireDAC Client/Server Pack <sup>7</sup>	Opcional	✓	✓
RAD Server (Developer License) <sup>8</sup>		✓	✓
DataSnap multi-tier SDK <sup>9</sup>		✓	✓
Herramientas de modelado de datos <sup>10</sup>			✓
	<a href="#">Más información &gt;</a>	<a href="#">Más información &gt;</a>	<a href="#">Más información &gt;</a>
	* New User <del>\$1,405.00</del> Total: \$1,264.50	* New User <del>\$2,514.00</del> Total: \$2,986.90	* New User <del>\$5,623.00</del> Total: \$4,498.40