UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Diseño e implementación de un laboratorio para la gestión y monitoreo remoto de una red de distribución de agua y de variables meteorológicas del campus UTPL

TRABAJO DE TITULACIÓN.

AUTORES: Cumbicos Romero, José Alberto;
Suárez Loján, Johan Nicolás.

DIRECTOR: Quiñones Cuenca, Manuel Fernando, Mg.

LOJA – ECUADOR

2018

2018

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Mgtr.
Manuel Fernando Quiñones Cuenca
DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: “Diseño e implementación de un laboratorio para la gestión y monitoreo remoto de una red de distribución de agua y de variables meteorológicas del campus UTPL”, realizado por Cumbicos Romero José Alberto y Suárez Loján Johan Nicolás, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, octubre de 2018

f.)……

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Nosotros Cumbicos Romero José Alberto y Suárez Loján Johan Nicolás declaramos ser autores del presente trabajo de titulación: “Diseño e implementación de un laboratorio para la gestión y monitoreo remoto de una red de distribución de agua y de variables meteorológicas del campus UTPL”, de la Titulación Electrónica y Telecomunicaciones, siendo Mg. QUIÑONES CUENCA MANUEL FERNANDO director del presente trabajo; y eximimos expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de nuestra exclusiva responsabilidad.

Adicionalmente declaramos conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

f.)……
Autor: Cumbicos Romero José Alberto.
Cédula: 0703631572

f.)……
Autor: Suárez Loján Johan Nicolás.
Cédula: 1105857864

DEDICATORIA

A mi padre que en paz descanse, a mi madre y hermana.

José Alberto Cumbicos Romero.

A mis amados padres Boris y Gaby, a hermano Kevin y a mi abuelo Leoncio.

Johan Nicolás Suarez Loján

AGRADECIMIENTO

Agradezco a Dios por la vida, lo que he recibido y lo que está por venir. A mi madre y hermana por el apoyo constante durante mi formación académica y personal. A mis abuelos por sus consejos y educación basada en valores. A mi familia que siempre me está ayudando y motivando a mejorar y salir adelante.

Finalmente agradezco a mis amigos y compañeros de estudio.

José Alberto Cumbicos Romero.

Agradezco a mis padres, por su confianza y apoyo durante mi formación. A mi abuelo por sus consejos y su valiosa guía. A mi hermano por su amor, paciencia y respaldo.

Gracias a mi universidad por permitirme ser un profesional en lo que me apasiona. Gracias a cada maestro que hizo parte de este proceso, de manera especial al Ing. Manuel Quiñones, Ing. Marco Morocho y al Ph.D. Holger Benavides.

Finalmente, gracias a todos mis amigos y a todas las personas que me brindaron su apoyo durante la realización de este trabajo de titulación.

Johan Nicolás Suárez Loján

COLABORACIÓN

El desarrollo e implementación del sistema propuesto se logra con el apoyo del Departamento de Geología y Minas e Ingeniería Civil de la UTPL. Agradecemos la colaboración de: PhD. Holger Benavides, Sr. Carlos Vivanco y Sr. Esteban Eras.

Para la implementación y pruebas del sistema propuesto se contó con la ayuda de la Unidad Municipal de Agua Potable y Alcantarillado de Loja (UMAPAL). Agradecemos la colaboración de: Ing. George Buele, Ing. Carlos Jimbo, Sr. Edwin Álvarez y el Sr. Byron Cabrera por los permisos, adecuaciones e instalación de un nodo sensor en el tanque de reserva Época Bajo.

TERMINOLOGÍA

ADC	Analog Digital Converter
AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
ANATEL	Agência Nacional de Telecomunicações
ANSI	American National Standards Institute
API	Application Programming Interface
ASIC	Application Specific Integrated Circuit
AT&T	American Telephone and Telegraph
AWG	American Wire Gauge
CE	Conformité Européene
CoAP	Constrained Application Protocol
CSV	Comma-Separated Values
DNS	Domain Name System
EPROM	Erasable Programmable Read-Only Memory
FCC	Federal Communications Commission
FS	Full Scale
GPIO	General Purpose Input/Output
GPM	Galones por minuto
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
I2C	Inter-integrated Circuit
IC	Industry Canada
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IoT	Internet of Things
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
MAC	Media Access Control
MANET	Mobile Ad Hoc Network
MQTT	Message Queuing Telemetry Transport
NC	Normal Close
NO	Normal Open
NPT	National Pipe Thread
NTP	Network Time Protocol
PC	Personal Computer
PE	Polyethylene
PIR	Pasive Infra Red
PRDA	Prototivo de red de Distribución de Agua
PSI	Pounds per Square Inch

PTCRB	PCS Type Certification Review Board
PVC	Polyvinyl chloride
PVDF	Polyvinylidene fluoride
PWM	Pulse-width modulation
QoS	Quality of Service
RCM	Regulatory Compliance Mark
RDA	Red de Distribución de Agua
REST	Representational State Transfer
RFID	Radio Frequency Identification
RH	Relative Humidity
RP-SMA	Reverse Polarity-SubMiniature A
RTC	Real Time Clock
SI	Sistema Internacional
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TFTP	Trivial file transfer Protocol
TLS	Transport Layer Security
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VDC	Volts Direct Current
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPA2	Wi-Fi Protected Access 2
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
XMPP	Extensible Messaging and Presence Protocol

INDICE DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

RESUMEN

El agua es un recurso vital y limitado, las redes para su distribución presentan algunos problemas como fisuras, rajaduras, roturas o reventamientos. Por este motivo, se diseña e implementa una Red de Sensores Inalámbricos en un Sistema de Distribución de Agua, desarrollando un sistema robusto de monitoreo y gestión de este recurso. Se utilizan recursos de software y hardware abierto para la construcción de los elementos de la red. Para la comunicación inalámbrica se utilizan los módulos de radio Xbee900 Pro con el protocolo DigiMesh. Los datos recopilados por los nodos sensores Smart Security y Smart Agriculture de Libelium se publican en un servidor del Internet de las cosas para su visualización en tiempo real. Se utiliza el protocolo de capa de aplicación MQTT. Esta red fue validada y desplegada en el campus de la Universidad Técnica Particular de Loja y forma parte del proyecto Smart Water Network.

PALABRAS CLAVES: Distribución de agua, Libelium, MQTT, WSN, Sensores, Smart Water.

ABSTRACT

Water is a vital limited resource and the distribution networks present problems such as cracks, breaks or bursts. That is why in this work a Wireless Sensor Network is designed and implemented to monitor a Water Distribution System, allowing the creation of a strong management system of this resource. Open hardware and software platforms are used for the construction of the network elements. The Xbee900 Pro radio modules are used for the wireless communication using the DigiMesh protocol. The data collected from the sensor nodes Smart Security & Smart Agriculture by Libelium are published on an Internet of Things server for their real time visualization. Application protocol MQTT is used. This WSN was validated and deployed in the Universidad Técnica Particular de Loja campus and it is part of the Smart Water Network project.

KEYWORDS: Water Distribution, Libelium, MQTT, WSN, Sensors, Smart Water.

INTRODUCCIÓN

En la actualidad los dispositivos que utilizan Internet y brindan servicios en la nube son cada vez más utilizados. El desarrollo de aplicaciones usando Redes de Sensores Inalámbricos (WSN por sus siglas en inglés) es un tema novedoso e interesante. Además, permite la implementación del Internet de las Cosas [1] [2]. Las WSN cuentan con sensores autónomos distribuidos que monitorean las condiciones físicas o ambientales [3]. Algunas consideraciones generales para el diseño de cada nodo de la red son: la tarjeta para la adquisición de datos, el módulo para la comunicación inalámbrica, el sistema de energización y los sensores que utilizarán.

El agua es un recurso vital, muy importante para garantizar el bienestar y desarrollo de un país. Sin embargo, es un recurso limitado y las redes para su distribución presentan algunos problemas como fisuras, rajaduras, roturas o reventamientos. Estos problemas generan pérdidas al sistema, pérdidas del recurso (agua) y también pérdidas económicas. Las técnicas de monitoreo tradicionales son altamente ineficientes [4][5]. Por estas razones, la utilización de una WSN en un sistema de distribución de agua permitirá tener un sistema robusto de monitoreo y gestión de este recurso.

En el presente trabajo se presenta el diseño e implementación de una Red de Sensores Inalámbricos para el monitoreo de la red de distribución de agua y de variables meteorológicas en el campus de la Universidad Técnica Particular de Loja (UTPL). Esta WSN forma parte del proyecto Smart Water Network: “Conformación de nudos de monitoreo remoto de la red de Distribución de agua potable para el campus UTPL” del área de Geología, Minas e Ingeniería Civil. La WSN recopila algunas variables como: flujo y presión de agua en tuberías, nivel horizontal de líquido, detección de fugas de agua en línea y en punto. Conjuntamente, se recolectan variables meteorológicas de: la temperatura y humedad ambiental, la precipitación pluvial y la velocidad y dirección del viento.

En el desarrollo de este trabajo se realiza una revisión bibliográfica del estado actual de las tecnologías, estándares y protocolos para el diseño de la WSN. Se construyen, acondicionan y programan los distintos equipos que conforman la red. Todos los datos recolectados se almacenan en un servidor IoT en Internet. Posteriormente, se despliegan los nodos sensores en diferentes puntos del campus de la Universidad Técnica Particular de Loja para realizar pruebas y validar los resultados.

CAPÍTULO I

ALCANCE DEL TRABAJO DE TITULACIÓN

1.1. Problemática

En la ciudad de Loja, las redes de agua potable del casco urbano central presentan falencias. Un alto nivel de pérdidas de agua, un elevado porcentaje de tuberías con más de 40 años de uso, un parque de contadores antiguo, un sistema de redes de abastecimiento caótico tanto en su topología, distribución geométrica y ubicación. Por otra parte, un porcentaje del agua que ingresa a la red se pierde por fugas, conexiones ilegales, contadores de agua mal operados, entre otros, que originan un suministro de agua intermitente y poco eficaz [6].

El sistema de distribución de agua del campus de la UTPL muestra un panorama similar. Se estima que la calidad y cantidad de agua distribuida no cumple con los mínimos requeridos en determinados espacios y tiempos. Otros indicadores como: la particular variación de niveles de agua en los depósitos, las elevadas cifras de pago mensual por planillas y las notorias bajas de presión en varios nudos de consumo, revelan que la red presenta fugas de agua [7].

Los métodos tradicionales de monitoreo (medidores analógicos, inspecciones visuales) son altamente ineficaces. Debido a ello, la implementación de un sistema de monitoreo robusto, utilizando técnicas modernas como las WSN, permitirá automatizar la adquisición de información en tiempo real de la red de distribución. Además de optimizar los costos de operación, manteniendo las presiones y la calidad del agua requeridas.

1.2. Justificación

En el trabajo de titulación se diseñan y construyen los elementos de la red de sensores inalámbrica y se validan sus componentes de software y hardware. El uso de esta WSN permite recolectar datos de la red de distribución de agua en tiempo real. Estos datos junto con diferentes metodologías para la detección de fugas desarrollarán un sistema de gestión de redes de distribución de agua escalable hacia industrias o municipios.

Además, este sistema es afín con los objetivos del Plan Nacional del Buen Vivir respecto a la infraestructura y acceso a Agua Potable. Específicamente con los siguientes:

- “Mejorar la calidad de vida de la Población”.
- “Garantizar el acceso universal, permanente, sostenible y con calidad a agua segura y a servicios básicos de saneamiento, con pertinencia territorial, ambiental, social y cultural”.

1.3. Objetivos.

1.3.1. Objetivo general

- Diseñar e implementar una WSN para la gestión y monitoreo remoto de una red de distribución de agua y de variables meteorológicas en un ambiente de laboratorio.

1.3.2. Objetivos específicos

- Realizar revisión bibliográfica del estado actual de tecnologías, estándares y protocolos para WSN.
- Programar la tarjeta Waspmote para la recolección y envío de datos de los sensores de interés a un *gateway*, para su posterior almacenamiento en un servidor en la nube.
- Diseñar y desplegar la red de sensores inalámbricos.
- Realizar pruebas de la red y analizar los resultados obtenidos.

1.4. Metodología

Este trabajo de titulación es desarrollado en cuatro fases: una primera fase de investigación en la que se toma en cuenta todas las tecnologías y protocolos necesarios en una WSN. Una segunda fase de diseño y construcción de la red, en la cual se bosqueja y construye cada elemento de la WSN. La tercera fase de implementación de la red, en la que se realiza una validación de los datos recolectados por los sensores contra datos de equipos de medición de referencia y se ejecuta un primer despliegue de la red en un ambiente de laboratorio para la recolección de datos de las distintas variables. Finalmente, en la fase de pruebas y resultados se demuestra el funcionamiento de la WSN.

CAPÍTULO II

ESTADO DEL ARTE

2.1. Trabajos relacionados

En los últimos años muchos investigadores han demostrado el potencial de las WSNs para el monitoreo de sistemas de distribución de agua. Utilizan diferentes técnicas de monitoreo, de adquisición de datos y de algoritmos para el procesamiento de los datos recolectados.

Whittle *et al* [5] [8] en el proyecto *WaterWiSe@SG* exhiben un sistema capaz de detectar rompimientos y fugas en un sistema de distribución de agua en una sección del centro de Singapur. En un primer documento denominado *"WaterWiEe@sg: a testbed for continuous monitoring of the water distribution system in Singapore"* se describe la primera fase del proyecto. Despliegan una pequeña WSN que recolecta datos hidráulicos. Desarrollan un banco de pruebas donde validan sus prototipos, en hardware y software; y se determinan algunas técnicas de procesamiento para detectar y localizar eventos [5]. Un trabajo posterior: *"Sensor Networks for Monitoring and Control of Water Distribution Systems"* expone la capacidad del sistema para detectar y ubicar rompimientos, y además la de identificar zonas de riesgo en la red de distribución de agua. Se concluye que un sistema de monitoreo provee información crítica de la red de distribución, la cual permite optimizar los costos de operación, manteniendo las presiones y la calidad del agua [8].

Ayadi *et al* [9] refieren los distintos métodos utilizados para detectar fugas. Las técnicas de monitoreo utilizando sensores pueden ser aplicadas dentro o fuera de las tuberías. Indican que las WSN pueden ser extremadamente útiles para monitorear redes de distribución, convirtiéndose en una herramienta precisa y eficaz. Adicionalmente, Xhafa *et al* [10] en su artículo "*Automation Control on Water Supply Networks"*, presentan una metodología para reducir las pérdidas en redes de distribución de agua. Realizan un estudio en Lipljan-Kosovo junto con la empresa RWC Prishtina. Utilizan *dataloggers* gprs de presión y de flujo para la detección de fugas. Se concluye que el uso de sistemas de monitoreo reduce las pérdidas de agua y también pérdidas económicas.

2.2. Sistema de distribución de agua

Un Sistema de Distribución de Agua es el conjunto de tuberías, válvulas y bombas, a través del cual el agua potable es trasladada desde la planta de tratamiento hacia los hogares, oficinas e industrias. El diseño del sistema de distribución para el servicio de agua potable depende del plano de calles, topografía y la localización de las fuentes de captación. La Red de Distribución de Agua (RDA) también incluye: tanques de almacenamiento de agua, medidores del consumo e hidrantes para control de incendios [11][12].

Los requerimientos más importantes que la RDA debe cumplir son: Proveer a cada usuario el caudal y la presión adecuada de agua; y, que el agua entregada cumpla con la calidad establecida.

2.2.1. Caudal o flujo

Las tuberías permiten el transporte del agua desde los nudos fuente (inyección) hasta los nudos de consumo. Las líneas hidráulicas son el enlace entre los sistemas de captación y los sistemas de distribución. La ubicación de la fuente determina si el agua es trasladada mediante gravedad o es bombeada. Se asume que el agua se mueve por las tuberías a un flujo uniforme [12].

La metodología más aplicada para la medición y estimación del flujo se basa en una simplificación de la ecuación de continuidad. Establece que para un líquido incompresible, el caudal (Q) es igual al producto del área (A) de la sección transversal de la corriente y su velocidad media (v) [13]. Se puede expresar mediante:

$$Q = A \cdot v \tag{1}$$

Generalmente, el caudal se expresa en pies cúbicos por segundo o metros cúbicos por segundo [m^3/s]. Otra medida común son los galones por minuto [GPM]. En el presente trabajo se utilizará como medida el litro por minuto [l/min]. En un sistema de distribución municipal la demanda de agua varía a lo largo del día. El monitoreo del caudal en una red permite obtener patrones de la demanda de agua [12]. Posterior a un análisis, cualquier cambio en esta tendencia puede indicar la existencia de problemas en el sistema de distribución de agua.

2.2.2. Presión

La presión (p) se define como la cantidad de fuerza (F) aplicada por unidad de área (A). Matemáticamente se expresa:

$$p = \frac{F}{A} \tag{2}$$

En el SI la unidad de presión es el pascal, un pascal es un newton de fuerza sobre un metro cuadrado [14]. Esta unidad de medida es muy pequeña, en una RDA la presión se encuentra alrededor de los 300 kPa; por lo tanto, el uso de libras por pulgada cuadrada (psi) es más recomendable y es la variable que se utilizará en este trabajo (1 psi equivale a 6 894.76 pascales).

La presión de un sistema de distribución municipal oscila entre los 60 psi a 75 psi para edificios comerciales y entre 40 psi para áreas residenciales. En casos específicos pueden existir secciones de la red que alcancen presiones altas de 100 psi. No obstante, en nuevos diseños de RDA se prefiere trabajar con presiones bajas. Operar a presiones bajas reduce las fugas en las líneas y el desperdicio de agua [12]. La presión interna de tuberías es una variable de gran interés para el monitoreo de una red de distribución de agua. Mediante el monitoreo de la presión es posible detectar fugas y roturas en la tuberías [15].

2.3. Estación meteorológica

Una estación meteorológica es una instalación que tiene los instrumentos y equipos necesarios para la medición de las condiciones atmosféricas. Entrega la información necesaria para el estudio del clima. Las mediciones que son tomadas por la estación incluyen: temperatura, humedad, cantidad de precipitación pluvial y velocidad y dirección del viento.

2.3.1. Temperatura

En términos físicos, la temperatura es la energía cinética promedio de las moléculas. Las moléculas más rápidas en movimiento tienen temperaturas más altas. Los informes de temperatura estándar están en grados Celsius (°C). Sin embargo, Estados Unidos y otras naciones utilizan la escala Fahrenheit (°F). Diversos derivados de la temperatura se informan comúnmente tales como la temperatura por hora, la temperatura máxima y mínima dentro de un período de 24 horas, y la temperatura media diaria, mensual, estacional y anual [16].

2.3.2. Dirección del viento

Los vientos se describen por dirección, velocidad y ráfagas y se miden en relación con la rotación de la Tierra [17]. La dirección del viento puede ser representada usando direcciones cardinales alrededor de una brújula (por ejemplo, noreste) o grados alrededor de un círculo que se mueve en el sentido de las agujas del reloj comenzando y terminando con el norte. Los vientos predominantes (o las condiciones de viento más comunes durante un período de tiempo especificado) se representan a menudo en una rosa del viento que muestra las frecuencias de la dirección del viento en un lugar [16].

2.3.3. Velocidad del viento

Es el cambio de la distancia en el tiempo o la velocidad relativa de la atmósfera local. Antes de la llegada de anemómetros precisos, se utilizó la escala de viento graduada de Beaufort para hacer una evaluación cualitativa de la fuerza del viento en base a las condiciones de la superficie del mar, que van desde 0 (calma) hasta 12 (huracanes); La moderna escala de Beaufort atribuye las velocidades del viento a las descripciones categóricas en kilómetros por hora [18]. Además de las unidades estándar SI e inglés, las velocidades del viento se reportan regularmente en nudos donde 1 nudo es igual a 0.51 m por segundo.

2.3.4. Precipitación

En meteorología, la precipitación es cualquier forma de hidrometeoro que cae de la atmósfera y llega a la superficie terrestre. Este fenómeno incluye lluvia, llovizna, nieve, aguanieve, granizo, pero no neblina ni rocío, que son formas de condensación y no de precipitación. La cantidad de precipitación sobre un punto de la superficie terrestre es llamada pluviosidad. La

precipitación pluvial se mide en mm, que es la cantidad de agua por metro cuadrado en una hora. También es el equivalente de un litro de agua por metro cuadrado [19].

2.4. Redes de sensores inalámbricos.

Las Redes de Sensores Inalámbricos (WSN) son las más usadas para la recolección de datos. Una WSN se puede describir como un conjunto de sensores que crean una red para monitorear de forma cooperada un ambiente físico determinado.

Una WSN no se limita a un ambiente específico, teniendo diversos campos de aplicación. Son ampliamente utilizadas para monitorear y controlar parámetros ambientales, industriales, médicos, calidad de agua. Con la información obtenida se puede controlar y tomar decisiones más acertadas para solucionar problemas. Además, la intervención de personal es mínima, porque los nodos sensores pueden ser programados para que tomen decisiones automáticas en base a las mediciones que realizan [20].

Los nodos realizan mediciones periódicas a través de los sensores y transmiten los datos recolectados a un *gateway* usando enlaces de radio. Haciendo uso del Internet se puede enviar esta información a sistemas remotos de procesamiento, para su análisis, visualización y almacenamiento [21].

2.4.1. Características.

Las principales características que presenta una red de sensores se detalla a continuación [22].

- **Bajas tasas de transmisión:** son necesarias ya que la información se envía en tramas pequeñas, cuya longitud dependerá del número de sensores que se encuentre conectados en el nodo.
- **Bajo ciclo de trabajo:** la mayor parte del tiempo el nodo se encuentra en modo se suspensión y se “despierta” para realizar el proceso de medición y de transmisión de datos. Como el nodo se mantiene en reposo la mayoría del tiempo permite reducir el consumo de energía.
- **Tamaño restringido:** el entorno donde se ubica el nodo debe ser lo menos invasivo, es decir que tiene que ser pequeño para no ocupar mayor espacio. En redes corporales es un parámetro crítico.
- **Bajo costo:** es el parámetro que toma en cuenta el usuario final para la implementación. Por lo que los dispositivos deben ser económicos sin comprometer la calidad y precisión de los sensores.

- **Energía limitada:** es un factor importante que va a limitar las prestaciones del nodo en cuanto a número de sensores, procesamiento y transmisión de datos. Es necesario buscar eficiencia en el consumo de energía.
- **Localización:** un nodo sensor no debe estar aislado de la red, por lo tanto, tiene que estar dentro del radio de alcance del nodo central o de otro nodo que funcione como enrutador.
- **Escalabilidad:** una WSN tiene que trabajar independientemente del tamaño y el número de nodos que se conecten a la red. También se debe adaptar a los cambios de la topología cuando se elimina, agrega o mueve un nodo sensor.

2.4.2. Arquitectura del nodo sensor.

El nodo sensor es el elemento principal en una WSN y puede monitorear uno o varios fenómenos físicos. El nodo no se limita a la recopilación de datos, también dispone de subsistemas de procesamiento, comunicación, almacenamiento y energía. Por lo tanto, el hardware está diseñado para cumplir estos requerimientos específicos. El nodo sensor es un sistema embebido que está compuesto por un microcontrolador, sensores, transmisores de radio y baterías que se describen a continuación [23]. La Figura 2.4.1 muestra un diagrama de bloques con los componentes de un nodo sensor.

- **Microcontrolador:** es considerado como una computadora con características limitadas, incluye un procesador, memoria, puertos de entrada y salida para conectar dispositivos externos.
- **Sensores:** dependen de la aplicación y de las propiedades físicas que van a monitorear y permiten medir la temperatura, humedad, luz entre otros. A medida que aumenta la calidad y precisión también sube el costo.
- **Transmisores de radio:** son módulos que permiten la comunicación a través de enlaces de radio. Estos componentes envían o reciben datos y son los que consumen la mayor energía. En algunas placas los módulos de comunicación vienen integrados.
- **Baterías:** son la fuente de energía del nodo para que pueda realizar las tareas programadas y son limitadas por el tamaño y capacidad. Es un subsistema crítico por lo que se debe emplear mecanismo para que la energía no se agote.

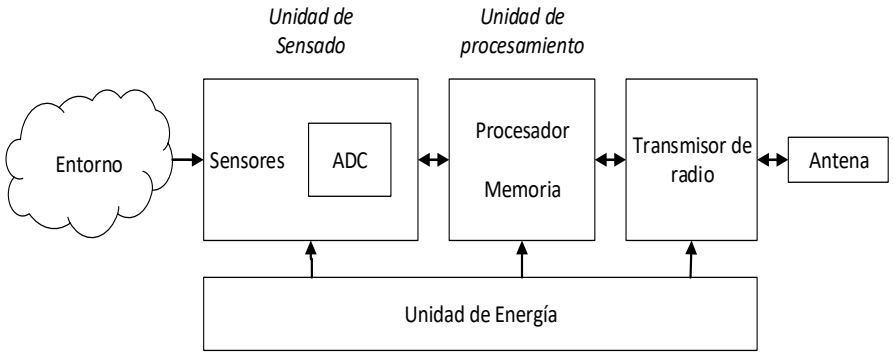


Figura 2.4.1. Diagrama de bloques de un nodo sensor.
Fuente y elaborado por los autores, basados en [24].

2.4.3. Protocolos de comunicación de IoT.

Los nodos de una WSN cumplen varios roles y utilizan varios protocolos para comunicarse unos con otros. En las capas inferiores utilizan tecnologías de radio como: DigiMesh, RFID, WLAN, WPAN, ZigBee, entre otras [25]. En la capa de aplicación, existe una gran cantidad de protocolos para IoT. Cada uno se enfoca en resolver un problema específico de la comunicación, de esta manera se tienen algunos protocolos como: HTTP, MQTT, CoAP, REST, XMPP y AMQP [26].

Estos protocolos tienen los siguientes requerimientos:

- Es necesario que el código del protocolo quepa dentro de la pequeña memoria de los dispositivos IoT.
- Deben ser de bajo consumo energético al momento de procesar los datos y enviarlos a otra fuente.
- Los protocolos deben ser operables en distintos ambientes. Es decir, deben seguir estándares.
- Ser "*open source*" o de código abierto. De esta manera pueden ser ampliamente aceptados y utilizados con dispositivos de distintos fabricantes [27].

2.4.3.1. *DigiMesh*

DigiMesh es un protocolo inalámbrico de red desarrollado por Digi. Un aspecto exclusivo de DigiMesh frente a otros similares como ZigBee® o Z-Wave, es que todos los dispositivos en una red DigiMesh son del mismo tipo. La arquitectura de red es más sencilla ya que no es necesario diferenciar entre nodos finales, coordinadores, routers, etc. Todos los dispositivos en la red son capaces de enrutar, dormir (para eficiencia energética) y de comunicarse inalámbricamente utilizando la red de malla [28].

Entre algunas de sus características principales están:

- Configuración y expansión de la red más sencilla
- DigiMesh está disponible en la banda de 2.4 GHz y en bandas interiores como la de 900MHz. Lo que permite alcanzar mayores distancias (+60km)
- Características de seguridad como encriptado 128-bit AES.

2.4.3.2. *HTTP, Hypertext Transfer Protocol*

HTTP es un protocolo web de mensajería. Fue desarrollado por Tim Berners-Lee y estandarizado en 1997 por la IETF. HTTP es un protocolo basado en texto y no define el tamaño de la carga útil del encabezado ni del mensaje, estos factores dependen del servidor o de la programación. Utiliza TCP como protocolo de transporte predeterminado y TLS/SSL para seguridad. Este protocolo no define QoS, y requiere soporte adicional para ello [25].

HTTP presenta algunos problemas cuando se aplica en IoT. Este protocolo debe transferir un gran número de paquetes pequeños, lo que causa un gran consumo de recursos de red y retrasos debido a la sobrecarga del protocolo. Además, el direccionamiento IP depende de la localización física, lo cual complica el control de la red [29].

2.4.3.3. *MQTT, Message Queuing Telemetry Transport*

Es uno de los protocolos de capa de aplicación *publisher/subscriber* más utilizados en WSN. El funcionamiento general es bastante sencillo: el cliente se suscribe a un *topic* (tema) con un *broker* y el *Publisher* envía datos relacionados de ese *topic* al *broker,* finalmente el cliente suscrito consume los datos asociados a ese *topic*. El *topic* está en formato jerárquico [27]. MQTT trabaja con el protocolo TCP, y adicionalmente tiene tres formatos de QoS:

- **QoS 0:** Sin transmisión asegurada, el mensaje puede ser enviado o no.
- **QoS 1:** Transmisión asegurada, el mensaje es enviado al receptor una vez por lo menos.
- **QoS 2**: Servicio asegurado en aplicaciones, el mensaje es enviado exactamente una vez sin pérdida ni por duplicado.

2.5. Plataforma IoT ThingSpeak

ThingSpeak es una plataforma de Internet de las Cosas (IoT) que permite almacenar, visualizar y analizar datos de sensores. Para enviar información al servidor se puede utilizar protocolos de comunicación como HTTP y MQTT con calidad de servicio nivel 0 [30]. La visualización de los datos es organizada en canales y dentro de los mismos puede existir hasta ocho variables. Para realizar análisis la plataforma puede integrarse con MATLAB y aprovechar las herramientas que ofrece este software. Asimismo, ThingSpeak dispone la opción de generar alertas al cumplir ciertas condiciones y realizar notificaciones mediante

Twitter. Además, la aplicación de *TalkBack* permite crear comandos que pueden ser utilizados por otros dispositivos para encender o apagar actuadores [31].

Para utilizar esta plataforma se requiere la creación de una cuenta. En su versión gratuita es posible enviar tres millones de mensajes al año y el intervalo de actualización de mensajes es de 15 segundos. Solamente permite tres suscripciones simultaneas de MQTT y compartir a tres usuarios la visualización de los canales. Las versiones de paga no están limitadas y son más recomendables para proyectos grandes. Cada canal creado puede contener hasta 8 campos y tiene las *apikeys* específicas para la escritura y lectura de datos en el canal.

2.6. Plataformas hardware *open source*

Con el objetivo de reducir el tiempo de desarrollo en los prototipos, existen plataformas hardware disponibles con diferentes características. Los costos varían según la robustez, velocidad de procesamiento, cantidad de memoria y número de pines que tenga el dispositivo. La elección del hardware depende de las necesidades del proyecto. Para el presente trabajo se ha decidido trabajar con las herramientas open source: Libelium, Dragino MS14, Módulo IoT M328W.

2.6.1. Tarjeta Waspmote

La tarjeta Waspmote es una plataforma *Open source* para construir redes inalámbricas de bajo consumo creada por la empresa Libelium. Se programa con un API proporcionado por la empresa fabricante. Al ser una plataforma modular nos permite agregar dispositivos de acuerdo con los requerimientos de monitoreo y envío de datos. El resumen de las características que tiene la tarjeta Waspmote se presenta en la Tabla 2.6.1.

Tabla 2.6.1. Características de la tarjeta Waspmote

Descripción	Especificaciones
Microcontrolador	Atmega1281
Frecuencia	14.7456 MHz
SRAM	8 KB
EPROM	128 KB
Peso	20 g
Dimensiones	73.5 x 51 x 13 mm
Rango de temperatura	[-10 ºC+65 ºC]
Reloj	RTC (32 KHz)
Entradas y salidas	7 analógicas, 8 digitales, 1 PWM, 2 UART, 1 I2C, 1USB, 1 SPI
Batería	3.3-4.2 V
Carga USB	5 v – 480 mA
Panel solar	6-12 v – 330 mA

Fuente: Elaborado por los autores, basado en [32].

2.6.2. Dragino MS14

El Draguino MS14 es un dispositivo de IoT que permite trabajar con Linux en proyectos que utilizan microcontroladores. Es una placa de bajo costo con hardware y software abierto, ejecuta el sistema operativo OpenWrt que es una distribución confiable de Linux para sistemas embebidos. La administración se realiza por Web, SSH o Serie. Su diseño modular permite reducir costos y tiempos de desarrollo, con la opción de montar una placa hija con un MCU. El objetivo de este módulo es resolver problemas de conectividad y acceder a servicios de Internet mediante una red WiFi o Ethernet [33]. La Tabla 2.6.2 muestra las principales características técnicas del dispositivo.

Tabla 2.6.2. Características del Dragino MS14

Descripción	Especificaciones
Procesador	400 MHz, 24k MIPS
Flash	16 MB
RAM	64 MB
Entrada de alimentación	9 ~ 12 VDC
Interfaz RJ45	2 x 10 m/100 m
Puerto USB	1 x USB 2.0
Protocolo WiFi	802.11 b/g/n
Interfaces	SPI /PCM/UART/I2C

Fuente: Elaborado por los autores, basado en [33].

2.6.3. Módulo IoT M328W

El Módulo IoT M328W está diseñado para una fácil programación e instalación en proyectos IoT donde se requiera implementar diferentes protocolos inalámbricos de comunicación (ZigBee, bluetooth, 433 MHz, etc). Posee un socket XBee estándar. Además, tiene un área de prototipado para implementar pequeños circuitos de forma rápida y sencilla. Puede ser instalado en un módulo MS14, posee un microcontrolador ATmega328P de ATMEL que puede ser programado y depurado con el IDE de Arduino vía red Wifi [33]. La Tabla 2.6.3 muestra las principales características técnicas del módulo.

Tabla 2.6.3. Características del módulo M328W

Descripción	Especificaciones
Microcontrolador	ATMEGA328P
Voltaje de operación	3.3 VDC
Flash	32 KB
EEPROM	1 KB
Reloj	16 MHz
I/O digitales	13
Entradas analógicas	6
Bee socket	1

Fuente: Elaborado por los autores, basado en [33].

2.6.4. Módulo XBee

El módulo de comunicación Xbee es utilizado en redes de sensores para enviar información a través de enlaces inalámbricos. Trabajan en la banda ISM y permiten implementar tecnologías como ZigBee o DigiMesh. Los dispositivos presentan ventajas como bajo consumo energético, tamaño reducido y tasas de transmisión óptimas para aplicaciones de WSN. Para la configuración se utiliza el software multiplataforma "*XCTU*" que tiene una interfaz gráfica que resume todos los parámetros del módulo [34]. El modelo empleado es XBEE09P y sus características se muestran en la Tabla 2.6.4**.**

Tabla 2.6.4. Características del módulo de comunicación XBEE09P

Descripción	Especificaciones
Tasa de datos	156 kbps
Rango en interiores	140 m
Rango en exteriores con línea de vista y antena de alta ganancia	11 km
Potencia de transmisión	50 mW (+17 dBm)
Sensibilidad del receptor	-100 dBm
Configuración	API, comandos AT
Banda de frecuencias	900 MHz ISM
Conector para antena	RP-SMA
Voltaje de alimentación	3.0 - 3.6 VDC

Fuente y elaborado por los autores, basados en [35].

CAPÍTULO III

DISEÑO Y CONSTRUCCIÓN

3.1. Requerimientos del sistema

Previo al proceso de diseño y desarrollo del sistema, se han propuesto algunos requerimientos que permitirán cumplir con los objetivos planteados. Los cuales se detallan a continuación:

- Los nodos desplegados deben monitorear las variables de interés en la red de distribución de agua y enviar los datos periódicamente a un dispositivo coordinador.
- El *Gateway* debe recibir la información de todos los nodos y publicarlos a un servidor remoto para su almacenamiento y visualización en tiempo real.
- Desde la plataforma IoT realizar el encendido y apagado de actuadores, por ejemplo, una electroválvula.
- Utilizar tecnologías de comunicación acorde al ámbito de aplicación y disponibilidad.
- Hacer uso de componentes de hardware y software abierto.

3.2. Diseño de la WSN

El sistema propuesto consta de varios nodos distribuidos en la zona de interés a monitorear y un dispositivo denominado *gateway* que sirve como puerta de enlace entre los nodos y la plataforma IoT, como se aprecia en el esquema de una red de sensores inalámbricos de la Figura 3.2.1. Se utiliza una topología de red tipo estrella. Mediante módulos de radio XBee, con protocolo DigiMesh, los nodos se conectan con el *gateway.* Con la infraestructura de acceso a Internet de la UTPL, el gateway realiza el envío de datos al servidor ThingSpeak. El protocolo de capa de aplicación empleado entre el *gateway* y el servidor es MQTT. Para la visualización de los datos recolectados, los usuarios acceden al servidor mediante una página web.

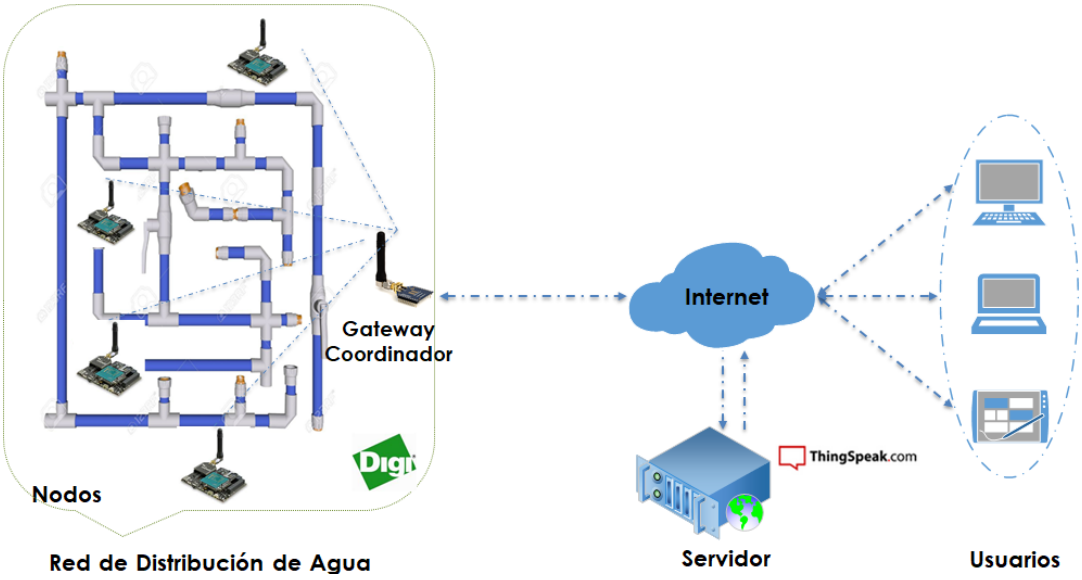


Figura 3.2.1. Red de Sensores Inalámbricos.
Fuente y elaborado por los autores, basados en [36].

3.2.1. Estructura de tramas

Las tramas encapsulan los datos antes de ser transmitidos, para ser interpretados de forma correcta por los dispositivos que intervienen en la red. El *gateway* está programado para procesar un formato especifico de trama, cuando la información no cumple con este formato los datos son descartados.

3.2.1.1. *Trama entre los nodos y el gateway*

El formato de la trama que se utiliza para enviar la información consta de dos partes cabecera y carga útil, como se ve en la Figura 3.2.2. La cabecera que contiene un identificador único para cada nodo y sirve para que el g*ateway* conozca quién envió la trama. La segunda parte es la carga útil que incluye todos los valores de los sensores y es creada con la sintaxis que debe tener la trama para subir los datos a ThingSpeak. Esta plataforma permite almacenar un máximo de 8 campos por canal [37].

CABECERA		CARGA UTIL						
ID nodo	&	field1=valor	&	field2=valor	&	…	&	field8=valor

Figura 3.2.2. Formato de la trama de envío desde el nodo al *Gateway*.
Fuente y elaborado por los autores.

Cuando la trama llega al *gateway*, se toma la carga útil y mediante el protocolo MQTT los datos son publicados en la plataforma ThingSpeak. En la Figura 3.2.3 se muestra un ejemplo de trama.

```
a&field1=90&field2=12.321&field3=122.0&field4=80&field5=50.2&
```

Figura 3.2.3. Ejemplo de trama de envío desde el nodo al *Gateway*.
Fuente y elaborado por los autores.

3.2.1.2. *Trama entre plataforma IoT y nodos.*

Esta trama es empleada para encender y apagar la electroválvula. Es creada en la plataforma ThingSpeak con el formato mostrado en la Figura 3.2.4 a**.** La cabecera es utilizada por el *Gateway* para identificar al nodo que debe enviar la trama. El comando es una cadena de caracteres para ejecutar la acción que corresponda. En la Figura 3.2.4 b se muestra un ejemplo de la trama.

CABECERA		COMANDO
ID nodo	:	string

a)

a:cerrar

b)

Figura 3.2.4. Trama para accionar la electroválvula. a) Estructura y b) Ejemplo.
Fuente y elaborado por los autores.

3.3. Diseño del *Gateway*

Para el diseño de este dispositivo, se hace uso de tres componentes, como se ve en la Figura 3.3.1. En primer lugar, está el Dragino MS14 donde se ejecuta el programa principal y se encarga de procesar la información que puede llagar desde los nodos o la plataforma. El segundo componente es la placa IoT M328W que se configura en modo “consola BEE” para comunicarse y controlar al módulo de comunicación desde el Dragino MS14. El tercer componente es el módulo de radio XBee, tiene la función de realizar la comunicación bidireccional y de forma inalámbrica entre los nodos al *gateway*.

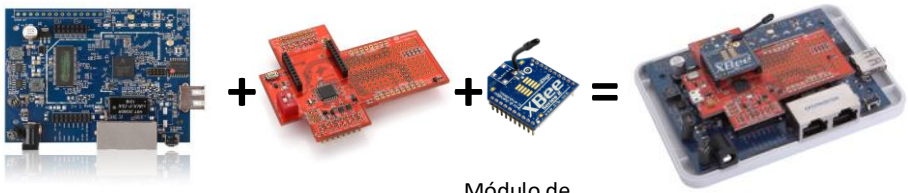


Figura 3.3.1. Componentes hardware del *Gateway*.
Fuente y elaborado por los autores, basados en [33].

3.3.1. Preparación del hardware y software.

El proceso de instalación del sistema operativo OpenWrt en el Dragino MS14 se detalla en el ANEXO B.1**,** la versión que se utiliza es la IoT-4.3.2. Por motivos de seguridad y evitar que usuarios no autorizados ingresen al dispositivo se realizan algunas configuraciones que son detalladas en el ANEXO B.2.

Para almacenar los datos en la plataforma ThingSpeak se utiliza la librería *paho-mqtt* versión 1.4.0 de Python. La misma, va a permitir la conexión al *bróker* MQTT para publicar los mensajes [38]. Las peticiones a la plataforma son realizadas mediante HTTP y con la librería *requests* versión 2.18.4 [39]. La instalación se realiza con la herramienta *pip* que está integrada en el sistema y facilita la descarga e instalación de paquetes, los comandos a utilizar respectivamente son: # *pip install paho-mqtt* y # *pip install requests*.

El intercambio de información entre la placa principal y el módulo de radio se realiza mediante comunicación serie. Para transmitir y recibir los datos a través del puerto UART del Dragino MS14 se utiliza la librería *XBee* versión 2.3.2 de Python [40]. El comando para instalar es el siguiente: # *pip install XBee.* Los problemas encontrados en la librería y durante el proceso de instalación se solucionan en el ANEXO C.

Para ensamblar el XBee sobre el módulo IoT M328W es necesario habilitar el modo “consola BEE”. De esta forma, se logra establecer la comunicación de forma correcta entre el Dragino MS14 y el XBee. La configuración de los siguientes pines de propósito general o GPIO (General Purpose Input/Output), permite establecer el modo mencionado [41]:

- El GPIO24 en bajo para desconectar el UART del microcontrolador mega328p.
- El GPIO27 en alto para conectar el UART del XBee.
- El GPIO20 en alto para poner al mega328p en estado de reset.

El conjunto de instrucciones para configurar los GPIOs es guardado en un archivo con el nombre “set_gpio.sh”. En la Figura 3.3.2 se muestran las líneas de código que contiene.

set_gpio.sh

```
#!/bin/sh

echo 24 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio24/direction
echo 0 > /sys/class/gpio/gpio24/value

echo 27 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio27/direction
echo 1 > /sys/class/gpio/gpio27/value

echo 20 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio20/direction
echo 1 > /sys/class/gpio/gpio20/value
```

Figura 3.3.2. Líneas de código para configurar los GPIOs.
Fuente y elaborado por los autores, basados en [42].

3.3.2. Algoritmo de funcionamiento

El algoritmo es desarrollado en el lenguaje de programación Python y el nombre del script es “nodo_central.py”. En la Figura 3.3.3 se muestra el diagrama de flujo del algoritmo y a continuación se describe el funcionamiento:

1. Se inicia incluyendo las librerías necesarias y definiendo variables que serán utilizadas durante la ejecución. Se crea una tabla con un identificador y las claves de escritura (APIKEY) y canal de la plataforma ThingSpeak. Después, se establece la comunicación serial entre el Dragino MS14 y el XBee, configurando el puerto (“ttyATH0”) y la velocidad en baudios entre los dispositivos (115200).
2. El puerto serie se pone en modo escucha, esperando recibir información de los nodos mediante el XBee. Este proceso cuenta con un tiempo de espera de 10 segundos. Si alguna información es recibida durante este tiempo, se lee la cabecera de la trama y se toma el identificador de nodo para buscar si está registrado en la tabla anteriormente creada. Si el identificador no es encontrado la trama es rechazada, caso contrario, se abre un *socket* y se publica el mensaje en la plataforma ThingSpeak mediante MQTT.
3. Si al abrir el *socket* se producen excepciones del tipo *gaierror* y *timeout*, el programa realiza un intento adicional. Este número puede ser modificado en el código.
4. Al agotarse el tiempo de espera o finalizado de procesar los datos, se pasa a verificar si existen tramas en ThingSpeak mediante peticiones HTTP utilizando el método GET. Cuando exista alguna trama se revisa la cabecera para identificar a que nodo pertenece y se procede a enviar la información a través del módulo de comunicación. Finalizado este proceso se repite el punto 2.

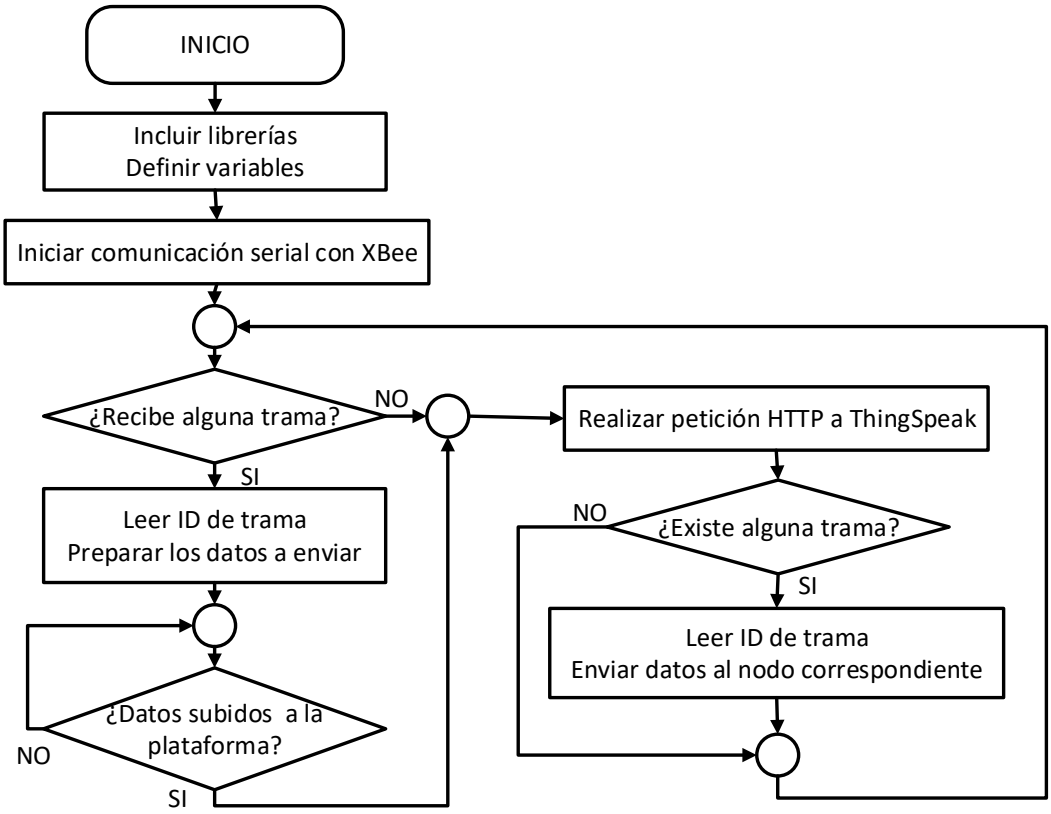


Figura 3.3.3. Diagrama de flujo del algoritmo del *Gateway*.
Fuente y elaborado por los autores.

Los scripts creados ("set_gpio.sh" y "nodo_central.py") son configurados para que se ejecuten en el arranque del sistema. De esta forma, se inician de forma automática al proporcionar alimentación al *Gateway*. El procedimiento que se debe realizar es el siguiente: dentro de la carpeta */etc/init.d/* se crea un archivo con el nombre "*gateway*", su contenido se muestra en la Figura 3.3.4**.** Los scripts deben tener permisos de ejecución (véase ANEXO B.3) y para que se inicien junto con el sistema es necesario utilizar el comando: # */etc/init.d/gateway enable*. Para evitar conflictos en el manejo del GPIO24 se deshabilita del arranque el fichero "iotd" con el siguiente comando: # */etc/init.d/iotd disable.*

gateway

```
1  #!/bin/sh /etc/rc.common
2
3  START=99
4
5  start()
6  {
7      /root/set_gpio.sh
8      /root/nodo_cental.py &
9  }
```

Figura 3.3.4. Contenido del archivo "gateway".
Fuente y elaborado por los autores, basados en [43].

3.3.3. Construcción

Los componentes de hardware y software que se utilizan en el desarrollo del *gateway* son integrados para formar un sistema embebido. En la Figura 3.3.5 se muestran los elementos instalados dentro de una caja IP65 (resistente al agua y polvo). El cable de alimentación y el de red salen a través de un conector. El cargador tiene una salida de 12 VDC y soporta corrientes de hasta 1 A. Para la conexión a la red de Internet que tiene el Campus se utiliza un cable Ethernet categoría 6. También se emplea un *pigtail* con conector RPSMA para la instalación de la antena omnidireccional de 8 dBi en la parte externa de la caja.

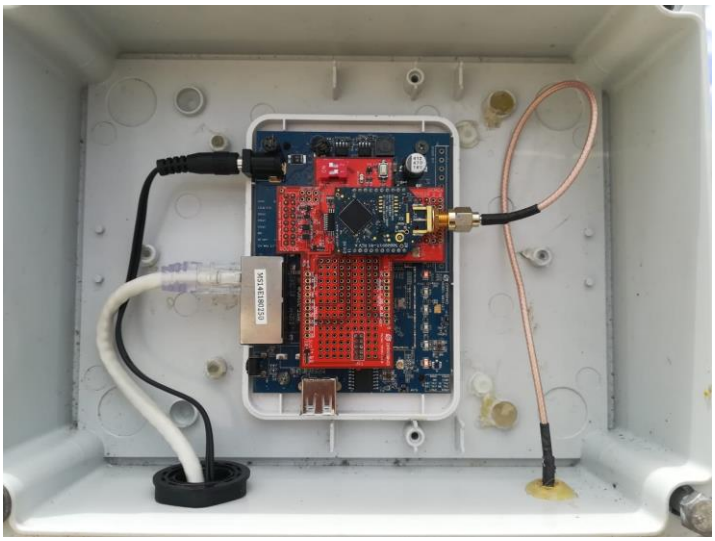


Figura 3.3.5. Elementos del *Gateway* instalados.
Fuente y elaborado por los autores.

3.4. Descripción del nodo

El nodo es el componente de la WSN encargado de recopilar los datos que se requieren monitorear y de enviarlos al *Gateway*. Una WSN puede tener uno o más nodos desplegados dentro de su área de cobertura. A continuación, se presenta el equipo que tiene la función de nodo dentro de la red de sensores.

3.4.1. Waspmote Plug and Sense!

Plug and Sense! es una línea de productos de la empresa Libelium. Esta línea simplifica el trabajo de los desarrolladores para que puedan enfocarse más en los servicios y aplicaciones, dejando la parte electrónica en un segundo plano. Estos dispositivos permiten desplegar una WSN de manera fácil y escalable, con costos de mantenimiento mínimos. Entre las características del producto se encuentran:

- Encapsulado robusto con protección IP65.
- Sockets externos para la conexión de los sensores.

- Panel solar externo.
- Soporte para distintas tecnologías de radio.
- Puerto USB para reprogramar el nodo.
- Certificaciones: CE (Europa), FCC (USA), IC (Canadá), ANATEL (Brasil), RCM (Australia), PTCRB (USA), AT&T (USA).

El Waspmote Plug and Sense está conformado por la tarjeta Waspmote, una placa de sensores, un módulo de radio y una batería. Existen diferentes modelos y configuraciones, dependiendo del tipo de sensor a ser utilizado. Cada modelo utiliza diferentes circuitos de acondicionamiento, como consecuencia la placa de sensores es diferente. El sensor permitido para cada socket depende del modelo. La distribución de los sockets se muestra en la Figura 3.4.1. El módulo de radio depende del entorno y de la aplicación de la red que se va a desplegar. El dispositivo Waspmote Plug and Sense puede integrar algunos módulos de radio para la comunicación inalámbrica como: Xbee, WiFi, 4G, Sigfox y LoRaWAN. Para una descripción de todos los modelos y de los sensores que se pueden utilizar ver [44].

Figura 3.4.1. Distribución de sockets para los sensores para Waspmote Plug and Sense! Fuente y elaborado por [44].

3.4.1.1. *Plug & Sense! Smart Security*

Plug and Sense! Smart Security es el dispositivo equipado con la placa de sensores *events-sensor-board_3.0.* y con el módulo de radio XBee 900 PRO. Este modelo permite la conexión de los sensores indicados en la Tabla 3.4.1.

Tabla 3.4.1. Sensores permitidos para Smart Security

Socket	Sensores permitidos para cada socket	
	Sensor	Modelo
A, C, D o E	Temperatura, humedad y presión (atmosférica)	9370-P
	Luminosidad	9325-P
	Ultrasonido (medir distancia)	9246-P
	Presencia – PIR	9212-P
	Nivel de líquido	9239-P, 9240-P
	Presencia de líquido (punto y línea)	9243-P, 9295-P
	Efecto de hall	9207-P
B	Caudalímetro	9296-P, 9297-P, 9298-P
F	Conexión para relé	9270-P

Fuente y elaborado por los autores, basados en [45].

3.4.1.1. Plug & Sense! Smart Agriculture

Los modelos de los módulos *Smart Agriculture* permiten el monitoreo de múltiples parámetros ambientales. Las principales aplicaciones de este Plug & Sense! son: agricultura de precisión, sistemas de riego, invernaderos, estaciones meteorológicas, entre otras [46]. En este trabajo de titulación se utiliza este modelo como una estación meteorológica. Permite la conexión de los sensores indicados en la Tabla 3.4.2.También está equipado con el módulo de radio XBee 900 PRO.

Tabla 3.4.2. Sensores permitidos para Smart Agriculture

Socket	Sensores permitidos para cada socket	
	Sensor	Modelo
A	Temperatura + humedad (sensirion)	9247
B	Presión atmosférica	9250
C	Temperatura del suelo	86949
	Humedad del suelo	9248
D	Estación meteorológica ws-3000 (anemómetro + veleta de viento + pluviómetro)	9256
E	Humedad de suelo	9248
F	Humedad de hojas	9249
	Humedad de suelo	9248

Fuente y elaborado por los autores, basados en [46].

3.4.2. Algoritmo de funcionamiento

El nodo se puede programar de algunas maneras. El algoritmo que se utiliza depende del tipo y número de sensores y también de los intervalos de medición. Los sensores de Libelium se encuentran en librerías ya instaladas. Los sensores adaptados requieren de funciones adicionales.

3.4.2.1. *Algoritmo general del nodo*

Es el algoritmo que se ejecuta indefinidamente en el nodo de la red. En la Figura 3.4.2 (izquierda) se muestra el diagrama de flujo del algoritmo y a continuación se describe su funcionamiento:

1. Se incluyen las librerías necesarias. Se ingresa la dirección MAC del módulo Xbee que está en el *Gateway* para garantizar un envío seguro. Se definen las variables y las clases dependiendo de los sensores que se van a conectar. En las clases se especifica el modelo del sensor (para la clase *flowClass*), o el socket en el que se conectará.
2. Se enciende la paca de sensores y el módulo RTC, en caso de realizar un seguimiento por el puerto serial también se debe habilitar la comunicación USB con la línea USB.ON().
3. Se establece el tiempo de envío con una alarma.
4. Luego se miden los valores instantáneos de los sensores conectados y se almacenan en las variables correspondientes.
5. Se realizan algunas operaciones como: obtener valores mínimos, máximos, sacar promedio, comparar umbrales, entre otras. Dependiendo del ambiente en el que se instala.
6. Si aún no se cumple el tiempo de envío, se espera cierto tiempo para realizar la siguiente medición. Los sensores pueden realizar mediciones según su tiempo de respuesta. Por ello este *delay* depende de los sensores que se conecten al nodo.
7. Cuando el tiempo de envío se cumple el algoritmo entra a la subrutina de envío de datos que se explicará más adelante. Se limpian todas las variables y se establece el tiempo para el próximo envío.

3.4.2.2. *Subrutina de envío de datos*

La subrutina de envío es la encargada de armar la trama con los datos de las variables y enviarla usando el módulo de radio Xbee al *Gateway.* En la Figura 3.4.2 (derecha) se muestra el diagrama de flujo y a continuación de detalla el funcionamiento del algoritmo.

1. Se enciende el módulo Xbee y se ingresa un *delay* de 3 segundos para su inicialización.
2. Se crea la trama utilizando la librería *WaspBody.h.* Se ingresa la variable que se va a enviar dentro de la función *body.addField().* También se puede presentar por el puerto serial con la función *body.showBody().*
3. Se crea una variable para la persistencia del envío.
4. La variable *n* sirve para reintentar el envío 3 veces en caso de error.

5. La trama se envía con la línea *error = xbeeDM.send(RX_ADDRESS, body.buffer, body.length);* Luego se compara si existe error. En caso de error la variable n incrementa en uno y se reintenta el envío, este comportamiento persiste por tres veces. Si la trama no se envía en los tres intentos, esa trama se pierde y la subrutina finaliza. En el caso contrario, si el envío es correcto, la subrutina finaliza y continúa el algoritmo principal.

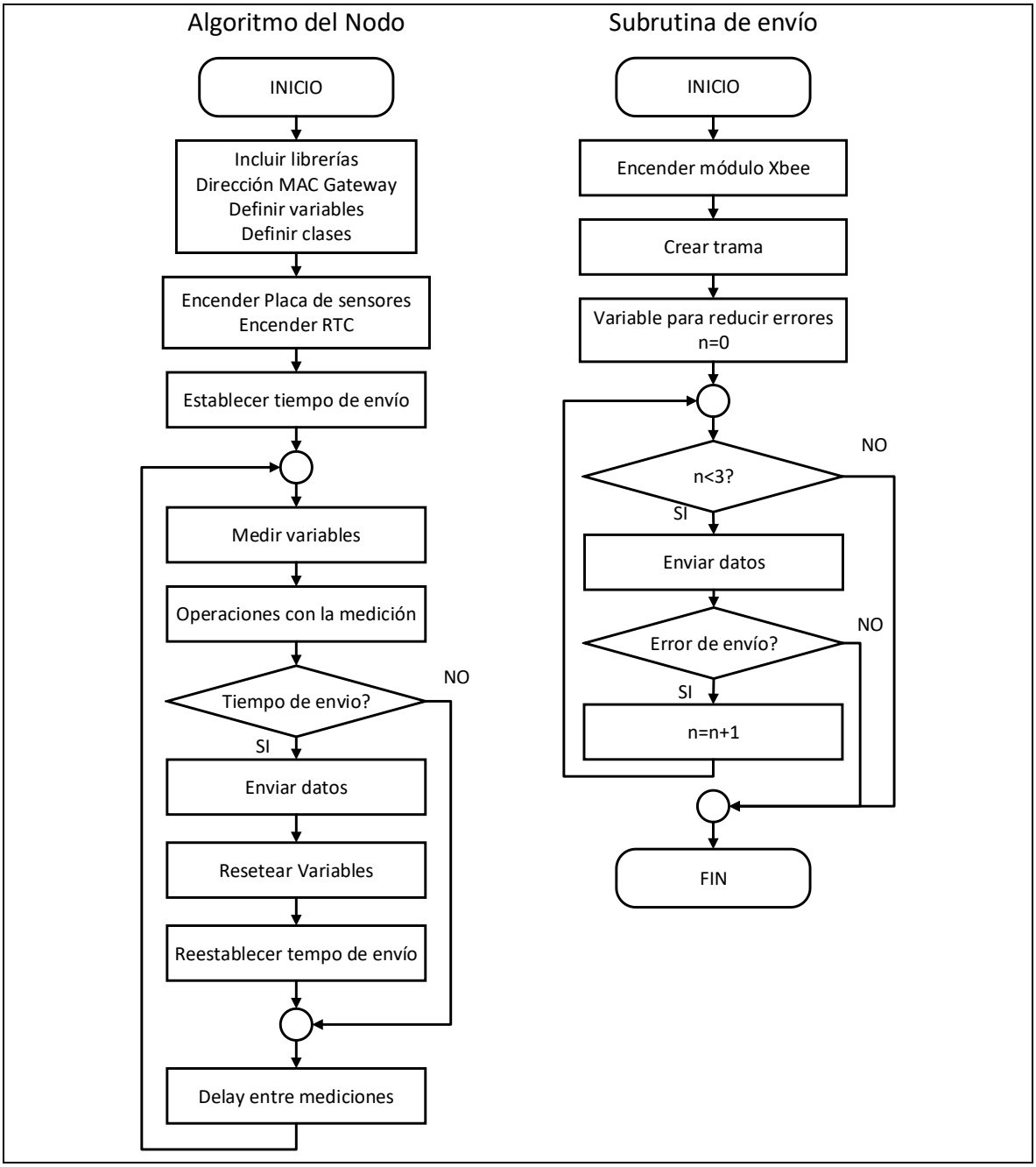


Figura 3.4.2. Diagramas de flujo de: Algoritmo de funcionamiento general del nodo (izquierda). Subrutina para el envío de tramas al *Gateway* (derecha).
Fuente y elaborado por los autores.

3.5. Especificaciones de sensores

Los sensores son los dispositivos que “traducen” las señales no eléctricas, como el caudal y presión en las tuberías, a señales eléctricas que el nodo puede interpretar [47]. Al procesar estas señales se obtiene la información que el sistema de monitoreo requiere. Los dispositivos encargados de recopilar esta información son los sensores que se detallan en esta sección.

3.5.1. Sensor de presión MLH150PSB01A

El sensor fabricado por Honeywell de la serie MLH permite medir presiones en ambientes exigentes. Combina la tecnología ASIC (Application Specific Integrated Circuit) con un diseño de diafragma metálico aislado. Está construido con conectores eléctricos estándar (Metri-Pack 150) y con el puerto de presión 1/4-18 NPT (ANSI B1.20.1). Entrega una salida Radiométrica de 0.5 VDC a 4.5 VDC con un voltaje de excitación de 5.0 VDC. La señal de salida es proporcional al rango de medición de este sensor (0 a 150 psi) [48]. Algunos datos adicionales se pueden ver en la Tabla 3.5.1.

Tabla 3.5.1. Características del sensor de presión MLH150PSB01A

Descripción	Especificaciones
Presión de Operación	150 psi
Presión de Prueba	450 psi
Presión de rompimiento	1500 psi
Exactitud	± 0.50% FS
Tiempo de respuesta	< 2 ms
Temperatura de funcionamiento	-40ºC A 125ºC
Voltaje de alimentación	5.0 VDC
Señal de Salida	DE 0.5 VDC A 4.5 VDC
Escala completa (Span)	4 VDC
Peso	57 g

Fuente y elaborado por los autores, basados en [48].

3.5.2. Sensor de caudal WTP100-400

El sensor de Seametrics WTP100-400 es un medidor de turbina para instalaciones en línea. Consta de una única parte móvil, un rotor helicoidal. El cuerpo del sensor es de PVC Schedule 80 para tuberías de 4”. Puede ser alimentado con voltajes desde 6 VDC a 24 VDC y entrega una salida de 0/160 pulsos por segundo dependiendo del flujo que atraviesa al rotor. El rango de medición es de 6 GPM a 600 GPM (22.7 l/min a 2 271 l/min) [49][50]. Las principales características se muestran en la Tabla 3.5.2.

Tabla 3.5.2. Características del sensor de caudal WTP100-400

Descripción	Especificaciones	
Voltaje de alimentación	6 A, 24 VDC	
Presión de máxima	150 psi a 24˚ C	
Exactitud	+/- 1% FS	
Tiempo de respuesta	1 S	
Temperatura máxima	50˚ C	
Rango de medición	22.7 – 2 271 l/min	
Tipo de salida	0/160 pulsos de corriente de sumidero (*sinking current*)	
Conexión a Tubería	4”	
Materiales	Cuerpo	PVC
	Soporte de turbina	PVC
	Rotor	KYNAR (PVDF)
	Eje	Cerámica de zirconio
	Rodamiento	Zafiro y rubí
Cable	#22 AWG	
Factor K	6.0	

Fuente y elaborado por los autores, basados en [49].

3.5.3. Sensor de caudal YF-G1

El sensor de flujo YF-G1 entrega una señal de salida que consiste en una serie de pulsos digitales cuya frecuencia es proporcional al caudal del líquido que atraviesa al sensor. La señal digital se encuentra en el rango de 0Hz a 100Hz. Se instala en línea con la tubería, el cuerpo del sensor es plástico y cuenta con dos extremos roscados para tuberías de 1” [45]. Las características se detallan en la Tabla 3.5.3.

Tabla 3.5.3. Características del sensor de caudal YF-G1

Descripción	Especificaciones
Voltaje de operación	3.3 VDC – 24 VDC
Presión máxima	290 psi
Exactitud	±3% FS
Tiempo de respuesta	1 s
Temperatura de trabajo	0ºC – 80ºC
Rango de medición	2-100 l/min
Conexión a tubería	1”

Fuente y elaborado por los autores, basados en [45].

3.5.4. Sensor de caudal YF-S201

El sensor YF-S201 mide el flujo de agua de manera similar al YF-G1. Se instala en línea con la tubería y gracias a un molinillo determina la cantidad de agua que se mueve a través de él. El sensor magnético de efecto de hall entrega un pulso eléctrico con cada revolución del molinillo [51]. Mediante el conteo de los pulsos entregados es posible calcular el caudal de líquido. Este sensor es poco preciso y las mediciones pueden ser afectadas por la presión del

fluido o la orientación del sensor. Cuenta con dos extremos roscados de ½". Su uso es recomendable para tareas de medición básicas, sus características se muestran en la Tabla 3.5.4.

Tabla 3.5.4. Características del sensor de caudal YF-S201

Descripción	Especificaciones
Voltaje de operación	5VDC a 18VDC
Consumo de Corriente	15mA a 5VDC
Presión máxima	280 psi
Exactitud	±10% FS
Tiempo de respuesta	1 s
Temperatura de trabajo	-25ºC a 80ºC
Rango de medición	1-30 l/min
Conexión a tubería	1/2"

Fuente y elaborado por los autores, basados en [52].

3.5.5. Sensor de detección de fugas 9295-P

El sensor 9295-P consta de una unión de dos materiales semiconductores a manera de un cable trenzado. Este sensor detecta líquidos conductores presentes en cualquier punto a lo largo del cable. Es un sensor resistivo, lo que indica que disminuye su resistividad a mayor cantidad de agua presente. De esta manera se puede utilizar para detectar fugas a lo largo de una tubería. Sus características se muestran en la Tabla 3.5.5.

Tabla 3.5.5. Características del sensor de detección de fugas 9295-P

Descripción	Especificaciones
Longitud	5 metros de sensor + 2 metros de cable jumper
Peso	18 g
Límite de fuerza de tiro	60 kg
Diámetro del cable	5.5 mm
Resistencia del núcleo	3 ohm/100 m
Temperatura de trabajo	75ºC
Líquidos detectables	Agua
Material	PE + Aleación

Fuente y elaborado por los autores, basados en [45].

3.5.6. Sensor de detección de líquidos 9243-P

Este sensor funciona variando su resistencia entre sus dos contactos frente a la presencia de líquidos. Luego, un *reed switch* es conmutado de "abierto" a "cerrado", o a "abierto" nuevamente cuando el líquido desaparece. Tiene las características que se muestran en la Tabla 3.5.6.

Tabla 3.5.6. Características del sensor de detección de líquidos 9243-P

Descripción	Especificaciones
Voltaje máximo de operación	100 VDC
Temperatura de trabajo	+5ºC ~ +80ºC
Líquidos detectables	Agua

Fuente y elaborado por los autores, basados en [45].

3.5.7. Sensor de nivel de líquido horizontal PTFA3415

La operación del PTFA3415 se basa en el estado de un *switch,* que puede estar "abierto" o "cerrado", dependiendo del nivel del líquido en su extremo. El sensor está construido para usarse en líquidos comestibles y algunos ácidos. Se trata de un sensor de dos estados, la única información que entregan es:

- El nivel del líquido está bajo el punto en donde está instalado el sensor, o
- El nivel de líquido está por encima del punto de instalación.

Sus características se muestran en la Tabla 3.5.7.

Tabla 3.5.7. Características del sensor PTFA3415

Descripción	Especificaciones
Nivel de medición	Horizontal
Líquidos	Agua
Material	Propylene
Temperatura de Operación	-10ºC a +80ºC

Fuente y elaborado por los autores, basados en [45].

3.5.8. Sensor de temperatura y humedad (SHT75)

El sensor SHT75 fabricado por Sensirion incorpora un sensor capacitivo para la medición de la humedad ambiental relativa y otro sensor para la temperatura ambiental. Sus especificaciones se muestran en la Tabla 3.5.8.

Tabla 3.5.8. Características del sensor SHT75

Descripción	Especificación
Alimentación	2.4VDC a 5.5VDC
Temperatura	
Rango de Medición	-40ºC a +123.8ºC
Resolución	0.04ºC
Exactitud	±0.4ºC (rango 0ºC a +70ºC), ±4ºC (rango -40ºC a +125ºC)
Tiempo de respuesta	5 s
Humedad	
Rango de Medición	0 a 100% RH
Resolución	0.05% RH
Exactitud	±1.8% RH
Tiempo de respuesta	8 s

Fuente y elaborado por los autores, basados en [46].

3.5.9. Estación meteorológica WS-300

La estación meteorológica WS-300 permite medir parámetros básicos para la medición del clima. Cuenta con tres sensores diferentes: un anemómetro, una veleta de viento y un pluviómetro.

3.5.9.1. *Anemómetro*

El anemómetro consiste en un *reed switch* normalmente abierto que se cierra durante un corto período de tiempo cuando los brazos del anemómetro completan un giro. Su salida es una señal digital cuya frecuencia será proporcional a la velocidad del viento. Detalles adicionales se muestran en la Tabla 3.5.9.

Tabla 3.5.9. Características del anemómetro

Descripción	Especificación
Sensibilidad	2.4 km/h /vuelta
Rango de velocidad del viento	0-240 km/h
Altura	7.1 cm
Longitud del brazo	8.9 cm
Conector	RJ11

Fuente y elaborado por los autores, basados en [46].

3.5.9.2. *Veleta de viento*

La veleta de viento consiste en un eje que gira libremente sobre una plataforma dotada de una red de ocho resistencias conectadas a ocho interruptores que están normalmente cerrados hasta cuando un imán en el eje actúa sobre ellos, lo que permite distinguir 16 posiciones diferentes (el equivalente a una resolución de 22.5°). La resistencia equivalente de la veleta de viento, junto con una resistencia de 10 kΩ, forman un divisor de tensión,

alimentado a 3.3 VDC a través de un interruptor digital. Sus especificaciones se encuentran en la Tabla 3.5.10.

Tabla 3.5.10. Características de la veleta de viento

Descripción	Especificación
Altura	8.9 cm
Longitud	17.8 cm
Precisión	22.5°
Rango de resistencia	688 Ω a 120 kΩ

Fuente y elaborado por los autores, basados en [46].

3.5.9.3. *Pluviómetro*

El pluviómetro consiste en un pequeño recipiente que, una vez lleno (0.28mm de agua aproximadamente), cierra un interruptor, vaciándose automáticamente. El resultado es una señal digital cuya frecuencia es proporcional a la intensidad de la precipitación. Más detalles aparecen en la Tabla 3.5.11.

Tabla 3.5.11. Características del pluviómetro

Descripción	Especificación
Altura	9.05 cm
Longitud	23 cm
Capacidad del balancín	0.28 mm de lluvia

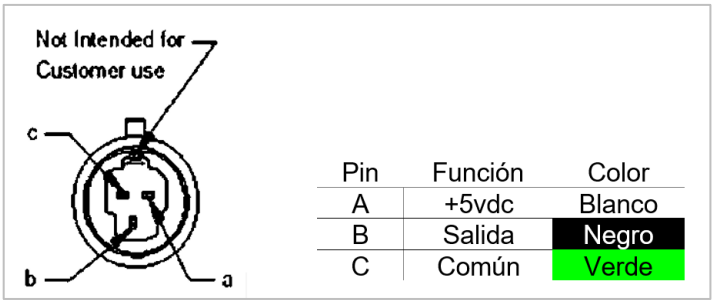
Fuente y elaborado por los autores, basados en [46].

3.6. Integración de sensores al nodo Smart Security

Algunos sensores requieren de un acondicionamiento previo para integrarse al nodo Smart Security ya que no son comercializados por Libelium. Para su acondicionamiento ha sido necesaria la elaboración de pequeños circuitos electrónicos y de nuevas conexiones dentro del nodo.

3.6.1. Sensor de presión MLH150PSB01A

El MLH150PSB01A es un sensor de comportamiento muy similar al de algunos sensores especificados para la placa *events-sensor-board_3.0,* por ello requiere solamente de cableado adicional en el nodo para funcionar. Este sensor utiliza el conector Metri-Pack 150, por ello es necesaria la elaboración de un cable jumper. En la Figura 3.6.1 se muestran los pines en el sensor, su función y el código de colores utilizado en el cable jumper. Para más detalles ver [48].



Pin	Función	Color
A	+5vdc	Blanco
B	Salida	Negro
C	Común	Verde

Figura 3.6.1. Distribución de pines en el sensor MLH150PSB01A y código de colores.
Fuente y elaborado por [48].

En el otro extremo del cable jumper se utiliza un conector eléctrico de tres pines con protección IP68 para la conexión con el nodo.

3.6.1.1. *Conexión del sensor a la placa de sensores*

La placa de sensores *events-sensor-board_3.0.* cuenta con terminales para la conexión de otros sensores. La salida del sensor MLH150PSB01A se puede conectar a cualquiera de las entradas analógicas de la placa. Es decir, en cualquiera de los Sockets A, C, D o E (en la placa: Input 1, 2, 3 o 4). Además, es necesario el voltaje de alimentación y tierra. Esta conexión se muestra en la Figura 3.6.2.

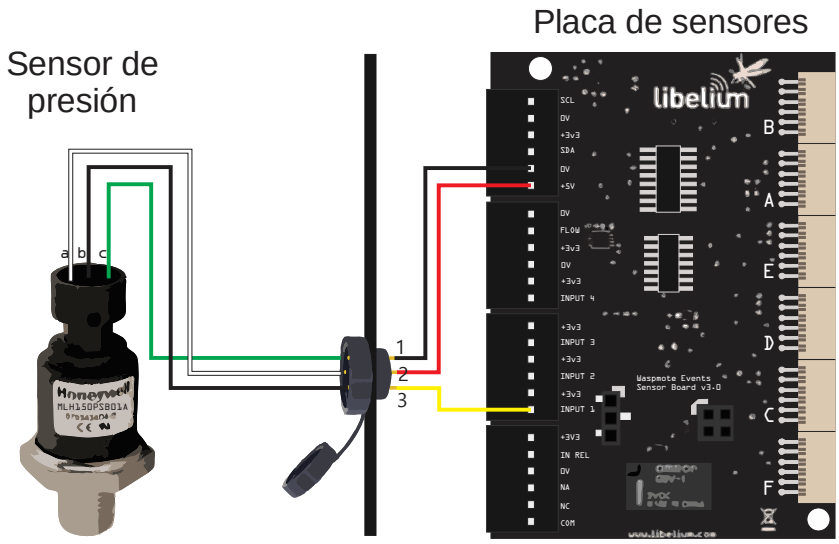


Figura 3.6.2. Conexión del sensor MLH150PSB01A con la placa de sensores *events-sensor-board_3.0.*
Fuente y elaborado por los autores.

3.6.1.2. *Diseño de algoritmo para adquisición de datos*

Una vez que el sensor se encuentra conectado al nodo, es necesario diseñar una función que traduzca los niveles de voltaje que entrega el sensor a la presión correspondiente que está midiendo. En primer lugar, se obtiene la curva característica del voltaje en función de la

presión. Luego se iguala con la ecuación del ADC de la tarjeta de adquisición y finalmente se diseña el algoritmo.

La salida del sensor de presión es radiométrica, es decir, el voltaje de salida es proporcional a la presión medida. En la Tabla 3.6.1 se muestra esta relación.

Tabla 3.6.1. Relación entre voltaje y presión

Voltaje de salida [v]	Presión [psi]
0.5	0
4.5	150

Fuente y elaborado por los autores, basados en [48].

Como resultado de esta relación lineal se obtiene

$$y = 37.5x - 18.75 \tag{3}$$

Donde y es el valor de presión y x es el nivel de voltaje a la salida del sensor.

El ADC de la tarjeta lee valores de 0 a 1 023 pin analógico. Estos valores corresponden al nivel de voltaje en el pin analógico. El voltaje se puede obtener de la ecuación:

$$x = \frac{a \cdot 3.3}{1\,023} \tag{4}$$

Donde a es el valor en el pin analógico y x es el voltaje proporcional. Con el uso de estas dos ecuaciones se puede obtener el valor de presión. Además, se incluye una alarma cuando el sistema sobrepasa el rango de medición del sensor. El algoritmo diseñado se muestra en la Figura 3.6.3.

```
/////////función presión instantánea//////////////
float presionf(int analog) {
  float presion = 0;
  float voltaje = 0;
  voltaje = float(analog) * 3.3 / 1023;
  presion = (37.5 * voltaje) - 18.75;
  if(presion<0){
    presion=0.0;
    }
  if (presion > 150) {
    PWR.setSensorPower(SENS_5V, SENS_OFF);
    USB.print(F("Presión Excesiva"));
    str="Presion Excesiva";
    alm=1;
  }
  return (presion);
}
```

Figura 3.6.3. Algoritmo para la medición del sensor de presión.
Fuente y elaborado por los autores.

3.6.2. Sensor de caudal WTP100-400

El sensor WTP100-400 es de uso industrial y entrega a su salida un cierto número de pulsos de corriente (*sinking current*) proporcionales al caudal que lo atraviesa. Además, requiere de una alimentación mínima de 6Vdc. Es necesario utilizar una fuente de alimentación externa, ya que el nodo cuenta solamente con fuentes de alimentación de 3.3Vdc y 5Vdc. Para la lectura de los pulsos de corriente también es necesario utilizar una resistencia *pull up* [53], [54], y un divisor de voltaje para que la señal de entrada a la placa de sensores sea detectada adecuadamente. El circuito diseñado con este fin se muestra en la Figura 3.6.4. El sensor cuenta con 5 metros de cable #22 AWG, y en su extremo se ha instalado un conector eléctrico de cuatro pines para conectarse al Molex J3 de la Figura 3.6.4.

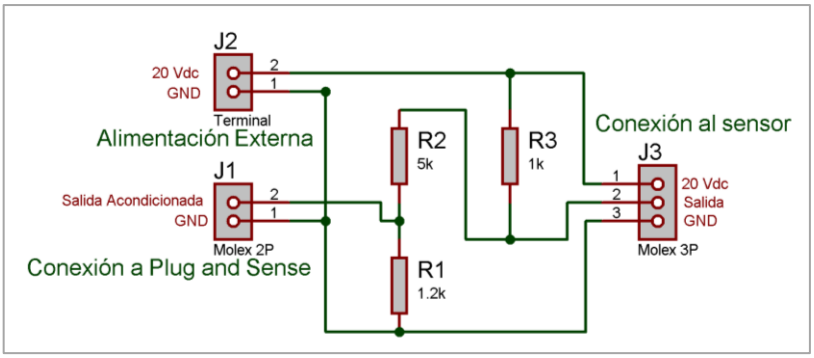


Figura 3.6.4. Circuito de acondicionamiento para el sensor WTP100-400.
Fuente y elaborado por los autores.

3.6.2.1. *Detección de pulsos*

Una vez construido el circuito, es necesario conocer la forma en la que los pulsos son generados para elaborar la función que los detectará y contabilizará. La turbina tiene un elemento ferromagnético en dos aspas opuestas. Cuando este elemento pasa por un punto del sensor la señal cambia de estado momentáneamente, generando un pulso. Si el sensor es atravesado por un flujo constante de agua los pulsos se generan proporcionales al volumen y a la velocidad del líquido. Es decir, la frecuencia de la señal indica cuantos pulsos existen para un determinado caudal. Es posible realizar mediciones de la señal de salida del circuito con el osciloscopio. En la Figura 3.6.5 se muestra la forma de la señal de salida, donde se distinguen tres pulsos que cambian de estado alto a bajo.

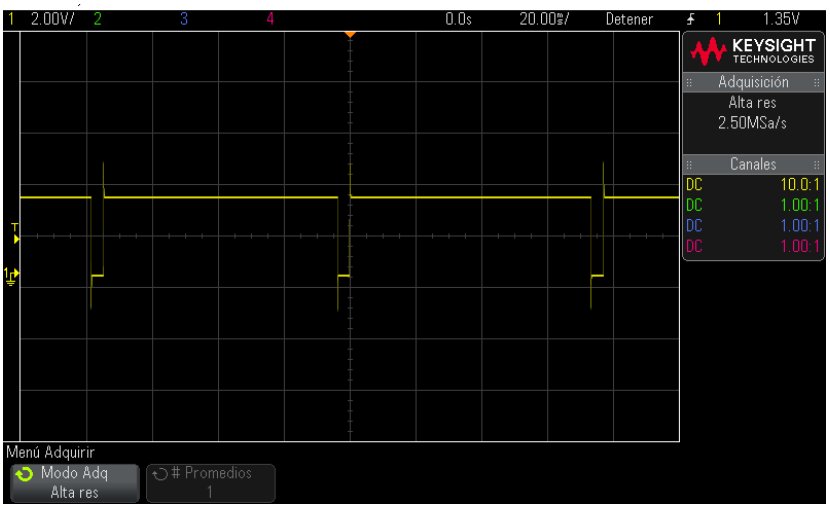


Figura 3.6.5. Forma de la señal de salida del sensor WTP100-400.
Fuente y elaborado por los autores.

La función para contar el número de pulsos se encuentra en la librería *WaspSensorEvent_v30.h.* La función aumenta un contador cada vez que detecta flancos de bajada. Para corroborar los datos que entrega la función *flowReading()* con los pulsos que genera el sensor, se realiza una prueba con el osciloscopio. Se conecta el sensor al nodo y para un flujo constante se cuentan los pulsos entregados en ambos equipos.

El inverso del período es la frecuencia, o el número de pulsos en un segundo. En la Figura 3.6.6 se observa la forma de un pulso La señal medida tiene una frecuencia de 9.23Hz (9 pulsos por segundo). Los resultados que entrega la función *flowReading()* son similares, como se aprecia en la Figura 3.6.7.

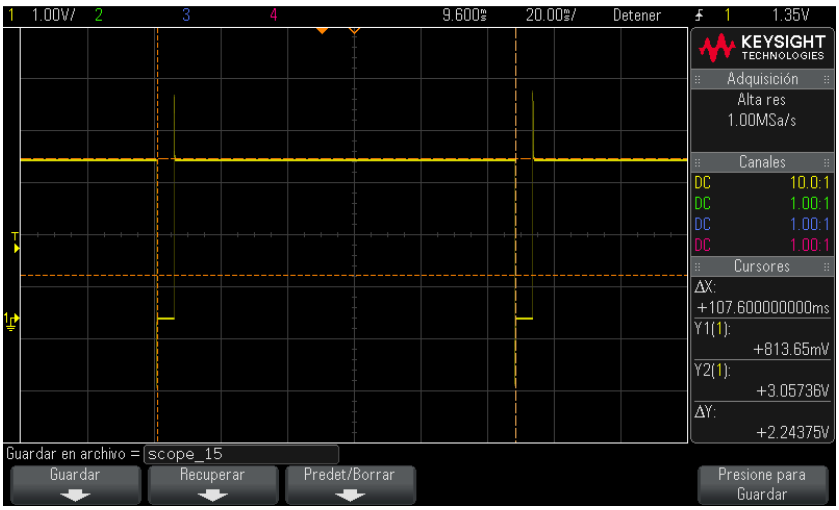


Figura 3.6.6. Medición del período de un pulso.
Fuente y elaborado por los autores.

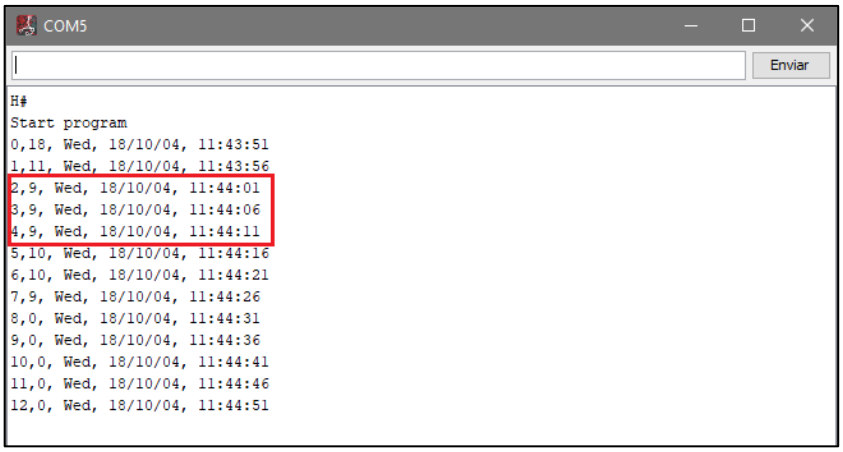


Figura 3.6.7. Número de pulsos detectados por la función *flowReading().*
Fuente y elaborado por los autores.

3.6.2.1. Cálculo del caudal

Cuando los pulsos son detectados por el nodo, es necesaria una ecuación que traduzca el número de pulsos al caudal correspondiente. Los sensores de Seametrics traen incluido un "factor K" en la etiqueta del dispositivo. El factor K representa el número de pulsos que el sensor genera por un galón americano de flujo [55]. En el caso del WTP100-400, $K = 6.0$ por lo tanto el caudal se obtiene mediante:

$$Q = K \cdot n \tag{5}$$

Donde Q es el caudal (en GPM), K es el factor multiplicativo y n es el número de pulsos entregados en un segundo.

Es necesario realizar una conversión para presentar el caudal en litros por minuto (l/min). Esta unidad es más utilizada en el medio local y es la unidad que se utiliza en el presente trabajo. Para realizar la conversión se utilizan las siguientes equivalencias:

6 pulsos → 1 gal

6 pulsos → 3.785 l

6 pulsos/s → 3.785 l/s

6 pulsos/s → 227.124 l/min

De estas equivalencias se deduce que si el sensor detecta seis pulsos en un segundo entonces existe un caudal de 227.1246 litros por minuto. Luego, utilizando la ecuación (5) se tiene:

$$227.1246\ {}^{l}/_{min} = K \cdot 6$$

$$K = 37.854$$

El nuevo factor K se introduce en el algoritmo del nodo, específicamente en la función *flowReading()* de la librería *WaspSensorEvent_v30.cpp*. La tarjeta Waspmote es compatible con otros sensores de flujo de funcionamiento similar. Editar la librería simplifica su uso futuro y requiere menos líneas de código en el algoritmo general del nodo.

3.6.3. Circuito para electroválvula B11MN-DN15

Esta electroválvula solenoide requiere de un voltaje de alimentación de 20 VDC. Se puede controlar desde el nodo utilizando un relé. La placa *events-sensor-board_3.0* cuenta con un relé de 1 A 24 VDC que se podría utilizar para activar una sola electroválvula. Para activar más de una electroválvula se requiere utilizar un módulo de relé. En este trabajo se utiliza un módulo de dos relés, el cual se ubica en la "caja de acondicionamiento". Sus características se muestran en la Tabla 3.6.2.

Tabla 3.6.2. Características Módulo de Relés

Descripción	Especificación
Voltaje de Operación	5 VDC
Señal de control	TTL (3.3 VDC o 5 VDC)
Canales	2
Modelo Relé	SRD-05VDC-SL-C
Voltaje máximo	10 A/250 VAC, 10 A/30 VDC
Corriente máxima	10 A (NO), 5 A (NC)
Tiempo de acción	5 ms

Fuente y elaborado por los autores, basados en [56].

El módulo de relés se alimenta con la fuente de 5 VDC del nodo. Las señales de activación se conectan a entradas digitales. La electroválvula se conecta en la entrada normalmente cerrada. El diagrama de conexión se muestra en la Figura 3.6.8.

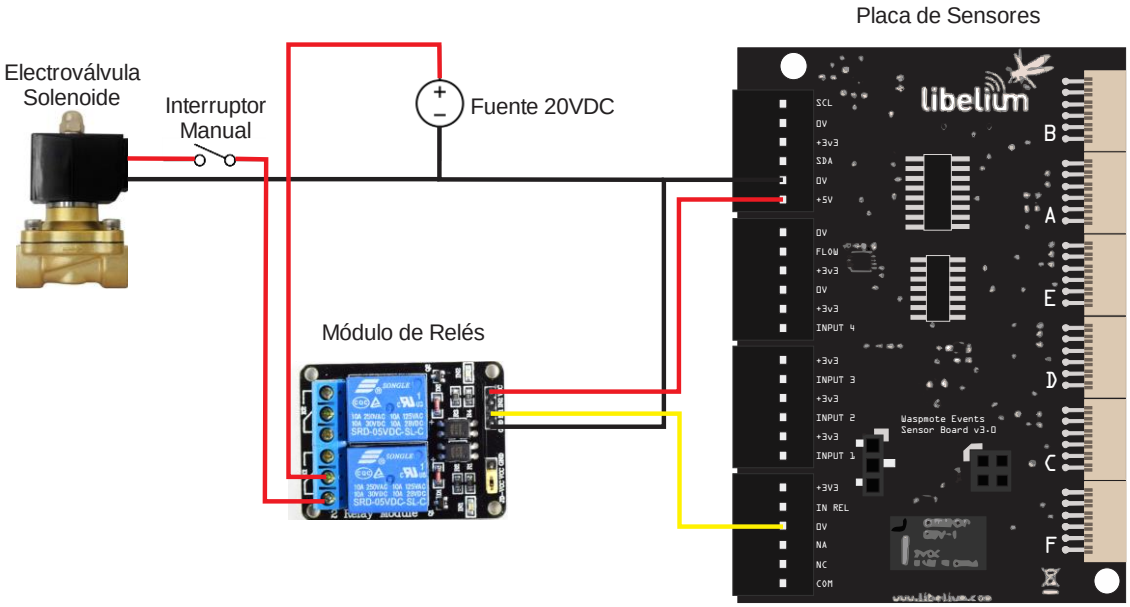


Figura 3.6.8. Diagrama de conexión de la electroválvula.
Fuente y elaborado por los autores.

3.6.4. Caja de acondicionamiento

La caja de acondicionamiento hace que las conexiones en ambientes industriales o en el campo sean más sencillas La caja de acondicionamiento contiene el circuito para el sensor WTP100-400 y el módulo de relés para controlar las electroválvulas, véase Figura 3.6.9.

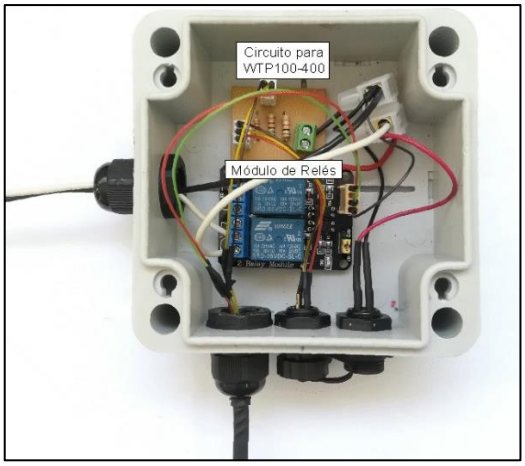


Figura 3.6.9. Caja de Acondicionamiento.
Fuente y elaborado por los autores.

Consiste en una caja plástica con protección IP65, con tres sockets para la conexión de: el sensor WTP100-400, la electroválvula B11MN-DN15, una interfaz para conectarse al nodo y una fuente de poder externa que se conecta a la red eléctrica de 110Vac y entrega a su salida 20Vdc regulados. La distribución de los sockets se muestra en la Figura 3.6.10. Los detalles para su conexión se detallan en la Tabla 3.6.3.

Figura 3.6.10. Distribución de sockets en la Caja de Acondicionamiento.
Fuente: Elaborado por los autores.

Tabla 3.6.3. Conexiones de la Caja de Acondicionamiento

Sockets	Conexión
A	Socket B del Plug & Sense Smart Security
B	Sensor WTP100-400
C	Fuente de alimentación
D	Salida para electroválvula

Fuente y elaborado por los autores.

CAPÍTULO IV

VALIDACIÓN Y RESULTADOS

4.1. Descripción del área de estudio.

El sistema fue desplegado en el campus de la UTPL. Para determinar los lugares donde se instalan los nodos primero se hizo un estudio de la red de agua potable del Campus Universitario, realizado por el Sr. Esteban Eras de la titulación de Ingeniería Civil. Los puntos de control identificados se muestran en la Figura 4.1.1.

Para la ubicación del *Gateway* se eligió la parte sur de la terraza del edificio 9. Los parámetros considerados para la elección del lugar fueron:

- Línea de vista con los puntos de control para evitar la pérdida de paquetes.
- Fácil acceso para instalar y realizar tareas de mantenimiento al equipo.
- Altura adecuada para tener cobertura total en la zona a monitorear.

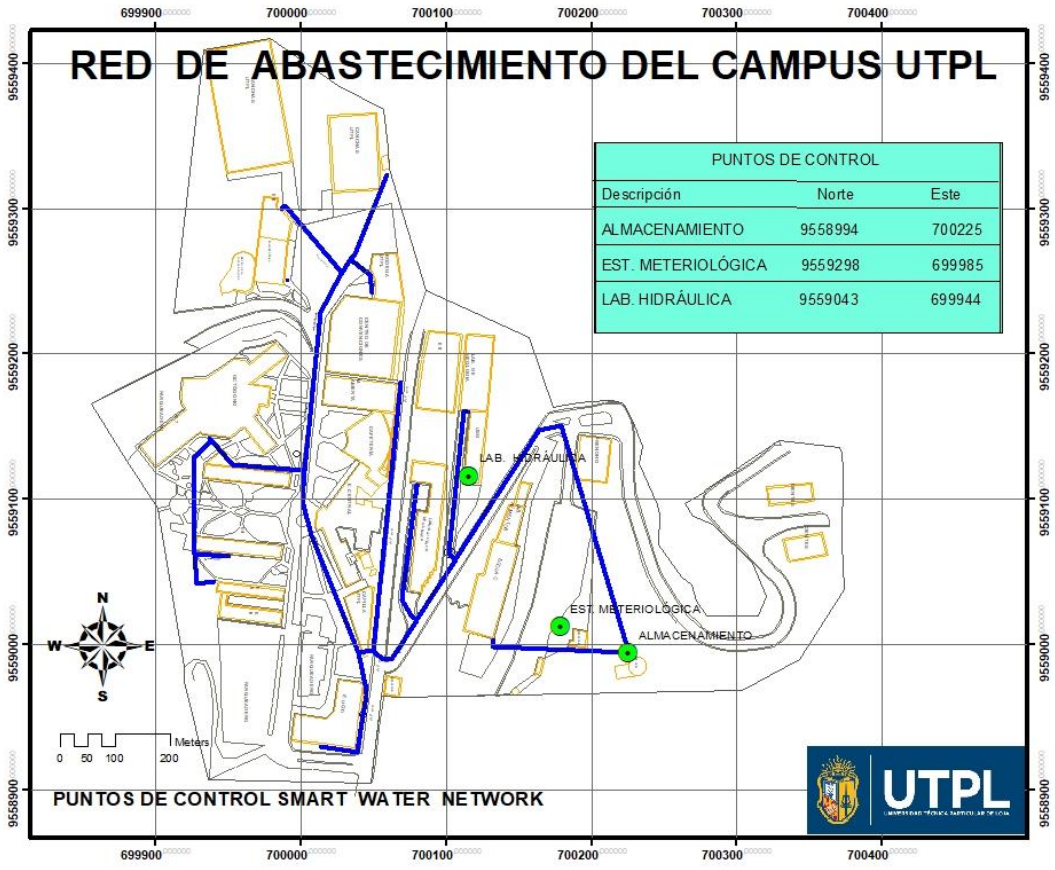


Figura 4.1.1. Ubicación de los puntos de control.
Fuente y elaborado por Sr. Esteban Eras.

4.2. Validación de sensores

El proceso de validación se realizó en los sensores de presión y caudal, los mismos que previamente se acondicionaron para ser integrados en el nodo. Para comprobar que los resultados sean correctos, se compararon los valores obtenidos de los sensores contra

equipos de medición del laboratorio de Hidráulica de la UTPL. Los sensores manufacturados por Libelium también fueron validados, especialmente para comprobar que no tengan daños.

4.2.1. Ambientes para la validación

Se determinaron dos ambientes para validar el sistema. El primero fue el Prototipo de red de distribución (PRDA), ubicado en el laboratorio de Hidráulica del Departamento de Geología, Minas e Ingeniería Civil de la UTPL. El segundo fue el tanque de reserva V-300M3-EPOCA BAJO de la ciudad de Loja, que forma parte del sistema de distribución de agua del Municipio de Loja.

4.2.1.1. *Prototipo de Red de Distribución de Agua (PRDA)*

El PRDA ha sido diseñado e implementado en el Laboratorio de Hidráulica por Esteban Eras y Carlos Vivanco, estudiantes de la titulación de Ingeniería Civil. Este banco hidráulico cuenta con: un sistema de bombeo en serie y paralelo, un sifón para la instalación de los caudalímetros y derivaciones para los sensores de presión, como se ve en la Figura 4.2.1. El prototipo permitió simular presiones y caudales similares a los de un sistema de distribución de agua.

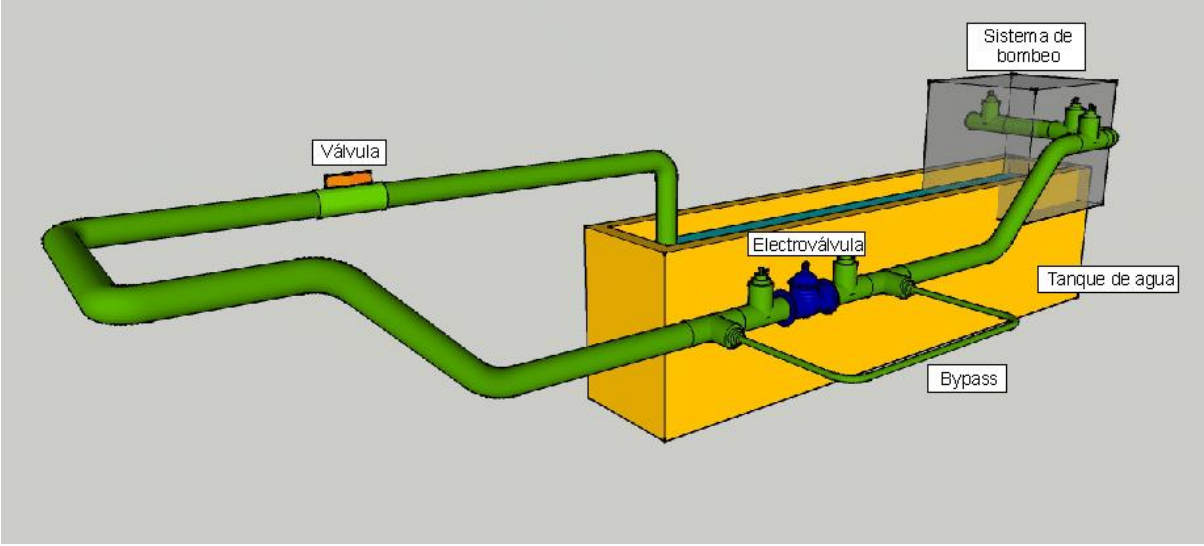


Figura 4.2.1. Prototipo de Red de Distribución de Agua (PRDA).
Fuente y elaborado por los autores.

4.2.1.2. *Tanque de reserva V-300M3-EPOCA BAJO*

El tanque de almacenamiento está ubicado en la ciudadela Época. Es un tanque con una capacidad de 300 m^3 y tiene un caudal promedio a su entrada de 483.35 l/min. Se eligió este espacio de pruebas ya que el PRDA no fue capaz de generar los caudales necesarios para la validación del sensor WTP100-400.

4.2.2. Validación de sensor de presión

Para la medición de la presión en las tuberías, en una sección del prototipo de red de distribución (PRDA) se colocó una serie de accesorios, en forma de candelabro, en el que se conectaron: una válvula de aire, un manómetro, el instrumento de referencia y el sensor de presión MLH150PSB01A.

El instrumento de referencia que se utilizó fue el *datalogger* OM-PL. El cual es un dispositivo que muestrea la presión en tuberías. Se puede configurar para registrar datos con un tiempo mínimo de un segundo hasta un dato cada 18 horas. Además, permite configurar alarmas de valores máximos y mínimos con los umbrales que se requieran. Se conecta mediante un software específico al computador para permitir la visualización de los datos de presión. El sensor con el que viene equipado tiene un rango de 500 psi, una exactitud de ±0.60% FS y una resolución de 0.15 psi. Trabaja hasta temperaturas de 54ºC. Para más especificaciones se puede revisar el manual del fabricante [57].

A continuación se describe el proceso de validación: Se conectó el sensor MLH150PSB01A al nodo sensor. El sensor MLH150PSB01A y el *datalogger* Omega se instalaron en el candelabro como se muestra en la Figura 4.2.2. Una válvula de aire aseguró que todo el aire sea purgado de la tubería. También se instaló un manómetro para la comprobación visual de los valores de presión.

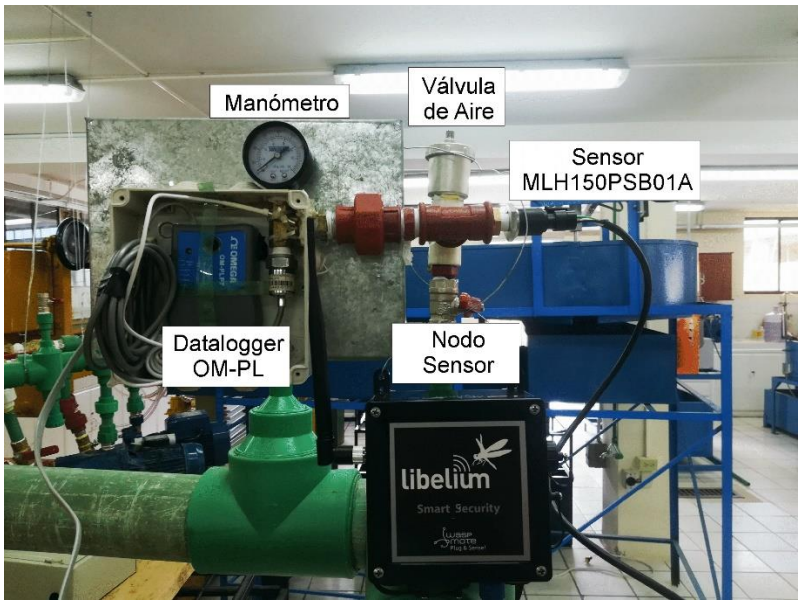


Figura 4.2.2. Conexión de los sensores de presión en el PRD.
Fuente y elaborado por los autores.

Se realizaron mediciones de presión cada tres segundos desde las 10:46:06 hasta las 11:08:45 (22 minutos y 39 segundos). Al final del proceso se obtuvieron 454 mediciones por cada sensor. Se hizo variar la presión mediante el uso de válvulas y desfogues, de 0 psi a

aproximadamente 80 psi. De esta manera los valores medidos se encuentran en el tercio medio del rango de medición. A continuación, en la Figura 4.2.3 se muestra una gráfica a lo largo del tiempo de los valores medidos por estos dos sensores.

Como se aprecia en la Figura 4.2.3 los valores son en mayor parte similares. El sensor Honeywell MLH150PSB01A es más sensible a los pequeños cambios debido a su menor tiempo de respuesta. Los picos en 10:59:21, 11:03:00 y 11:03:18 se debieron a cambios bruscos de la presión de muy poco tiempo (menores a 1 segundo). La sincronización no es completamente exacta entre los dos instrumentos, por ello no fueron registrados por los dos sensores simultáneamente.

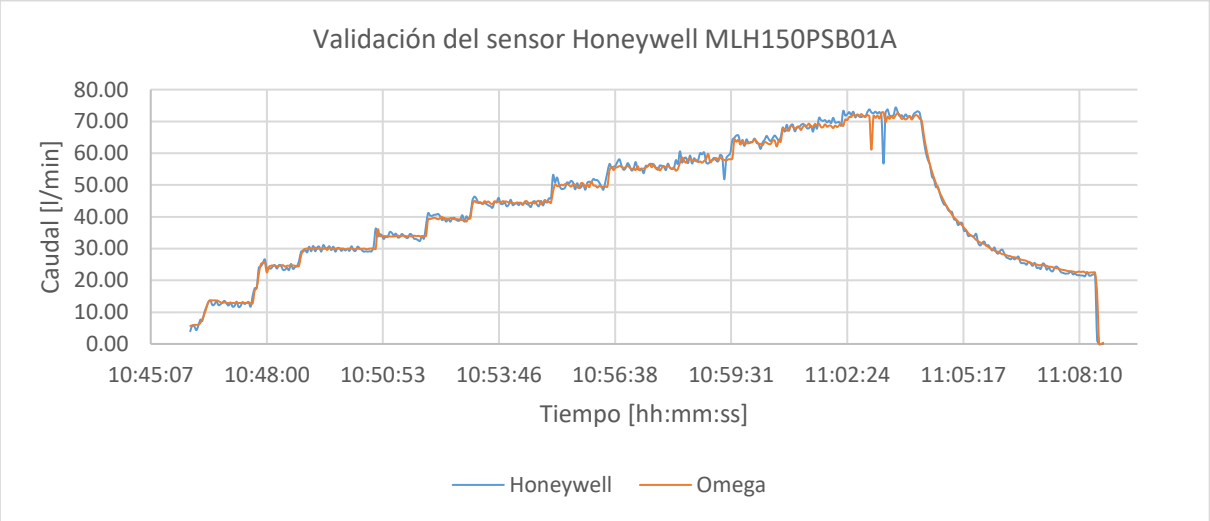


Figura 4.2.3. Gráfica comparativa entre sensor Honeywell y Omega durante el proceso de validación. Fuente y elaborado por los autores.

Se calculó el error relativo del sensor con respecto al instrumento de referencia. El error relativo es representado en la Figura 4.2.4 mediante un diagrama de caja. El desplazamiento de la gráfica de caja hacia la parte inferior indica que el error relativo es mínimo. Además, se observa que el 75% de los datos tienen un error menor al 3.0% y el error medio se ubica en 1.9%. También existen valores atípicos, los cuales están lejos de la tendencia general de los datos y corresponden a los cambios bruscos de la presión.

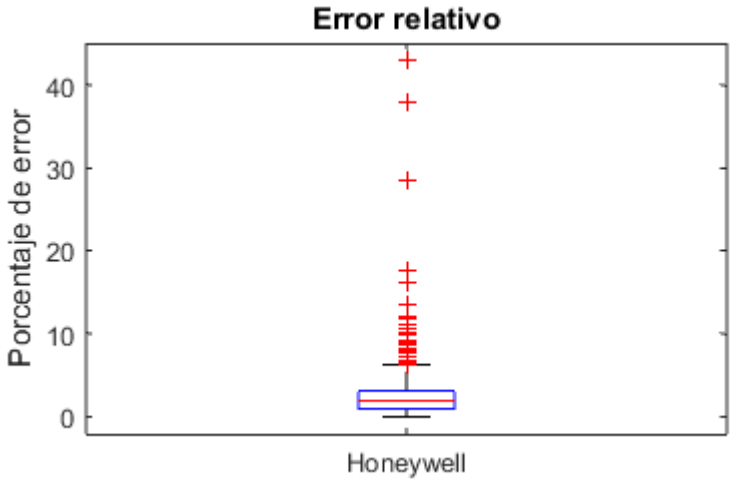


Figura 4.2.4. Diagrama de caja de la validación del sensor Honeywell.
Fuente y elaborado por los autores.

El sensor MLH150PSB01A es un dispositivo que tiene una precisión y exactitud aceptable en la medición. Además, su diseño para ambientes industriales lo convierte en un sensor robusto y de fácil instalación. Su integración al nodo Smart Security fue posible y entregó resultados confiables como se ha mostrado.

4.2.3. Validación de sensor WTP100-400

El sensor WTP100-400 se instaló en el tanque de reserva V-300M3-EPOCA BAJO. El sensor fue ubicado en la cámara de válvulas, en la tubería de entrada al tanque. Es necesario que existan 1.10 m y 0.5 m de tubería libre de accesorios antes y después del punto de instalación, respectivamente. Se adaptaron dos bridas a los extremos del caudalímetro para facilitar su instalación. En la Figura 4.2.5 se observa el sensor instalado entre la válvula de acceso y la válvula de llenado del tanque de reserva con todas las adecuaciones realizadas.

Figura 4.2.5. Sensor WTP100-400 instalado.
Fuente y elaborado por los autores.

Para validar las mediciones de este sensor se emplearon dos instrumentos de referencia. El primero fue el flujómetro Polysonics DCT7088 de la marca Thermo. El segundo fue un prototipo para la medición del nivel de agua desarrollado en la UTPL.

El flujómetro Polysonics DCT7088 utiliza sondas no invasivas que se abrazan al tubo. Tiene una gran flexibilidad ya que se puede configurar para diferentes diámetros y materiales de tubería. Cuenta con un *datalogger* para almacenar los datos de caudal. Permite la medición de caudales que tengan una velocidad de hasta 12 m/s en tuberías de 25 mm a 5 m con una exactitud de ±0.5% de la velocidad.

El prototipo para la medición del nivel de agua en ríos es un dispositivo que entrega la distancia entre el punto en el que se encuentra instalado hasta el agua. Su instalación en el tanque de reserva permitió conocer su nivel en cualquier instante de tiempo. Este dispositivo utiliza el sensor HRXL-MaxSonar-WRS MB7364. Es un sensor robusto para la medición de profundidad, con resolución de 1 mm y un rango máximo de detección de 5 m.

La validación del sensor se realizó en dos fases. En la primera se conectó el WTP100-400 y el Polysonics DCT7088 en la tubería de entrada al tanque de reserva y se tomaron datos por una hora, cerrando progresivamente la válvula de acceso para reducir el caudal. La conexión y ubicación del sensor y los instrumentos de referencia se muestra en la Figura 4.2.6.

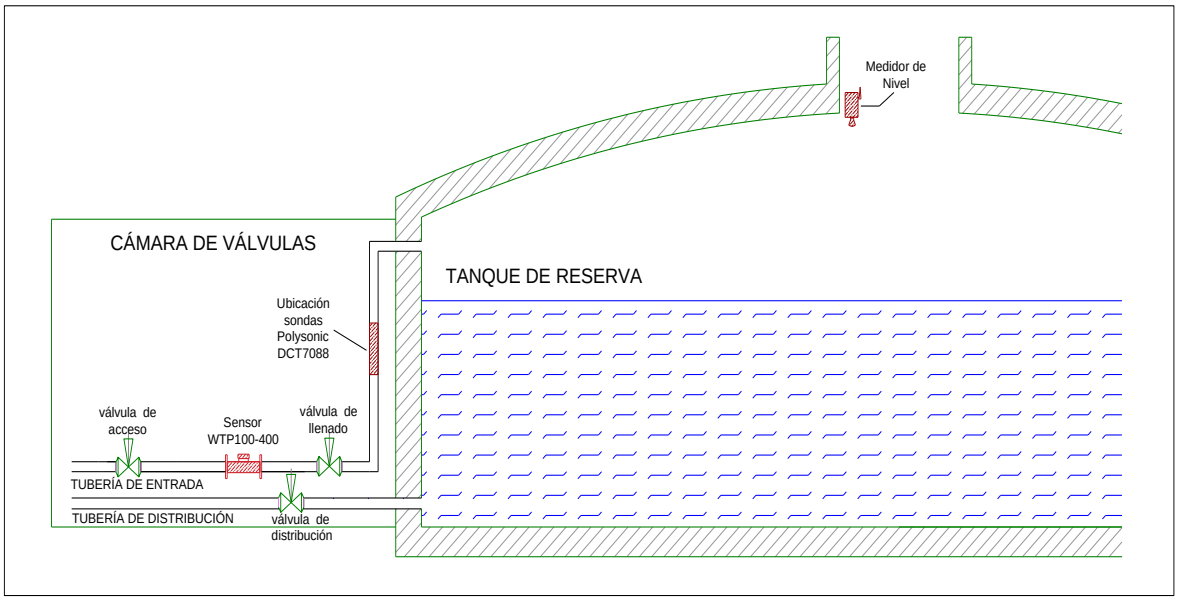


Figura 4.2.6. Instalación de los dispositivos para el proceso de validación.
Fuente y elaborado por los autores.

Se recopilaron datos cada cinco segundos desde las 11:52:15 hasta las 12:45:20 (53 minutos con 5 segundos). Se obtuvieron 638 mediciones de caudal por cada equipo en un rango de caudal entre 567.8 l/min y 75.7 l/min. Los resultados de la primera fase de validación se muestran en la Figura 4.2.7.

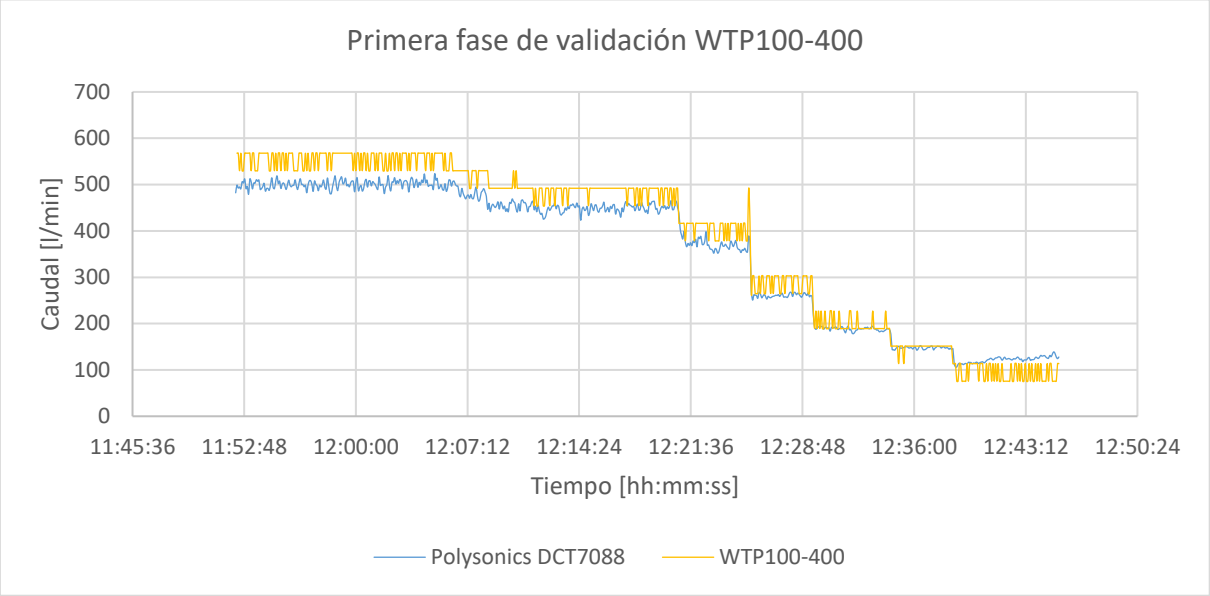


Figura 4.2.7. Resultados de la primera fase de validación.
Fuente y elaborado por los autores.

La segunda fase de validación consistió en medir el nivel del tanque en tiempos determinados para calcular el caudal de ingreso al tanque de reserva, es decir se realizó un aforo. El medidor de nivel se instaló en la compuerta de acceso al tanque de reserva, como se mostró en la Figura 4.2.6. El proceso de medición con este instrumento se describe a continuación:

1. Previo al proceso de validación, se cerró la válvula de acceso al tanque y se abrió la válvula de distribución durante aproximadamente dos horas para reducir el nivel de agua en el tanque.
2. Se cerró la válvula de distribución. El cierre de esta válvula implica suspender el servicio de agua a los usuarios, por lo que solamente se realizaron pruebas por el lapso de una hora.
3. Se abrió la válvula de acceso completamente para permitir la entrada de un flujo constante de agua al tanque de reserva.
4. Se esperaron 10 minutos para que se estabilizaran las turbulencias en la superficie del agua y se midió el nivel inicial del tanque. Simultáneamente se empezaron a recopilar mediciones con el sensor WTP100-400 cada cinco segundos.
5. Se realizaron las mediciones de nivel cada cinco minutos.
6. Tras 30 minutos se cierra parcialmente la válvula de acceso para medir otro valor de caudal. Las mediciones con los instrumentos se realizan de manera similar.
7. Una vez finalizada la prueba se abrieron las válvulas de acceso y distribución para que la RDA continúe con su funcionamiento normal.

La altura registrada por el medidor de nivel sirve para calcular el volumen de agua que ingresa al tanque de reserva. El caudal se obtiene dividiendo el volumen para el tiempo. Los resultados de la segunda fase de validación se muestran en la Figura 4.2.8.

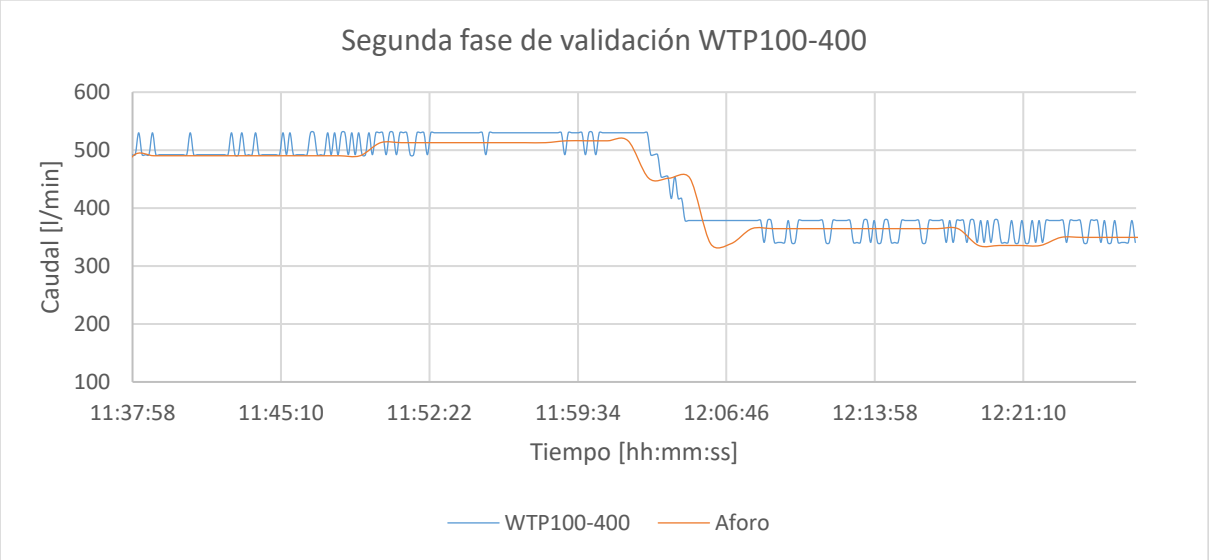


Figura 4.2.8. Resultados de la segunda fase de validación.
Fuente y elaborado por los autores.

Para este sensor también se calculó el error relativo con respecto al aforo realizado. El error relativo de la segunda fase de validación se muestra en la Figura 4.2.9. Se puede observar una asimetría al no estar la mediana en el centro de la caja, esto se debe a que los valores superiores a la media se encuentran dispersos y los inferiores están estrechamente agrupados. El error medio del conjunto de datos se ubica en 3.8% y existen valores atípicos que corresponden a la disminución del caudal entre las 12:02:00 y 12:05:00 de la Figura 4.2.8**.** Cuando la transición termina, los valores se estabilizan.

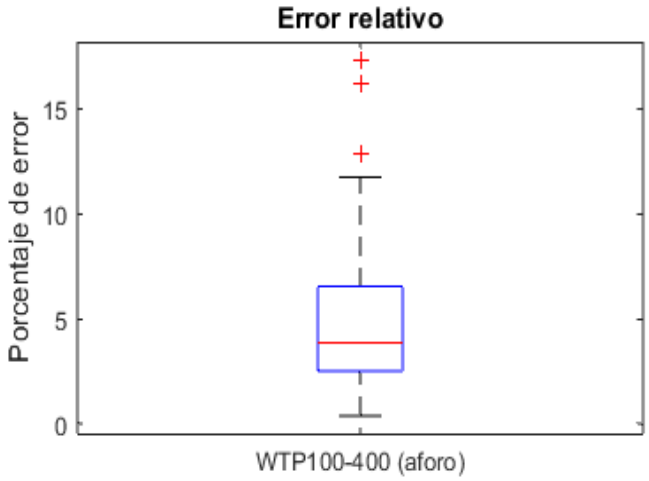


Figura 4.2.9. Diagrama de caja de la validación del sensor WTP100-400.
Fuente y elaborado por los autores.

4.2.4. Validación de sensores de Libelium

Los sensores fabricados por Libelium se conectaron directamente al nodo. Además, como ya cuentan con funciones desarrolladas, su programación fue sencilla. Estos sensores fueron instalados siguiendo las especificaciones de la Tabla 3.4.1.

4.2.4.1. *Sensor YF-G1*

La validación para los sensores de fuga se realizó mediante aforo. Fueron instalados en el PRDA en una sección de *bypass.* Dejando más de 10 y 5 diámetros caudal antes y después del sensor respectivamente. Los aforos se realizaron a un caudal bombeado constante.

Durante el proceso de validación se detectó una incongruencia de datos entre la fórmula entregada por el fabricante y el valor aforado. Por ello se instaló el Polysonics DCT7088 para comparar las mediciones. Se realizaron mediciones cada 5 segundos desde las 16:45:50 hasta las 17:35:55 (50 minutos y 5 segundos), obteniendo 602 valores de caudal por cada sensor.

El error en la medición es evidente como se observa en la Figura 4.2.10 y Figura 4.2.11. Por lo tanto, se propone el uso de una ecuación lineal de ajuste, obtenida tras este proceso de validación.

$$y = 0.9163x - 0.6656 \tag{6}$$

Donde y es caudal en l/min, x es el número de pulsos que el sensor detecta cuando lo atraviesa cierto flujo de agua. En la Figura 4.2.10 también se aprecian los valores que serían adquiridos con el uso de la ecuación (6).

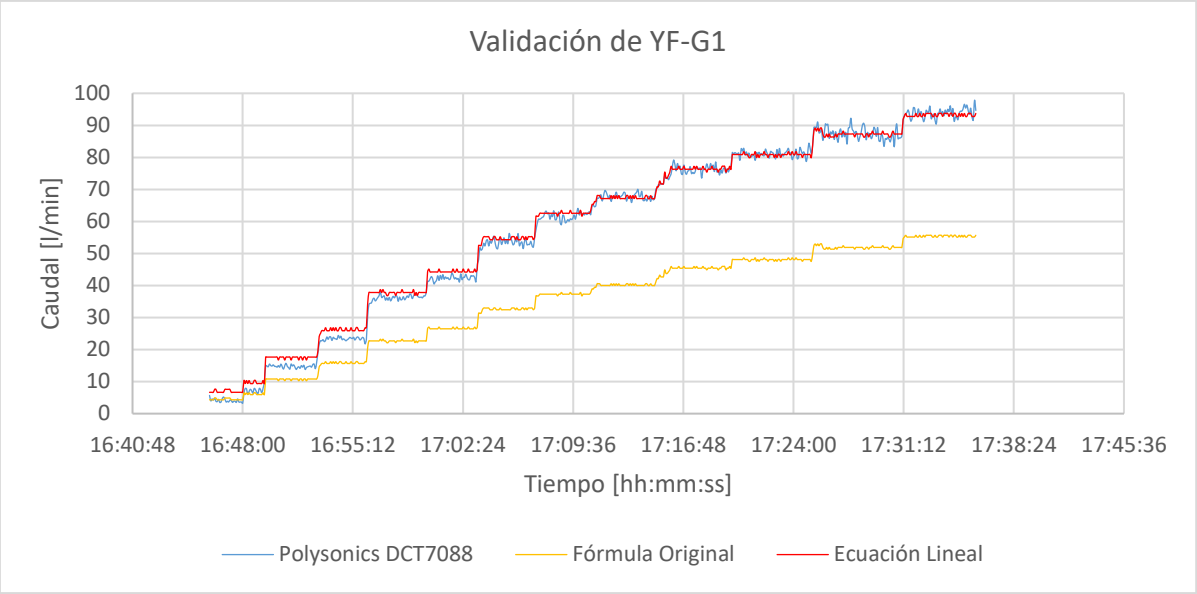


Figura 4.2.10. Resultados de la validación del sensor YF-G1.
Fuente y elaborado por los autores.

El error relativo que presentó el sensor se muestra en el diagrama de caja de la Figura 4.2.11. La gráfica de caja de la izquierda esta desplazada hacia la parte superior, indicando un alto porcentaje de error. A la derecha se muestra el error de la ecuación de ajuste, donde la tendencia del conjunto de datos esta próxima a 0%. Para esta gráfica, la mediana disminuye 2.3% frente al 39.5% de error que presentaba la fórmula original. Los datos atípicos corresponden principalmente a caudales bajos, donde las mediciones del instrumento de referencia y el sensor se encuentran alejadas. Sin embargo, conforme aumenta el caudal el error va disminuyendo y las gráficas son similares como se mostró en la Figura 4.2.10.

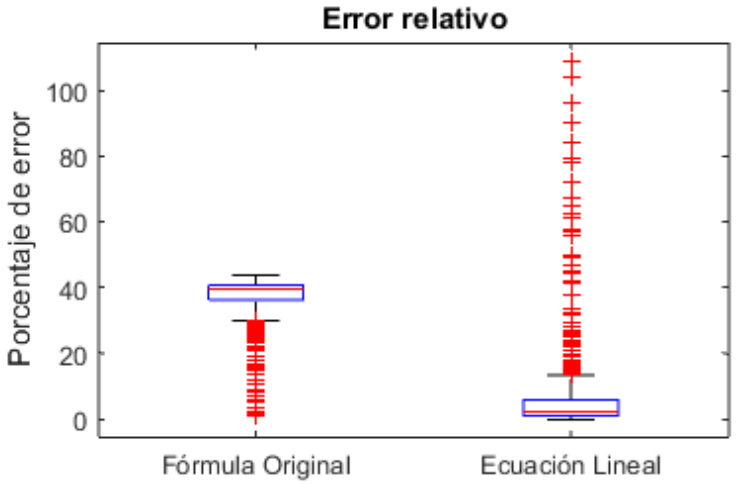


Figura 4.2.11. Diagrama de caja de la validación del sensor YF-G1.
Fuente y elaborado por los autores.

4.2.4.2. *Sensor YF-S201*

El proceso de validación de este sensor fue similar al descrito en el punto anterior. Se instaló en el PRDA con tubería de ½". Se realizaron mediciones cada 5 segundos desde las 10:17:58 hasta las 10:33:48 (15 minutos y 50 segundos), obteniendo 191 puntos de medición. Las mediciones de este sensor tuvieron una exactitud aceptable como se aprecia en la Figura 4.2.12.

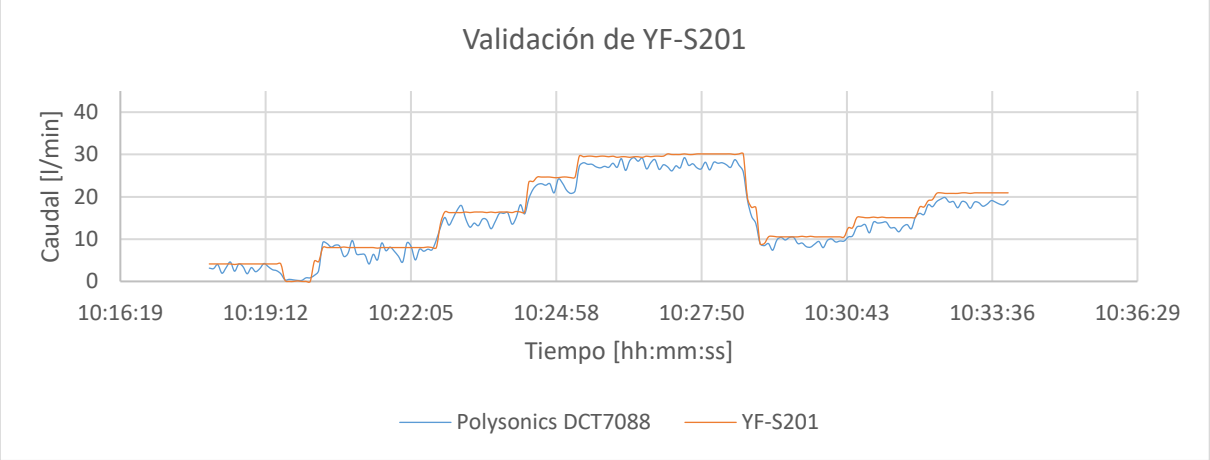


Figura 4.2.12. Resultados de la validación del sensor YF-S201.
Fuente y elaborado por los autores.

Luego se obtuvo el error relativo de este instrumento respecto al Polysonics DCT7088. Se obtuvo un error medio de 11.6%. Frente a este error se desarrolló la ecuación de ajuste que se presenta a continuación.

$$y = 0.9344x - 0.496 \quad (7)$$

Donde y es el valor de caudal corregido y x es el valor obtenido originalmente. El error relativo de la medición se muestra en la Figura 4.2.13 en forma de diagramas de caja. La gráfica de la izquierda presenta un desplazamiento hacia la parte superior y un error medio de 11.6%. Mediante la ecuación de ajuste este error es disminuido a 6.8%, como se muestra a la derecha de la Figura. Sin embargo, este valor es mayor que el obtenido en los sensores validados anteriormente, esto debido a que el caudalímetro presenta una baja exactitud (véase Tabla 3.5.4). Los valores atípicos se deben a que no se logró estabilizar las mediciones del instrumento de referencia y los valores varían presentando oscilaciones, como se mostró en la Figura 4.2.12.

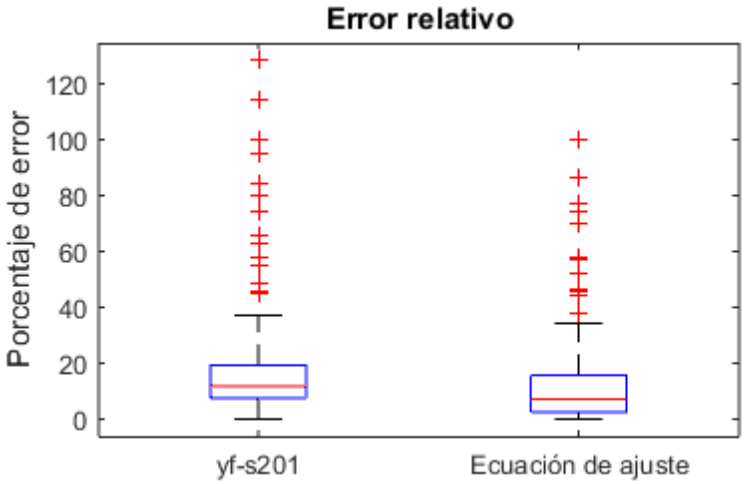


Figura 4.2.13. Diagrama de caja de la validación del sensor YF-S201.
Fuente y elaborado por los autores.

4.2.4.3. *Sensor 9295-P (línea)*

Para la validación de sensor de fugas en línea se puso al sensor en contacto con el agua. Durante el proceso de validación fue detectada una falla de hardware, el cableado del sensor no es correcto. Debido a la versión de la placa de sensores fue necesario conectar los dos cables del sensor a los cables amarillo y café del conector. Luego de este cambio el proceso de validación fue satisfactorio. Los resultados se observan en la Figura 4.2.14.

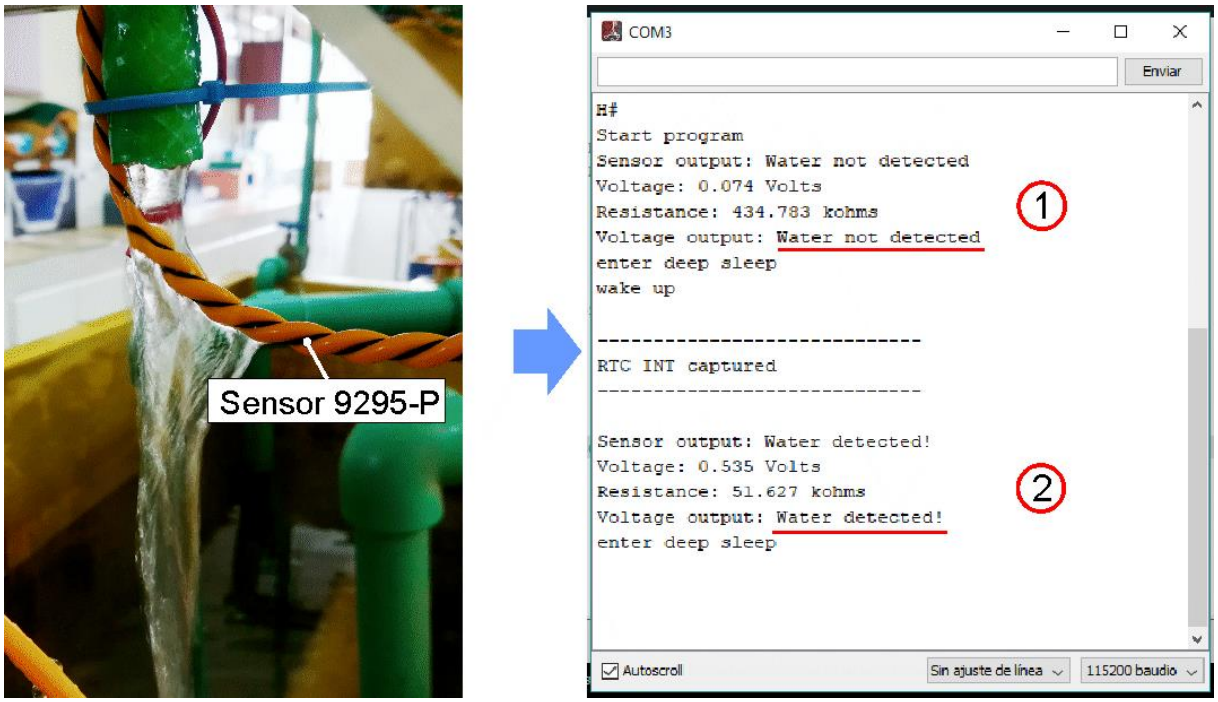


Figura 4.2.14. Validación de sensor 9295-P.
Fuente: Elaborado por los autores.

4.2.4.4. *Sensor 9243-P*

Para la validación de sensor de fugas se colocó al sensor en contacto con agua. El nodo sensor muestra una alarma cuando el sensor detecta agua. El proceso de validación fue satisfactorio como se puede observar en la Figura 4.2.15.

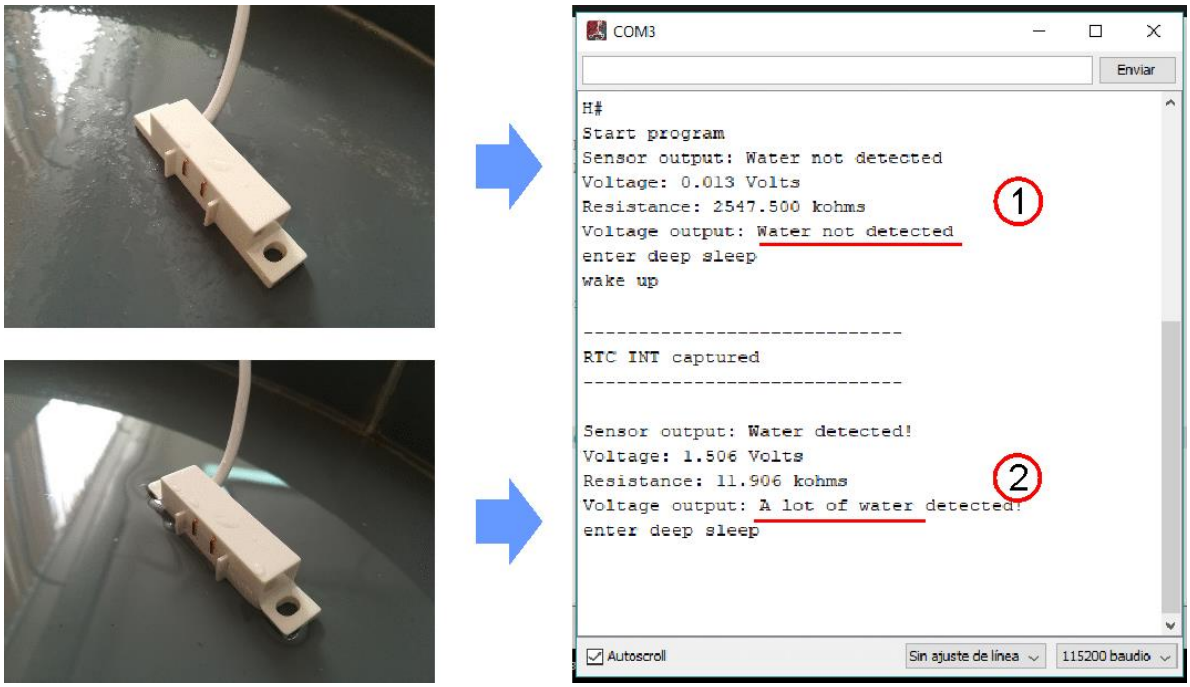


Figura 4.2.15. Validación de sensor 9243-P.
Fuente y elaborado por los autores.

4.2.4.5. *Sensor de nivel PTFA3415*

El sensor de nivel se instaló en el tanque del PRDA. Se leyó la salida del sensor con el nivel de agua bajo el sensor (Figura 4.2.16 (1)) y cuando el tanque alcanzó el nivel del sensor (Figura 4.2.16 (2)). El proceso de validación fue positivo.

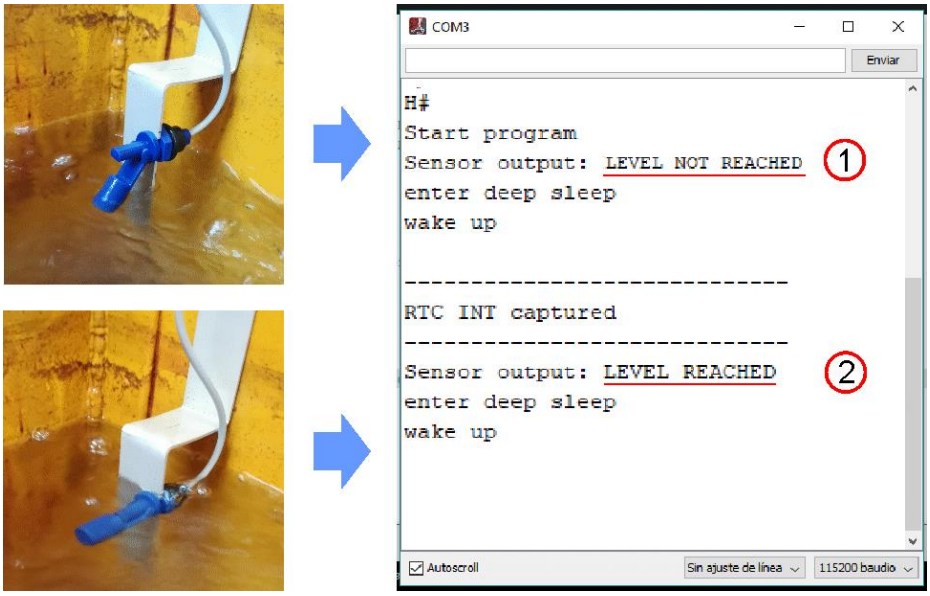


Figura 4.2.16. Validación sensor de nivel PTFA3415.
Fuente y elaborado por los autores.

4.3. Despliegue de la WSN

Una vez validados los sensores, se despliegan varios nodos para recolectar datos y comprobar su envío a la plataforma. En primer lugar, se instala el *gateway* para brindar cobertura al área de estudio. Luego, se despliegan cuatro nodos para recolectar las variables de interés. Finalmente se prueba el envío correcto de los datos recolectados a la plataforma IoT ThingSpeak.

4.3.1. Instalación del *Gateway*

Antes de realizar la instalación se realizaron algunas adecuaciones que son mencionadas a continuación. Para la alimentación del dispositivo fue necesario el uso de un cargador DC con salida de 12V y conectado a la red eléctrica de 120V. También se requirió de un punto de red para acceder a Internet. Todas las adecuaciones necesarias en el campus fueron realizadas por el departamento de Infraestructura

Se colocó el *Gateway* sobre una estructura metálica sujeto al mástil por medio de una abrazadera. En la parte superior se ubicó la antena omnidireccional de 8 dBi. Los cables de alimentación y de red fueron empotrados en un tubo para protegerlos de la intemperie y se conectaron en la parte inferior. El *Gateway* instalado se observa en la Figura 4.3.1.

Figura 4.3.1. *Gateway* instalado, vista frontal.
Fuente y elaborado por los autores.

4.3.2. Nodo del prototipo de red de distribución de agua

El nodo para el monitoreo del prototipo de red de distribución de agua se instaló en el laboratorio de Hidráulica del campus UTPL para recolectar datos durante cuatro días, véase Figura 4.3.2. Este nodo realizó las mediciones de caudal, de presión, de detección de fugas y de nivel del tanque.

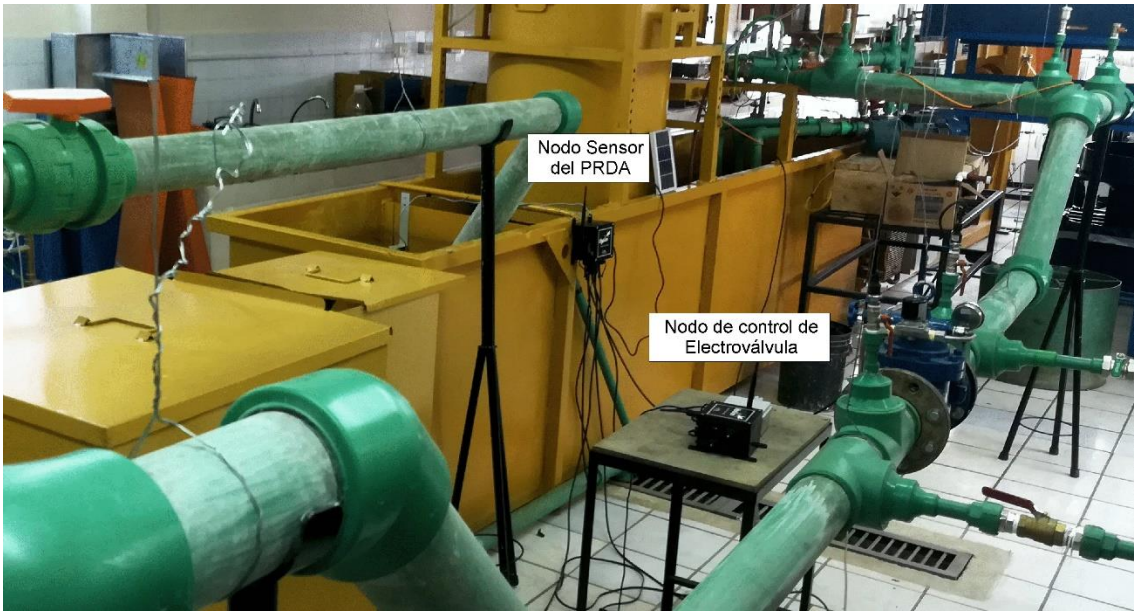


Figura 4.3.2. Nodos instalados en el PRDA.
Fuente y elaborado por los autores.

Los sensores se instalaron en diferentes puntos del PRDA, para recolectar los datos requeridos. Los sensores instalados se muestran en la Figura 4.3.3. Un sensor de presión se instaló antes de la electroválvula y otro fue instalado a su salida. El sensor de nivel se ubicó en el tanque de reserva. El sensor de caudal fue instalado en el *bypass*. Finalmente, el sensor para detectar fugas se instaló a lo largo de la tubería como se ve en la Figura 4.3.3(e). En la Figura 4.3.3(d) se simuló una fuga en una derivación del PRDA. La conexión de un panel solar no fue posible debido a la ubicación física del dispositivo, sin embargo, el nodo tuvo la autonomía energética suficiente para el tiempo que duraron las pruebas.

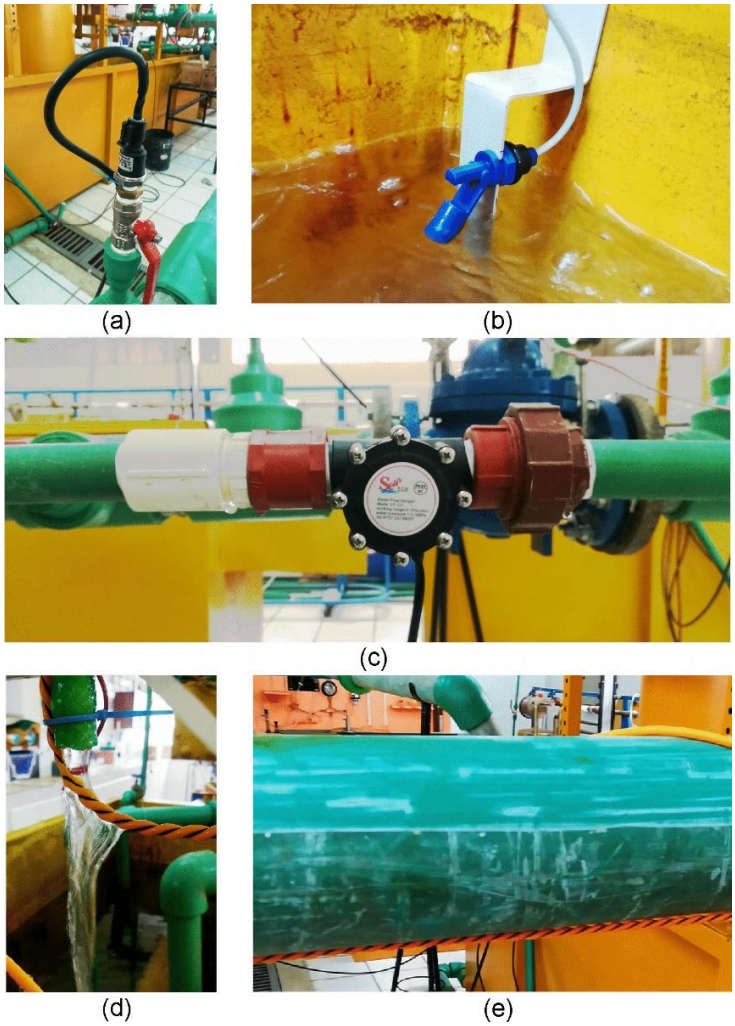


Figura 4.3.3. Sensores instalados. (a) Sensor de presión. (b) Sensor de nivel. (c) Caudalímetro. (d) y (e) Sensor de fugas en línea para tuberías.
Fuente y elaborado por los autores.

La conexión de los sensores al nodo siguió las especificaciones de la Tabla 4.3.1. Además, fue el orden en el que se enviaron los datos y como se presentaron en la plataforma IoT.

Tabla 4.3.1. Especificaciones de conexión del nodo

Socket	Sensor	Modelo	Campo ThingSpeak
A	Fugas en línea	9295-P	status
B	Caudal	YF-G1	field8 (instantáneo)
C	Presión	MLH150PSB01A	field3 (instantáneo)
D	Presión	MLH150PSB01A	field2(instantáneo), field4 (mínimo), field5 (máximo), field6 (promedio)
E	Nivel	PTFA3415	field7
-	Batería	-	field1

Fuente y elaborado por los autores.

Para este nodo, el tiempo de envío fue establecido en cinco minutos en el algoritmo de ejecución (ver ANEXO D). Durante este tiempo se realizaron mediciones constantes cada segundo de los valores de presión que entregaba el sensor del socket D. De estas mediciones

se obtuvieron los valores mínimos, máximos y el promedio en este intervalo. Cuando se cumplía el tiempo de envío, todas las variables se enviaban al gateway. Para el sensor de fugas, la información se enviaba en el campo de estado usando las siguientes siglas: SF (Sin Fugas), CF (Con Fugas) y MF (Muchas Fugas). El sensor de nivel envió un uno (1) cuando el nivel era alcanzado y un cero (0) en el caso contrario. El caudal se envió en l/min y los valores de presión en psi. En la Figura 4.3.4 se muestran los resultados de dos días de pruebas de este nodo.

Figura 4.3.4. Resultados de pruebas del nodo del PRDA.
Fuente y elaborado por [31].

4.3.3. Nodo de control de electroválvula

Junto al nodo del PRDA se instaló otro nodo como se mostró en la Figura 4.3.2. Tiene la función de controlar el flujo de agua activando una electroválvula La conexión se realizó como se muestra en la Figura 4.3.5. El nodo (3) se conectó con la caja de acondicionamiento (2). La caja de acondicionamiento tiene conectada la fuente de 20VDC para alimentación y la electroválvula. La electroválvula puede ser controlada manualmente, con el uso de un interruptor (1). También se puede accionar remotamente desde la plataforma de ThingSpeak en Internet.

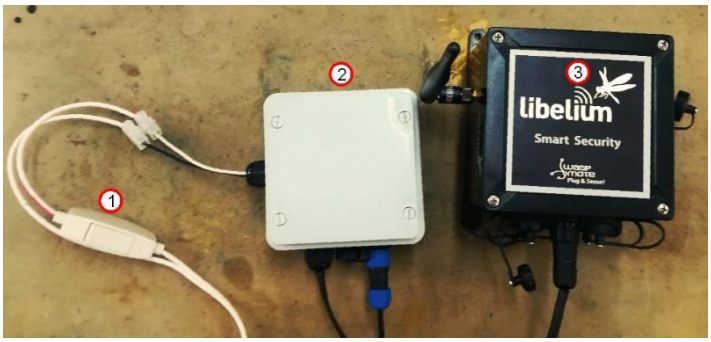


Figura 4.3.5. Conexión del nodo de control de Electroválvula.
Fuente y elaborado por los autores.

El algoritmo que este nodo ejecutó fue ligeramente diferente, ya que era necesario que esté en modo de recepción esperando un mensaje del *gateway*. Para que este proceso se ejecute, el *gateway* accedió periódicamente al canal de ThingSpeak y cuando el botón de ABRIR o CERRAR válvula era accionado en la plataforma, se envió una trama con el comando respectivo únicamente a este nodo. El nodo procesó el comando recibido y ejecutó las acciones para ABRIR o CERRAR la electroválvula. Una vez realizada la acción se envió un acuse de recibo con el estado actual de la electroválvula, que puede ser visualizado en el canal de ThingSpeak. Este proceso se observa en la Figura 4.3.6.

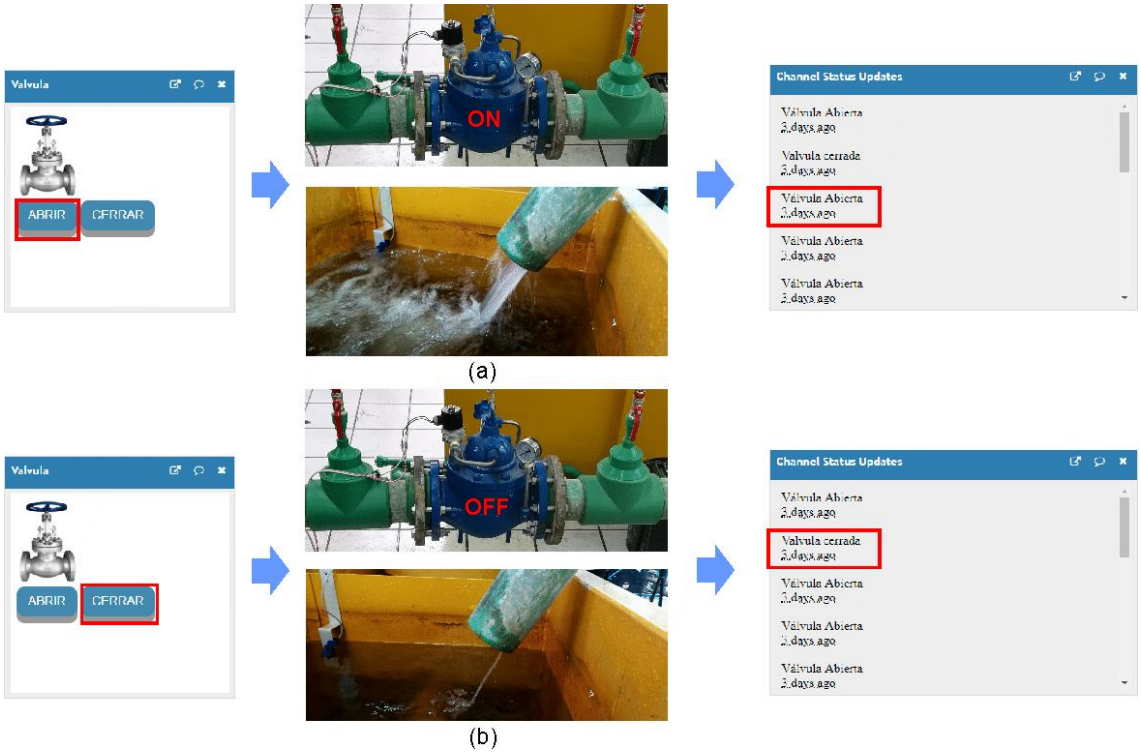


Figura 4.3.6. Proceso de funcionamiento del nodo de control de electroválvula.
Fuente y elaborado por los autores, basados en [31].

4.3.4. Nodo de variables meteorológicas

El nodo para la recolección de variables meteorológicas fue instalado en el campus de la UTPL. En este nodo se conectaron los sensores de temperatura y humedad ambiental, junto con la estación meteorológica WS-300, siguiendo las especificaciones de la Tabla 4.3.2.

Tabla 4.3.2. Especificaciones de conexión de nodo de variables meteorológicas

Socket	Sensor	Modelo	Campo ThingSpeak
D	Precipitación	WS-3000	field2
D	Velocidad de viento	WS-3000	field3
D	Dirección de viento	WS-3000	field4
A	Temperatura	SHT75	field5
A	Humedad	SHT75	field6
-	Batería	-	field1

Fuente y elaborado por los autores.

Para su ubicación se buscó en una zona despejada de edificios y árboles. La parte alta del campus, entre el edificio de Prendho y el tanque de reserva de agua de la universidad, cumple con los requerimientos para la instalación. Adicionalmente, en este punto existen otras estaciones meteorológicas instaladas y se utilizó la infraestructura existente para ubicar el nodo. Sus coordenadas son 3°59'15.37"S, 79°11'48.68"O y la distancia entre el *gateway* y el nodo es de 177m. En la Figura 4.3.7 se observa el nodo y sus sensores instalados.

Figura 4.3.7. Instalación de Nodo de Variables Meteorológicas.
Fuente y elaborado por los autores.

Para este nodo, el tiempo de envío fue establecido en 10 minutos. Se enviaron las mediciones instantáneas de todas las variables descritas en la Tabla 4.3.2. Este nodo sensor lleva recopilando datos por 376 días y el canal del servidor de ThingSpeak tiene actualmente 46 223 entradas de datos. En la Figura 4.3.8 se muestran los resultados de cinco días de pruebas de este nodo. Es posible visualizar todos los datos exportando el archivo CSV con todas las entradas de canal.

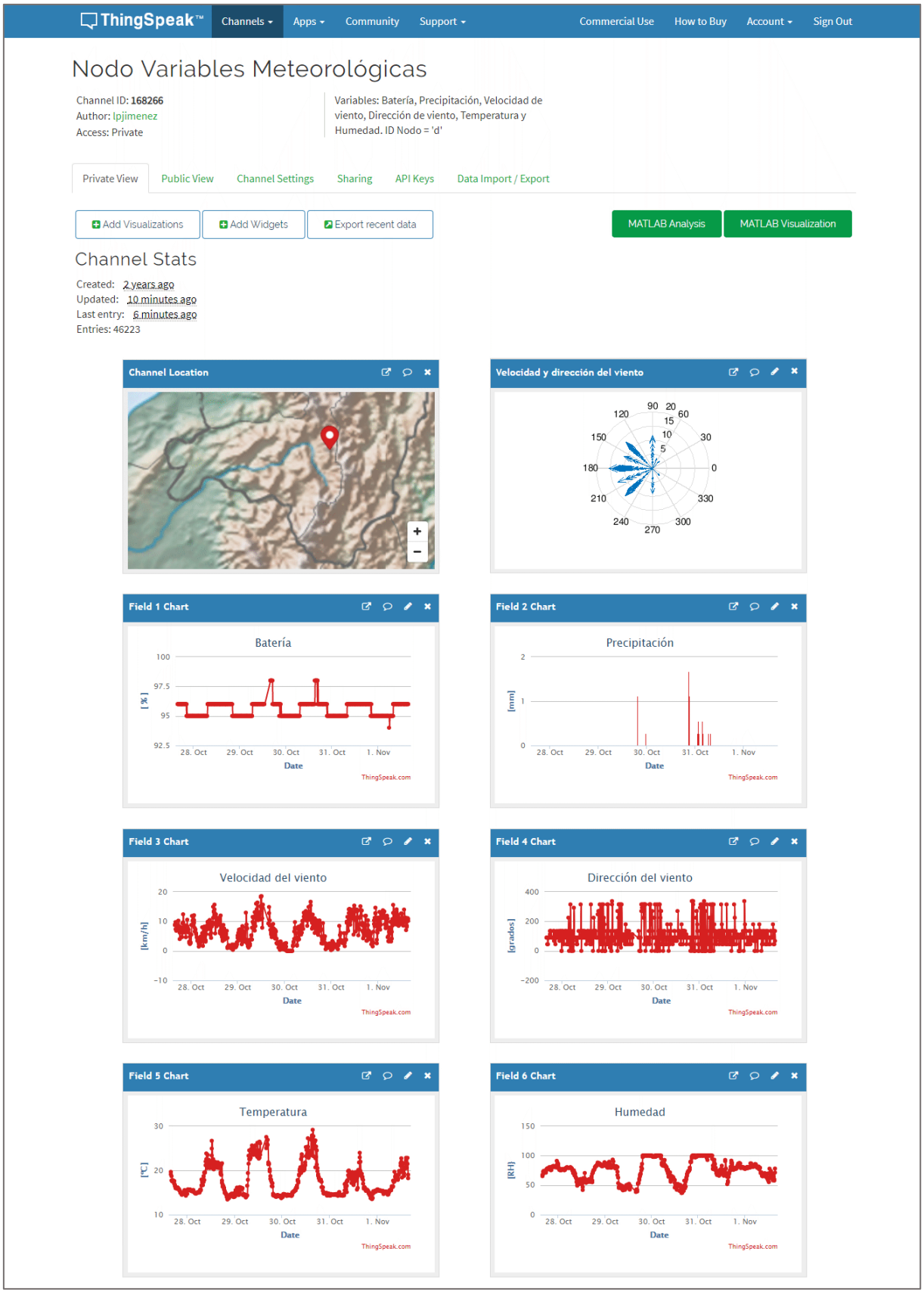


Figura 4.3.8. Resultados de pruebas del Nodo de Variables Meteorológicas.
Fuente y elaborado por [31].

4.3.5. Nodo Época bajo

Un nodo fue instalado en el tanque de reserva V-300M3-EPOCA BAJO para aprovechar el espacio de validación del caudalímetro y realizar pruebas. Este punto se encuentra fuera del área de cobertura inicial del *Gateway*. La distancia desde este punto de control hasta el *Gateway* ubicado en la UTPL es de 3 km, el nivel de señal se ve afectado y no era posible establecer la comunicación entre los dos puntos con el hardware por defecto. La opción de menor costo para solucionar el problema fue cambiar la antena omnidireccional del nodo por una de mayor ganancia y directividad. La antena que se utilizó es la Yagi–Uda de 900MHz. Para verificar que el enlace era factible se realizó la simulación en el software Radio Mobile, ver ANEXO E. Como resultado se obtuvo un margen de 17.2 dB. Este valor es superior a los mínimos recomendados, los cuales van desde 10 dB a 15 dB [58]. Por esta razón, el enlace es viable. Mediante la simulación también se obtuvieron los parámetros de azimut y elevación para alinear la antena.

Para la instalación de los equipos, primero se ubicó un mástil con su respectivo tensor para evitar la inclinación por el peso de la antena y permitir un mejor ajuste en el ángulo de elevación. Por motivos de seguridad el nodo se ubicó en el interior de la cámara de válvulas y mediante cable coaxial RG-58 con sus respectivos conectores en los extremos se conectó la antena de transmisión. El panel solar también se ubicó en el exterior, en un lugar despejado para evitar sombras que afecten el aprovechamiento de la radiación solar, ver Figura 4.3.9.

Figura 4.3.9. Instalación del panel solar y antena en la ciudadela Época.
Fuente y elaborado por los autores.

Los datos recopilados mediante este nodo sirvieron para verificar que el envío y almacenamiento de la información era realizado de forma correcta. También se evaluó el consumo energético cuando el módulo de comunicación XBee permanece encendido. Durante las pruebas efectuadas se presentaron inconvenientes que fueron corregidos y se detallan a continuación.

4.3.5.1. *Envío de información a la plataforma IoT.*

Los datos obtenidos en las mediciones fueron enviados a la plataforma ThingSpeak para la visualización. En la Figura 4.3.10 se muestran los niveles de batería recolectados, donde se puede observar los ciclos de carga y descarga. Durante los primeros días de pruebas se identificaron problemas en el enlace de radio entre el nodo y el *gateway*. El inconveniente fue causado por interferencias debido al uso de la frecuencia libre de 900MHz (banda ISM). Además, la línea de vista no está totalmente despejada debido a edificios y construcciones que existen en el lugar, como consecuencia existió la pérdida de información. El problema se solucionó reintentando el envío de la trama hacia el gateway en caso de error.

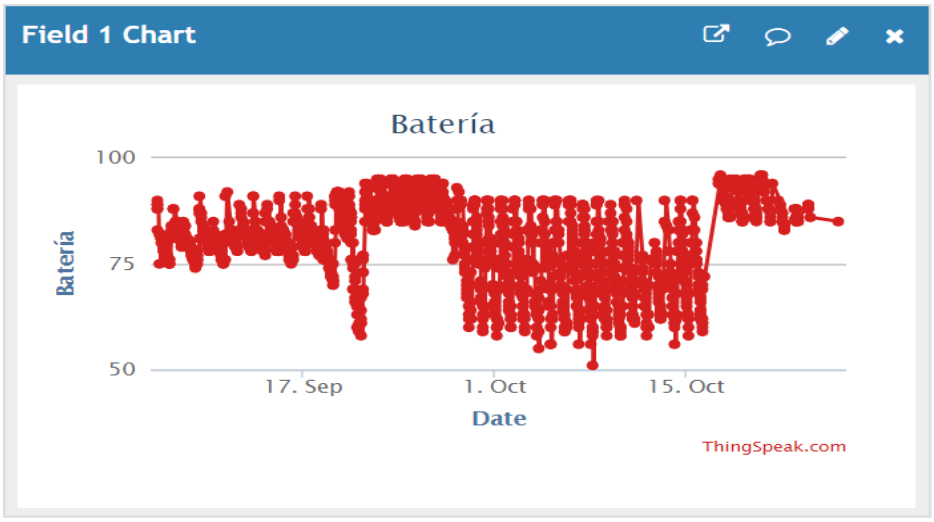


Figura 4.3.10. Niveles de batería recolectados en la plataforma ThingSpeak.
Fuente y elaborado por [31].

4.3.5.2. *Consumo energético*

Previo al análisis de consumo energético, se identificó que existía ineficiencia en la captación de energía solar. Al momento de adquirir los equipos, los soportes del panel solar tenían un ángulo de inclinación de 45°. Como consecuencia se obtenía un pobre aprovechamiento de la radiación solar. En la Figura 4.3.11 se observa que la batería se cargaba solo durante la mañana hasta aproximadamente las 11H00. Durante las horas siguientes no existía almacenamiento de energía y la batería comenzaba a descargarse.

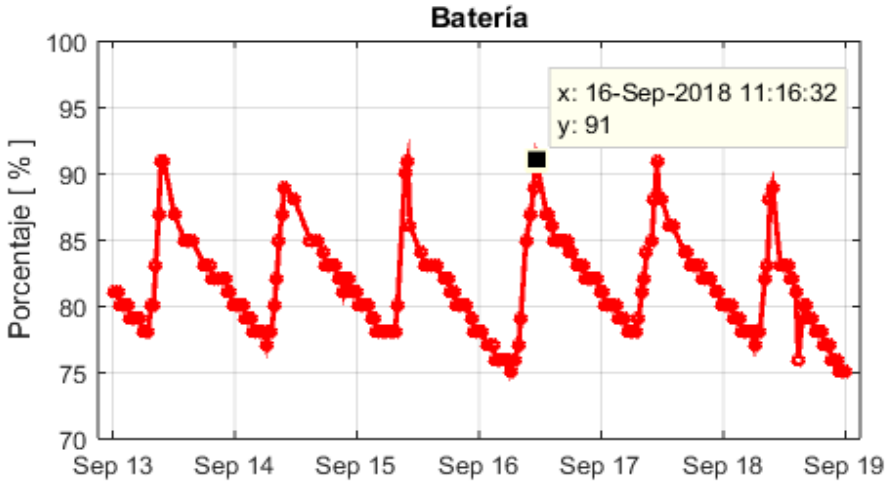


Figura 4.3.11. Niveles de batería con ángulo de inclinación de 45°.
Fuente y elaborado por los autores, basados en [31].

Para tener una mejor captación de energía, los rayos solares deben incidir perpendicularmente sobre el panel. Esto se logra modificando el ángulo de inclinación que varía en función de la latitud del lugar y durante las estaciones de invierno o verano, ver Tabla 4.3.3. La latitud de la ciudad de Loja es de 3.98° Sur, entonces, la inclinación óptima para la instalación del panel solar sería de 15° y su orientación hacia al Norte.

Tabla 4.3.3. Inclinación del panel solar en función de la latitud

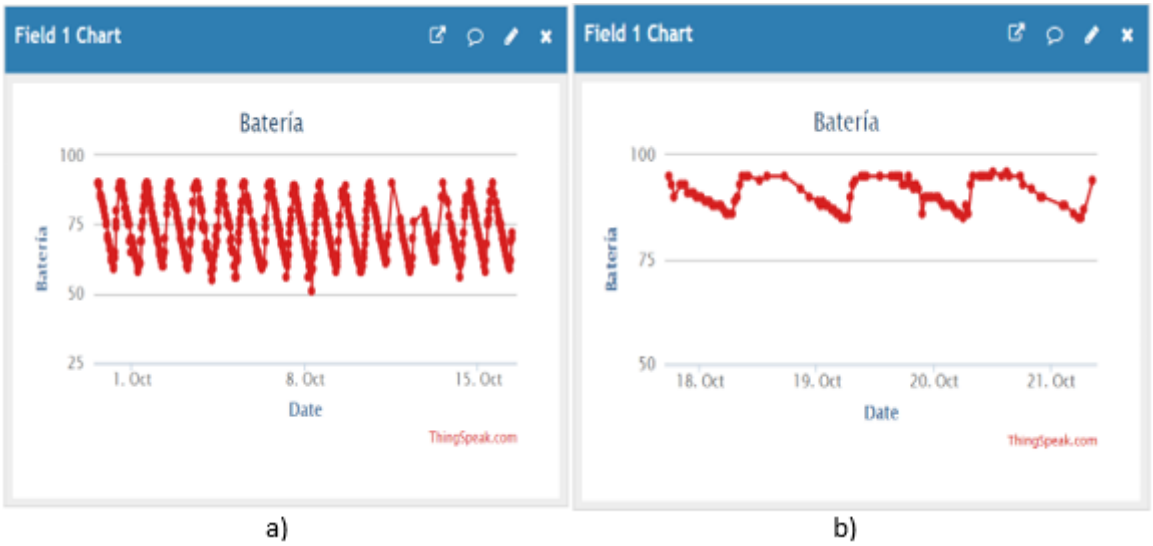
Latitud del lugar	Angulo en invierno	Angulo en verano
0° a 15°	15°	15°
15° a 25°	Latitud	Latitud
25° a 30°	Latitud + 5°	Latitud - 5°
30° a 35°	Latitud + 10°	Latitud - 10°
35° a 40°	Latitud + 15°	Latitud - 15°
> 40°	Latitud + 20°	Latitud - 20°

Fuente y elaborado por los autores, basados en [59].

Instalando el panel con las indicaciones descritas se mejoró el aprovechamiento de energía solar. Esto se traduce en un mayor tiempo de autonomía para el nodo sensor. Otros cambios también pueden ser realizados para extender su autonomía energética. Éstos se describen a continuación.

En la Figura 4.3.12a se muestra los niveles de batería para el caso crítico, donde el módulo de comunicación XBee permaneció todo el tiempo encendido. Este sería el caso para el nodo de Control de Electroválvula. Este nodo no puede ser apagado porque está en modo espera para recibir alguna trama desde el *gateway* y activar la electroválvula. De los datos recopilados en 17 días de pruebas, la batería comienzo a cargarse durante el día hasta alcanzar el nivel máximo del 90%. En la noche, los niveles de batería descienden hasta el 55% y a partir de este valor comienzo nuevamente el proceso de carga.

En la Figura 4.3.12b se muestra el consumo cuando el módulo de comunicación solamente fue encendido cada 30 minutos durante la transmisión de los datos. Para este caso, el proceso de carga se mantuvo constante en el día y durante la noche, existió un consumo del 10% de la batería. Comparando con los datos anteriores, se observa que los niveles de descarga fueron menores, como resultado el tiempo autonomía energética es extendido. Mantener el módulo de comunicación XBee siempre encendido representa un consumo de 3.5 veces superior que al activarlo solo durante la transmisión.



a) b)

Figura 4.3.12. Niveles de batería, a) XBee siempre encendido y b) XBee encendido solo para transmitir. Fuente y elaborado por [31].

CONCLUSIONES

La implementación de una red de sensores inalámbricos en el laboratorio de hidráulica de la Universidad Técnica Particular de Loja permite la gestión y monitoreo remoto de la red de distribución de agua, debido a que entrega información en tiempo real de: presión y caudal en tuberías, nivel de agua, fugas y variables meteorológicas.

Los nodos de Waspmote permiten la conexión de sensores de otros fabricantes. El sensor de presión MLH150PSB01 de Honeywell y el caudalímetro WTP100-400 de Seametrics han sido correctamente integrados utilizando la caja de acondicionamiento. También pueden controlar la electroválvula B11MN-DN15 como se ha mostrado.

El gateway permite la recolección de los datos de los nodos sensores y su envío a Internet para ser visualizados en la plataforma IoT ThingSpeak. Además, es capaz de enviar señales de control a los nodos sensores para la activación de las electroválvulas.

El uso de hardware y software abierto utilizado en los diferentes elementos de la red, permite la interoperabilidad entre dispositivos de varios fabricantes.

Los datos que miden los sensores han sido validados y se han desarrollado funciones para que funcionen en el nodo sensor. El error calculado en la medición es muy bajo en la mayoría de los casos, con el uso de las ecuaciones de ajuste presentadas. Esto permite contar con datos altamente confiables y exactos de las variables monitoreadas.

Los cuatro nodos sensores desplegados en los diferentes puntos del campus de la UTPL tienen autonomía energética, cuando su panel solar es instalado conforme a las indicaciones del presente trabajo. Los nodos funcionan de acuerdo con los requerimientos planteados como se visualiza en los resultados de las pruebas realizadas con esta WSN.

RECOMENDACIONES

Para evitar descargas profundas en la batería que reduzcan la vida útil, se recomienda que la misma esté cargada al momento de realizar las pruebas. Cuando los nodos se instalen en el área de estudio, la batería debe estar al 100% de su capacidad.

El modelo XBEE09P permite una longitud de trama máxima de 100 bytes de carga útil. Al superar este valor, el nodo presenta problemas para enviar la información al gateway. Para evitar este inconveniente es recomendable limitar la longitud de la trama.

Se recomienda seguir las indicaciones de cada fabricante para la instalación de los sensores para prevenir daños materiales, prolongar su vida útil y garantizar que las mediciones adquiridas sean exactas.

Para una correcta comunicación inalámbrica entre los nodos sensores y el *gateway* debe existir línea de vista entre las antenas de ambos dispositivos y los dispositivos se deben ubicar en el área de cobertura. Obstáculos como: árboles, edificios o planchas metálicas pueden causar la pérdida de datos de la red.

El panel solar debe ser ubicado en una zona despejada con una inclinación de 15º y orientado al Norte. De esta forma puede captar la mayor cantidad de energía solar y prolongar el tiempo de autonomía energética de los nodos sensores.

La electroválvula utilizada en este trabajo tiene un modo de operación “normalmente cerrada” y debe estar energizada para permitir el paso de agua. Instalarla a la salida del tanque va a representar un alto consumo de energía, debido a que la mayor parte del tiempo estará abierta. Por este motivo, se recomienda el uso de una válvula con modo de operación normalmente abierta.

En caso de un corte de energía, la electroválvula NC va a bloquear el flujo de agua. Para evitar este inconveniente, se recomienda utilizar un respaldo de energía mediante un banco de baterías.

BIBLIOGRAFÍA

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Comput. Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] M. a. Song, Zhenyu; Lazarescu, Mihai T.; Tomasi, Riccardo; Lavagno, Luciano; Spirito, *Internet of Things Applications: Challenges and Opportunties*. 2014.

[3] S. Saseendran and V. Nithya, “Automated water usage monitoring system,” *2016 Int. Conf. Commun. Signal Process.*, pp. 99–103, 2016.

[4] E. Kyriakides, *Studies in Computational Intelligence 565 Intelligent Monitoring , Control , and Security of Critical Infrastructure Systems*. .

[5] A. J. Whittle *et al.*, “WaterWiSe@SG: A Testbed for Continuous Monitoring of the Water Distribution System in Singapore,” in *Water Distribution Systems Analysis 2010*, 2011, pp. 1362–1378.

[6] Municipio de Loja, “REGENERACIÓN URBANA, ACTUALIZACIÓN DEL DISEÑO DEL SISTEMA DE AGUA POTABLE,” 2015.

[7] H. Benavides, “Comunicación oral.” 2018.

[8] A. Whittle, M. Allen, A. Preis, and M. Iqbal, “Sensor Networks for Monitoring and Control of Water Distribution Systems,” in *6th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII 2013)*, 2013.

[9] A. Ayadi, O. Ghorbel, A. Obeid, M. S. Bensaleh, and M. Abid, “Leak Detection in Water Pipeline by Means of Pressure Measurements for WSN,” pp. 1–6, 2017.

[10] S. Xhafa, I. Avdullahu, and M. Ahmeti, “Automation Control on Water Supply Networks,” *IFAC-PapersOnLine*, vol. 49, no. 29, pp. 175–179, 2016.

[11] L. Mays, *WSO: Water Transmission and Distribution*, Fourth Edi. American Water Works Association, 2010.

[12] N. K. Shammas and L. K. Wang, *Water engineering*. Wiley, 2016.

[13] J. D. Wilson, *Física*, 1st ed. Pearson Education, 2007.

[14] S. H. Quasim, *SI UNITS IN ENGINEERING AND TECHNOLOGY*, 1st ed. Great Britain: Pergamon Press, 1977.

[15] I. Stoianov, L. Nachman, S. Madden, T. Tokmouline, and M. Csail, “PIPENET: A Wireless Sensor Network for Pipeline Monitoring,” *Inf. Process. Sens. Networks, 2007. IPSN 2007. 6th Int. Symp.*, pp. 264–273, 2007.

[16] J. S. M. Coleman and K. T. Law, “Meteorology,” in *Reference Module in Earth Systems and Environmental Sciences*, Elsevier, 2015, p. .

[17] C. D. Ahrens, *Essentials of Meteorology: An Invitation to the Atmosphere*. Cengage Learning, 2011.

[18] Elizabeth Mills, *Weather Studies: Introduction to Atmospheric Science 6th Ed*. .

[19] glosarios@servidor-alicante.com, “Precipitación pluvial [Precipitation] (Ecología),” *glosarios@servidor-alicante.com*, Aug-2015. .

[20] J. S. Rueda and J. M. T. Portocarrero, “Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una postura clarificadora,” *Rev. Colomb. Comput.*, vol. 18, no. 2, pp. 58–74, 2017.

[21] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: theory and practice*. 2010.

[22] L. Farrugia, *Wireless Sensor Network*. New York: NOVA, 2011.

[23] F. Dressler, *Self-organization in sensor and actor networks.* John Wiley & Sons, 2008.

[24] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, 2010.

[25] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," *2017 IEEE Int. Symp. Syst. Eng. ISSE 2017 - Proc.*, 2017.

[26] A. Al-Fuqaha, A. Khreishah, M. Guizani, A. Rayes, and M. Mohammadi, "Toward better horizontal integration among IoT services," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 72–79, 2015.

[27] A. Chaudhary, S. K. Peddoju, and K. Kadarla, "Study of Internet-of-Things Messaging Protocols Used for Exchanging Data with External Sources," *2017 IEEE 14th Int. Conf. Mob. Ad Hoc Sens. Syst.*, pp. 666–671, 2017.

[28] Digi Internationa, "WIRELESS MESH NETWORKING: ZIGBEE ® VS. DIGIMESH ®."

[29] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IoT," *ICCEREC 2016 - Int. Conf. Control. Electron. Renew. Energy, Commun. 2016, Conf. Proc.*, pp. 1–6, 2017.

[30] ThingSpeak, "MQTT Basics - MATLAB & Simulink." [Online]. Available: https://la.mathworks.com/help/thingspeak/mqtt-basics.html. [Accessed: 21-Nov-2018].

[31] "IoT Analytics - ThingSpeak." [Online]. Available: https://thingspeak.com/. [Accessed: 01-Feb-2017].

[32] Libelium, "Libelium." [Online]. Available: http://www.libelium.com/products/waspmote/hardware/.

[33] Dragino, "Welcome to Dragino Wiki Page." [Online]. Available: http://wiki.dragino.com/index.php?title=Main_Page.

[34] DIGI, "XCTU User Guide," 2018. [Online]. Available: https://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm. [Accessed: 11-Sep-2018].

[35] Digi, "XBee®/XBee-PRO® 900 OEM RF Modules 900 OEM RF Modules." Digi International Inc., p. 40, 2008.

[36] M. Quiñonez, "Tecnologías Inalámbricas de Comunicación: MANET-WSN." Loja, p. 112, 2017.

[37] ThingSpeak, "Update channel data with HTTP GET or POST - MATLAB Write Data." [Online]. Available: https://la.mathworks.com/help/thingspeak/writedata.html. [Accessed: 16-Nov-2018].

[38] PyPi, "paho-mqtt." [Online]. Available: https://pypi.org/project/paho-mqtt/. [Accessed: 01-May-2018].

[39] PyPi, "requests." .

[40] PyPi, "XBee PyPi." [Online]. Available: https://pypi.org/project/XBee/. [Accessed: 01-May-2018].

[41] Dragino, "M328W." [Online]. Available: http://wiki.dragino.com/index.php?title=M328W.

[42] Dragino, “Ms14 gpios - Wiki for Dragino Project.” [Online]. Available: http://wiki.dragino.com/index.php?title=Ms14_gpios. [Accessed: 20-Aug-2018].

[43] OpenWrt, “OpenWrt Project: Init Scripts.” [Online]. Available: https://openwrt.org/docs/techref/initscripts. [Accessed: 20-Aug-2018].

[44] Libelium, “Waspmote Technical Guide.” p. 170, 2013.

[45] Libelium Comunicaciones Distribuidas S.L., “Events Board v30 / Plug & Sense! Security Guide.” 2017.

[46] Libelium, “Agriculture sensor board 2.0.” .

[47] TU Delft OCW, “Sensors,” *2016*. [Online]. Available: https://ocw.tudelft.nl/course-lectures/sensors/. [Accessed: 01-Oct-2018].

[48] Honeywell, “Heavy Duty Pressure Transducers Datasheet,” 2016. [Online]. Available: www.honeywell.com.

[49] Seametrics, “WT-SERIES Turbine Meters,” 2014.

[50] Seametrics, “Inline Turbine Meter Instructions,” 2014.

[51] Hobby Electronics, “YF-S201 Hall Effect Water Flow Meter / Sensor |.” [Online]. Available: http://www.hobbytronics.co.uk/yf-s201-water-flow-meter. [Accessed: 21-Aug-2018].

[52] Hobby Electronics, “YF-S201 Hall Effect Water Flow Meter / Sensor.” [Online]. Available: http://www.hobbytronics.co.uk/yf-s201-water-flow-meter. [Accessed: 22-Aug-2018].

[53] National Instruments, “What is the Difference Between Sinking and Sourcing Digital I/O?” [Online]. Available: https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019LBbSAM. [Accessed: 25-Sep-2018].

[54] “Sourcing and sinking current | Liudr’s Blog.” [Online]. Available: https://liudr.wordpress.com/2011/10/16/sourcing-and-sinking-current/. [Accessed: 25-Sep-2018].

[55] Seametrics, “FT400-SERIES Rate/Total Indicator Instructions.”

[56] Naylamp Mechatronics, “Módulo Relay 2CH 5VDC.” [Online]. Available: https://naylampmechatronics.com/drivers/31-modulo-relay-2-canales-5vdc.html. [Accessed: 27-Oct-2018].

[57] OMEGA, “OM-PL SERIES User’s Guide.” p. 32, 2015.

[58] E. Pietrosemoli, M. Zennaro, C. Fonda, S. Okay, and C. Aichele, *Redes inalámbricas en los países en desarrollo*, 4th ed. 2013.

[59] M. P. Aparicio, *Energía solar fotovoltaica: cálculo de una instalación aislada*. Marcombo, 2010.

[60] Dragino, “MS14N-P.” [Online]. Available: http://www.dragino.com/products/mother-board/item/113-ms14n-p.html. [Accessed: 13-Feb-2018].

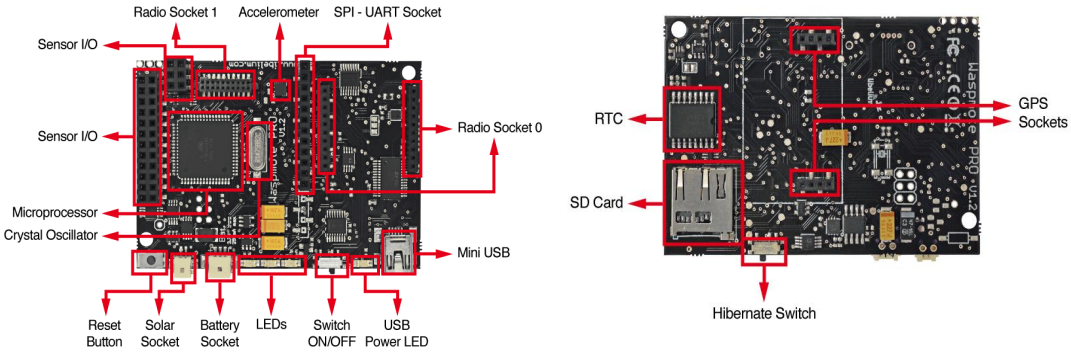
[61] Digi, “XBee®/XBee-PRO® 900 OEM RF Modules 900 OEM RF Modules.” Digi International Inc., p. 40, 2008.

ANEXOS

ANEXO A

ESQUEMÁTICOS DE TARJETAS UTILIZADAS

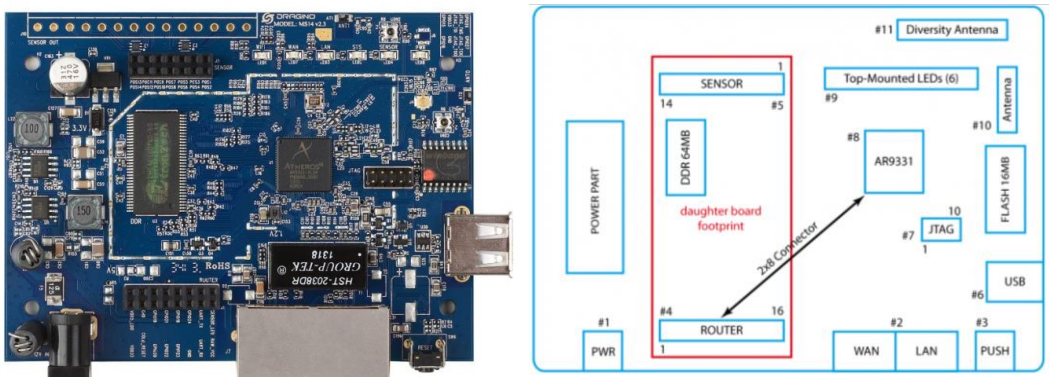
A.1. Tarjeta Waspmote



a) Frontal. b) Posterior.

Fuente: [32].

A.2. Draguino MS14



a) Tarjeta b) Esquemático

Fuente: [60]

A.3. XBee09P

Fuente: [61]

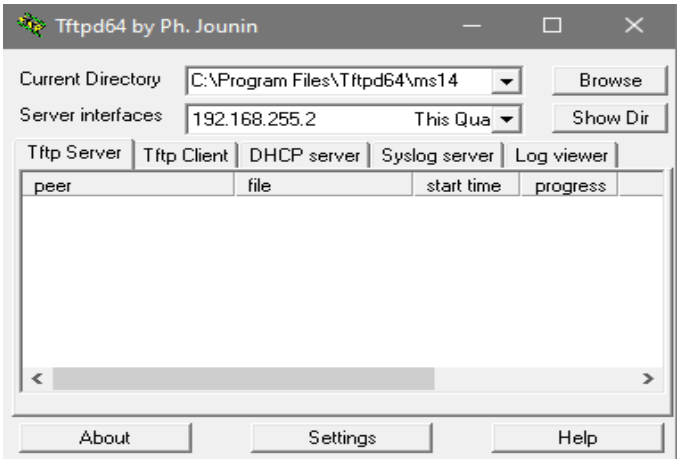
ANEXO B

ACTUALIZACIÓN Y CONFIGURACIÓN EN DRAGUINO

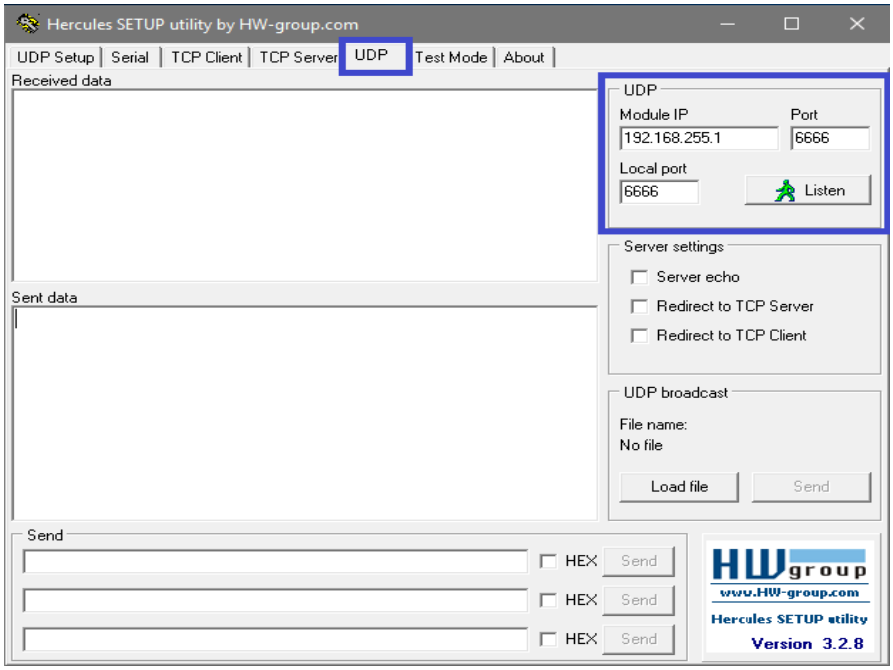
B.1. Actualización de firmware MS14

Existen varias formas de actualizar el firmware a la placa Dragino MS14 como se muestra en el siguiente enlace. Para el presente trabajo se utiliza el modo U-boot NetConsole, que a diferencia de los otros modos es más tedioso y requiere la instalación de otros programas. Sin embargo, este método es ideal en los siguientes casos: no se tiene las claves para acceder a la administración, se ha modificado el funcionamiento del botón de reset o ha ocurrido un error durante la actualización del firmware. A continuación, se explica los pasos que se deben seguir:

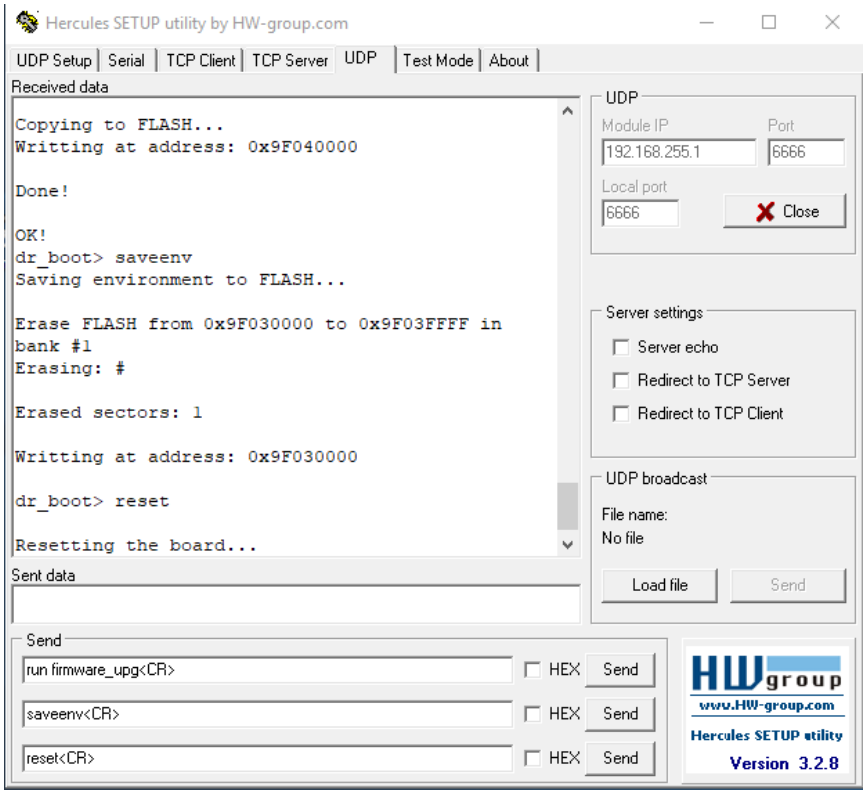
- Configurar la PC con la dirección IP 192.168.255.2 con máscara 255.255.255.0. Es necesario deshabilitar el firewall de Windows.
- Instalar un servidor TFTP (se recomienda tftp32), crear una carpeta en el directorio de instalación y guardar el firmware más reciente, descargado de la siguiente página. Renombrar el archivo a “firmware.bin”
- Configurar el servidor tftpd32 seleccionando la carpeta y la ip anteriormente configuradas, como se ve en la siguiente figura.



- Descargar la herramienta Hercules. Configurar como se muestra en la siguiente figura y presionar el botón “Listen”
- Usando un cable de red conectar el puerto RJ45 de la PC al puerto LAN del dispositivo.



- Presionando el botón de reset se procede a conectar el cable de alimentación, todos los leds empezaran a parpadear. Si el proceso se ha realizado correctamente, después de 10 segundos aparecerán algunos mensajes en Hercules como muestra la siguiente figura.

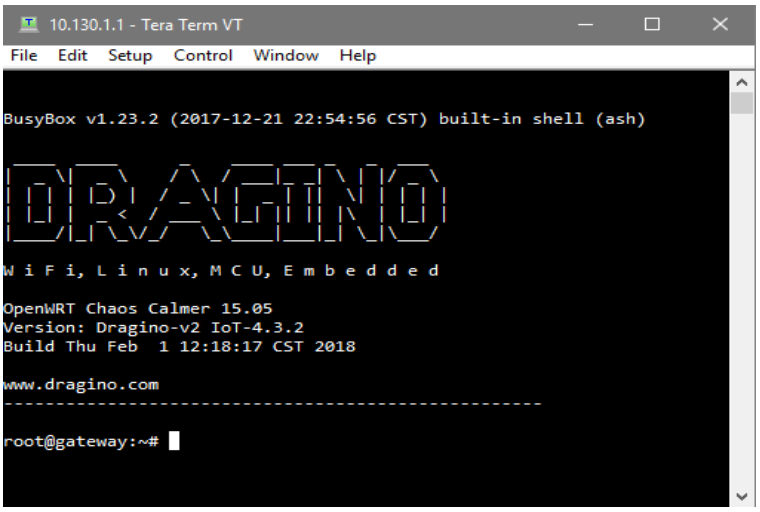


B.2. Configuraciones iniciales

Cuando un nuevo firmware es instalado en el MS14, se crea una red wifi con nombre “dragino-xxxxxx” que no tiene contraseña de acceso. Para realizar configuraciones en el dispositivo es necesario conectarse a la red y en un navegador ingresar la dirección IP “10.130.1.1” donde se pedirá iniciar sesión para acceder a la interfaz Web de administración. Por defecto, el nombre de usuario es “root” y la contraseña “dragino”. Las configuraciones que se realizan son las siguientes:

1. En System -> Administration cambiar la contraseña para ingresar por vía SSH, Web GUI o RS232.
2. En Network -> Access Point habilitar la encriptación WPA2-AES, cambiar el SSID y la contraseña.
3. En System -> System modificar el hostname y timezone.

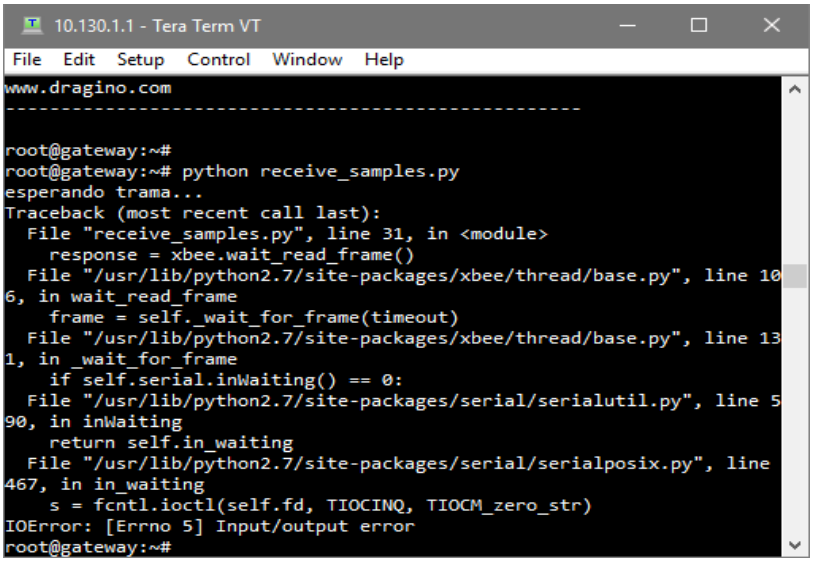
Otra forma de administrar y personalizar el sistema a través de comandos es ingresando por vía SSH. Para ello, es necesario un emulador de terminal que soporte el protocolo SSH (se recomienda Tera Term). En host se escribe la dirección ip 10.130.1.1, el usuario es “root” y la contraseña es la establecida en el punto 1. La siguiente figura muestra el mensaje de bienvenida que aparece cuando se inicia sesión de forma correcta.



Para la comunicación entre módulo de radio Xbee y la placa Dragino MS-14 es necesario deshabilitar el modo consola, en la interfaz Web hay que dirigirse a Sensor -> PowerUART y en “UART Operation Mode” elegir “Disable Linux Console”. Sin embargo, se puede presentar el siguiente problema con el puerto serie (UART): al momento de establecer la comunicación serial con el XBee se produce el siguiente mensaje de error:

File “/usr/lib/python2.7/site-packages/serial/serialposix.py”, line 467, in in_waiting
s = fcntl.ioctl(self.fd, TIOCINQ, TIOCM_zero_str).

El tipo de error producido es: **IOError: [Errno 5] Input/output error,** el cual ocurre porque otro proceso está haciendo uso del puerto.



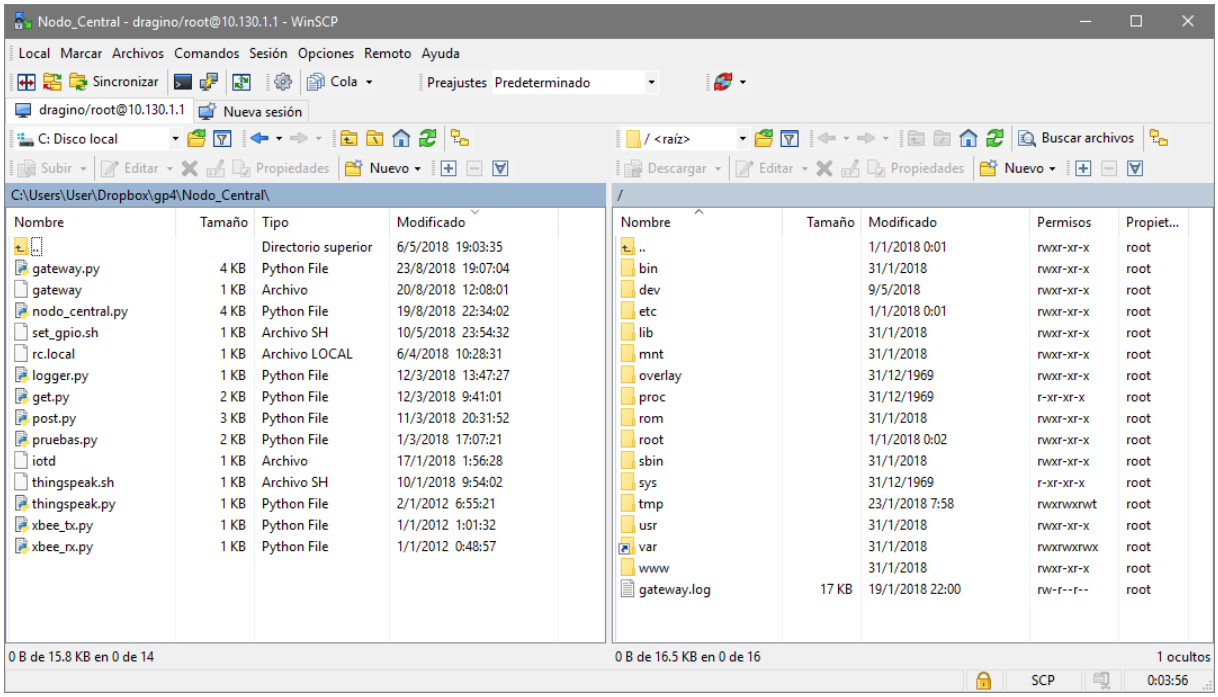
La solución es obtenida de la wiki de OpenWRT, disponible en el siguiente enlace. La misma consiste en editar dos archivos:

El primero se encuentra en la ruta */etc/sysctl.conf* en el cual se debe agregar al principio del fichero la siguiente línea: **kernel.printk = 0 4 1 7**

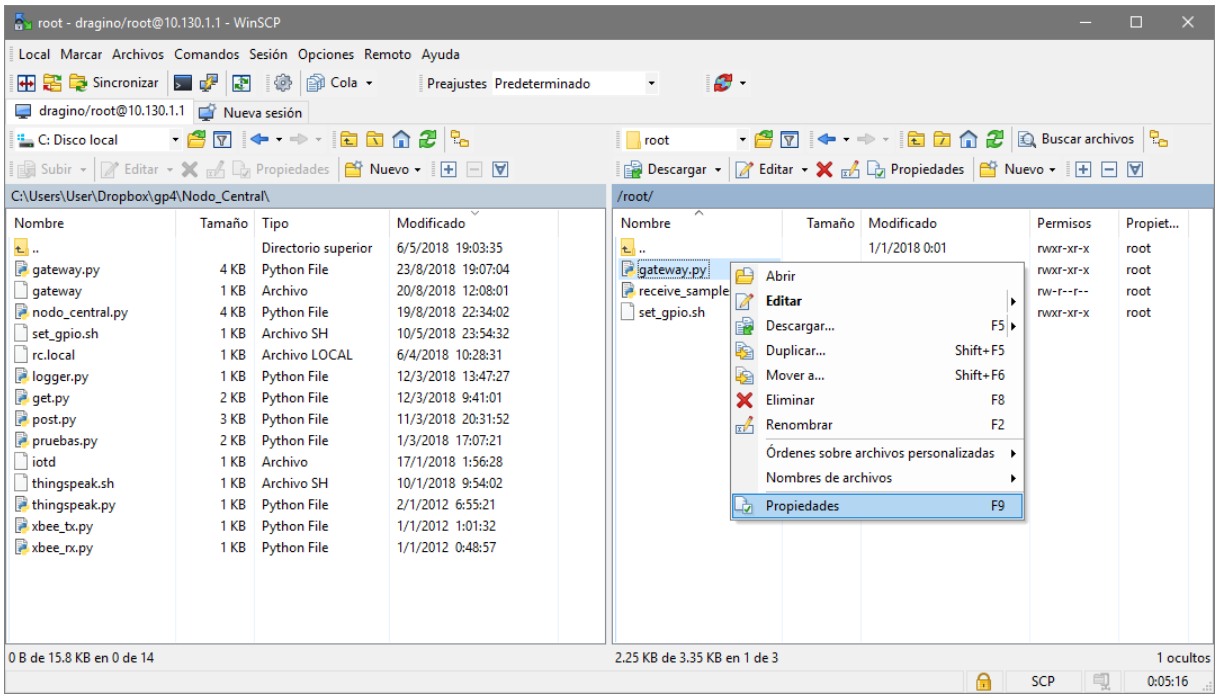
El segundo se encuentra en la ruta /etc/inittab en el cual se debe buscar y comentar la línea: #::askconsole:/bin/ash –login.

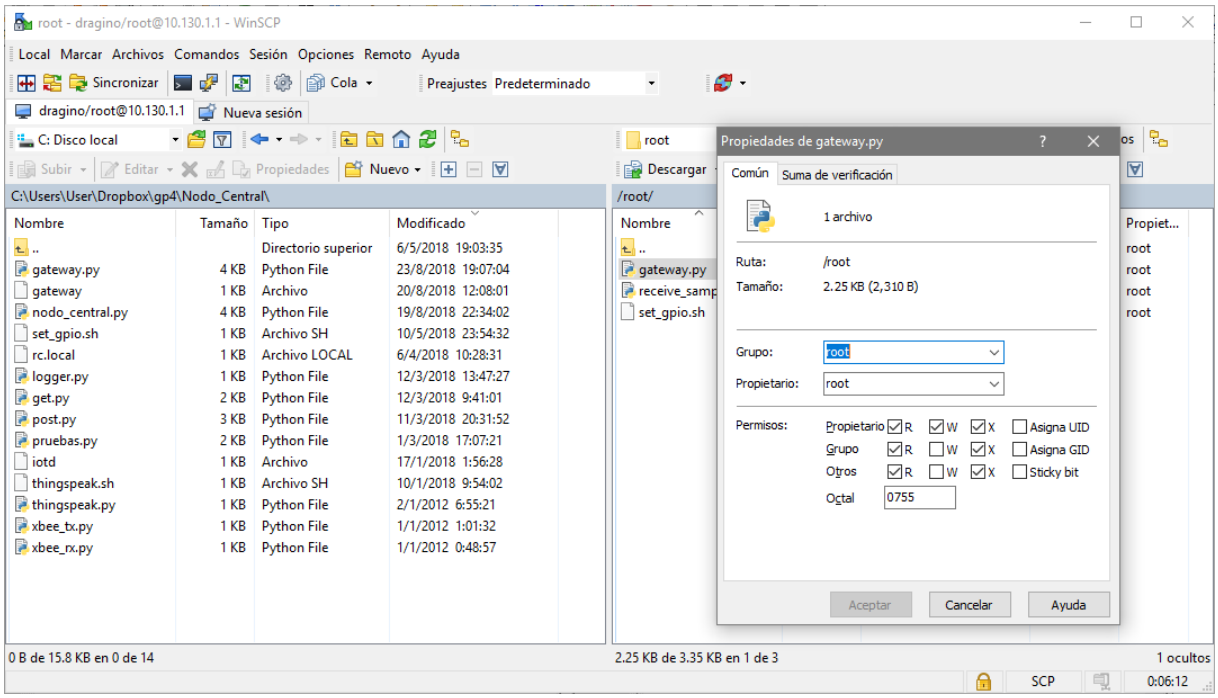
B.3. Manejo de archivos en el Dragino

Para facilitar la navegación por el árbol de directorios del sistema OpenWrt se recomienda utilizar WinSCP. El programa muestra una interfaz gráfica con todas las carpetas disponibles, similar al explorador de archivos de Windows, como en la siguiente figura.



La herramienta entre otras cosas permite realizar lo siguiente: crear y editar scripts, transferir archivos desde Windows a OpenWrt y viceversa de forma sencilla como arrastrar y soltar. Para gestionar los permisos de los scripts se debe hacer clic derecho en el archivo de interés y en propiedades modificar los permisos, ver las siguientes figuras.



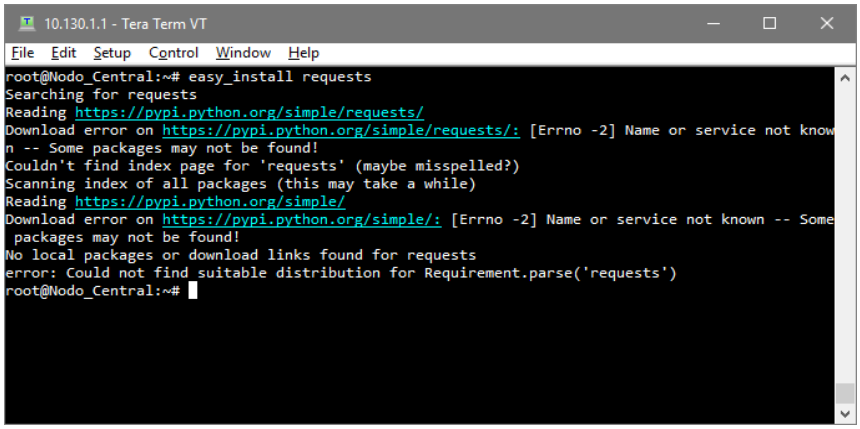


ANEXO C

PROBLEMAS EN COFIGURACIÓN DE GATEWAY

C.1. Problema con los servidores DNS

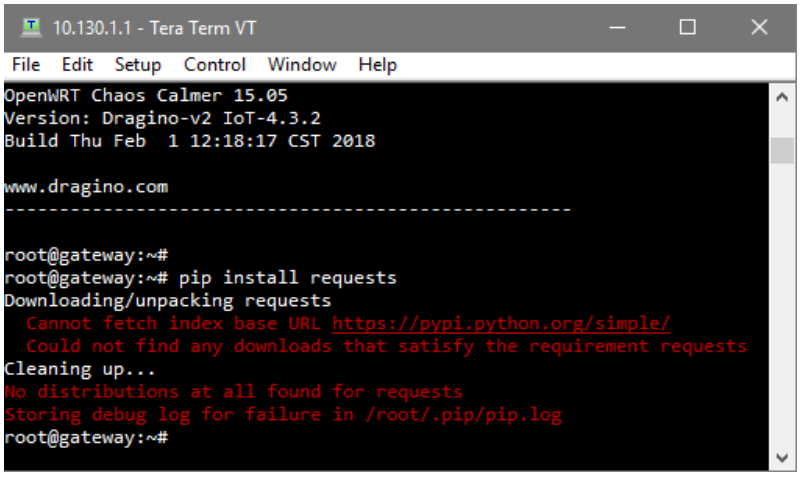
El error se presenta al instalar módulos de Python con easy_install, el mensaje de error es el siguiente: **[Errno -2] Name or service not known – Some packages may not be found!** Ver la siguiente figura.



La solución consiste en agregar manualmente la dirección ip de los servidores DNS que utiliza la Universidad. Con algún editor de archivos (por ejemplo: nano) se abre el fichero: # *nano /etc/resolv.conf* y se escribe las líneas *nameserver 172.16.50.58* y *nameserver 172.16.50.54*.

C.2. Problema con la verificación de certificados.

El error ocurre cuando se accede a páginas seguras en Internet (https) y es causado por una incorrecta configuración en la fecha y hora del sistema, podemos verificar ejecutando el comando: # *date* en la terminal. La placa Dragino no posee un circuito RTC, pero tiene habilitado un cliente NTP para la actualización automática de la hora. Sin embargo, la red del Campus tiene implementado algunas políticas de seguridad que impiden la actualización. El mensaje de error con *easy_install* es el siguiente: **[SSL: CERTIFICATE_VERIFY_FAILED] unknown error (_ssl.c581) – Some packages may not be found!** Utilizando pip install el error es: **Could not find any downloads that satisfy the requeriment.** Ver la figura que sigue.

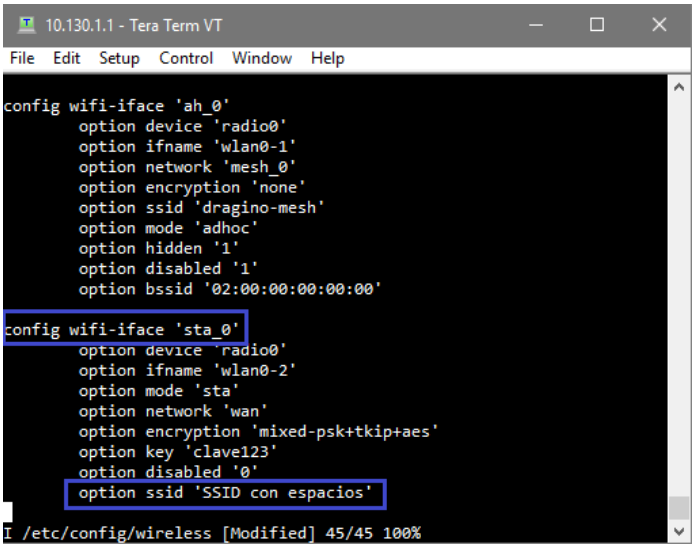


La solución fue sincronizar manualmente la fecha, en la interfaz Web de administración hay que dirigirse a System → System y en Local Time presionamos el botón “Sync with browser” y guardar los cambios realizados. Se debe mencionar que esta fecha se volverá a desconfigurar cuando se apague el Dragino MS14.

C.3. Conectarse a una red WiFi con espacios.

En la interfaz Web de administración en Network → Internet Access se puede configurar el modo de acceso a Internet. Cuando se elige a través de un cliente WiFi el nombre de la red no debe tener espacios ya que no es reconocido en la interfaz Web. Una solución es cambiar el nombre de la red a una que no tenga espacios, sin embargo, a veces no es posible porque no se tiene el nombre de usuario y contraseña ára acceder a la administración del router.

Otra solución es configurar en la interfaz Web en Network → Internet seleccionar via “WiFi Cliente” y configurar los parámetros que nos solicita, teniendo en cuenta que el SSID hay que escribirlo sin espacios, luego guardamos y aplicamos los cambios. Después, se accede al terminal por vía SSH, haciendo uso de algún editor de archivos (por ejemplo: nano) se abre el siguiente fichero: # *nano /etc/config/wireless.* Se baja hasta la sección *config wifi-iface 'sta_0'* para buscar la línea *option ssid* y añadir los espacios al nombre de la red, ver figura siguiente. Finalmente, se guarda los cambios y se reinicia el dispositivo ejecutando # *reboot*.



C.4. Problema con la librería XBee

Esta librería presenta un problema cuando la trama tiene caracteres de escape. Entonces, el bucle de la librería se queda esperando más bytes y no imprime la trama recibida.

La solución consiste en editar el script que encuentra en la siguiente ruta */usr/bin/python2.7/site-packages/xbee/frame*, buscar la función *fill* y comentar las líneas desde la 133 hasta la 138 y agregar el código como se muestra en la figura. Estas líneas son tomadas de versiones anteriores de la misma librería.

```
124     def fill(self, byte):
125         """
126         fill: byte -> None
127 
128         Adds the given raw byte to this APIFrame. If this APIFrame is mar
129         as escaped and this byte is an escape byte, the next byte in a ca
130         to fill() will be unescaped.
131 
132 
133         if self._unescape_next_byte:
134             byte = intToByte(byteToInt(byte) ^ 0x20)
135             self._unescape_next_byte = False
136         elif self.escaped and byte == APIFrame.ESCAPE_BYTE:
137             self._unescape_next_byte = True
138             return
139         """
140         if self._unescape_next_byte:
141             byte = intToByte(byteToInt(byte) ^ 0x20)
142             self._unescape_next_byte = False
143         elif byte == APIFrame.ESCAPE_BYTE:
144             self._unescape_next_byte = True
145             return
146 
147         self.raw_data += intToByte(byteToInt(byte))
```

ANEXO D

CÓDIGO PARA EL WASPMOTE IDE

D.1. Código del nodo del PRDA

```
/*
------ Waspmote Pro Code PRDA --------
en este nodo, el socket D se usa para el sensor de presion
A->fugas
C y D->analog5 y analog4 -> presión
B->caudal
E->nivel
*/

// Librerías
#include <WaspBody.h>
#include <WaspXBeeDM.h>
#include <WaspSensorEvent_v30.h>

//definición de variables
uint8_t nivel = 0;
char nodeID[] = "e";
char RX_ADDRESS[] = "0013A20040A523B2";
uint8_t error;
int aux = 0;
float presion = 0.0;
float presion2 = 0.0;
float value = 0.0;
float flujo = 0.0;
float fuga = 0;
char* str = "";
float mini = 101.0;
float maxi = 0.0;
float prespro = 0.0;
int i = 0;
bool pro = false;

//clases, definir el sensor conectado a cada socket
flowClass yfg1(SENS_FLOW_YFG1);
liquidLevelClass liquidLevel(SOCKET_E);
liquidPresenceClass liquidPresence(SOCKET_A);

//definiciones para sensores de deteccion de líquidos
#define DRY 0.0
#define WET 0.1
#define VERY_WET 1.0

////////////////////////////////////////////////////
//              CONFIGURACION INICIAL             //
////////////////////////////////////////////////////
void setup()
{
// put your setup code here, to run once:
USB.ON();
USB.println(F("Start program"));
body.setID( nodeID );
PWR.setSensorPower(SENS_5V, SENS_ON);
// Turn on the sensor board
Events.ON();
delay (100);
```

```
//mediciones de prueba
aux = analogRead(ANALOG4);
aux = analogRead(ANALOG5);
delay (100);
leer();
// Tiempo para realizar el primer envio
RTC.ON();
RTC.setAlarm1("00:00:00:10", RTC_OFFSET, RTC_ALM1_MODE2);
Events.attachInt();
}

////////////////////////////////////////////////////
//    FUNCIÓN PARA VALOR INSTATÁNEO DE PRESION    //
////////////////////////////////////////////////////
void instantaneo() {
value = read_presion(ANALOG4);
}

////////////////////////////////////////////////////
//    FUNCIÓN PARA VALOR MÁXIMO DE PRESION        //
////////////////////////////////////////////////////
void maximo() {
if (value > maxi) {
maxi = value;
}
}

////////////////////////////////////////////////////
//    FUNCIÓN PARA VALOR MÍNIMO DE PRESION        //
////////////////////////////////////////////////////
void minimo() {
if (value < mini) {
mini = value;
}
}

////////////////////////////////////////////////////
//    FUNCIÓN PARA VALOR PROMEDIO DE PRESION      //
////////////////////////////////////////////////////
void promedio() {

if (pro) {
prespro = prespro / i;
i = 0;
    //USB.print(F("Promedio: "));
    //USB.print(prespro);
    } else {
    prespro = prespro + value;

    }
    }

    ////////////////////////////////////////////////////
    //                  PROGRAMA PRINCIPAL            //
    ////////////////////////////////////////////////////
    void loop()
    {
    instantaneo();
    maximo();
    minimo();
    promedio();
```

```
117.      i++;
118.      //comprobacion de tiempo de envio
119.      if (intFlag & RTC_INT)
120.      {
121.      // Disable interruptions from the board
122.      Events.detachInt();
123.      USB.println(F("-----------------------------"));
124.      USB.println(F("RTC INT captured"));
125.      USB.println(F("-----------------------------"));
126.      leer();
127.      pro = true;
128.      promedio();
129.      enviar_datos();
130.      i = 0;
131.      mini = 101.0;
132.      maxi = 0.0;
133.      prespro = 0.0;
134.      i = 0;
135.      pro = false;
136.      // Intervalo de medicion y envio de datos
137.      RTC.ON();
138.      RTC.setAlarm1("00:00:05:00", RTC_OFFSET, RTC_ALM1_MODE2);
139.      // clear flag
140.      intFlag &= ~(RTC_INT);
141.      // Enable interruptions from the board
142.      Events.attachInt();
143.      }
144.      delay(1000);
145.      }
146.      ////////////////////////////////////////////////////
147.      //  FUNCIÓN PARA LECTURA DE VARIABLES DE SENSORES   //
148.      ////////////////////////////////////////////////////
149.      void leer() {
150.
151.      USB.println(F("------------- Read process ---------------"));
152.
153.      ////////////////////////////////////////
154.      //           sensor nivel             //
155.      ////////////////////////////////////////
156.      // Read the sensor level
157.      nivel = liquidLevel.readliquidLevel();
158.
159.      // Print the info
160.      if (nivel == 1)
161.      {
162.      USB.println(F("Level output: LEVEL REACHED"));
163.      }
164.      else
165.      {
166.      USB.println(F("Level output: LEVEL NOT REACHED"));
167.      }
168.      ////////////////////////////////////////
169.      //      sensor presión                //
170.      ////////////////////////////////////////
171.
172.      presion = read_presion(ANALOG4);
173.      presion2 = read_presion(ANALOG5);
174.
175.      ////////////////////////////////////////
176.      //          sensor flujo              //
177.      ////////////////////////////////////////
```

```
178.      flujo = yfg1.flowReading();
179.      if (flujo < 0) {
180.      flujo = 0.0;
181.      }
182.      // Print the flow read value
183.      USB.print(F("Flow: "));
184.      USB.print(flujo);
185.      USB.println(F(" l/min"));
186.
187.      ////////////////////////////////////
188.      //      sensor fugas linea        //
189.      ////////////////////////////////////
190.      // Read the sensor level
191.      fuga = liquidPresence.readliquidPresence();
192.      // Print the info
193.      if ((DRY <= fuga) && (fuga <= WET))
194.      {
195.      USB.println(F("Leakage output: Water not detected"));
196.      str = "SF";
197.      }
198.      else if ((WET <= fuga) && (fuga <= VERY_WET))
199.      {
200.      USB.println(F("Leakage output: Water detected!"));
201.      str = "CF";
202.      }
203.      else if (fuga >= VERY_WET)
204.      {
205.      USB.println(F("Leakage output: A lot of water detected!"));
206.      str = "MF";
207.      }
208.      }
209.
210.      ////////////////////////////////////////////////////
211.      //          FUNCIÓN PARA MEDIR DE PRESION         //
212.      ////////////////////////////////////////////////////
213.
214.      float read_presion(uint8_t analog) {
215.      float formula = 0.0, adc = 0.0;
216.      adc = analogRead(analog);
217.      formula = adc * 3.3 / 1023;
218.      formula = (37.5 * formula) - 18.75;
219.      if (formula < 0) {
220.      formula = 0.0;
221.      }
222.      if (formula > 100) {
223.      PWR.setSensorPower(SENS_5V, SENS_OFF);
224.      USB.print(F("Presión Excesiva"));
225.      }
226.      return formula;
227.      }
228.
229.      ////////////////////////////////////////////////////
230.      //              FUNCIÓN DE ENVÍO DE DATOS         //
231.      ////////////////////////////////////////////////////
232.      void enviar_datos()
233.      {
234.      USB.println(F("***** SEND DATA *****"));
235.      xbeeDM.ON();
236.      //xbeeDM.wake();
237.      delay(2000);
238.
```

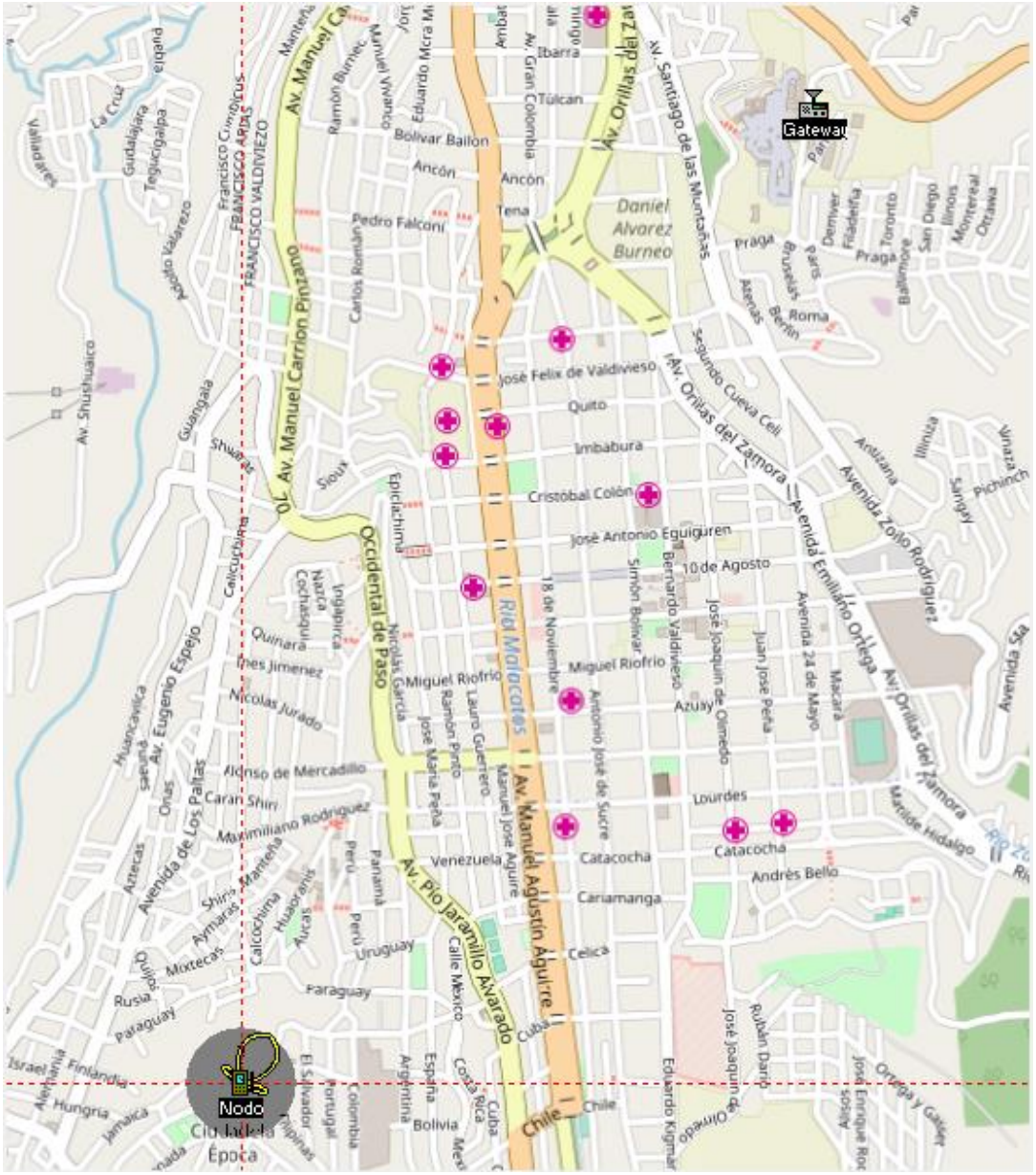
```
239.      // Create new frame
240.      body.createBody();
241.      body.addField( PWR.getBatteryLevel() );
242.      body.addField( presion,1 );
243.      body.addField( presion2 ,1);
244.      body.addField( mini,1);
245.      body.addField(maxi,1);
246.      body.addField(prespro,1);
247.      body.addField( nivel );
248.      body.addField( flujo ,1);
249.      body.addStatus( str );
250.      body.showBody();
251.      //  USB.println(F(sequenceNumber));
252.      uint8_t n = 0;
253.      while (n < 4) {
254.      error = xbeeDM.send( RX_ADDRESS, body.buffer, body.length);
255.      // check TX flag
256.      if ( error == 0 )
257.      {
258.      USB.println(F("send ok"));
259.      break;
260.      }
261.      else
262.      {
263.      USB.println(F("send error"));
264.      delay(200);
265.      n++;
266.      }
267.      }
268.      xbeeDM.OFF();
269.      // Reinicio de variables
270.      }
```

ANEXO E

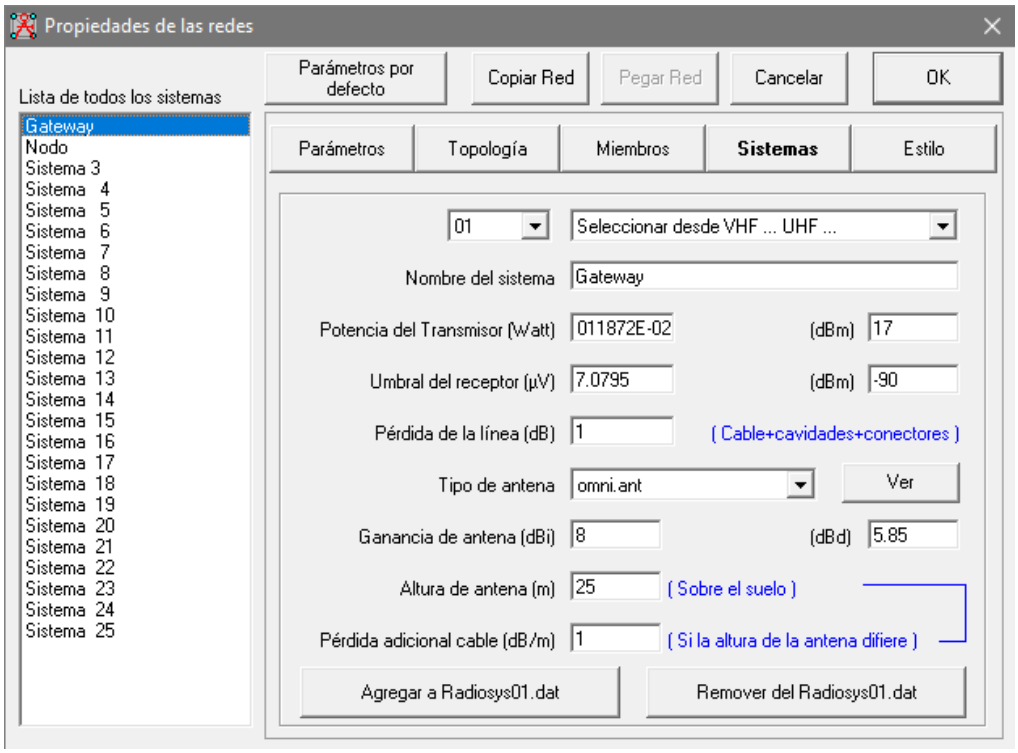
SIMULACIÓN EN RADIO MOBILE

Simulación del enlace entre el nodo ubicado en la ciudadela Época y el gateway ubicado en la UTPL.

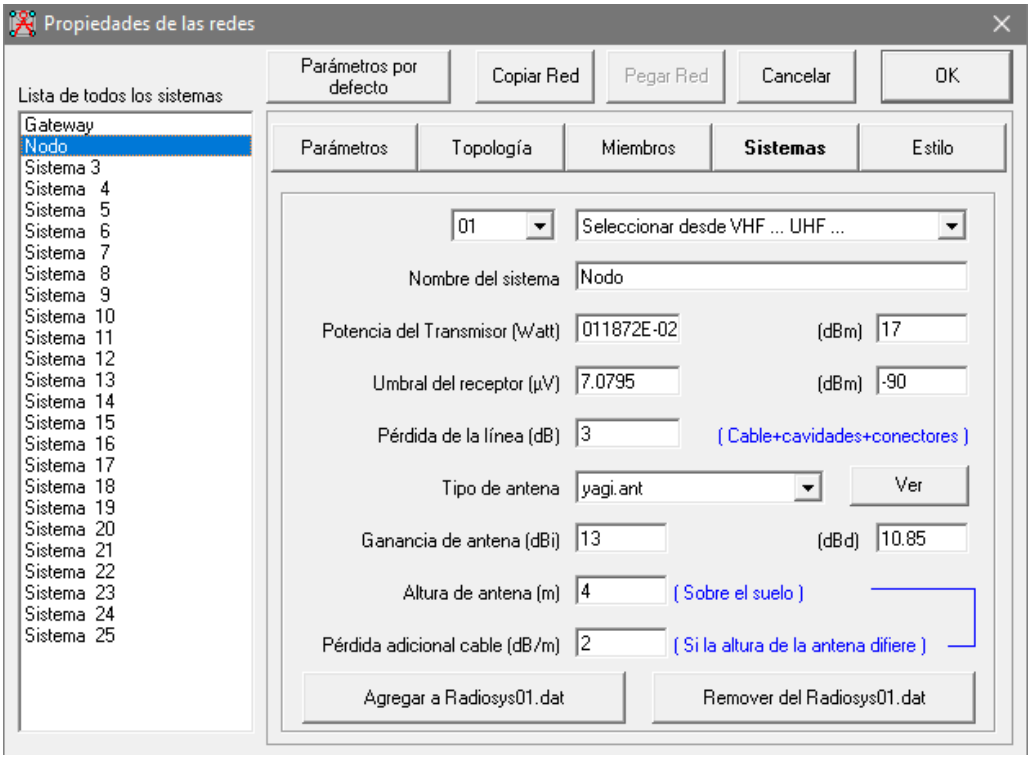
Ubicación geográfica de los puntos.



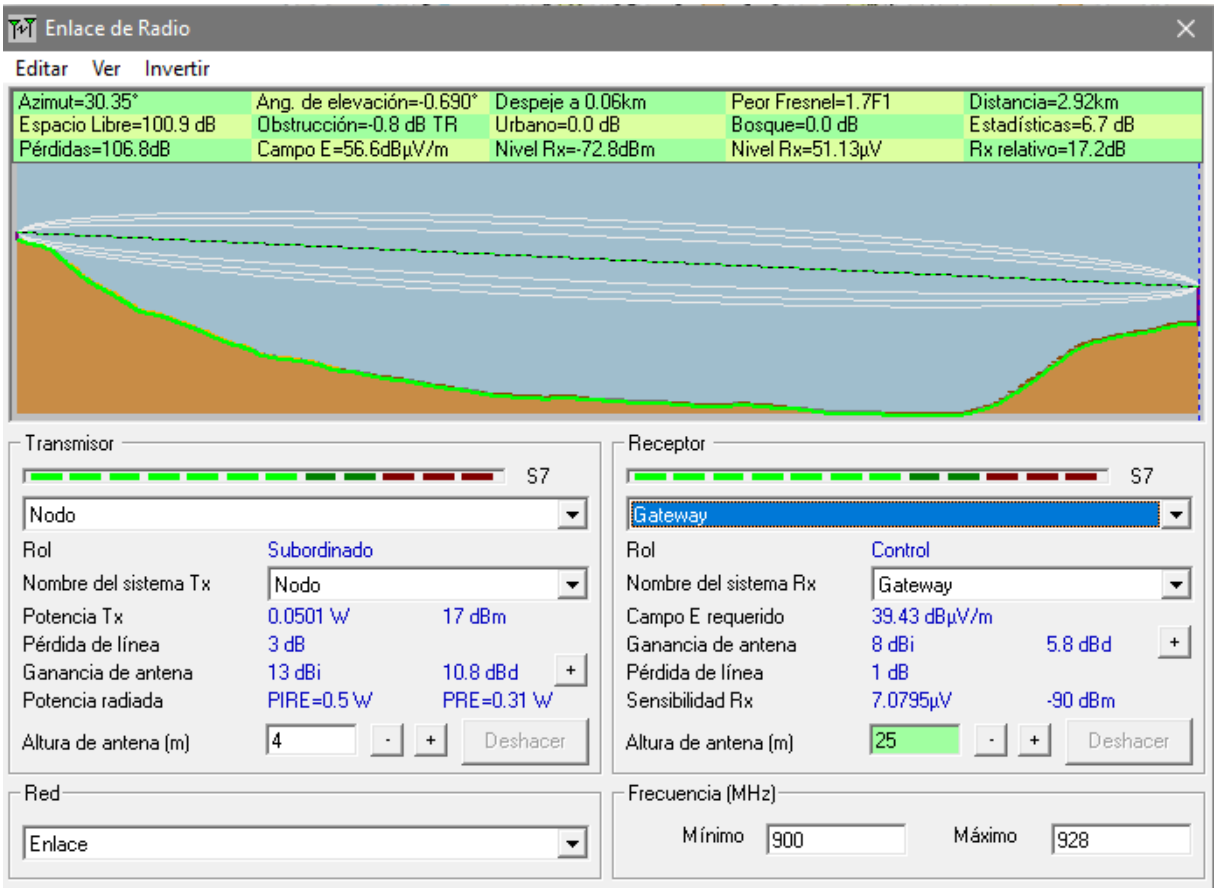
Propiedades del sistema ubicado en la UTPL.



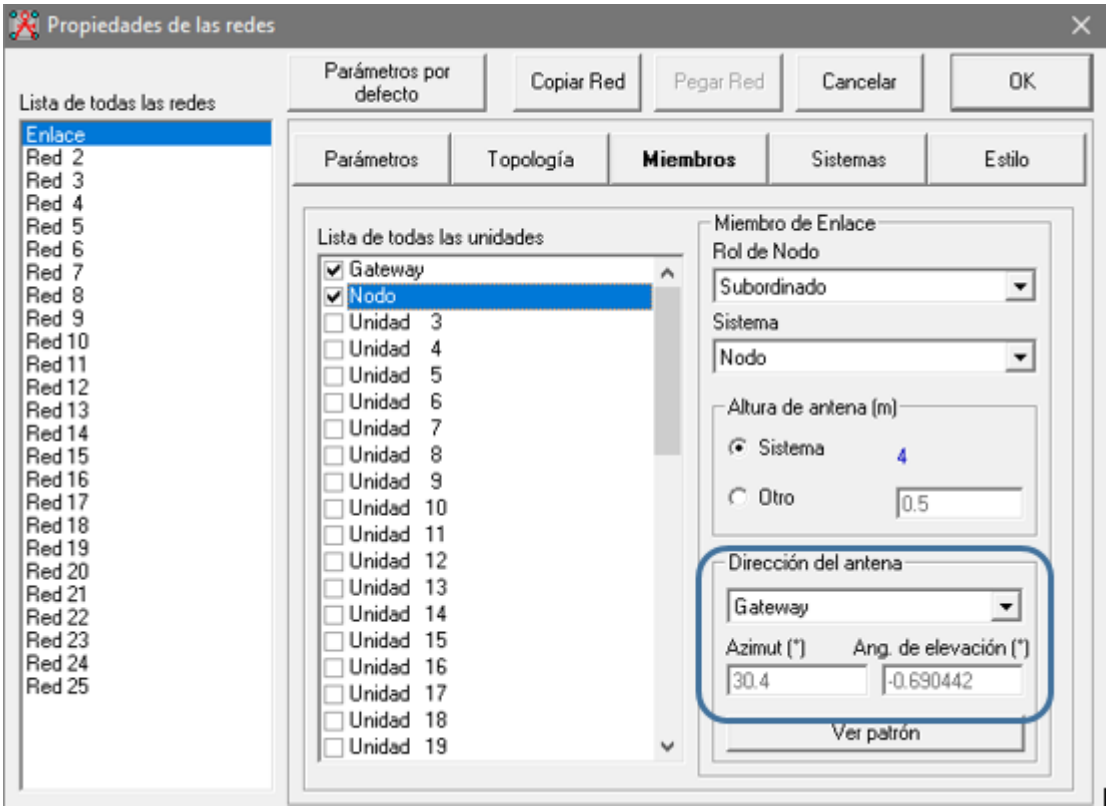
Propiedades del sistema ubicado en la ciudadela Época.



Resultados del enlace de radio.



Parámetros para la alineación de la antena


Propiedades de las redes
Lista de todas las redes
Enlace
Red 2
Red 3
Red 4
Red 5
Red 6
Red 7
Red 8
Red 9
Red 10
Red 11
Red 12
Red 13
Red 14
Red 15
Red 16
Red 17
Red 18
Red 19
Red 20
Red 21
Red 22
Red 23
Red 24
Red 25
Parámetros por defecto
Copiar Red
Pegar Red
Cancelar
OK
Parámetros
Topología
Miembros
Sistemas
Estilo
Lista de todas las unidades
Gateway
Nodo
Unidad 3
Unidad 4
Unidad 5
Unidad 6
Unidad 7
Unidad 8
Unidad 9
Unidad 10
Unidad 11
Unidad 12
Unidad 13
Unidad 14
Unidad 15
Unidad 16
Unidad 17
Unidad 18
Unidad 19
Miembro de Enlace
Rol de Nodo
Subordinado
Sistema
Nodo
Altura de antena (m)
Sistema
4
Otro
0.5
Dirección del antena
Gateway
Azimut (°)
Ang. de elevación (°)
30.4
-0.690442
Ver patrón