

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Presentado por:

ROGELIO ROMANÍ FLORES

**DISEÑO DE UN PROTOTIPO PARA EL MONITOREO
DEL CONSUMO EFICIENTE DE AGUA EN UNA
INSTITUCIÓN EDUCATIVA**

Asesor:

M.Sc. CARLOS YINMEL CASTRO BULEJE

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

ANDAHUAYLAS – APURÍMAC - PERÚ

2019



APROBACIÓN DEL ASESOR

Quién suscribe:

M.Sc. CARLOS YINMEL CASTRO BULEJE por la presente:

CERTIFICA,

Que, el Bachiller en Ingeniería de Sistemas **ROGELIO ROMANÍ FLORES** ha culminado satisfactoriamente el informe final de tesis intitulado: "**DISEÑO DE UN PROTOTIPO PARA EL MONITOREO DEL CONSUMO EFICIENTE DE AGUA EN UNA INSTITUCIÓN EDUCATIVA**" para optar el Título Profesional de **INGENIERO DE SISTEMAS**.

Andahuaylas, 05 de Agosto de 2019.

M.Sc. Carlos Yinmel Castro Buleje
Asesor

Bach. Rogelio Romani Flores
Tesisista



DECLARACIÓN JURADA DE AUTENTICIDAD

Yo, Rogelio Romaní Flores, identificado con DNI N° 47152776 de la Escuela Profesional de Ingeniería de Sistemas.

Declaro bajo juramento que el informe final de tesis intitulado: DISEÑO DE UN PROTOTIPO PARA EL MONITOREO DEL CONSUMO EFICIENTE DE AGUA EN UNA INSTITUCIÓN EDUCATIVA.

Es auténtico y no vulnera los derechos de autor. Además, su contenido es de entera responsabilidad del autor (es) del informe final de tesis, quedando la UNAJMA exenta de toda responsabilidad en caso de atentar contra la Ley de propiedad intelectual y derechos de autor.

Andahuaylas, 05 de Agosto de 2019.


.....
Firma
N° DNI: 47152776
E-mail: rromaflores@gmail.com
N° Celular: 995 843 378



FACULTAD DE INGENIERÍA

**ACTA DE SUSTENTACIÓN
DE TESIS**

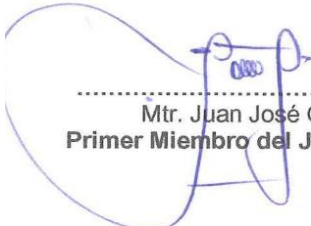
En la Av. José María Arguedas del Local Académico SL01 (Ccoyahuacho) en el auditorio de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional José María Arguedas ubicado en el distrito de San Jerónimo de la Provincia de Andahuaylas, siendo las 07:00 horas del día 23 de julio del año 2019, se reunieron los docentes: Dra. Norma Lorena Catacora Flores, Mtr. Juan José Oré Cerrón, Mg. Humberto Silvera Reynaga, en condición de integrantes del Jurado Evaluador del Informe Final de Tesis intitulado: "DISEÑO DE UN PROTOTIPO PARA EL MONITOREO DEL CONSUMO EFICIENTE DE AGUA EN UNA INSTITUCIÓN EDUCATIVA", cuyo autor es el Bachiller en Ingeniería de Sistemas **ROGELIO ROMANÍ FLORES**, el asesor MSc. Carlos Yínmel Castro Buleje, con el propósito de proceder a la sustentación y defensa de dicha tesis.

Luego de la sustentación y defensa de la tesis, el Jurado Evaluador **ACORDÓ: APROBAR** por **MAYORÍA** al Bachiller en Ingeniería de Sistemas **ROGELIO ROMANÍ FLORES**, obteniendo la siguiente calificación y mención:

Nota escala vigesimal		Mención
Números	Letras	
14	catorce	Bueno

En señal de conformidad, se procedió a la firma de la presente acta en 03 ejemplares.


.....
Dra. Norma Lorena Catacora Flores
Presidente del Jurado Evaluador


.....
Mtr. Juan José Oré Cerrón
Primer Miembro del Jurado Evaluador


.....
Mg. Humberto Silvera Reynaga
Segundo Miembro del Jurado Evaluador



APROBACIÓN DEL JURADO DICTAMINADOR

LA TESIS: DISEÑO DE UN PROTOTIPO PARA EL MONITOREO DEL CONSUMO EFICIENTE DE AGUA EN UNA INSTITUCIÓN EDUCATIVA; para optar el Título Profesional de Ingeniero de Sistemas, ha sido evaluada por el Jurado Dictaminador conformado por:

PRESIDENTE: Dra. Norma Lorena Catacora Flores

PRIMER MIEMBRO: Mtr. Juan José Oré Cerrón

SEGUNDO MIEMBRO: Mg. Humberto Silvera Reynaga

Habiendo sido aprobado por MAYORIA, en la ciudad de Andahuaylas el día 23 del mes de Julio de 2019.


Andahuaylas, 06 de Agosto de 2019.



Dra. Norma Lorena Catacora Flores
PRESIDENTE DEL JURADO DICTAMINADOR



Mtr. Juan José Oré Cerrón
PRIMER MIEMBRO DEL JURADO DICTAMINADOR



Mg. Humberto Silvera Reynaga
SEGUNDO MIEMBRO DEL JURADO DICTAMINADOR

DEDICATORIA

Para ti mamá que fuiste mi apoyo incondicional durante mi formación personal y profesional. Es admirable tu coraje y valentía porque supiste sobreponerte a las adversidades por darnos lo mejor a tus hijos. Y no me alcanzará las palabras, ni la forma de expresar por todo lo que has hecho por mí. Solamente me aferro agradecerte eternamente y darte lo mejor de mí, que Dios te guarde por siempre.

Rogelio Romani

AGRADECIMIENTOS

Me complace agradecer al creador divino por su amor y bendición de permitir lograr mi objetivo.

Agradezco a mi madre, a mi abuelita María y todos mis hermanos(as), quienes fueron mi apoyo incondicional a lo largo de mi formación personal y profesional. Porque atreves de sus consejos y palabras de ánimo hicieron este sueño realidad, siempre de una u otra forma están acompañándome en todos mis metas. Por ello solo me queda agradecerles siempre estaré con ustedes los amo FAMILIA.

Mi agradecimiento a la Universidad Nacional José María Arguedas, a la Escuela Profesional de Ingeniería de Sistemas y en ella, al M.Sc. Carlos Yinmel Castro Buleje como asesor de la tesis por su comprensión y apoyo incondicional; y a todos los demás distinguidos ingenieros por su loable labor como docentes formadores de profesionales que mi brindaron un valioso aporte para mi formación profesional, lo que me permite constituirme como persona elemental para la sociedad.

De igual forma agradecer a la Institución Educativa Industrial Federico Villareal por permitirme de información, espacios y elementos para el desarrollo del presente trabajo de tesis por ello extendiendo mi gratitud de agradecimiento.

Para iniciar una meta, hay que ponerle voluntad y compromiso. Para lograrlo el éxito es la perseverancia.

Gracias a todos que, sumando sus esfuerzos, se logró el objetivo.!!!

ÍNDICE GENERAL

APROBACIÓN DEL ASESOR.....	ii
DECLARACIÓN JURADA DE AUTENTICIDAD.....	iii
ACTADE SUSTENTACIÓN DE TESIS.....	iv
APROBACIÓN DEL JURADO DICTAMINADOR.....	v
DEDICATORIA.....	vi
AGRADECIMIENTOS.....	vii
RESUMEN.....	xv
ABSTRACT.....	xvi
CHUMASQA.....	xvii
INTRODUCCIÓN.....	1
CAPÍTULO I. PROBLEMA DE INVESTIGACIÓN.....	3
1.1. Realidad problemática.....	3
1.2. Formulación del problema.....	5
1.3. Objetivos.....	5
1.3.1. Objetivo general.....	5
1.3.2. Objetivos específicos.....	5
1.4. Justificación.....	6
1.5. Viabilidad de la investigación.....	6
1.5.1. Viabilidad técnica.....	6
1.5.2. Viabilidad económica.....	7
1.5.3. Viabilidad social.....	7
1.6. Limitación del estudio.....	7
CAPÍTULO II. ANTECEDENTES.....	8
2.1. Antecedentes nacionales.....	8
2.2. Antecedentes internacionales.....	9
CAPÍTULO III. MARCO TEÓRICO.....	14
3.1. Prototipo.....	14
3.2. El agua.....	14
3.2.1. Agua potable.....	14
3.3. Monitoreo eficiente del agua.....	15
3.4. Medidores convencionales de recursos hídricos.....	15
3.4.1. Desplazamiento o volumétricos.....	16
3.4.2. De turbina o velocidad.....	17
3.5. Medidores de agua electrónicos.....	18

3.5.1.	De efecto Hall.....	18
3.6.	Arduino	18
3.6.1.	Arduino uno	19
3.7.	Tecnologías inalámbricas.....	20
3.7.1.	Bluetooth	20
3.7.2.	Wi-Fi	20
3.7.3.	Wi-Máx	20
3.8.	Estándar IEEE 802.15.4.....	20
3.9.	Estándar Zigbee	21
3.9.1.	Topología Zigbee.....	23
3.10.	Estándar propietario Digimesh	26
3.10.1.	Nodos	27
3.11.	Cuadro comparativo entre Zigbee Y Digimesh.....	27
3.12.	Xbee	29
3.12.1.	Series de módulos Xbee	31
3.12.2.	Modos de operación de los módulos Xbee	33
3.13.	Programa para configurar Xbee.....	33
3.14.	Sensor	34
3.14.1.	Sensor de flujo	35
3.15.	Base de datos	37
3.15.1.	MySQL	37
3.16.	Lenguaje de programación	37
3.16.1.	PHP.....	37
3.17.	Metodología ágil programación extrema (XP)	38
3.17.1.	Fases de la programación extrema	38
CAPÍTULO IV. METODOLOGÍA DE LA INVESTIGACIÓN		42
4.1.	Planificación	42
4.1.1.	Requisitos funcionales.....	43
4.1.2.	Requisitos no funcionales	50
4.1.3.	Roles del proyecto.....	51
4.1.4.	Plan de iteraciones y entregas	51
4.1.5.	Requerimientos del prototipo.....	51
4.2.	Diseño	52
4.2.1.	Primera iteración.....	52
4.2.2.	Segunda iteración.....	55
4.2.3.	Tercera iteración.....	56

4.2.4. Cuarta iteración.....	58
4.3. Desarrollo.....	59
4.3.1. Primera iteración.....	59
4.3.2. Segunda iteración.....	77
4.3.3. Tercera iteración.....	78
4.3.4. Cuarta iteración.....	84
4.4. Pruebas.....	86
4.5. Puesta en marcha del prototipo	92
CAPÍTULO V. RESULTADOS.....	96
CAPÍTULO VI. DISCUSIÓN.....	100
CONCLUSIONES.....	102
RECOMENDACIONES.....	103
REFERENCIAS BIBLIOGRÁFICAS	104
ANEXOS	108

ÍNDICE DE TABLAS

Tabla 1: Características de Arduino Uno.....	19
Tabla 2: Diferencias entre ZigBee y DigiMesh	28
Tabla 3: Descripción de pines de los módulos Xbee y Xbee pro	30
Tabla 4: Serie 1 versus Serie 2 para XBees regulares	32
Tabla 5: Características del sensor de flujo YS-S201	36
Tabla 6: Historia 01 del cliente	43
Tabla 7: Historia 02 del cliente	44
Tabla 8: Historia 03 del cliente	44
Tabla 9. Historia 04 del cliente	44
Tabla 10. Historia 05 del cliente.....	44
Tabla 11. Requerimientos no funcionales.....	50
Tabla 12: Roles del Proyecto.....	51
Tabla 13: Plan de entregas del proyecto.....	51
Tabla 14: Requerimiento de hardware y software.....	52
Tabla 15. Parámetros de configuración del nodo coordinador.....	60
Tabla 16. Parámetros de configuración del nodo sensor.....	62
Tabla 17. Estructura de la trama de petición.....	64
Tabla 18. Estructura de la trama de envío de datos.....	66
Tabla 19: Resultados de calibración del sensor para el nodo 01	68
Tabla 20: Prueba de aceptación de medir el consumo de agua.....	86
Tabla 21: Prueba de aceptación de solicitar datos de la medición.....	87
Tabla 22: Prueba de aceptación de transmitir medición.....	87
Tabla 23: Prueba de aceptación de registrar medición en la base de datos.....	88
Tabla 24: Prueba de aceptación de administrar usuarios.....	89
Tabla 25: Prueba de aceptación de administrar sensores.....	90
Tabla 26: Prueba de aceptación de administrar nodos.....	91

ÍNDICE DE FIGURAS

Figura 1: Historial de consumo por hora y semanal de agua.....	13
Figura 2: Medidor de disco oscilante.....	16
Figura 3: Medidor de pistón oscilante.	16
Figura 4: Medidor de turbina para chorro único.	17
Figura 5: Medidor de turbina para chorro múltiple	17
Figura 6: Arduino Uno.....	19
Figura 7: Capas de protocolo de red inalámbrica Zigbee.	21
Figura 8: Aplicaciones de Zigbee.....	22
Figura 9: Conexión típica usando Xbee.....	23
Figura 10: Topología estrella.	24
Figura 11: Topología en árbol	24
Figura 12: Topología mesh.....	25
Figura 13: Nodos de red DigiMesh.....	27
Figura 14: Red Punto a Punto y Punto a Multipunto.	29
Figura 15: Módulo XBee XB24-Z7WIT-004 de Digi.....	30
Figura 16: Conexiones mínimas del XBEE	31
Figura 17: Programa XCTU.....	34
Figura 18: Funcionamiento del sensor de caudal.	35
Figura 19: Sensor de flujo YF-S201.....	36
Figura 20: Proceso XP.	42
Figura 21: Módulos emisor y receptor.....	43
Figura 22: Diagrama de casos de uso.	45
Figura 23: Diagrama de objetos.....	45
Figura 24: Diagrama de secuencia registrar medición.....	46
Figura 25: Diagrama de secuencia visualizar consumo.....	47
Figura 26: Diagrama de secuencia administrar usuarios.....	48
Figura 27: Diagrama de secuencia administrar sensor.....	49
Figura 28: Diagrama de secuencia administrar nodo.....	50
Figura 29: Conexión del nodo sensor.	53
Figura 30: Conexión esquemático del nodo sensor.....	53
Figura 31: Conexión del nodo coordinador.	54
Figura 32. Conexión esquemática del nodo coordinador.	55
Figura 33. Esquema de la base de datos.....	55
Figura 34. Interfaz general del sistema web.	56
Figura 35. Interfaz de inicio de sesión.	57
Figura 36. Interfaz de registro de nodos.....	57
Figura 37. Interfaz de reportes.	58
Figura 38. Topología Física de la red.	59
Figura 39. Conexión del Xbee al ordenador.	60
Figura 40. Actualización del firmware del nodo coordinador.....	61
Figura 41. Registro de parámetros del nodo coordinador en el software XCTU.	61
Figura 42. Actualización del firmware del nodo sensor.....	62
Figura 43. Registro de parámetros del nodo sensor en el software XCTU.....	63
Figura 44: Prueba de comunicación entre el coordinador y el nodo sensor.....	63
Figura 45. Generación de trama API en el nodo coordinador.....	64

Figura 46. Datos en la estructura de la trama del nodo sensor.	67
Figura 47. Código ASCII.	72
Figura 48: Prueba generar trama de petición.	73
Figura 49: Prueba enviar trama de petición.	74
Figura 50: Prueba de respuesta de la trama de petición.	74
Figura 51: Prueba del envío de datos al servidor de base de datos.	76
Figura 52. Interfaz del servidor local XAMPP	77
Figura 53. Base de datos con las respectivas tablas	77
Figura 54. Arquitectura MVC para el sistema web.....	78
Figura 55. Interfaz de desarrollo Sublime Text.....	79
Figura 56. Página de inicio de sesión.	79
Figura 57. Sección de visualización del consumo de agua.....	80
Figura 58: Ventanas emergentes de eliminar y restaurar.....	81
Figura 59: Formulario de registro y/o modificación de usuarios.	81
Figura 60: Formulario de registro y/o modificación de sensores.	82
Figura 61: Formulario de registro y/o modificación de nodos.	83
Figura 62: Sección de reportes del consumo de agua.....	84
Figura 63. Topología lógica de la red.....	85
Figura 64: Prueba del enlace entre el nodo coordinador y el nodo sensor.....	86
Figura 65: Prueba de petición y transmisión de datos.	88
Figura 66: Prueba de registro y almacenamiento de datos.....	89
Figura 67: Instalación del sensor de flujo.	92
Figura 68: Nodo 01 instalado.	93
Figura 69: Nodo coordinador instalado.....	94
Figura 70: Conexión del nodo coordinador con el servidor de BD.....	95
Figura 71: Visualización del monitoreo del consumo de agua en tiempo real.	95
Figura 72: Prototipo desarrollado.....	96
Figura 73: Base de datos desarrollado.	96
Figura 74: Red inalámbrica ZigBee implementado.....	97
Figura 75: Sistema web local desarrollado.	97
Figura 76: Reporte del consumo de agua, abril 2019.	98
Figura 77: Incidente 05 de Abril del 2019.....	99
Figura 78: Incidente 30 de Abril del 2019.....	99

ÍNDICE DE CÓDIGOS

Código 1: Código para la trama de petición.....	66
Código 2: Código de trama para el nodo sensor.....	68
Código 3: Código para obtener los pulsos del sensor.....	69
Código 4: Código para medir el flujo de agua.	70
Código 5: Código para la escucha de la petición del coordinador.	70
Código 6: Código para el envío de datos del nodo sensor.	71
Código 7: Código para la recuperación de los datos del nodo sensor.	72
Código 8: Código para el envío de datos al servidor.	75
Código 9: Código del controlador para registrar datos en la base de datos.....	76

RESUMEN

En el presente trabajo de tesis se desarrolló un prototipo para el monitoreo del consumo eficiente de agua en la Institución Educativa Industrial Federico Villareal, el cual se instaló en la tubería principal que conecta con diferentes grifos de agua de la Institución Educativa, de tal forma que cuando se abre un grifo se calcule con exactitud el total de litros de agua que se consumieron durante la apertura del grifo, el desarrollo del prototipo ofrece una mejor medición y monitoreo constante del consumo. Por otro lado, la Institución Educativa resalta su imagen sobre otras Instituciones mostrando su cultura en el cuidado del recurso más importante para la vida.

Como parte fundamental en el diseño del prototipo se consideró el estudio de las diferentes redes inalámbricas y se optó en utilizar los dispositivos Xbee basado en el estándar Zigbee, por sus características de bajo costo y consumo de energía. Con los dispositivos Xbee se realizó la interconexión inalámbrica del nodo sensor instalado en la tubería principal que conecta los grifos de agua con el nodo coordinador para la captura y procesamiento de datos. Finalmente, el nodo coordinador envía los datos capturados a la base de datos para que el usuario, por medio del sistema web pueda monitorear el consumo de agua en tiempo real.

El hardware del prototipo consta del diseño de un módulo emisor y otro módulo receptor utilizando el microcontrolador Arduino uno, sensor de flujo y los dispositivos Xbee. Donde el módulo emisor es el encargado de medir el consumo de agua y transmitirlo al módulo receptor el cual está conectado directamente a un servidor local para el almacenamiento de los datos. Finalmente se desarrolló un sistema web utilizando el lenguaje de programación en PHP para el procesamiento de los datos y la visualización del mismo.

El trabajo de tesis se realizó con enfoque tecnológico, ya que se aplicaron los conocimientos adquiridos en el desarrollo tecnológico del prototipo. El cual ofrece automatizar un proceso que en la actualidad no es posible realizar con facilidad, como es el caso de monitorear el consumo del agua en la Institución Educativa Industrial Federico Villareal.

Palabras clave: Prototipo, consumo de agua, Xbee, Zigbee, nodo, monitoreo.

ABSTRACT

In this thesis work a prototype was developed for the monitoring of efficient water consumption in the Federico Villareal Industrial Educational Institution, which was installed in the main pipeline that connects with different water taps of the Educational Institution, so that When a tap is opened, the total liters of water consumed during the tap opening are calculated exactly, the prototype development offers a better measurement and constant consumption monitoring. On the other hand, the Educational Institution highlights its image on other Institutions showing its culture in the care of the most important resource for life.

As a fundamental part in the design of the prototype, the study of the different wireless networks was considered and it was decided to use Xbee devices based on the Zigbee standard, due to their low cost and energy consumption characteristics. With the Xbee devices, the wireless interconnection of the sensor node installed in the main pipeline that connects the water taps with the coordinating node for data capture and processing was performed. Finally, the coordinating node sends the captured data to the database so that the user, through the web system, can monitor the water consumption in real time.

The prototype hardware consists of the design of a transmitter module and another receiver module using the Arduino Uno microcontroller, flow sensor and Xbee devices. Where the sending module is responsible for measuring water consumption and transmitting it to the receiving module which is directly connected to a local server for data storage. Finally, a web system was developed using the programming language in PHP for data processing and visualization.

The thesis work was carried out with a technological approach, since the knowledge acquired in the technological development of the prototype was applied. Which offers to automate a process that is currently not possible to perform easily, as is the case of monitoring water consumption in the Federico Villareal Industrial Educational Institution.

Key words: Prototype, water demand, Xbee, Zigbee, node, monitoring.

CHUMASQA

Kay kunan investigación qillqa ruwayninwanmi qispichisqa kachkan huk prototipo nisqan sutiyuq, astawanqa kay prototipom yanapan yaku yupayta kay Institución Educativa Industrial Federico Villareal yachay wasipi runakuna yakuta pilamanta orqoptinku, kay prototipoqa tinkisqam kachkan chay tubería yaku apaq nisqanpa chaupinpi chaylla yakuta horqoptinku yupananpaq. Chay yupasqanta runakuna qawaspan yachananpaq qaykatas yakuta orquchkanku chayta, chaymanqina allinllata yakuta orqonankupaq qinaspa astawanqa masta yakuta cuidanankupaq.

Chaynallataq kay prototipo nisqan ruwaynimpí ñawinchasqa karqa qayllas wayrantakama información apaq tecnología nisqan, chay lliuchanmanta allkasqa karun Xbee nisqan sutiyuq tecnología. Astawanqa kay akllanapaq qawarisqa karqa pisi qollqillapaq valorniyuq inaspapas mana ancha energía eléctrica nisqanta tukunampaq. Kay Xbee tecnología nisqanwanmi wayantakama apachin chay prototipopa yaku yupasqa informacionta huk prototipo umalliq nisqankama. Chaymantataq kay prototipo umalliq chayachin chay información chaskisqanta huk computadora personal nisqankama, chay computadorapiñataq runa qawachkan qaykatas yakuta orqochkanku kunan pacha.

Kay prototipo nisqan ruwaypiqa aswanqa iskay módulom ruwasqa kachkan, chay modulokunapam sutinmi hukninpa emisor nisqan hukninpañataqmi receptor nisqan. Kay modulokunaqa ruwakurqa micro-controlador arduino uno, sensor de flujo y el radio Xbee nisqankunawanmi. Chaynapi kay modulo emisor nisqan yakuta yupanampaq hinaspa apachinqa chay yupasqanta modulo receptor nisqankama, kay modulo receptor nisqantaqmi sapa chaskisqanta waqaychan huk computadora personal nisqanpi. Kay computadorapa ukumpiñataqmi churasqa kachkan huk sistema web nisqan, payñataqmi chay yaku yupasqa waqaychasqata sumaqta ñawinchachin chaylla runakuna qawayta munaptin.

Kay investigación qellqam ruwakun chay enfoque tecnológico nisqanman hina, chayman hinam lliw yachaykuna churasqa karun kay prototipo nisqan qispinanpaq.

Chaymi kay prototipo kunanqa yakuta yupanqa kay Institución Educativa Industrial Federico Villareal yachay wasipi.

Rimaykuna kichaq: Prototipo, consumo de agua, Xbee, Zigbee, nodo, monitoreo.

INTRODUCCIÓN

El sustento principal de la vida de todos los seres vivos en el planeta es el AGUA, prácticamente el uso de este recurso está relacionado en todas las actividades que realiza el hombre, el hombre bajo la dependencia del agua, es consciente en su uso eficiente. La realidad indica que no son conscientes en el uso, a raíz de esta inconciencia del hombre el agua en la actualidad parece estar escanciando, sin embargo, la demanda del agua crece exponencialmente.

En la actualidad el problema del agua es común en los hogares, empresas, Instituciones Educativas, donde no se sabe con exactitud cuánto es el consumo de agua; por lo que esta información solamente lo manejan las empresas suministradoras y muchas veces el recibo del consumo no refleja lo consumido, debido a que hay fugas o goteos de agua no controlados. Para realizar el monitoreo y obtener la información exacta del consumo es necesario el diseño de sistemas que permitan hacer la medición y el almacenamiento de datos. A través del monitoreo, se puede ver en tiempo real la evolución del consumo instantáneo. La Institución Educativa Industrial Federico Villareal también no es ajeno al problema, por ende, surge la idea de adaptar un prototipo para el monitoreo del consumo eficiente de agua en dicha Institución Educativa. Esta adaptabilidad de soluciones tecnológicas permite no solamente el monitoreo sino también le permite a la Institución saber con exactitud cómo y cuándo se consume el agua. Teniendo las informaciones estadísticas se puede tomar decisiones más efectivas para un ahorro de consumo.

El presente trabajo de tesis está basado en una tecnología emergente llamada red de sensores inalámbricos, que cada vez tiene mucha relevancia por las grandes posibilidades de desarrollo que ofrece en cualquier área donde se quiera implementar. Esta tendencia tecnológica está enfocada en integrar: procesos, personas, objetos, datos; de una organización e incluso para el monitoreo de los objetos en los hogares es posible la automatización mediante el uso de sensores y comunicaciones inalámbricas, además son ideales por su bajo costo y facilidad de implementación, sobre todo por que convierten aspectos físicos del entorno en señales eléctricas que las computadoras pueden procesar. Las tecnologías inalámbricas permiten utilizar sensores en ambientes hostiles o de difícil acceso para medir los datos que con las tecnologías tradicionales sería casi imposible obtener.

El trabajo de tesis consta en el diseño de un prototipo utilizando microcontrolador Arduino Uno, sensor de flujo y un radio Xbee, donde el Xbee es un módulo de comunicación inalámbrica que permite enviar los datos leídos por el sensor al sistema principal. Mientras tanto el sistema consiste en una interfaz creada en PHP para la visualización de los datos,

además se empleó como gestor de base datos MySQL para el almacenamiento de los datos, la interfaz visual está ligada al sistema de comunicación inalámbrica que opera bajo el protocolo de comunicación ZigBee.

La metodología usada para la administración y desarrollo de la tesis fue la Programación Extrema, por su característica de agilidad y flexibilidad permitió el desarrollo a base de la retroalimentación continua en cada fase del proyecto.

La estructura adecuada del presente trabajo de tesis, se constituye de la siguiente manera:

En el capítulo 1 se fundamenta la realidad problemática, la formulación del problema, el objetivo general, los objetivos específicos, justificación de la tesis, viabilidad y limitaciones de la tesis. El capítulo 2 describe los antecedentes del trabajo de tesis a nivel nacional e internacional, teniendo en cuenta la aplicación de los sensores y las redes inalámbricas de bajo costo en los proyectos antecesores. El capítulo 3 presenta la base teórica que sustenta la aplicación y desarrollo del trabajo de tesis a base del conocimiento.

En el capítulo 4 se explica la metodología de la investigación y su desarrollo fue según la metodóloga ágil de programación extrema, el cual ofrece agilidad al trabajo de tesis desarrollándose iterativamente bajo las cuatro fases que comprende la metodología de programación extrema y además ofrece la realimentación continua en cada fase para garantizar el desarrollo y el éxito del trabajo de tesis. El capítulo 5 presenta los resultados obtenidos y el capítulo 6 está relacionado a la discusión de los resultados.

Se realizó la comprobación del cumplimiento de los objetivos y se concluye según los resultados obtenidos en la ejecución del trabajo de tesis. En seguida, se presentan las recomendaciones necesarias para el avance en el futuro del trabajo de tesis. También se presenta los recursos bibliográficos utilizados en el planteamiento y desarrollo del trabajo de tesis, finalmente se presentan los anexos.

CAPÍTULO I. PROBLEMA DE INVESTIGACIÓN

1.1. Realidad problemática

Desde su génesis de los seres vivos, el recurso indispensable que guarda su supervivencia es el AGUA. Este elemento permite desarrollar la imaginación de las personas ofreciendo la oportunidad de crecer, desarrollarse, y aprovechar las sustancias que se producen en la naturaleza.

Nadie se imagina como sería la vida sin agua, creo que ninguno de los seres vivos sobreviviría. Por lo tanto, las personas deben asumir el compromiso de cuidar su uso, fomentar esta disciplina a las futuras generaciones ya que solo así se puede salvar que el agua se seque por completo.

Hoy en día muchas personas no tienen una cultura sobre el agua, al respecto la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO), 2016) menciona que la cultura del agua es necesario porque toda operación se articula en su uso del agua, la manifestación de la cultura del agua está relacionado en tener valores; en las normas y formas de organización; en las practicas tecnológicas. Además de esto depende sobre una cultura colectiva que comparten las personas de acuerdo al nivel socioeconómico que se encuentre. Cabe mencionar que las personas que viven en las zonas alto andinas valoran más el agua a través de sus costumbres. Y según el primer principio de la Ley 29338 Ley de Recursos Hídricos en el Perú (Congreso de la República, 2010), prescribe la valoración del agua y de gestión integral del agua, enunciando que, el agua tiene valor sociocultural, valor económico y valor ambiental, por lo tanto para garantizar el valor del agua se debe tener una gestión integrada por parte de las empresas suministradoras y el uso adecuado del mismo, para así tener el equilibrio entre éstos.

(Sevilla, 2015) el Perú se encuentra entre los diez países con mayores reservas de agua en el mundo, cuenta con 3 fuentes vertientes hidrográficas: vertiente del Atlántico (genera 98,2% de los recursos hídricos), vertiente del Pacífico (1,5% de los recursos hídricos) y vertiente del Titicaca (0,3%). Sin embargo, a pesar de la aparente abundancia de agua, el Perú se encuentra entre los cinco países más vulnerables al cambio climático. En cuanto al consumo del agua en el Perú, el sector agrario tiene mayor consumo del agua seguido por el sector poblacional.

Con frecuencia se escucha sobre dos de los problemas más comunes del mundo actual que es del agua y del ambiente, es casi evidente y coincide con los problemas

actuales que se vive. Por ejemplo es posible vivir sin celular, estar incomunicados, o incluso pasar meses sin comer sólido, pero no se puede pasar sin líquidos más de 72 horas (Agencia CTyS, 2014). En la actualidad existen muchos conflictos acerca de los escasos del agua, esto a veces explica sobre el aumento de la población. Sin embargo, no es el caso porque el supuesto escasez está relacionado a la inadecuada gestión hídrica, cultura del agua, etc. Según el Boletín Virtual de la Superintendencia Nacional de Servicios de Saneamiento (Superintendencia Nacional de Servicios de Saneamiento (SUNASS), 2016) informa que cada hogar limeño sin acceso al agua potable se ahorraría s/ 724 soles al año si estuviera conectado a la red de SEDAPAL. Mientras tanto, hay un importante sector de la población que derrocha agua durante los meses de verano, especialmente en la época de carnavales. Por lo que refleja que aún no hay práctica del uso eficiente del agua en los hogares. Otro de los problemas según noticias del BBC (2015) casi dos millones de personas se mueren al año por falta de agua potable. Y es probable que en 15 años la mitad de la población mundial viva en áreas en las que no habrá suficiente agua para todos. Actualmente se consume al año el 54% del agua dulce disponible. A mediados del siglo actual la población mundial alcanzará los 12 millones de habitantes, la demanda se habrá duplicado y las reservas hídricas del planeta llegaran a su tope (Superintendencia Nacional de Servicios de Saneamiento (SUNASS), 2016). Sobre estos problemas muchos de los organismos mundiales han advertido que, si la humanidad no realiza cambios importantes en sus usos y costumbres, en el año 2025 existirán muchas zonas de Latinoamérica (Perú, Chile), África y Asia, entre otros en los que se desencadenarán agudos conflictos por la escasez del agua (Autoridad Nacional del Agua (ANA), 2011).

El presente trabajo de tesis se desarrolló en la Institución Educativa Industrial Federico Villareal que se encuentra ubicado en el Jirón Ayacucho s/n del distrito de Andahuaylas de la Provincia de Andahuaylas del Departamento de Apurímac. Es una Institución Educativa estatal que brinda la educación en los niveles de primaria y secundaria. Actualmente cuenta con 36 docentes, 16 administrativos, 610 estudiantes. Sumando un total de 662 de población educativa.

Durante mucho tiempo y hasta la actualidad el deficiente monitoreo del consumo de agua es uno de los problemas no identificados en la Institución Educativa, la primera causal que origina el problema es debido a la falta de cultura que tienen los docentes, estudiantes y personal administrativo sobre la valoración del agua, muchas veces la gran mayoría de los estudiantes juegan con el agua o dejan semiabierto el grifo.

Otra causal es la falta de mecanismos tecnológicos que no ayuda en el monitoreo permanente de las instalaciones como: baños, lavadero de manos para detectar si hay alguna fuga del agua o los cerrojos tanto del inodoro como de los grifos no cierran bien el flujo de agua. Actualmente la Institución Educativa cuenta con un medidor tradicional del consumo de agua administrado por la empresa Emsap Chanka SCRL, la empresa realiza la lectura del medidor mensualmente para emitir la factura correspondiente a la Institución Educativa. El detalle que contiene la factura sobre el consumo de agua es de forma general correspondiente al mes, por lo que la Institución Educativa desconoce la información del consumo diario del agua que realiza. Por otra parte, el costo del consumo de agua es asumido por la Unidad de Gestión Educativa Local Andahuaylas a través de los fondos del estado. La comunidad Educativa al no asumir el costo del consumo no toma el interés en hacer un uso más eficiente ni la preservación de este recurso tan importante para la comunidad educativa.

Las consecuencias que genera el deficiente monitoreo del consumo de agua es: desperdicio y mayor costo del consumo, pero la consecuencia más crítica será los escases del agua que a futuro no muy lejano tendrá efecto en la comunidad educativa, En particular hoy en día en muchos centros educativos no tienen un monitoreo eficiente del consumo de agua por lo que hay desconocimiento en el uso. Es oportuno de tomar iniciativas para controlar el uso innecesario del recurso más importante para la vida. Por ello en el presente trabajo de tesis se diseñó un prototipo para el monitoreo del consumo eficiente de agua. De tal forma que la Institución Educativa pueda ver el flujo de consumo de agua diaria, mensual de manera gráfica y los detalles del consumo para tomar acciones en el momento oportuno cuando se presente una fuga o desperdicio del agua innecesaria, además de concientizar a los estudiantes sobre el cuidado del recurso hídrico.

1.2. Formulación del problema

Deficiente monitoreo del consumo de agua en la Institución Educativa Industrial Federico Villareal.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un prototipo que ayude en el monitoreo constante del consumo eficiente de agua en la Institución Educativa Industrial Federico Villareal.

1.3.2. Objetivos específicos

- ✓ Diseñar un prototipo para la medición y monitoreo del consumo de agua.

- ✓ Desarrollar una base de datos para almacenar la información del consumo de agua.
- ✓ Implementar la red inalámbrica Zigbee como el medio de transmisión de datos del prototipo a la base de datos.
- ✓ Desarrollar un sistema web local para la visualización del consumo de agua.

1.4. Justificación

Siendo un problema común para la sociedad el tema del agua que es una cuestión latente por resolver. Y al respecto muchos de los organismos mundiales, personas dedicadas a la investigación según sus estudios han advertido que si no se toma acciones sobre valorar el agua desde los hogares hasta las empresas encargadas de administrar el recurso hídrico en el mundo tendrán consecuencias a futuro de que el agua se seque gota a gota.

Por esta razón el presente trabajo de tesis cobra importancia en su desarrollo para contribuir un mecanismo tecnológico de monitoreo eficiente del consumo de agua en tiempo real en la Institución Educativa Industrial Federico Villareal, el cual permite ver el consumo del agua, registrando en el sistema web para que la I.E sepa sobre su consumo real. Por lo que beneficia a la Institución Educativa no solo en el monitoreo para ahorrar el agua sino también resalta una imagen institucional con el buen uso del agua respecto a los demás colegios, además la Institución Educativa está multiplicando el deber ciudadano sobre llevar una cultura del agua, inculcando a los estudiantes.

En el presente trabajo de tesis se empleó la tecnología Zigbee para formar una red inalámbrica de área personal, ya que fue el protocolo más adecuado en el desarrollo del trabajo de tesis que permite la transmisión inalámbrica de datos entre los nodos del prototipo a la base de datos.

1.5. Viabilidad de la investigación

1.5.1. Viabilidad técnica

Es viable técnicamente porque la adquisición de las herramientas de hardware es posible ya que se cuenta con los materiales requeridos disponible en el mercado listo para ser adquirido, de la misma manera las herramientas de software cuyas licencias de uso también están disponibles y son de uso gratuito.

1.5.2. Viabilidad económica

El desarrollo del presente trabajo de tesis es viable económicamente, ya que fue asumido en su totalidad por el autor que presenta el trabajo de tesis para la adquisición de los materiales requeridos.

1.5.3. Viabilidad social

La Institución Educativa Industrial Federico Villareal, es el beneficiario directo con el desarrollo del presente trabajo de tesis.

1.6. Limitación del estudio

El siguiente trabajo de tesis se limita en los siguientes:

- ✓ Algunas zonas del espacio geográfico limitan el alcance de los módulos de radio enlace como obstrucción de línea de vista, disminuyendo la señal de transmisión de datos. Para garantizar el envío de los datos al 100% se adquirió la cantidad necesaria de los dispositivos Xbee y luego instalar en el intermedio entre el nodo sensor y el nodo coordinador.
- ✓ La energización del nodo sensor y el nodo coordinador es una limitante puesto que los componentes de los nodos funcionan con voltajes menores a 5v. Por lo que las pilas son primeras opciones para energizar a los nodos, sin embargo, las pilas no tienen la carga suficiente para garantizar el tiempo de vida de los nodos. Para el proyecto se optó en utilizar los cargadores de Arduino con regulador de energía de corriente alterna a corriente de 5v y se alimentó con la energía comercial de la I.E.I Federico Villareal lugar donde se ejecutó el trabajo de tesis.

CAPÍTULO II. ANTECEDENTES

Revisando los antecedentes sobre el presente tema de investigación, se obtuvo avances tecnológicos. Como en algunos países Ecuador, Colombia vienen desarrollando nuevas herramientas tecnológicas que ayudan en la monitorización del consumo de agua potable. Por consiguiente, se detalla los estudios y avances nacionales, internacionales debido a su importancia que tiene el presente trabajo de tesis en esta área de investigación.

2.1. Antecedentes nacionales

Los autores Pimentel & Vislao (2014) de la Universidad Privada Antenor Orrego, Trujillo Perú, hicieron un “Estudio de la mejora en el sistema de lectura de medidores eléctricos de la división de agua potable y energía eléctrica del proyecto especial Chavimochic mediante una solución inalámbrica” en donde el objetivo de tesis fue:

Elaborar un estudio con la estimación de mejoras que podrían ser introducidas con la aplicación de una solución inalámbrica al sistema de lectura de datos en medidores eléctricos de la División de Agua Potable y Energía Eléctrica del Proyecto Especial CHAVIMOCHIC para mejorar la toma de datos de los medidores de energía eléctrica en la ciudad de Trujillo mediante una solución inalámbrica, tomando como alternativa la tecnología de GPRS, para el envío de datos a un servidor central de la empresa supervisora. Por su parte la metodología de solución explica:

Los medidores eléctricos instalados cuentan con opciones avanzadas, los cuales permite agregar nuevo hardware (tarjetas dedicadas). Por lo que se aprovecha esta posibilidad agregando una tarjeta RS-232 para la comunicación entre este y un modem externo con conexión GPRS. También se emplean el uso de tarjetas de comunicaciones seriales, módems celulares y servicios adicionales para la integración del sistema propuesto.

El estudio concluye estimando posibles mejoras para la empresa y el usuario, observándose que la solución inalámbrica mejoró tiempos de lectura y costos asociados. Con este tipo de solución, la toma de lecturas en los medidores eléctricos fue más exacta y ya no se tiene un error por defecto o exceso, como cuando se realizaba la toma de lecturas de manera manual.

Los autores Huamanyalli & Rúa (2014) de la Universidad Nacional de Huancavelica, realizaron la investigación sobre “Sistema remoto en red multipunto para las medidas de consumo de energía eléctrica domiciliaria”. El objetivo general de la tesis fue:

Realizar un prototipo del sistema de medición remota en red multipunto que adquiera la medida del consumo de energía eléctrica domiciliaria para su facturación correcta. La solución está basada en obtener, almacenar y transmitir los datos de la medición a una estación central para su posterior procesamiento y facturación correspondiente.

Para el módulo del medidor los autores utilizaron el microcontrolador PIC16F628A, el cual realiza la lectura del medidor digital por el puerto IrDA, obteniendo el consumo de energía actual en kwh, la guarda en una memoria y transmite al ser solicitado mediante una comunicación inalámbrica de Radio Frecuencia con modulación para el caso transmisor y con demodulación en el caso receptor a una frecuencia de 315 MHz, velocidad entre la estación central y los módulos remotos del medidor.

Mientras la parte central está diseñado a través del programa LabView 2012, es el encargado de solicitar los datos remotamente a un microcontrolador PIC18F2550 conectado con interface USB para comunicarse con un pc, obtener y procesarlos la información de los medidores.

En conclusión, frente a las necesidades comunes que tiene una sociedad, los autores tienen la solución como este, que con pocos requerimientos logran el objetivo de diseñar un módulo que no solamente beneficia a un lugar en estudio. Más aun tiende abrir nuevas posibilidades a futuras investigaciones como es el caso del presente proyecto que sigue al problema similar en la Institución Educativa Industrial Federico Villareal.

2.2. Antecedentes internacionales

Los autores Arévalo & Rodríguez (2016) de la Universidad Politécnica Salesiana, con el proyecto de investigación "Implementación de un sistema de supervisión y monitorización del consumo de energía eléctrica y agua potable, utilizando redes Het-Net para la transmisión de datos; con la finalidad de obtener información oportuna para una eficiente facturación y disminuir perdidas en la dotación de los servicios", plantearon como objetivo:

Implementar un sistema de supervisión y monitorización del consumo de energía eléctrica y agua potable, utilizando redes Het-Net para la transmisión de datos; con la finalidad de obtener información oportuna para una eficiente facturación y disminuir perdidas en la dotación de los servicios.

De acuerdo a su marco metodológico el proyecto busca dar solución al problema de medición tradicional del consumo de agua potable y energía eléctrica implementando

una red de infraestructura avanzada. Para lo cual utilizaron los dispositivos principales que forman parte de este tipo de redes:

- ✓ **Medidores inteligentes:** Para la medición de energía utiliza el sensor ACS712, el cual puede medir el consumo de energía watts-hora y para la medición de agua se usa el sensor de flujo YF-S201 de tipo efecto hall, el cual tiene la capacidad de medir litros por minuto.
- ✓ **Medios de transmisión GPRS y WIFI:** Como se trata de redes heterogéneas el proyecto emplea dos tipos de medios de transmisión, con la posibilidad de determinar cuál de ellos es mejor.
- ✓ **Protocolo MQTT (bróker y clientes):** Este protocolo es el encargado de recibir mensajes los medios de transmisión y conectarse a la base de datos.
- ✓ **Servidor central (cliente MQTT):** Encargado de la persistencia de registros de consumo de la base de datos
- ✓ **Base de datos (mongo DB):** En donde se almacena los registros de la lectura por parte de los medidores remotos.

Además de la construcción del medidor inteligente los autores también desarrollaron un sistema para aprovechar que los datos almacenados sean útiles tanto para la empresa como para los usuarios de energía eléctrica y agua potable.

Después de la implementación los autores obtuvieron como resultados: de la cobertura y el alcance de la red GPRS y Wi-Fi, el mejor entre estos dos fue de la red GPRS que permite mejor comunicación entre el medidor y el bróker central, a grandes distancias con una latencia mínima.

En conclusión, los autores sobresalen con el logro de la implementación de la red de medidores inteligentes. En donde el proceso de automatizar a gran escala la distribución y monitoreo del consumo de energía y agua potable, permite a las empresas suministradoras brindar un servicio de calidad y garantizar la conservación de los recursos a largo plazo.

En Colombia, los autores Guacaneme & Pardo (2016) de la Universidad Distrital Francisco José de Caldas, presentaron la tesis titulado "Diseño e implementación de un sistema de medición de consumo de energía eléctrica y agua potable remoto con interacción al usuario basado en el concepto internet de las cosas", donde plantearon como objetivo general:

Diseñar e implementar un sistema de medición de consumo de energía eléctrica y agua potable remoto con interacción al usuario basado en el concepto "Internet de las

cosas", que permita tomar las lecturas del consumo doméstico de energía y agua potable y transmitir a través de la comunicación inalámbrica Wi-Fi a un servidor en la nube para que el usuario tenga acceso a la información desde cualquier lugar. El proyecto está basado en el concepto de Internet de las Cosas, en el cual los autores tienen la finalidad de contribuir a los habitantes de una vivienda, que tengan acceso y control de su medidor de agua y energía eléctrica.

En cuanto a su metodología el proyecto consistió en diseñar tanto la parte de hardware y software requerimientos utilizados fueron:

a. Elementos para el hardware

Para el medidor de agua:

- ✓ Microcontrolador MCF51JM128-VLH.
- ✓ Memoria de almacenamiento EEPROM 24LC512.
- ✓ Sensor de efecto hall, etc.

Para el medidor de energía:

- ✓ Microcontrolador MCF51JM128-VLH.
- ✓ Sensor de energía ACS714.
- ✓ Memoria de almacenamiento EEPROM 24LC512.

Y para la comunicación y transporte de los datos se utiliza el dispositivo ESP8266EX, el cual soporta el tipo de red inalámbrica Wi-Fi.

b. Elementos para el software

- ✓ Almacenamiento de datos (MYSQL).
- ✓ Servicios de gestión de conexión y comunicación de dispositivos (Websocket).
- ✓ Servicios de consulta y procesamiento de datos (Node js).
- ✓ Aplicación a nivel de usuario (Bootstrap).

Finalmente, el proyecto desarrollado por los autores ha logrado el diseño e implementación de un sistema de medición de consumo de energía eléctrica y agua potable de una vivienda en particular. Donde el usuario es directamente beneficiado para interactuar desde cualquier lugar usando un dispositivo inteligente con características disímiles de hardware y software, a través del cual pueda observar su consumo diario tanto de energía como de agua.

En Ecuador el autor Chuquimarca (2014) de la Universidad Nacional de Loja, a través de su tesis "Diseño e implementación del prototipo de un sistema domótico para la

medición del consumo de agua potable a través de internet y correo electrónico”, el objetivo general fue:

Diseñar e implementar un sistema domótico que permita monitorear el consumo de agua potable a través de correo electrónico y acceso Web en una vivienda de la ciudad de Loja, donde el sistema consta de un módulo principal como Raspberry Pi, el cual es el encargado de procesar la información recibida a través del sensor de flujo y luego enviar a un servidor en internet que le proporciona acceso al usuario en una página web.

En cuanto al desarrollo de la tesis el autor diseño e implemento el hardware y software para su correcto funcionamiento utilizando los siguientes requerimientos:

a. Hardware

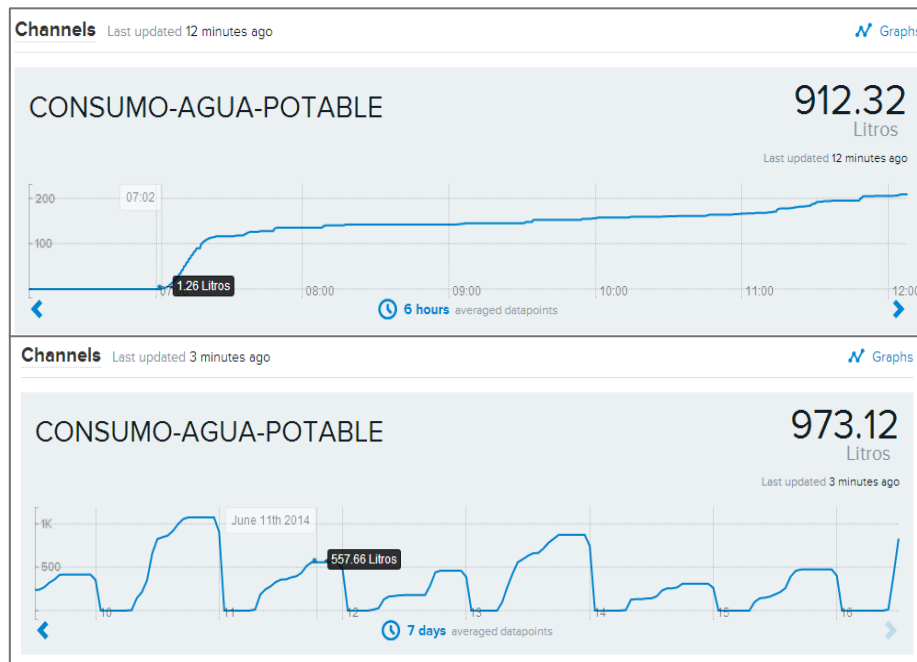
- ✓ Sensor de flujo de efecto hall.
- ✓ Raspberry Pi. como modulo principal.
- ✓ una pantalla LCD tipo resistiva.

b. Software

- ✓ Sistema operativo Linux usando la distribución Raspbian.
- ✓ Lenguaje de programación C/C++

El modulo desarrollado por el autor funciona ininterrumpidamente las 24 horas del día, siete días a la semana, excepto cuando la conexión a internet no existe o por falta de energía que alimente al sistema. Pero no obstante los datos leídos del medidor no se pierden cuando no hay conexión a internet. Cuando se restablece los datos no enviados se vuelven a enviar, lo que si ocasiona perdida de datos es cuando hay corte de energía en el sistema.

Mediante la plataforma Xively desarrollado por el autor se puede apreciar en la Figura 1 los litros de agua consumida durante una hora, en días, semanal y así el usuario puede consultar sobre su historial de consumo a la base de datos.



*Figura 1: Historial de consumo por hora y semanal de agua.
Fuente: (Chuquimarca, 2014)*

Finalmente, el autor concluye de forma satisfactoria con el desarrollo del proyecto de medición y monitoreo de consumo de agua potable residencial, donde el usuario cuenta con la información apropiada en el tiempo y espacio que lo desee. Esto a diferencia de los medidores tradicionales que tienen muchas desventajas al momento de obtener información. Esta investigación permite desarrollar tecnologías nuevas no solo para el campo residencial sino también en campos industriales.

Según Cázarez, y otros (2016) mediante su revista titulado “Sistema de tele medición de consumo de agua potable para aplicaciones residenciales y comerciales”. Plantearon diseñar e implementar un sistema electrónico enfocado en el monitoreo remoto del consumo de agua potable.

El proyecto consistió en diseñar una tarjeta electrónica, la cual incluye un microcontrolador Atmel 32U4 con bootloader Arduino y un módulo de comunicación Xbee con soporte para protocolo de comunicación en malla, además incluye un sensor de flujo de líquidos de tipo propela basado en efecto Hall, el cual puede medir consumos de hasta 30 litros por minuto e implementa un código de integración numérica para llevar control de lecturas de consumo acumulado y reportes por fecha y hora.

En la revista los autores concluyen con el desarrollo y pruebas del proyecto que permite automatizar el proceso de la medición del consumo de agua en las zonas residenciales y comerciales, empleando los requerimientos a menor costo y en poco tiempo.

CAPÍTULO III. MARCO TEÓRICO

3.1. Prototipo

Es un pre-diseño que sirve como modelo para la versión final de un producto que se desea obtener. Con los prototipos se realizan las simulaciones y pruebas hasta lograr que el producto final tenga los funcionamientos y requisitos deseados.

Los prototipos son representaciones de un diseño realizado antes de que existan los productos finales. Proporcionan los medios para examinar problemas de diseño y evaluar soluciones para su buen desempeño en el ámbito donde se va implementar, después de todas las pruebas se diseña el producto final tomando como modelo el prototipo. (Buchenau & Fulton Suri, 2000)

Los prototipos son ampliamente reconocidos como un medio central de exploración y expresión de diseños interactivos. Es una práctica común construir prototipos para representar diferentes estados de un diseño en evolución y explorar opciones. Sin embargo, dado que los sistemas interactivos son complejos, puede ser difícil o imposible crear prototipos de un diseño completo en las etapas formativas de un proyecto. (Houde & Hill, 1997)

3.2. El agua

Según la Real Academia Española, el agua (del latín aqua) es la sustancia formada por la combinación de un volumen de oxígeno y dos de hidrógeno, líquida, inodora, insípida, en pequeña cantidad incolora y verdosa o azulada en grandes masas. Es considerada como uno de los recursos naturales más fundamentales para el desarrollo de la vida, y junto con el aire, la tierra y la energía, constituye los cuatro recursos básicos en que se apoya el desarrollo de todos los seres vivos.

3.2.1. Agua potable

El autor Harper (2004) conceptualiza el servicio de agua potable como el agua que no tiene impurezas presentes en cantidad suficientes que motiven enfermedades u otros efectos en la comunidad, haciendo hincapié que calidad química y bacteriológica debe estar de acuerdo con las disposiciones normativas que rigen.

Según la Superintendencia Nacional de Servicios de Saneamiento (SUNASS, 2004) el agua potable, también llamada agua para consumo humano, es aquella que llega al consumidor y puede usarse de manera segura para beber, cocinar los alimentos y realizar la higiene personal. El tratamiento de potabilización del agua consiste en una serie de procesos llevados a cabo con el fin de eliminar las impurezas de ésta. En los tratamientos convencionales, los procesos de floculación, sedimentación y

filtración, destinados a clarificar el agua y eliminar su carga bacteriana, son las operaciones más caras e importantes de todo el procedimiento.

Es un derecho humano de primer orden, es un agua tratada según estándares de calidad.

3.3. Monitoreo eficiente del agua

El monitoreo eficiente del agua plantea cuantificar el uso del agua y emprender acciones que nos permitan generar un cambio en el comportamiento del uso hacia la eficiencia. El cual consiste en identificar el estado de instalaciones hidrosanitarias, inspeccionando si se presentan fugas, el estado en que se encuentran los sistemas ahorradores, la necesidad de su calibración y la presencia de daños en las unidades sanitarias (Alcaldía Mayor de Bogotá D.C, 2017).

La definición de monitoreo eficiente del agua implica a toda actividad que esté relacionada con utilizar el recurso de una mejor manera, hacer más o lo mismo con menos cantidad. También, se deben tomar medidas que permitan usar menos agua en cualquier proceso o actividad para la conservación y el mejoramiento de los recursos hídricos (Sánchez & Sánchez, 2004).

3.4. Medidores convencionales de recursos hídricos

Un medidor de agua es un dispositivo que, instalado en una tubería, permite conocer el flujo volumétrico (m^3/s) o caudal que está circulando por la misma. Los medidores tradicionales que se usan en la mayoría de hogares son de tipo mecánicos y estáticos al usuario, es decir no tienen ningún tipo de integración con las nuevas tecnologías de comunicación. Mientras tanto la evolución de los métodos de medición ha desarrollado tecnologías como sensores electromagnéticos, de efecto Hall, de efecto Doppler y otras que permiten fácilmente ser integradas a sistemas de procesamiento digital y con ello ampliar el abanico de servicios de los sensores tradicionales.

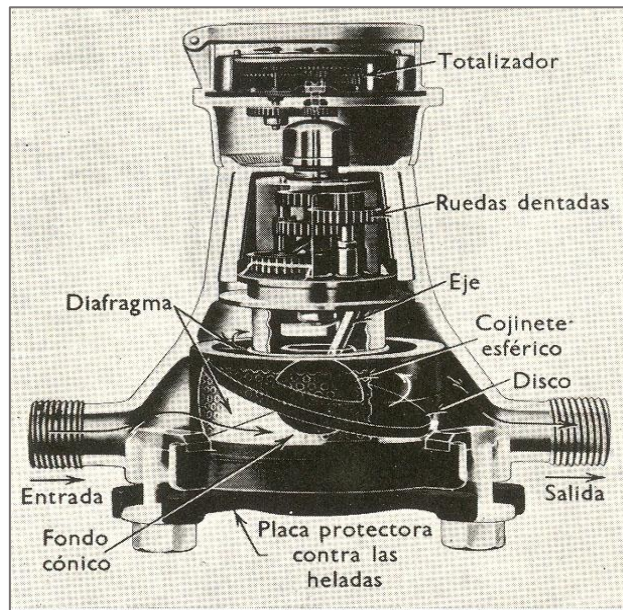
Las distintas variedades de medidores que se emplean en los sistemas de agua potable, para aforar el consumo domiciliario, se diferencian unas de otras, en los principios que han adoptado los fabricantes para su diseño y en la combinación de ellos que resulta al integrar todas sus partes. Las denominaciones con las cuales se conocen, generalmente tienen su origen, también en esos principios y combinaciones (Gómez, 1982).

Existen dos importantes grupos de medidores de agua convencionales que se detallan a continuación.

3.4.1. Desplazamiento o volumétricos

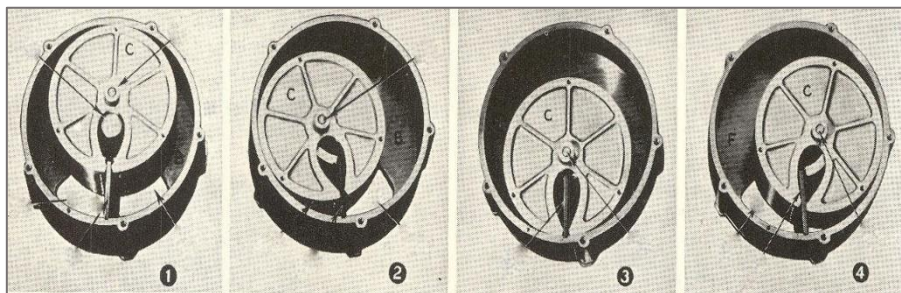
En estos medidores el gasto se determina subdividiendo la corriente total del fluido en fracciones de volumen conocido. La medición se efectúa mediante el recuento del número de fracciones en la unidad de tiempo. Los tipos de medidores por desplazamiento son: disco oscilante o nutativo y pistón oscilante.

El medidor de disco oscilante o nutativo: cuantifica la cantidad de agua cuando este penetra en el espacio entre el disco y las paredes de la cámara, empuja al disco hacia adelante imprimiéndole un movimiento de oscilación rotatoria.



*Figura 2: Medidor de disco oscilante.
Fuente: (Granger, 1965).*

Medidor de pistón oscilante: Se compone de una cámara de medida cilíndrica con una placa divisora que separa los orificios de entrada y salida. La única parte móvil es un pistón cilíndrico que oscila suavemente en un movimiento circular entre las dos caras planas de la cámara, y que está provisto de una ranura que desliza en la placa divisora fija que hace de guía del movimiento oscilante (Creus, 1997).



*Figura 3: Medidor de pistón oscilante.
Fuente: (Granger, 1965).*

3.4.2. De turbina o velocidad

Según Gómez (1982) los medidores de velocidad están constituidos todos fundamentalmente, por un rotor y un orificio, simple o compuesto que admite el agua. De acuerdo con la forma como actué esta y con la disposición de su entrada. Resultan en general cuatro tipos de medidores.

- ✓ Chorro único
- ✓ Chorro múltiple
- ✓ Woltman
- ✓ De hélice

Medidor de turbina a chorro único: Su funcionamiento es accionado cuando el agua entra al cuerpo del contador a través de una sola sección practicada a la periferia, este tipo de contador incorpora un difusor a la entrada de cámara de la turbina que permite que el agua penetre en la misma a través de diversos puntos. La velocidad de giro de la turbina depende de la velocidad de impacto del chorro de agua.

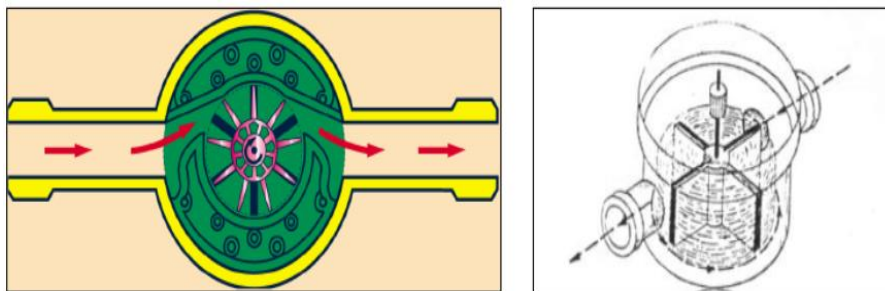


Figura 4: Medidor de turbina para chorro único.
Fuente: (Álvarez & Toledo, 2011).

Medidor de turbina a chorro múltiple: El mecanismo es accionado por varios chorros tangenciales de agua. Se distinguen del chorro único, en que la turbina está dentro de la cámara con varios orificios de entrada y salida, diametralmente opuesta.

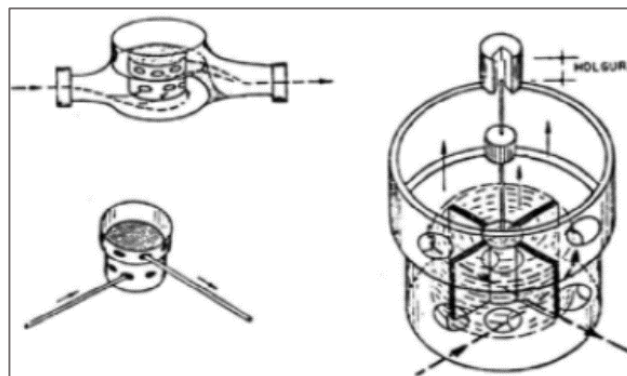


Figura 5: Medidor de turbina para chorro múltiple
Fuente: (Álvarez & Toledo, 2011).

Ambos tipos de medidores tanto chorro único y múltiple, miden el volumen de agua por la velocidad de la turbina.

3.5. Medidores de agua electrónicos

Actualmente existen medidores que están diseñadas para la tele lectura, consiste naturalmente en contabilizar el giro de los engranajes y convertirlos en señales de electrónicos de 0 y 1. Para luego que estas señales sean transmitidas hasta el sistema informático del gestor. Existen tres tipos de medidores electrónicos.

- ✓ Electromagnético.
- ✓ De efecto Hall.
- ✓ De efecto Doppler.

3.5.1. De efecto Hall

Descubierto en 1879, el fenómeno Hall, llamado así en honor al físico estadounidense Edwin Herbert Hall (Rodríguez, 2012), este efecto es la medición del voltaje transversal en un conductor cuando es puesto en un campo magnético. Mediante esta medición es posible determinar el tipo, concentración y movilidad de portadores. Es decir, el sensor de efecto Hall crea un voltaje saliente proporcional al producto de la fuerza del campo magnético y de la corriente.

Este tipo de medidores tienen un sensor que funciona a modo de switch ON/OFF. Cuando un imán pasa cerca del sensor se produce un voltaje en sus terminales.

Para el presente trabajo de tesis se usó un sensor basado en efecto hall. Por qué presenta un tipo de switch que abre y cierra cuando pasa el flujo de agua.

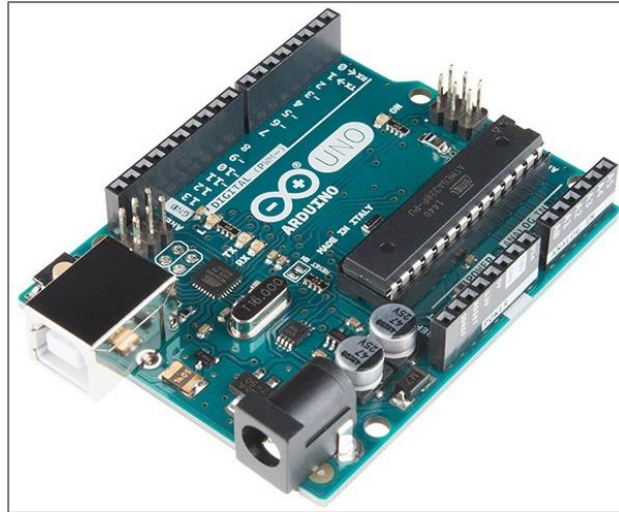
3.6. Arduino

Según Bordignon & Iglesias (2015) es un sistema electrónico de prototipado abierto, basado en software y hardware flexibles. Su versatilidad junto a un grupo amplio de componentes permite desarrollar de forma rápida (conceptos elementales de electricidad y electrónica) objetos interactivos digitales que son excelentes oportunidades de aprendizaje en niños y jóvenes, con el objetivo de potenciar su creatividad y su capacidad para resolver problemas complejos.

Físicamente, y a fines prácticos, Arduino es simplemente una placa que contiene un microprocesador ATmega, una serie de pines de entrada y salida de propósito general para datos analógicos y digitales, y una conexión USB que permite cargarle programas y establecer una comunicación con una PC. Posee además ciertas protecciones para hacer más difícil que se dañe la placa al experimentar con ella y se conecta, a través de los mencionados pines de entrada y salida, con los diversos sensores y actuadores

que permiten gran flexibilidad a la hora de crear objetos interactivos (Bordignon & Iglesias, 2015, pág. 31)

3.6.1. Arduino uno



*Figura 6: Arduino Uno.
Fuente: Elaboración propia.*

Arduino Uno es una placa basada en un microcontrolador Low Power Atmel ATMEGA 328P, el cual es de alto rendimiento, perteneciente a la familia de microcontroladores de 8 bits con CPU AVR de arquitectura RISC. Las características de dicho microcontrolador se muestran en la Tabla 1 (Hidalgo, 2015).

Tabla 1: Características de Arduino Uno

Voltaje de operación	5v
Voltaje de entrada (recomendado)	7 – 12v
Voltaje de entrada (limite)	6 – 20v
Pines I/O digitales	14 (6 de tipo PWM)
Pines de entradas analógicas	6 (sirven como digital)
Pines de salidas analógicas	0
Corriente continua por Pin I/O	40 mA.
Corriente en pin 3.3v	50 mA.
Memoria Flash	32 KB
SRAM	2KB
EEPROM	1KB
Frecuencia de reloj	16 MHz

Fuente: (Hidalgo, 2015).

Para el procesamiento de los datos leídos por los sensores se utilizó el Arduino uno. Porque tiene un menor costo y a su vez se adecua a los requerimientos del presente trabajo de tesis.

3.7. Tecnologías inalámbricas

Actualmente las redes inalámbricas tienen un gran desarrollo en el entorno de las personas, a tal punto que le facilita obtener información necesaria de manera inmediata. La comunicación inalámbrica es uno de los medios por el cual se transmite información entre un emisor y receptor.

3.7.1. Bluetooth

Bluetooth es una frecuencia de radio de disponibilidad universal que conecta entre sí los dispositivos habilitados para Bluetooth situados a una distancia de hasta 10 metros.

Por ejemplo, permite conectar una notebook o un teléfono celular con otras máquinas, teléfonos móviles, cámaras, impresoras, teclados, altavoces y otros dispositivos del ordenador (Olking, 2008) .

3.7.2. Wi-Fi

Wi-Fi o red de área local inalámbrica (WLAN) es una red de TI de tamaño medio que utiliza la frecuencia de radio 802.11a, 802.11b o 802.11g en lugar de cables y permite realizar diversas conexiones inalámbricas a Internet. Estas frecuencias fueron establecidas en 1997 por el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos). Si sabe dónde se encuentra una red Wi-Fi o WLAN, puede navegar por Internet, utilizar el correo electrónico y acceder a la red privada de una empresa. Esta es una buena opción para un empleado móvil que pasa fuera de su compañía (Olking, 2008).

3.7.3. Wi-Máx

Específicamente, la tecnología 802.16, denominada WiMAX, complementa la WLAN conectando hotspots (es una zona de cobertura Wi-Fi) con tecnología 802.11 a Internet y ofrece una alternativa inalámbrica para la conectividad de banda ancha de última generación a empresas y hogares”

Esta es una red muy costosa que aplica Microsoft verdaderamente podremos tener una banda ancha y no solo un ancho de banda, donde la velocidad de transmisión será mayor (Olking, 2008).

3.8. Estándar IEEE 802.15.4

De acuerdo al autor Farahani, (2008) es un estándar que define las capas físicas y de acceso al medio del modelo OSI para redes inalámbricas con tasas bajas de transmisión. Este estándar es la base sobre la que se define la especificación de ZigBee, también IEEE 802.15.4 se desarrolló independientemente del estándar

ZigBee, y es posible crear redes inalámbricas de corto alcance basadas únicamente en IEEE 802.15.4 y no implementar capas específicas de ZigBee. En este caso, los usuarios desarrollan su propia red aplicando las capas física y MAC del estándar IEEE 802.15.4.

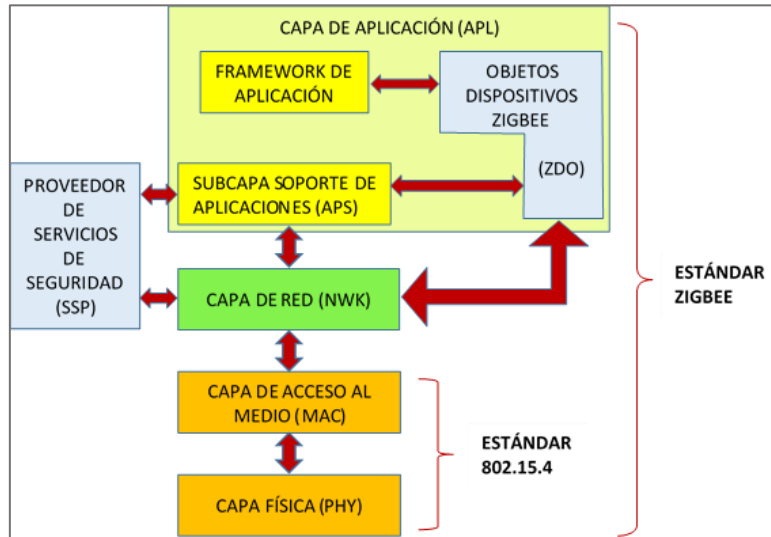


Figura 7: Capas de protocolo de red inalámbrica Zigbee.
Fuente: Elaboración propia.

El estándar IEEE 802.15.4 ofrece servicios de red, seguridad y de aplicaciones, cuyos principales requerimientos son: larga duración de las baterías, baja tasa de transmisión y bajo costo. Una de las formas comunes de establecer una red de comunicación (cableada o inalámbrica) es utilizar el concepto de capas de redes. Cada capa es responsable de ciertas funciones en red. Las capas normalmente pasan datos y comandos solo a las capas directamente arriba y debajo de ellos.

3.9. Estándar Zigbee

Zigbee de acuerdo Farahani, (2008) es un estándar que define protocolos para las capas de aplicación y red, mientras para las capas física y acceso al medio adopta del estándar IEEE 802.15.4, que define un conjunto de protocolos para la comunicación inalámbrica entre los dispositivos. Los dispositivos inalámbricos basados en Zigbee operan desde 868 MHz hasta 2.4 GHz de banda de frecuencia, la velocidad de datos máxima es de 250 Kbps. Zigbee está dirigido principalmente a aplicaciones alimentadas por batería donde baja velocidad de datos, bajo costo y larga duración de la batería son requisitos principales. El estándar ZigBee es desarrollado por ZigBee Alliance, que tiene cientos miembros de empresas, desde la industria de semiconductores y desarrolladores de software hasta el original de fabricantes de equipos (OEM) e instaladores. La Alianza ZigBee se formó en 2002 como una organización sin fines de lucro abierta a todos los que quieran unirse.

El concepto de usar la comunicación inalámbrica para recopilar información o realizar ciertas tareas de control y monitoreo dentro de una casa o una fábrica no son nuevas, debido a que existe varios estándares de redes inalámbricas de corto alcance, cada uno de los estándares tiene sus ventajas en particular. El estándar Zigbee esta específicamente desarrollado para abordar la necesidad de un costo muy bajo, baja velocidad de datos, conexiones inalámbricas con ultra bajo consumo de energía. La muy baja tasa de datos de ZigBee significa que no es la mejor opción para implementar una conexión inalámbrica a Internet o un auricular inalámbrico con calidad de CD donde más de 1Mbps es deseado. Sin embargo, si el objetivo de la comunicación inalámbrica es transmitir y recibir comandos y/o recopilar información de sensores como temperatura o humedad, ZigBee proporciona la mayor potencia y la solución más rentable en comparación con otros estándares. Las ventajas que tiene el estándar Zigbee son:

- ✓ Bajo costo.
- ✓ Ultra-bajo consumo de potencia.
- ✓ Uso de bandas de radio libres y sin necesidad de licencias.
- ✓ Instalación barata y simple.
- ✓ Redes flexibles y extensibles.

Zigbee puede ser usado en innumerables sistemas de aplicaciones orientadas a automatizar un proceso o la recolección de datos de los sensores, como se puede visualizar en la Figura 8, y en el presente trabajo de tesis se aprovechó su utilidad con fines de monitoreo eficiente del consumo de agua en la Institución Educativa Industrial Federico Villareal.



Figura 8: Aplicaciones de Zigbee.
Fuente: (Barneda, 2008).

El uso del protocolo Zigbee va desde reemplazar un cable por una comunicación serial inalámbrico, hasta el desarrollo de configuraciones punto a punto, multipunto, peer-to-

peer (todos los nodos conectados entre sí) o redes complejas de sensores. Una conexión típica se muestra en la Figura 9, donde se observa que cada módulo Xbee posee algún tipo de sensor, el cual entrega los datos para ser enviados a través de la red a un centro que administre la información.

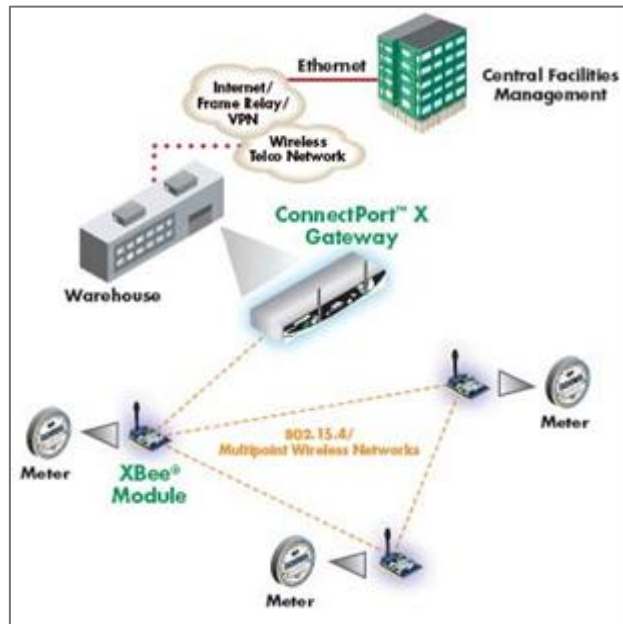


Figura 9: Conexión típica usando Xbee.
Fuente: (Oyarce, Aguayo, & Martin, 2008).

3.9.1. Topología Zigbee

Actualmente hay tres topologías que soporta la red Zigbee como: estrella, árbol y malla, cada una de estas topologías tienen sus ventajas a la hora de implementar y configurar. Además, independientemente de la topología en, todas las redes de protocolo ZigBee cada nodo permite igual acceso a la red. Por lo tanto, una red ZigBee en malla se considera una red de acceso múltiple de un solo salto. Llevando el concepto de acceso múltiple al otro par de redes. (Eady, 2007).

a. Topología en estrella.

La configuración de la topología en estrella está compuesta por un dispositivo coordinador al cual están conectados los dispositivos finales como se muestra en la Figura 10. Para transmitir un mensaje entre dos dispositivos finales, el mensaje debe ser enrutado únicamente por el coordinador. En esta topología se puede implementar hasta 6000 dispositivos. Pero con la desventaja de que, en algún momento, el coordinador deja de trabajar por algún problema técnico, entonces toda la red se cae, ya que todo el tráfico pasa por el coordinador de la estrella. (Eady, 2007).

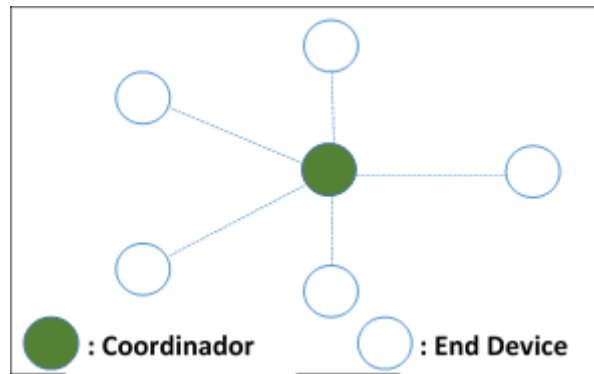


Figura 10: Topología estrella.
Fuente: Elaboración propia.

b. Topología en árbol

Una red de árbol, como se muestra en la Figura 11, parte de un dispositivo coordinador de la red ZigBee que actúa como raíz del árbol. Un coordinador o enrutador de ZigBee puede actuar como dispositivo principal y aceptar asociación desde otros dispositivos en la red. Un dispositivo conectado a un dispositivo principal es conocido como dispositivo infantil. Los mensajes destinados a un niño se pueden enrutar a través de su padre. Un dispositivo final ZigBee puede actuar solo como un niño porque carece de la capacidad de enrutamiento. La topología de árbol también se conoce como topología jerárquica. Si cualquier router no funciona, entonces sus hijos (dispositivos finales) también dejan de trabajar. (Farahani, 2008).

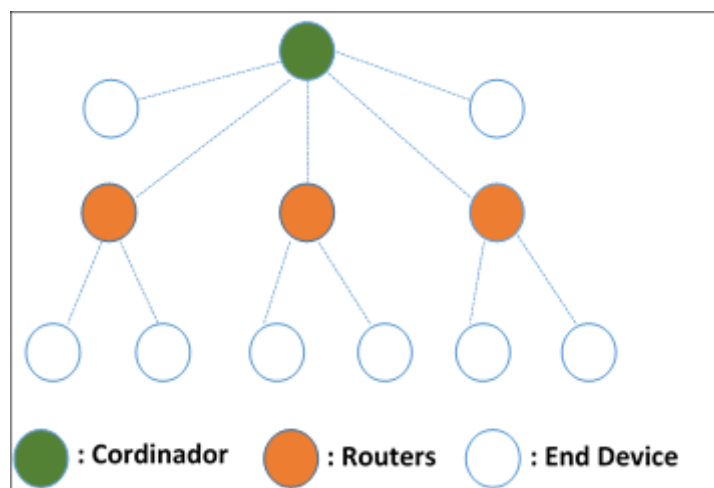


Figura 11: Topología en árbol
Fuente: Elaboración propia.

c. Topología mesh

En una topología de malla, en contraste con la topología de árbol, no hay jerarquía de relaciones. Aquí cualquier dispositivo puede intentar conectarse a cualquier otro dispositivo ya sea directamente o aprovechando los dispositivos intermedios con capacidad de enrutamiento para retransmitir el mensaje en nombre del originador

del mensaje. En topología de malla, la ruta desde la fuente hasta al destino se crea a pedido y se puede modificar según su entorno. La capacidad de una red en malla permite crear y modificar rutas dinámicamente, con el cual aumenta la confiabilidad de las conexiones inalámbricas. Si, por alguna razón, el dispositivo fuente no puede comunicarse con el dispositivo de destino usando una ruta previamente establecida, los dispositivos vecinos con capacidad de enrutamiento pueden cooperar para encontrar una ruta alternativa desde el dispositivo de origen al dispositivo de destino. (Farahani, 2008).

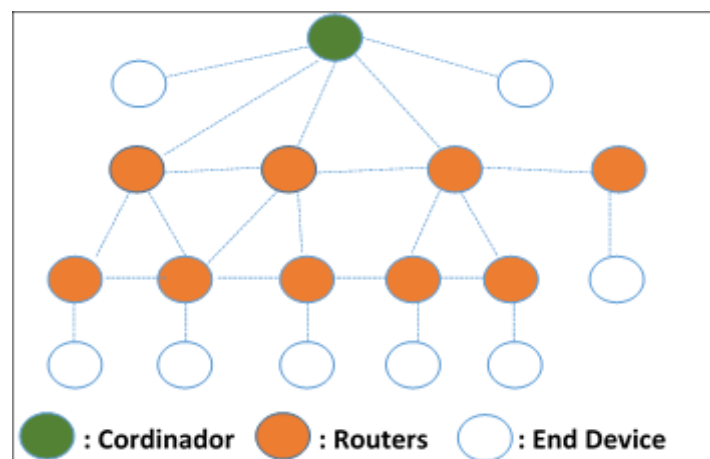


Figura 12: Topología mesh.
Fuente: Elaboración propia.

Existen tres tipos de nodos que forman una red ZigBee, el cual se describen a continuación (Faludi, 2010):

a. Zigbee Coordinador

Las redes ZigBee siempre tienen un solo coordinador este radio es responsable para formar la red, distribuir direcciones y administrar las otras funciones que define la red, es de función completa que es capaz de retransmitir mensajes, el coordinador es también el controlador principal de una red de área personal (PAN). Este dispositivo tiene la función principal de establecer una red asignado una dirección y PAN ID a la red, al cual los dispositivos puedan conectarse y mantener la tabla de enrutamiento actualizado. En tanto ya establecida la red, el coordinador hace las veces de Router, es decir enruta los mensajes entre los dispositivos de función reducida en una red.

b. Zigbee routers

Un Zigbee Router es un dispositivo de función completa, que permite extensión del rango físico de una red Zigbee. Este tipo de dispositivo permite que más nodos se unan a la red ya que el rango de radio de la red raíz es efectivo aumentando activamente, además puede cumplir funciones de dispositivos finales como es

monitorear sensores y ejecutar funciones de control. En una red Zigbee puede haber muchos dispositivos de tipo router.

c. Zigbee End

Los dispositivos finales son esencialmente versiones reducidas de un enrutador, pueden unirse a redes y enviar recibir información, sin embargo, no actúan como mensajeros entre cualquier otro dispositivo. Un dispositivo final tiene el menor tamaño de memoria y pocas capacidades y funciones de procesamiento, un dispositivo final es normalmente el menos costoso dispositivo de la red. Este dispositivo se conecta físicamente a un sensor o ejecuta control, funciones. El dispositivo final puede ser un dispositivo de función completa o de función reducida dependiendo del fin de la aplicación prevista para el dispositivo, su comunicación es requerida siempre de un router o un coordinador.

3.10. Estándar propietario Digimesh

DigiMesh es un protocolo de red propietario de la marca Digi basado en ZigBee soporta múltiples topologías como punto a punto, punto a multipunto y malla. También proporciona modos de ahorro de energía, por lo que es ideal para aplicaciones en las que el consumo de energía es algo crucial.

DigiMesh está disponible en plataformas con mayor rango de radio frecuencia y posee más opciones de velocidad de datos. Se usa en soluciones que requieren conectividad inalámbrica (Miñán, 2017).

Principales características del protocolo DigiMesh:

- ✓ **Self-healing:** referido a la capacidad de añadir o quitar un nodo de la red en cualquier momento sin que ésta falle.
- ✓ **Arquitectura punto a punto:** no existen jerarquías o relaciones padre-hijo entre los nodos.
- ✓ **Fácil de usar:** debido a que no hay jerarquías, una red de malla está más simplificada.
- ✓ **Route Discovery:** las rutas se descubren y se crean sólo cuando es necesario, eliminando la necesidad de mantener un mapa de la red.
- ✓ **Selective acknowledgments:** sólo los nodos de destino responderán a las peticiones de rutas.
- ✓ **Fiable:** las confirmaciones permiten el envío correcto de los datos.
- ✓ **Modo dormido:** soporta modos de bajo consumo con tiempos de encendido sincronizados, así como tiempos de hibernación variables.

3.10.1. Nodos

DigiMesh también diferencia entre 3 tipos de nodos, pero normalmente se suele usar el tipo Router. La ventaja es que se pueden construir redes homogéneas, es decir, al ser todos routers, los nodos pueden encaminar paquetes. Además, no existen relaciones padre-hijo, por lo que la red es más simple. Los nodos se pueden configurar para su uso en dispositivos que necesiten ahorrar energía (Miñán, 2017). En la siguiente Figura 13 se muestra un ejemplo de red DigiMesh homogénea.

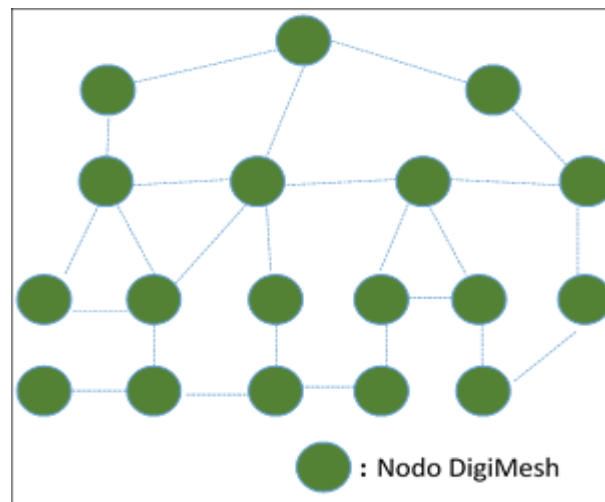


Figura 13: Nodos de red DigiMesh.
Fuente: Elaboración propia.

Con este tipo de redes se tiene ventajas como:

- ✓ Facilidad de implementar y configurar la red.
- ✓ Ofrece escalabilidad de la red.
- ✓ Fiabilidad ante las comunicaciones interrumpidas o nodos dañados.

3.11. Cuadro comparativo entre Zigbee Y Digimesh

Después de describir ambos protocolos, se realizó una comparación en la siguiente Tabla 2 donde cada uno de ellos tiene ventajas y desventajas según el escenario donde se implemente.

Tabla 2: Diferencias entre ZigBee y DigiMesh

Características	ZigBee Mesh	DigiMesh
Tipos de nodos.	3 tipos de nodos: Coordinador, Router, End Device.	Un solo tipo de nodo, flexibilidad a la hora de expandir la red.
Modo dormido.	Solo los End Device pueden hibernar.	Todos los nodos pueden hibernar, sin necesidad de sincronizar.
Actualización del Firmware Over the Air.	Si	No
Alcance de Rx y Tx.	Tiene un rango menor a 3.2 Km en saltos.	Rango hasta 6.5 Km entre saltos.
Carga útil rendimiento.	Hasta 80 bytes.	Hasta 256 bytes.
Tamaño del código.	Más largo. Menos espacio.	Más pequeño. Más espacio.
Frecuencias soportadas y tasas de RF.	2.4 GHz (250 Kbps), 900 MHz (40 kbps), 868 MHz (20 kbps).	2.4. GHz (250 kbps), 900 MHz (10, 125, 150 kbps).
Seguridad.	Encriptación AES.	Encriptación AES.
Interoperabilidad.	Interoperabilidad entre fabricantes.	Propietario.
Tolerancia a Interferencias.	Direct - Sequence Spread Spectrum (DSSS)	900 MHz: Frequency-Hopping Spread Spectrum (FHSS). 2.4 GHz: Direct-Sequence Spread Spectrum (DSSS).
Direccionamiento.	2 capas. MAC address (64 bit) y Network address (16 bit)	MAC address (64 bit)
Mantenimiento.	Más herramientas disponibles en el Mercado.	El direccionamiento más simple ayuda en la diagnosis y la configuración de la red.

Fuente: (Miñán, 2017).

En conclusión, según la Tabla 2 se tiene la opción elegir un protocolo u otro, de acuerdo a:

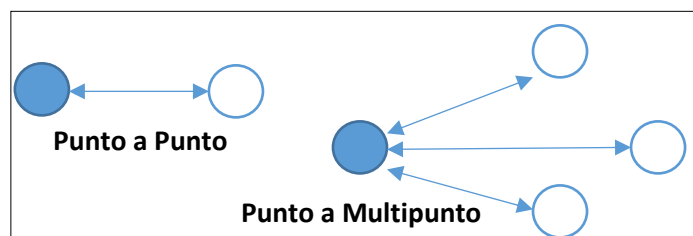
- ✓ Si se desea que la red sea abierta, es decir utilizar dispositivos de diferentes fabricantes y tener actualizaciones automáticas over-the-air, para ello el protocolo ZigBee es adecuado.
- ✓ Por otro lado, si se necesita ahorrar energía, simplificar el proceso de configuración y expansión de la red, más velocidad y mayor rango entre nodos, la opción es el protocolo DigiMesh.

Para efectos del presente trabajo de tesis se eligió el protocolo DigiMesh por sus características adecuados en el desarrollo de la tesis.

3.12. Xbee

XBee es propiedad de Digi, y está basado en el protocolo ZigBee. En pocas palabras, los XBees son módulos inalámbricos fáciles de usar. Según Digi, los módulos XBee son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos.

La radio XBEE transmite datos a 2,4 GHz, para crear redes punto a punto, o para redes punto a multipunto (Blum, 2014). La Figura 14 muestra ejemplos de ambas. Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible. En términos simples, los XBee son módulos inalámbricos fáciles de usar.



*Figura 14: Red Punto a Punto y Punto a Multipunto.
Fuente: Elaboración propia.*

La Figura 15 muestra el aspecto físico con la distribución de sus pines del módulo XBee.

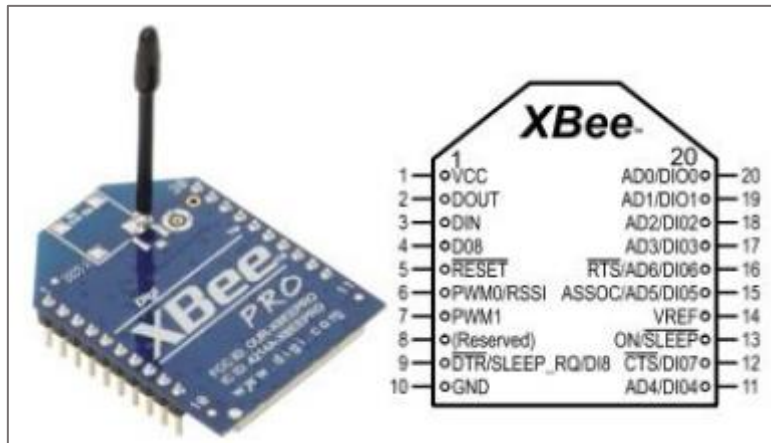


Figura 15: Módulo XBee XB24-Z7WIT-004 de Digi.
Fuente: Elaboración propia.

En la Tabla 3 se detalla la descripción de los 20 pines, de los cuales 9 pines son Entradas/Salidas las cuales pueden ser usadas como salida digital (entrega 0-LOW o 1-HIGH) o entrada digital (obtiene valores 0-LOW. Además 7 pines de los 9 pines tienen como conversor de ADC (Analogóico a Digital).

Tabla 3: Descripción de pines de los módulos Xbee y Xbee pro

Pin #	Nombre	Dirección	Descripción
1	Vcc	-	Tensión de alimentación
2	DOUT	Salida	UART salida de datos
3	DIN/CONFIG	Entrada	UART entrada de datos
4	DO8	Salida	Salida digital 8
5	RESET	Entrada	Modo reset
6	PWM0/RSSI	Salida	Salida PWM0/ indicador de señal recibida
7	PWM1	Salida	Salida PWM 1
8	[reserved]	-	No conectado
9	DTR/SLEEP_RQ/ DI8	Entrada	Pin de control de línea de sleep o Entrada digital 8
10	GND	-	Tierra
11	AD4/DIO4	Entrada/ Salida	Entrada analógica 4 o entrada/salida digital 4
12	CTS/DIO7	Entrada/ Salida	Control de flujo o entrada/salida 7
13	ON/SLEEP	Salida	Indicador módulo de estado
14	VREF	Entrada	Referencia de voltaje para entradas analógicas/digitales

15	Associate/AD5/DIO5	Entrada/ Salida	Indicador asociado/entrada analógica 5 o entrada/salida digital 5
16	RST/AD6/DIO6	Entrada/ Salida	Solicitud de envío de control de flujo/entrada analógica 6 o entrada digital 6
17	AD3/DIO3	Entrada/ Salida	Entrada analógica 3 o entrada/salida digital 3
18	AD2/DIO2	Entrada/ Salida	Entrada analógica 2 o entrada/salida digital 2
19	AD1/DIO1	Entrada/ Salida	Entrada analógica 1 o entrada/salida digital 1
20	AD0/DIO0	Entrada/ Salida	Entrada analógica 0 o entrada/salida digital 0

Fuente: Elaboración propia.

La Figura 16 muestra las conexiones mínimas que necesita el módulo XBee para poder ser utilizado. Luego de esto, se debe configurar según el modo de operación que se desea para la aplicación requerida por el usuario.

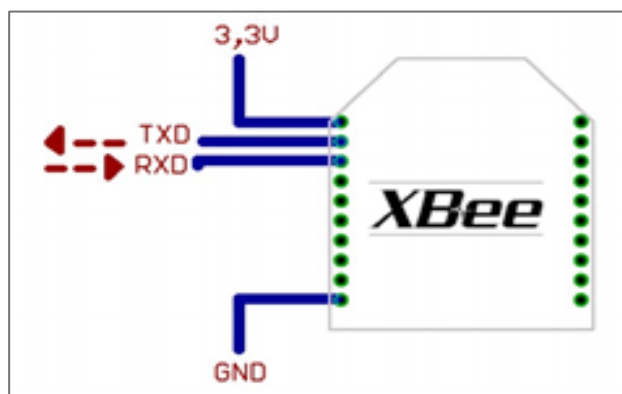


Figura 16: Conexiones mínimas del XBEE
Fuente: Elaboración propia.

El módulo requiere una alimentación desde 2.8 a 3.4V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún conversor adecuado para los niveles de voltaje.

3.12.1. Series de módulos Xbee

Digi International fabrica una deslumbrante variedad de radios con la marca XBee. Hay al menos 30 combinaciones diferentes de hardware de componentes, protocolos de firmware, etc. Además, presentan una serie de características que los hacen especialmente fáciles al momento de trabajar sobre todo para un

principiante. Los ingenieros profesionales a menudo prefieren XBees para la creación de prototipos con un esfuerzo reducido de desarrollo lo que los convierte en la opción más rentable. A continuación, se describe las características de las series 1 y 2.

a. Módulo XBee Serie 1

Los módulos Xbee de la serie 1 es bastante usado por lo que presentaba su fácil configuración hasta hoy en día, pero presenta desventajas entre la serie Pro. Esta serie 1 es ideal para reemplazos de cables simples y sistemas de menor tamaño. En esta serie no se implementa el protocolo Zigbee (Faludi, 2010).

b. Módulo XBee Serie 2

Esta serie 2 de hardware es compatible con el protocolo ZigBee completo, está diseñada para dar soporte a redes de sensores más grandes en mente y es esencial para las interacciones robustas con los sistemas basados en ZigBee. Que ahora se están implementando ampliamente en aplicaciones residenciales, académicas y comerciales. El hardware de esta serie tiene un rango un poco mejor y usa un poco menos de potencia que la serie normal; Sin embargo, estas pequeñas mejoras no serían una razón para elegir un formato sobre el otro. Ambas series usan la misma huella física y pueden intercambiarse fácilmente, a menudo con solo cambios menores en el software subyacente (Faludi, 2010). En la Tabla 4 se muestra un resumen de las similitudes y diferencias.

Tabla 4: Serie 1 versus Serie 2 para XBees regulares

Diferencias	Serie 1	Serie 2
Alcance de rango en interiores	30 metros	40 metros
Alcance de rango en línea de vista	100 metros	120 metros
Transmitir y recibir corriente	45/50 mA	40/40 mA
Firmware (típico)	802.15.4 punto a punto	ZigBee malla
Pines de entrada y salida digital	8	11
Pines de entrada analógica	7	4
Pines de salida analógicos (PWM)	2	Ninguno
Baja potencia, bajo ancho de banda, bajo costo, direccionable, estandarizado, pequeño y popular	Si	Si
Topologías de estrellas punto a punto	Si	Si
Topologías de malla y árbol	No	Si
Firmware único para todos los modos	Si	No

Requiere nodo coordinador	No	Si
Potencia de salida	1 mW	60 mW
Sensibilidad de receptor	-92 dBm	-100 dBm
Aplicaciones basadas en estándares	No	Si
Actualizado y apoyado activamente	Si	Si

Fuente: (Faludi, 2010).

3.12.2. Modos de operación de los módulos Xbee

Los radios XBee pueden comunicar en 2 modos diferentes que son modo transparente y modo API (Oyarce, Aguayo, & Martin, 2008).

a. Modo transparente

Este es el estado predeterminado para las radios XBee que usan firmware AT. Se llama transparente porque la radio simplemente pasa la información exactamente como la recibe. Modo transparente se utiliza para enviar datos a través del XBee a una radio de destino remota. Cuando el dato es recibido, esto es expedido mediante la serie de un puerto exactamente como es recibido (Faludi, 2010).

b. Modo API

Como se conoce la estructura del protocolo TCP/IP, no es bastante fácil de entender el formato API para radios XBee. Pues el modo API XBee usa las mismas estructuras del diseño TCP/IP, y hace las mismas funciones de encapsulación y des encapsulación de los datos. El modo API puede transmitir altamente estructurado de datos con rapidez, predeciblemente y seguramente. Cada frayme de datos son específicos y distintos para cada tipo de mensaje que recibimos de la radio XBee (Faludi, 2010).

3.13. Programa para configurar Xbee

XCTU, es un software multiplataforma de distribución libre que permite interactuar con los módulos Xbee mediante un interfaz gráfico. Esta aplicación sirve para configurar, inicializar, actualizar firmware y testear los módulos XBee, comunicándose por puerto serie a los módulos (Oyarce, Aguayo, & Martin, 2008).

Una ventaja de este software es que puedes ver rápidamente un resumen de todos los parámetros del módulo y una descripción de ellos.

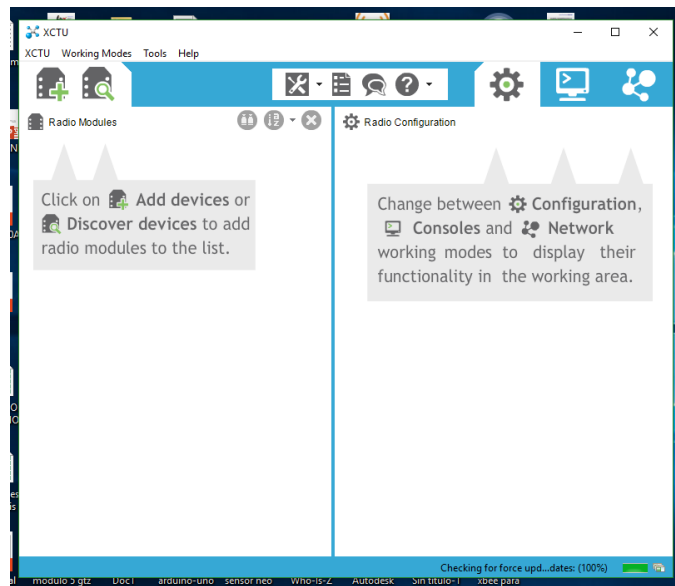


Figura 17: Programa XCTU.
Fuente: Elaboración propia.

3.14. Sensor

Es un dispositivo que está diseñado para recibir información del exterior, es decir magnitudes físicas como sonido, temperatura, caudal, movimiento, etc. Para luego transformarla en señales digitales y finalmente enviar a un controlador.

Se denomina sensor a un “sistema de medida” formado por el conjunto, transductor, sensor y acondicionador de señal, y que permite la medición de valores de características físicas, químicas, etc. Para tal objetivo adapta la señal obtenido desde un transductor que suele ser en el orden de los milivoltios a valores más comprensibles según el objetivo del sistema general a implementarse.

Se denomina transductor, en general, a todo dispositivo que convierte una señal de una forma física en una señal correspondiente, pero de otra forma física distinta. Es, por tanto, un dispositivo que convierte un tipo de energía en otro. Esto significa que la señal de entrada es siempre una energía o potencia, pero al medir una de las componentes de la señal suele ser tan pequeña que puede despreciarse, y se interpreta que se mide solo la otra componente.

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal de salida transducible que es función de la variable medida. Sensor y transductor se emplean a veces como sinónimos, pero sensor sugiere un significado más extenso: la ampliación de los sentidos para adquirir un conocimiento de cantidades físicas que, por su naturaleza o tamaño, no pueden ser percibidas directamente por los sentidos. Transductor, en cambio, sugiere que la señal de entrada y la de salida no deben ser homogéneas. (Pallás, 2003).

Los sensores obtienen información del mundo físico externo y la transforman en una señal eléctrica que puede ser manipulada por la circuitería interna de control.

3.14.1. Sensor de flujo

El sensor de caudal se compone de una válvula de plástico, un rotor (parte giratoria), y un sensor de efecto Hall. Cuando el agua fluye a través de los rodillos del rotor, este comienza a girar con mayor o menor velocidad dependiendo del caudal del líquido que fluye a través de él (Roman, 2010).

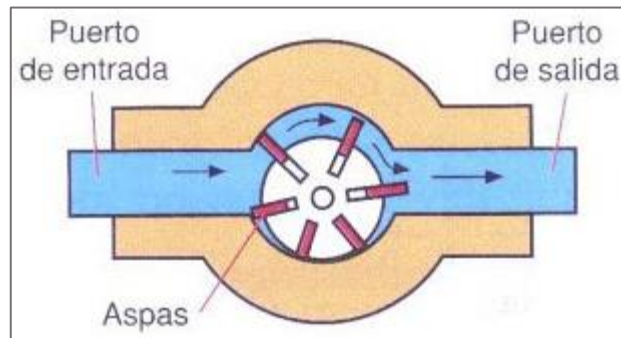


Figura 18: Funcionamiento del sensor de caudal.
Fuente: (Roman, 2010).

3.14.1.1. Sensor YF-S201

De acuerdo Marquez, (2014) es un sensor de flujo que se instala en la línea del agua, y utiliza un sensor de efecto Hall que permite medir la cantidad de líquido que se ha movido a través de él. El aspa de viento tiene un pequeño imán atado, y hay un sensor magnético de efecto Hall, en el otro lado del tubo de plástico se puede medir la cantidad de vueltas del aspa de viento que ha hecho a través de la pared de plástico. Este método permite que el sensor permanezca seguro y seco.

El sensor de flujo YF-S201 está constituido por una carcasa de plástico como se ve en la Figura 19, un rotor de agua y un sensor de efecto Hall. El funcionamiento de este sensor es muy simple, es decir, en el rotor tiene un pequeño imán adherido para poder registrar una vuelta cada vez que pasa por el sensor magnético de efecto Hall que hay en el otro lado del tubo, generándose pulsos de salida a una velocidad proporcional a la del flujo. En la parte posterior del sensor hay una flecha para indicar la dirección del flujo. Cada pulso en la salida del sensor equivale aproximadamente a 2,25 mililitros.



Figura 19: Sensor de flujo YF-S201.
Fuente: Elaboración propia.

Tabla 5: Características del sensor de flujo YS-S201

modelo	YF-S201
Sensor secundario	Efecto Hall
Voltaje de funcionamiento	5 a 18 VDC
Máximo consumo de corriente	15 mA a 5 VDC
Tipo de salida	5V TTL
Trabajo caudal	1 a 30 litros/minuto
Temperatura	-25 a +80 °C
Humedad	35% -80%
Precisión	$\pm 2\%$
Modo de detección	Vertical
Presión máxima de agua	2 MPa
Ciclo de trabajo de salida	50% $\pm 10\%$
Tiempo de subida de la señal de salida	0.04us
Tiempo de bajada de la señal de salida	0.18us
Velocidad de flujo (pulso)	Frecuencia (Hz) = 7,5 * Caudal (L / min)
Pulsos por litro	450
Longitud cable de conexión	15cm
Conexión entrada de caudal	Tubería 1/2" diámetro externo 0.78" roscado 1/2"
Tamaño	2.5" x 1.4" x 1.4
Detalle cable de conexión	Rojo: positivo de la alimentación (IN) Negro: negativo de la alimentación (GND) Amarillo: señal de salida (OUT)

Fuente: (Argüello González, 2016).

Según la hoja de datos del fabricante, por cada litro de agua que pasa por el sensor se obtiene aproximadamente 450 pulsos, entonces:

$$450 \text{ pulsos/min} = 1\text{L/min}$$

$$7.5 \text{ pulsos/seg} = 1\text{L/min}$$

$$7.5 \text{ Hz} = 1\text{L/min}$$

Esto quiere decir que si se tiene la frecuencia (pulsos por segundo), simplemente se divide entre 7.5 para obtener el número de litros/minuto. Para el caudal en litros/hora debe multiplicarse la frecuencia por 60 minutos que tiene una hora.

$$\text{Caudal (L/min)} = \text{Frecuencia}/7.5$$

$$\text{Caudal (L/hora)} = \text{frecuencia} * 60 / 7.5$$

3.15. Base de datos

Según la definición de Marqués, (2011) es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Estos datos han sido almacenados de manera estructurada para satisfacer los requisitos de información de una empresa u otro tipo de organización.

Teniendo la base de datos almacenada en algún lugar no es suficiente en términos de administración de datos. Un usuario necesita crear, modificar o eliminar datos, para ello existen aplicaciones que facilitan esas tareas comúnmente denominados Sistema de Gestión de Base de Datos (SGBD). Generalmente un SGBD define un lenguaje de datos para especificar el tipo de dato y el lenguaje de manejo de datos. Entre los tipos de sistemas de gestores de base de datos se seleccionó el MYSQL como gestor de la base de datos para el presente trabajo de tesis.

3.15.1. MySQL

Es uno de los gestores de base de datos, que entre sus ventajas tiene la velocidad y la robustez. Adoptando la definición de los autores Berni & De la Iglesia, (2010) es un SGBD sencillo y rápido que se adapta perfectamente a entornos en los que el volumen de datos sea de orden de megabytes.

3.16. Lenguaje de programación

Es una sintaxis de comunicación entre hombre – máquina, es decir las tareas que queremos que ejecute una máquina se lo decimos a través de un lenguaje de programación. Entre los tipos de lenguajes de programación tenemos: C/C++, Java, PHP, etc. Para el caso del trabajo de tesis se utilizó PHP.

3.16.1. PHP

PHP corresponde a las iniciales de Personal Home Page Tools (Herramientas para Páginas Iniciales Personales). Es un lenguaje de programación de tipo script

para entornos web con unas funciones muy semejantes a las ASP y JSP. Desde un punto de vista técnico es un lenguaje interpretado de alto nivel, además es un lenguaje incrustado en páginas HTML, es decir, es un lenguaje de programación que se introduce dentro de las páginas HTML. El código PHP se interpreta en el lado del servidor de web, desde donde se genera la página HTML. Su principal objetivo es hacer los desarrollos de aplicaciones basadas en la web se puedan escribir páginas que se generan dinámicamente de un forma sencilla y rápida (Berni & De la Iglesia, 2010).

3.17. Metodología ágil programación extrema (XP)

Es un enfoque de la ingeniería de software formulado por Kent Beck, considerado como el más destacado de los procesos ágiles de desarrollo de software, por que presenta más énfasis en la adaptabilidad que en la previsibilidad. (Bautista, 2012).

Según los autores Letelier & Penadés (2006). Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

3.17.1. Fases de la programación extrema

El autor Joskowicz, (2008) define el ciclo de vida de la Programación Extrema en 4 fases, las cuales son:

a. Planificación

En esta fase se hace una planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

La planificación comprende los siguientes aspectos básicos:

- ✓ **Las Historias de Usuarios:** en esta parte el cliente escribe en su propio lenguaje que o cuales son las funcionalidades del sistema, como descripciones cortas de lo que el sistema debe realizar.
- ✓ **El Plan de Entregas (Release Plan):** se establece el cronograma de entregas del desarrollo de las historias del cliente, algunas historias del cliente serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto.
- ✓ **Plan de Iteraciones (Iteration Plan):** las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.
- ✓ **Reuniones Diarias de Seguimiento (Stand – Up Meeting):** con las reuniones diarias se busca mantener la comunicación entre el equipo y compartir problemas y soluciones.

b. Diseño

Esta fase hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

- ✓ **Simplicidad:** Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione.
- ✓ **Soluciones “spike”:** Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de cliente, pueden utilizarse pequeños programas de prueba (llamados “Spike”), para explorar diferentes soluciones.
- ✓ **Recodificación:** La recodificación (“refactoring”) consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible.
- ✓ **Metáforas:** Una metáfora es un concepto que todos entienden, sin necesidad de mayores explicaciones. La metodología XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo. Por ejemplo, puede ser una guía para la nomenclatura de los métodos y las clases utilizadas en el diseño del código.

c. Desarrollo

Esta fase comprende la codificación del sistema. El desarrollo o codificación implica los siguientes aspectos:

- ✓ **Disponibilidad del cliente:** Uno de los requisitos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino el involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP.
- ✓ **Uso de estándares:** XP promueve la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo, y que facilite la recodificación.
- ✓ **Programación dirigida por las pruebas:** Las pruebas a los que se refiere esta práctica, son las pruebas unitarias, realizados por los desarrolladores. La definición de estos test al comienzo, condiciona o “dirige” el desarrollo.
- ✓ **Programación en pares:** XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.
- ✓ **Integraciones permanentes:** Los desarrolladores necesitan trabajar con la última versión. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones publicadas. Para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.
- ✓ **Propiedad colectiva del código:** En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, cualquier pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o recodificar.
- ✓ **Ritmo sostenido:** El concepto que se desea establecer con esta práctica es el de planificar el trabajo que mantenga un ritmo constante y razonable, sin sobrecargar al equipo de horas, tampoco tener horas muertas de trabajo.

d. Pruebas

Esta fase comprende los siguientes aspectos:

- ✓ **Pruebas unitarias:** Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código.
- ✓ **Detección y corrección de errores:** Cuando se encuentra un error (“bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.
- ✓ **Pruebas de aceptación:** Estas pruebas son creadas en base a las historias del cliente, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia del cliente ha sido correctamente implementada.

CAPÍTULO IV. METODOLOGÍA DE LA INVESTIGACIÓN

Para el desarrollo, gestión y administración del presente trabajo de tesis, se utilizó la metodología ágil de Programación Extrema. Lo que se logró con esta metodología es adaptabilidad, corrección y control iterativo en cada fase del desarrollo del proyecto. El ciclo de desarrollo en Programación Extrema, es como se muestra en la Figura 20.

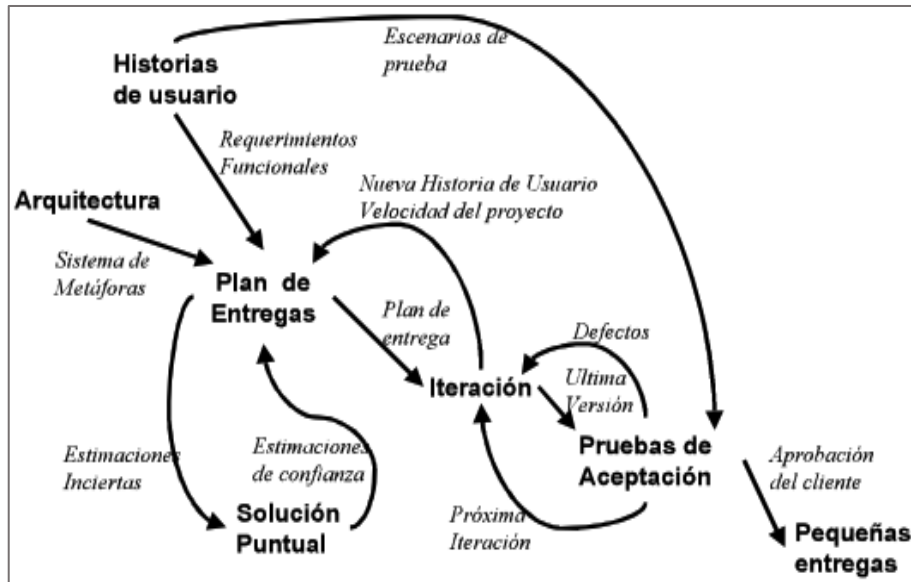


Figura 20: Proceso XP.
Fuente: (Orjuela & Rojas, 2008).

Por consiguiente, se desarrolló la tesis según las fases del modelo de Programación Extrema:

4.1. Planificación

Se requiere un prototipo para el monitoreo eficiente del consumo de agua en la Institución Educativa Industrial Federico Villareal. Para ello se implementó dos módulos, por un lado, el módulo emisor y por el otro el módulo receptor. Como se ve en la Figura 21 los componentes principales del módulo emisor son Arduino uno, sensor de flujo (YF-S201), radio Xbee. Este último es el dispositivo de transmisión inalámbrica. Mientras los componentes principales del módulo receptor son el radio Xbee, Arduino uno y finalmente un servidor para el almacenamiento de los datos.

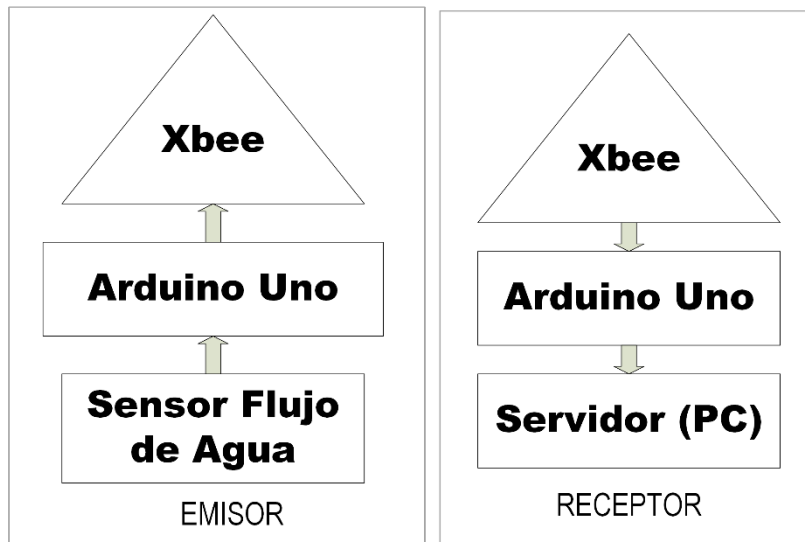


Figura 21: Módulos emisor y receptor.
Fuente: Elaboración propia.

4.1.1. Requisitos funcionales

El desarrollo de la tesis está embebido de hardware y software. La parte de hardware está conformada por el nodo sensor y el nodo coordinador, la parte de software comprende del sistema web que permite el monitoreo y consultar los datos del consumo de agua. Ambas partes se unen en la base de datos. Por consiguiente, se describe los requisitos funcionales que contempla la parte de hardware y software.

La Tabla 6, muestra la historia número 01 del cliente que hace referencia a las funcionalidades del nodo sensor de flujo.

Tabla 6: Historia 01 del cliente

Historia	Cliente	Requiere	Funcionalidad
01	Administrador/ Usuario	Se requiere de un nodo sensor que pueda medir el flujo de agua, cada vez que se consume.	Medir el consumo de agua. Transmitir medición.

Fuente: Elaboración propia.

La Tabla 7, muestra la historia número 02 del cliente que hace referencia a las funcionalidades del nodo coordinador de la red inalámbrica.

Tabla 7: Historia 02 del cliente

Historia	Cliente	Requiere	Funcionalidad
02	Administrador/ Usuario	Se requiere de un nodo coordinador que pueda recibir y registrar los datos enviados del nodo sensor.	Peticionar datos de la medición.
			Registrar medición en la base de datos.

Fuente: Elaboración propia.

La Tabla 8 muestra la historia número 03 del cliente que hace referencia al almacenamiento de los datos leídos por los sensores.

Tabla 8: Historia 03 del cliente

Historia	Cliente	Requiere	Funcionalidad
03	Administrador/ Usuario	Se requiere de una base de datos que pueda almacenar los datos leídos por el sensor de flujo.	Almacenar los datos enviados por el nodo sensor.

Fuente: Elaboración propia.

En la Tabla 9 se muestra la historia número 04 del cliente que hace referencia al monitoreo del consumo de agua a través de un sistema web.

Tabla 9. Historia 04 del cliente

Historia	Cliente	Requiere	Funcionalidad
04	Administrador/ Usuario	Se requiere un sistema web que permita el monitoreo del consumo de agua.	Visualizar consumo.
			Administrar usuarios.
			Administrar sensor
			Administrar nodo.
			Generar reportes.

Fuente: Elaboración propia.

La Tabla 10 muestra la historia número 05 del cliente, que hace referencia a la topología de la red inalámbrica.

Tabla 10. Historia 05 del cliente

Historia	Cliente	Requiere	Funcionalidad
05	Administrador/ Usuario	Se requiere que la red inalámbrica garantice la seguridad e integridad de los datos.	Formar la red inalámbrica del nodo sensor con el nodo coordinador mediante la topología punto a punto.

Fuente: Elaboración propia.

El diagrama de casos de uso de la Figura 22, representa la descripción de las acciones que realiza los diferentes actores con el sistema, para llevar a cabo algún proceso en circunstancias específicas.

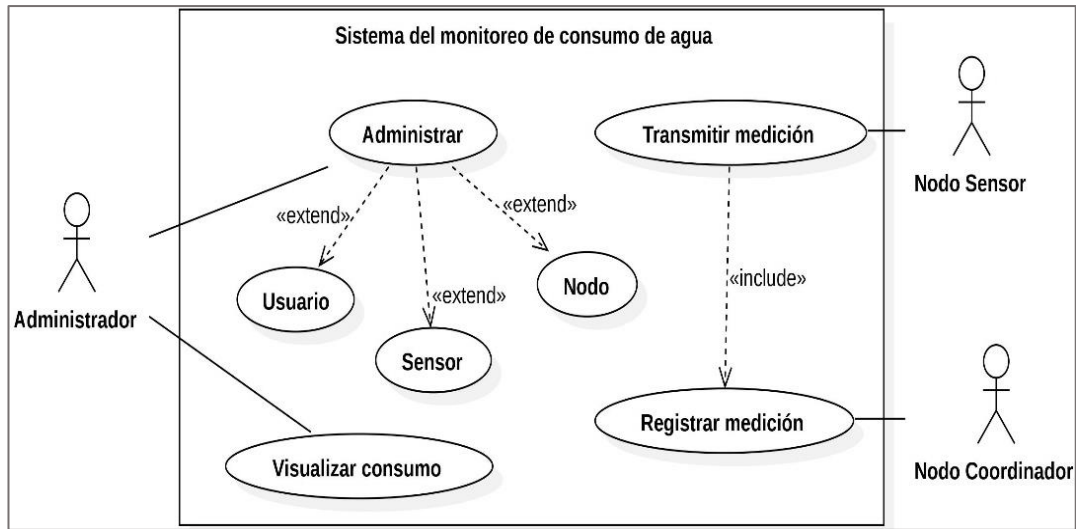


Figura 22: Diagrama de casos de uso.
Fuente: Elaboración propia.

El diagrama de objetos que presenta la Figura 23 muestra un nivel de abstracción que ayudó en el desarrollo del sistema. El diagrama presenta la relación de objetos que son instancias de clases con sus respectivos atributos con valores.

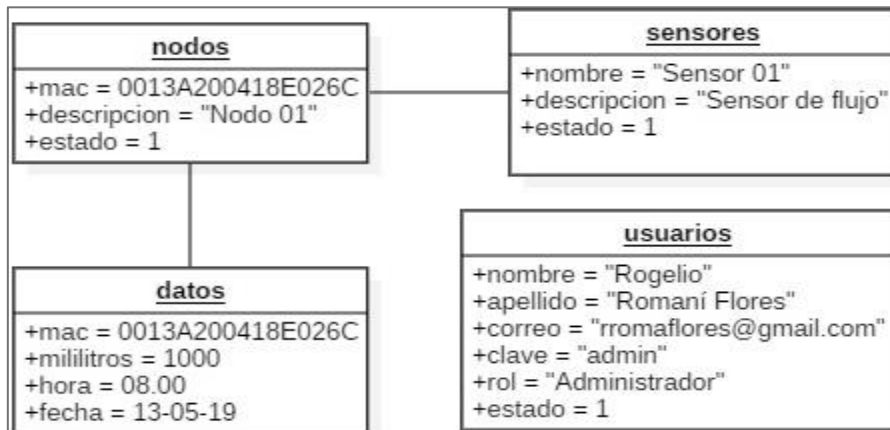


Figura 23: Diagrama de objetos.
Fuente: Elaboración propia.

A continuación, se desarrolló los diagramas de secuencia para cada acción descrito en el diagrama de casos de uso. Estos diagramas de secuencias indican la forma en la que los eventos provocan transiciones de un objeto a otro en tiempos diferentes.

En la Figura 24 se presenta el diagrama de secuencia que detalla la forma de interacción entre el objeto nodo sensor con el nodo coordinador para llevar a cabo la acción de transmitir medición hasta el objeto datos, el cual se encarga de registrar la medición del consumo de agua en la base de datos

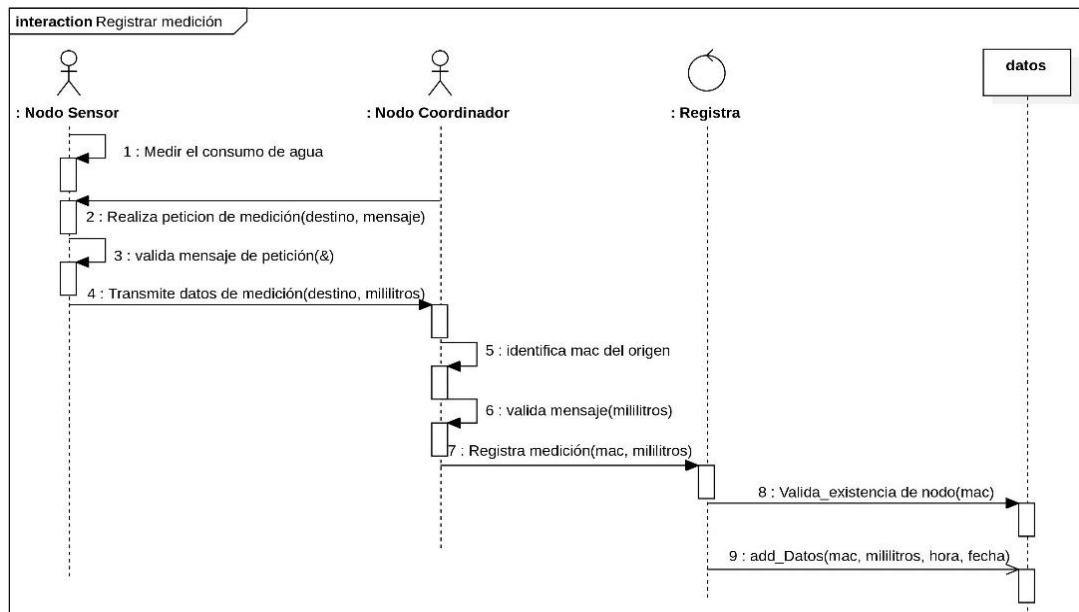


Figura 24: Diagrama de secuencia registrar medición.
Fuente: Elaboración propia.

El objeto administrador interactúa con el objeto datos para acceder al sistema, este acceso se realiza digitando los parámetros correo y clave. El objeto datos valida los parámetros enviados desde el objeto administrador para permitir o denegar el acceso al sistema. En caso de permitir, el sistema muestra al objeto administrador el consumo de agua del día y el grafico correspondiente.

El evento de generar reportes lo inicia el objeto administrador solicitando al objeto datos la información del consumo de agua correspondiente a un rango de fecha. El objeto datos busca la información solicitada y devuelve la información si es que existe en el rango de fecha indicada, caso contrario muestra un mensaje de que la información no se encuentra e indica que ingrese otro rango de fecha.

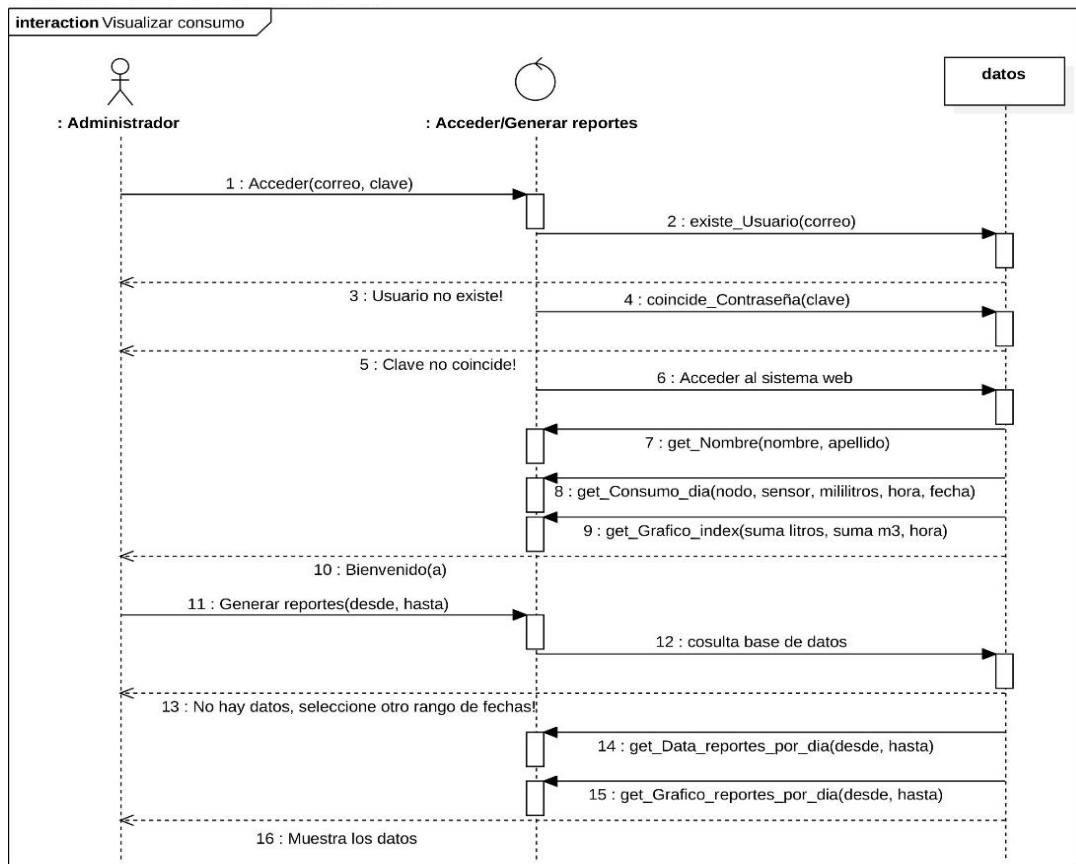


Figura 25: Diagrama de secuencia visualizar consumo.
Fuente: Elaboración propia.

La acción de administrar especificado en el diagrama de caso de uso corresponde a la administración de los usuarios, sensores, nodos. Los cuales se detalla a continuación indicando los eventos que realizan los objetos.

La Figura 26 representa la secuencia de los eventos de registrar, modificar, deshabilitar y habilitar un usuario, estos eventos se realizan en determinados tiempos por el actor administrador.

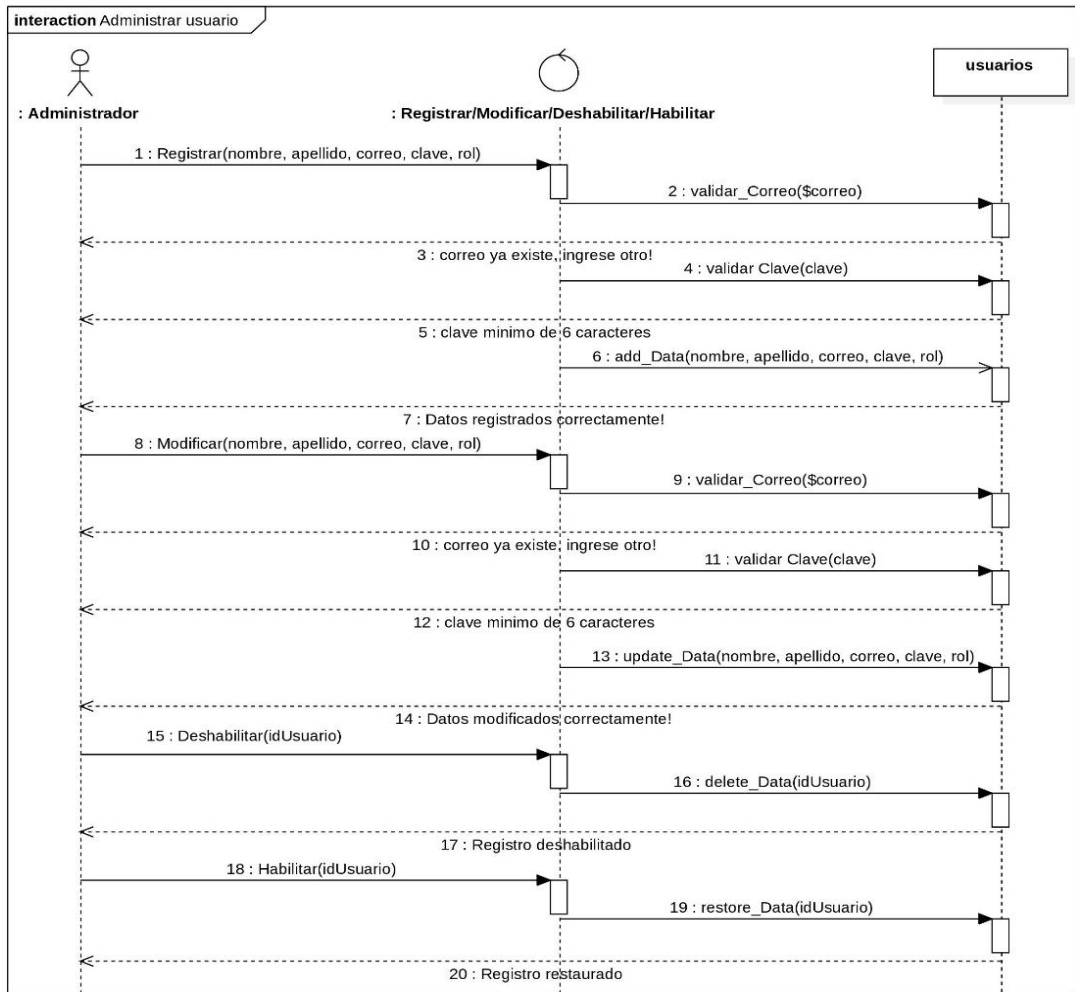


Figura 26: Diagrama de secuencia administrar usuarios.
Fuente: Elaboración propia.

La administración del sensor implica la realización de una de los cuatro eventos básicos como registrar, modificar, deshabilitar y habilitar. En la Figura 27 se describe la secuencia de estos eventos que los realizan los objetos.

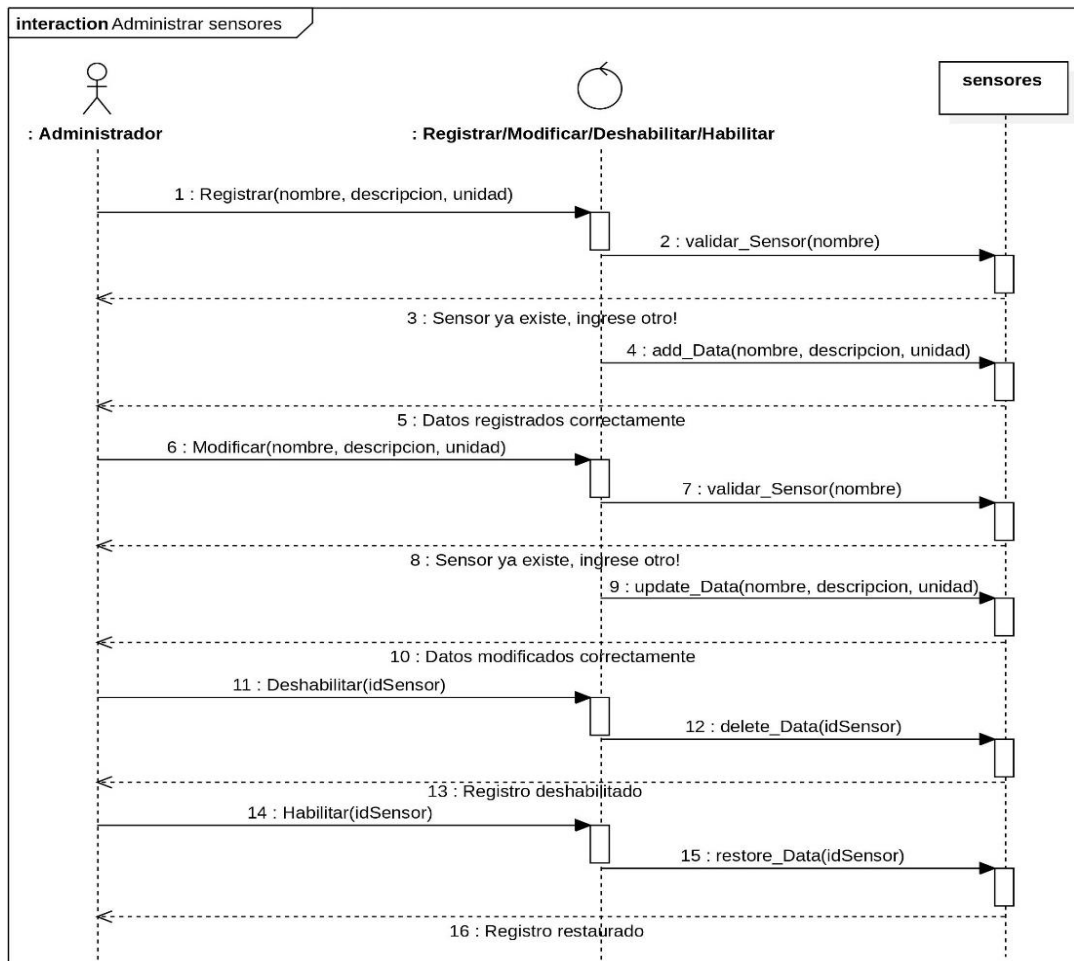


Figura 27: Diagrama de secuencia administrar sensor.
Fuente: Elaboración propia.

La administración del nodo implica la realización de una de los cuatro eventos básicos como registrar, modificar, deshabilitar y habilitar. En la Figura 28 se describe la secuencia de estos eventos que los realizan los objetos.

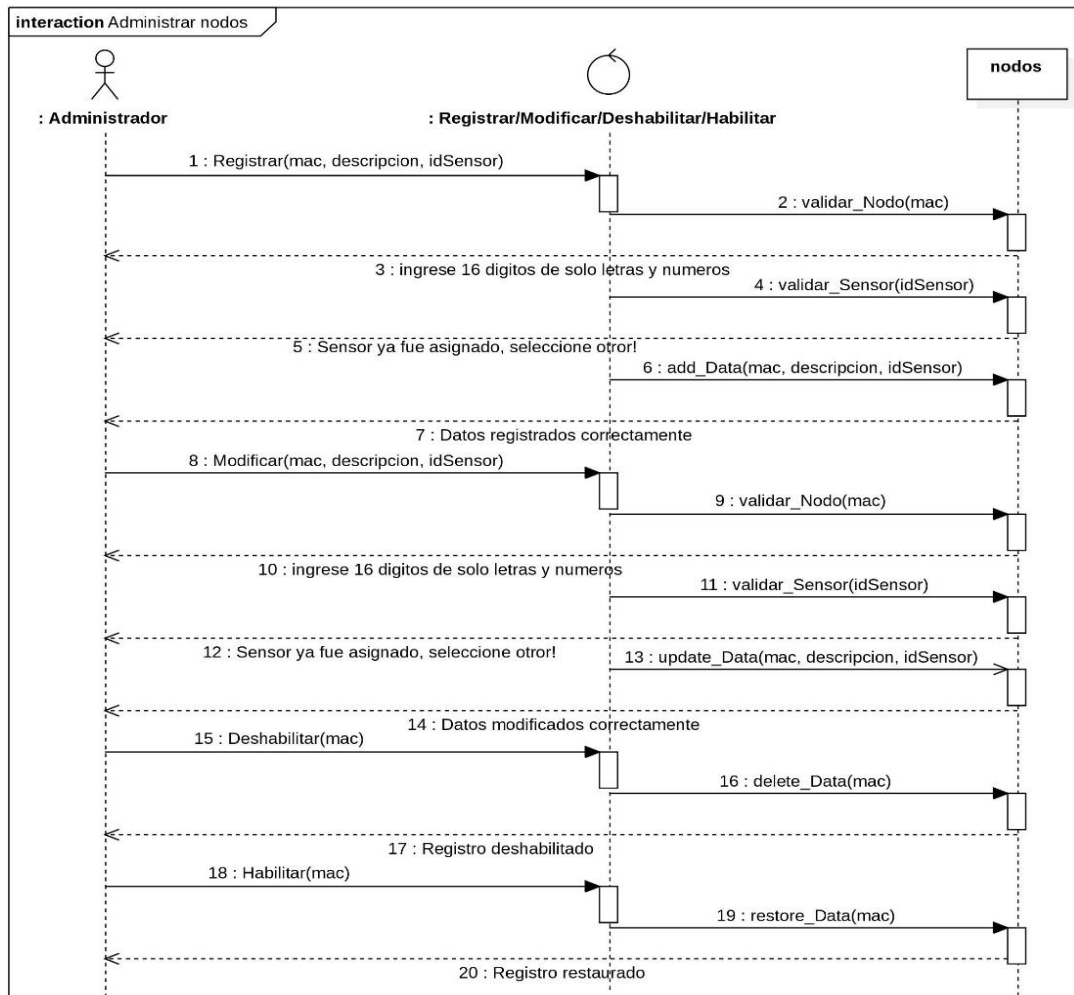


Figura 28: Diagrama de secuencia administrar nodo.
Fuente: Elaboración propia.

4.1.2. Requisitos no funcionales

Los requerimientos no funcionales se identificaron según las historias del cliente, los cuales se mencionan en la Tabla 11.

Tabla 11. Requerimientos no funcionales.

N°	Requerimiento No Funcional
01	El usuario necesita estar conectado a la red de la Institución Educativa para poder acceder al sistema web del monitoreo del consumo de agua.
02	El prototipo debe ofrecer facilidad en su uso por el usuario
03	El prototipo debe ofrecer disponibilidad para que el usuario pueda acceder en el momento que desee.
04	La interfaz del sistema web debe ser amigable para el usuario y poder utilizarse de forma fácil y sencilla.

Fuente: Elaboración propia.

4.1.3. Roles del proyecto

La Tabla 12 contiene los roles asignados en el desarrollo del presente trabajo de tesis.

Tabla 12: Roles del Proyecto

Rol:	Responsable:
Cliente	Director de la I.E.I Federico Villareal
Programador	Rogelio Romani Flores
Encargado de Pruebas (Tester)	Carlos Yinmel Castro Buleje
Encargado de Seguimiento(Tracker)	Carlos Yinmel Castro Buleje
Entrenador (Coach)	Carlos Yinmel Castro Buleje
Consultor	Rogelio Romani Flores
Jefe del Proyecto	Carlos Yinmel Castro Buleje

Fuente: Elaboración propia.

4.1.4. Plan de iteraciones y entregas

De acuerdo de las historias definidas por el cliente, se ha elaborado el cronograma de desarrollo de las historias por iteraciones y su entrega de las mismas. Como se muestra en la Tabla 13, el cual contiene el número de historias del cliente definidas en la sección anterior y se agrupan por iteraciones.

Tabla 13: Plan de entregas del proyecto.

Historias	Iteración	Fecha Inicio	Fecha Fin
Historia 01	1	01/01/19	31/01/19
Historia 02	1	01/01/19	31/01/19
Historia 03	2	01/02/19	06/02/19
Historia 04	3	07/02/19	11/03/19
Historia 05	4	12/03/19	18/03/19

Fuente: Elaboración propia.

4.1.5. Requerimientos del prototipo

En la Tabla 14 se puede ver el requerimiento de hardware y software para la implementación del prototipo.

Tabla 14: Requerimiento de hardware y software

Requerimientos de hardware		
N°	Descripción	Cantidad
1	Arduino uno	2
2	Sensor de flujo	1
4	Xbee	2
5	Shield Xbee	2
6	Shield Ethernet	1
Requerimientos de software		
N°	Descripción	Cantidad
1	Arduino	1
2	XCTU	1
3	XampServer	1
4	SublimeText	1
5	Fritzing	1

Fuente: Elaboración propia.

4.2. Diseño

Esta segunda fase de la metodología XP aplicado al trabajo de tesis, comprende el diseño del prototipo, diseño del sistema web, diseño de la base de datos, diseño de la topología física de la red. Los mencionados diseños han sido desarrollados por iteraciones como fue previsto en la fase de planificación.

4.2.1. Primera iteración

En esta iteración se diseñó el nodo sensor, nodo coordinador.

a. Diseño del nodo sensor

El nodo sensor es el responsable de leer los datos a través del sensor de flujo y transmitir al nodo coordinador, la transmisión se realiza vía radiofrecuencia, mediante tramas de datos, al nodo coordinador.

o Conexión del nodo sensor

En la Figura 29, se hace una conexión entre el Arduino uno y la tarjeta shield, los pines 0 y 1 permiten la comunicación serial entre ambas tarjetas. Por otro lado, se conecta los pines 0 (rx) y 1 (tx) de la tarjeta shield a los pines 2 (tx) y 3 (rx) del Xbee respectivamente para la recepción y transmisión de datos del sensor de flujo, el cual está conectado al pin digital 2 de la tarjeta shield.

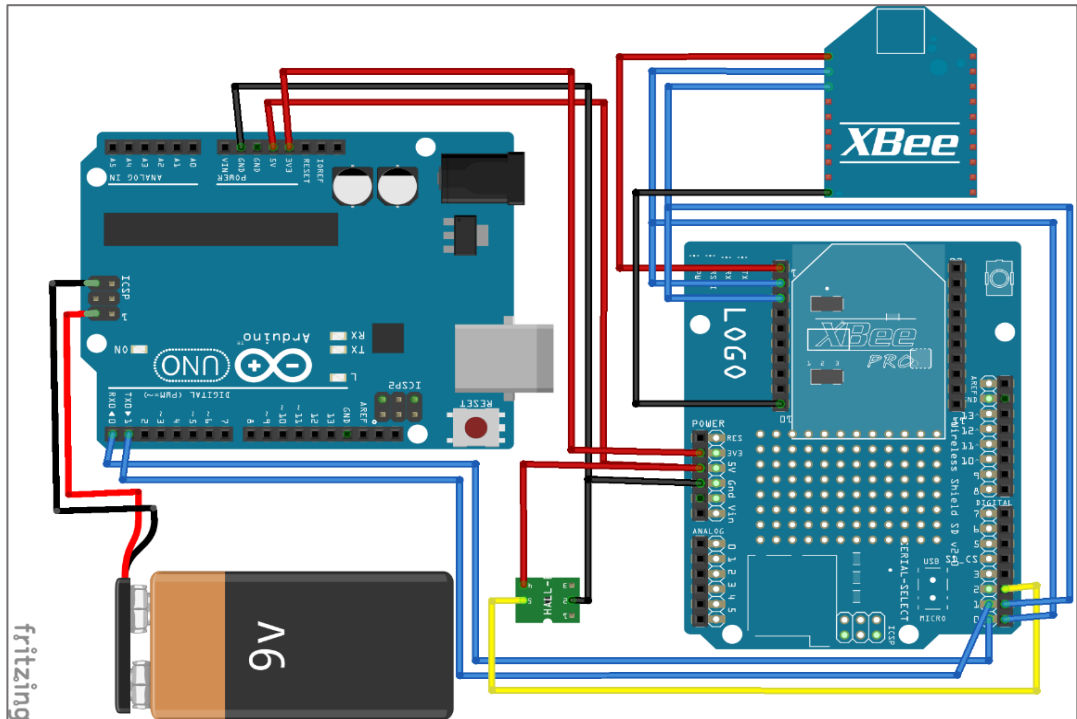


Figura 29: Conexión del nodo sensor.
Fuente: Elaboración propia.

o **Conexión esquemática del nodo sensor**

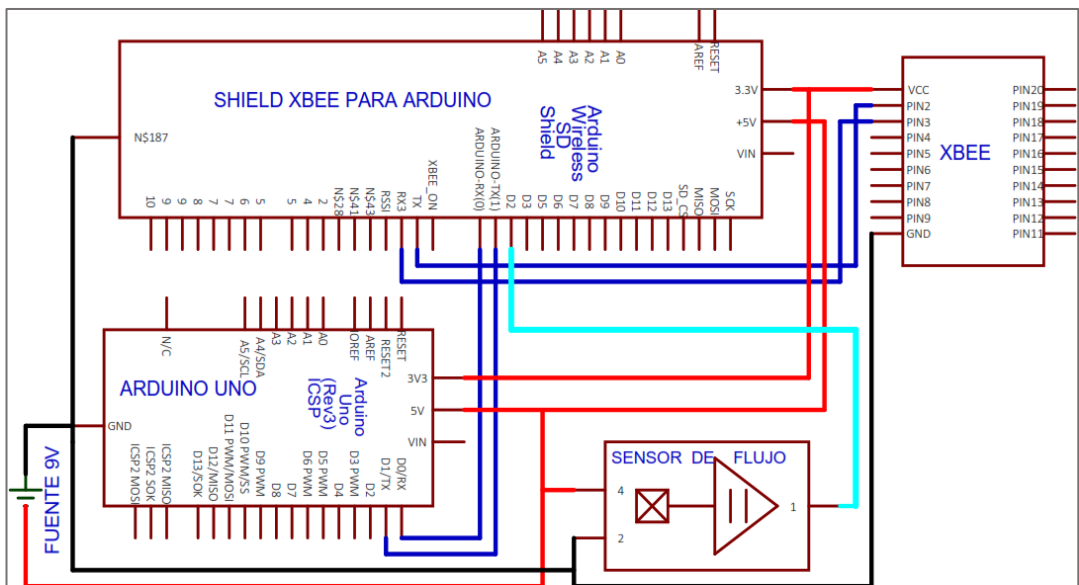


Figura 30: Conexión esquemático del nodo sensor.
Fuente: Elaboración propia.

b. Diseño del nodo coordinador

El nodo coordinador es el responsable de formar la red inalámbrica, para este módulo se empleó la tarjeta shield Ethernet para la conexión del Arduino con el servidor. El shield Ethernet contiene un conector RJ45, también se usó la tarjeta shield para el Xbee que permite la comunicación con el nodo sensor.

- **Conexión del nodo coordinador**

En la Figura 31, se hace una conexión de las tarjetas shield con el Arduino. El Arduino uno usa los pines digitales 10, 11, 12, 13 para comunicarse con la tarjeta shield Ethernet, los pines 0 y 1 del Arduino uno, de la tarjeta shield Ethernet, de la tarjeta shield Xbee se conectan entre sí para lograr una comunicación serial entre las tres tarjetas. Finalmente se conecta los pines 0 (rx) y 1 (tx) de la tarjeta shield a los pines 2 (tx) y 3 (rx) del Xbee respectivamente para la recepción y transmisión de datos.

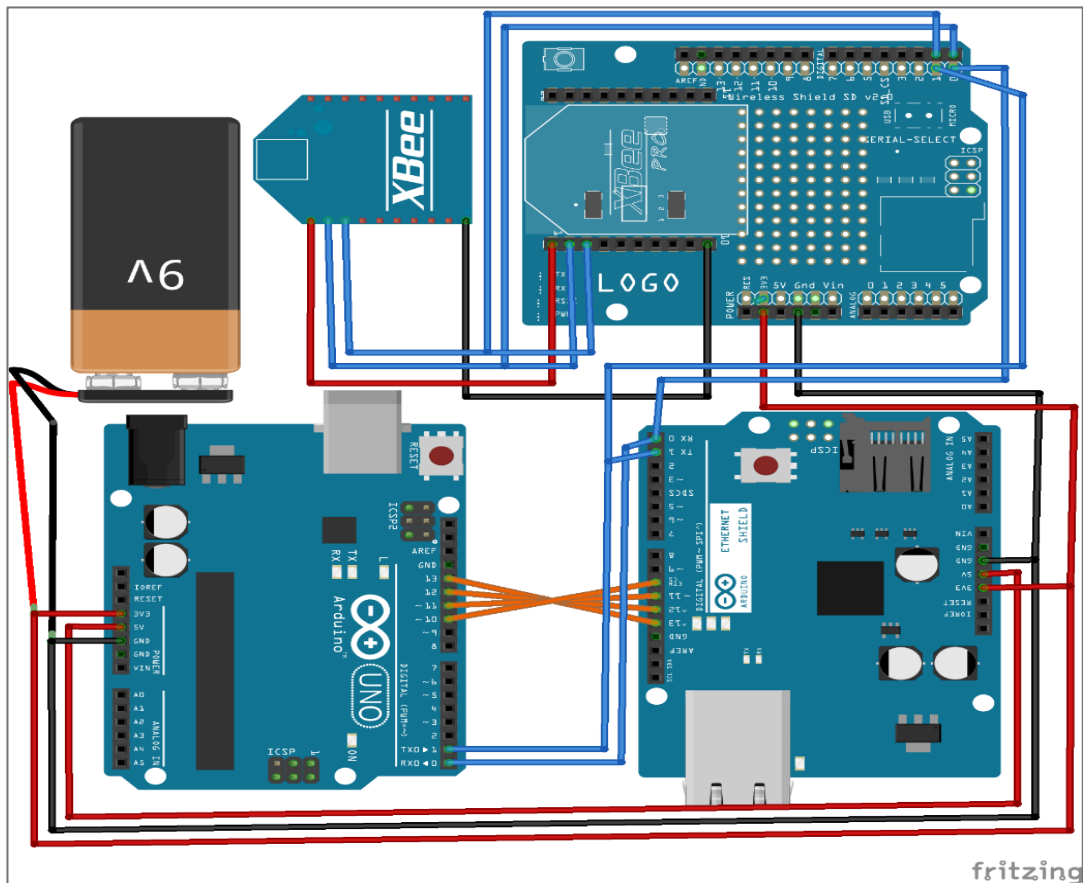


Figura 31: Conexión del nodo coordinador.
Fuente: Elaboración propia.

- **Conexión esquemática del nodo coordinador**

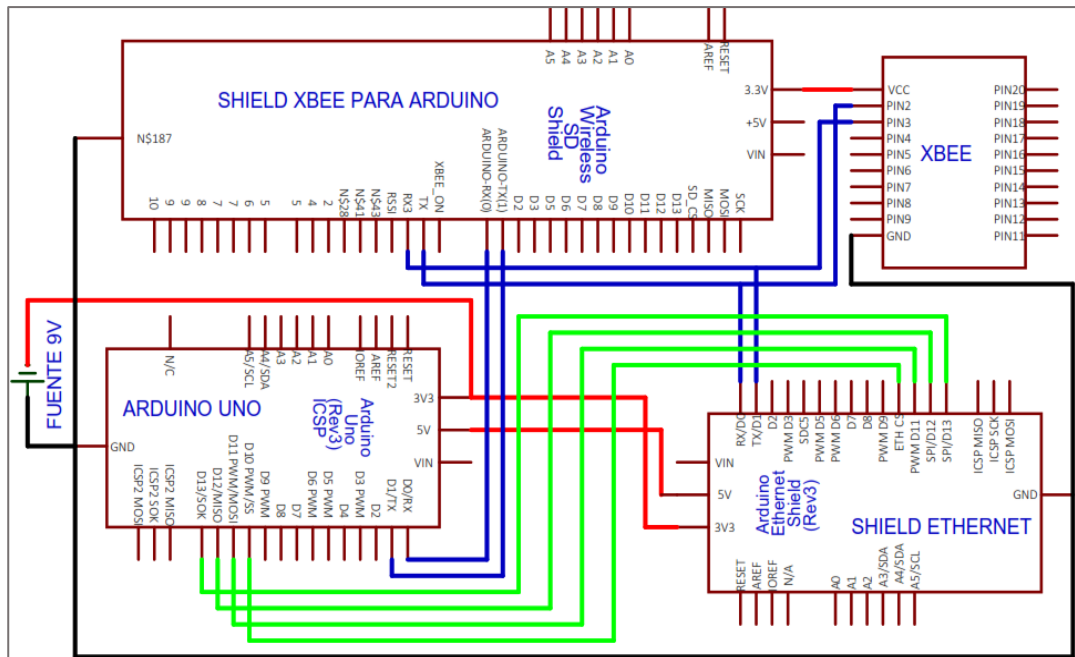


Figura 32. Conexión esquemática del nodo coordinador.
Fuente: Elaboración propia.

4.2.2. Segunda iteración

Para el almacenamiento de los datos enviados por el nodo sensor se ha diseñado la base de datos mediante el software MySQL Workbench como se muestra en la Figura 33. La estructura de la base de datos cuenta con cinco tablas, donde la tabla p_datos almacena los mililitros que se envían desde el nodo sensor. Además la tabla p_datos guarda relación de muchos a uno con la tabla p_nodo que guarda los datos de los nodos.

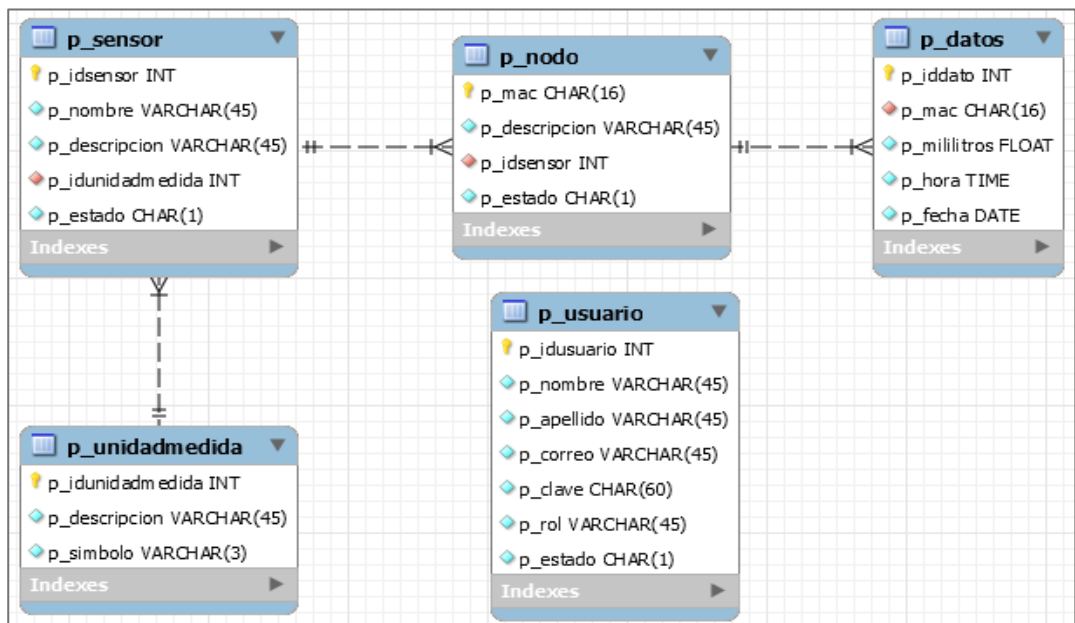


Figura 33. Esquema de la base de datos.
Fuente: Elaboración propia.

4.2.3. Tercera iteración

En esta iteración se diseñó la interfaz del sistema web que permite el monitoreo del consumo de agua a través de la red inalámbrica. Previamente la interfaz se ha diseñado de forma esquemática con la ayuda del programa Balsamiq Mockups, como se muestra en la Figura 34, en donde está definido las secciones que tiene el sistema.

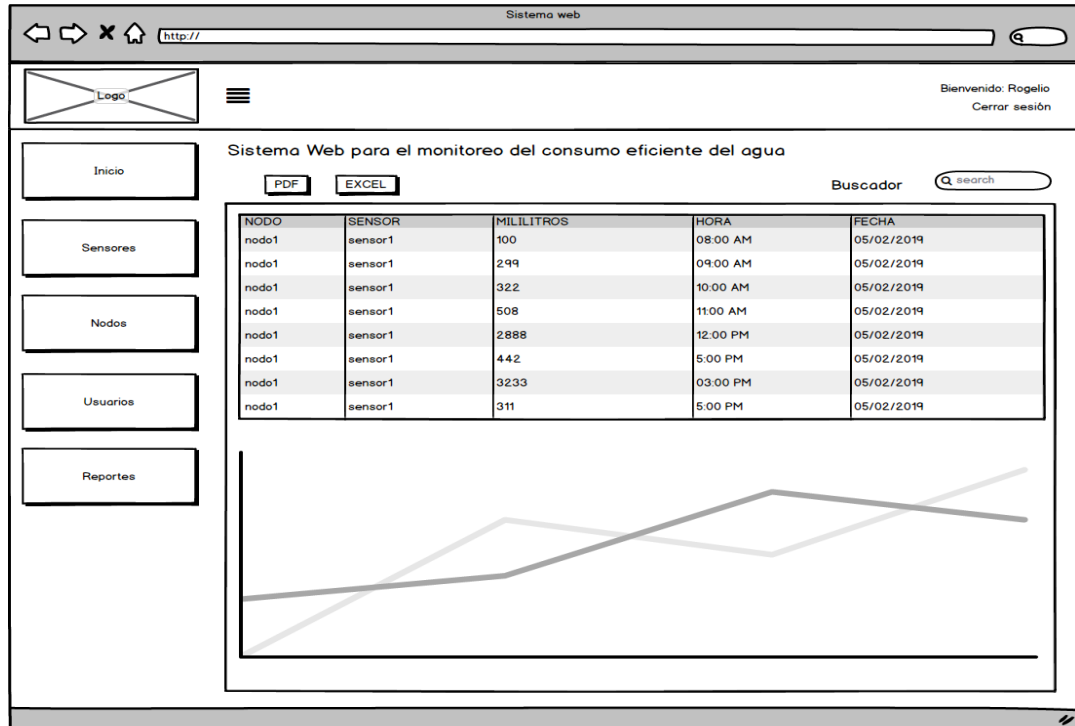


Figura 34. Interfaz general del sistema web.
Fuente: Elaboración propia.

El sistema tiene la seguridad para el acceso, para ello se ha diseñado la interfaz de inicio de sesión, que se muestra previamente a la página principal si el usuario aún no ha iniciado sesión. Como se observa en la Figura 35, la interfaz muestra un formulario para el ingreso de usuario y contraseña.

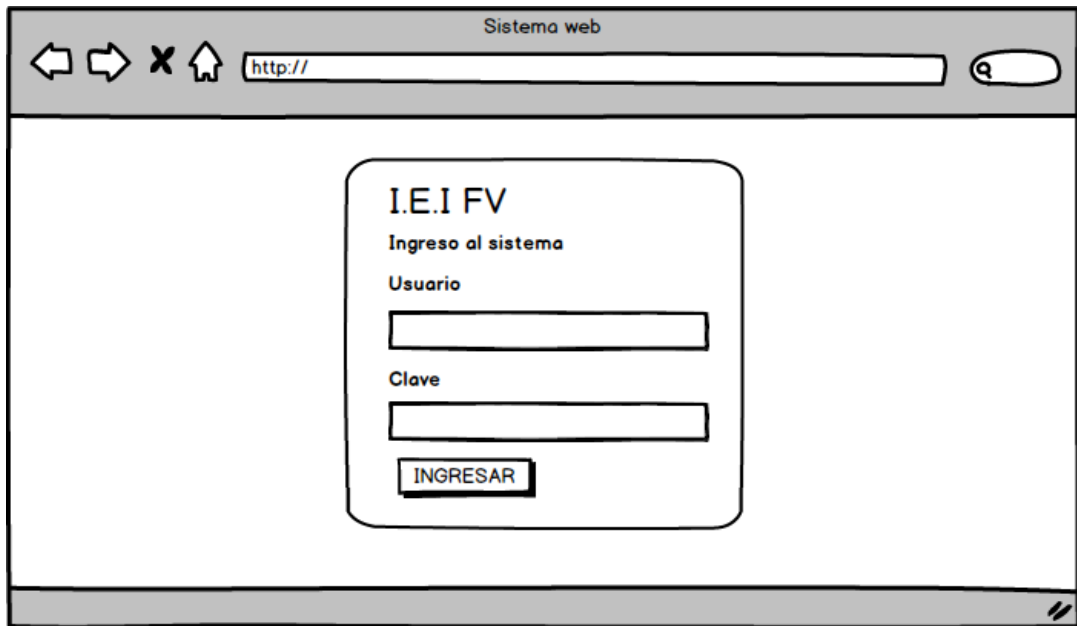


Figura 35. Interfaz de inicio de sesión.
Fuente: Elaboración propia.

La sección nodos del sistema permite registrar, actualizar nodos de la red inalámbrica. Para lo cual se ha diseñado una interfaz como se muestra en la Figura 36.

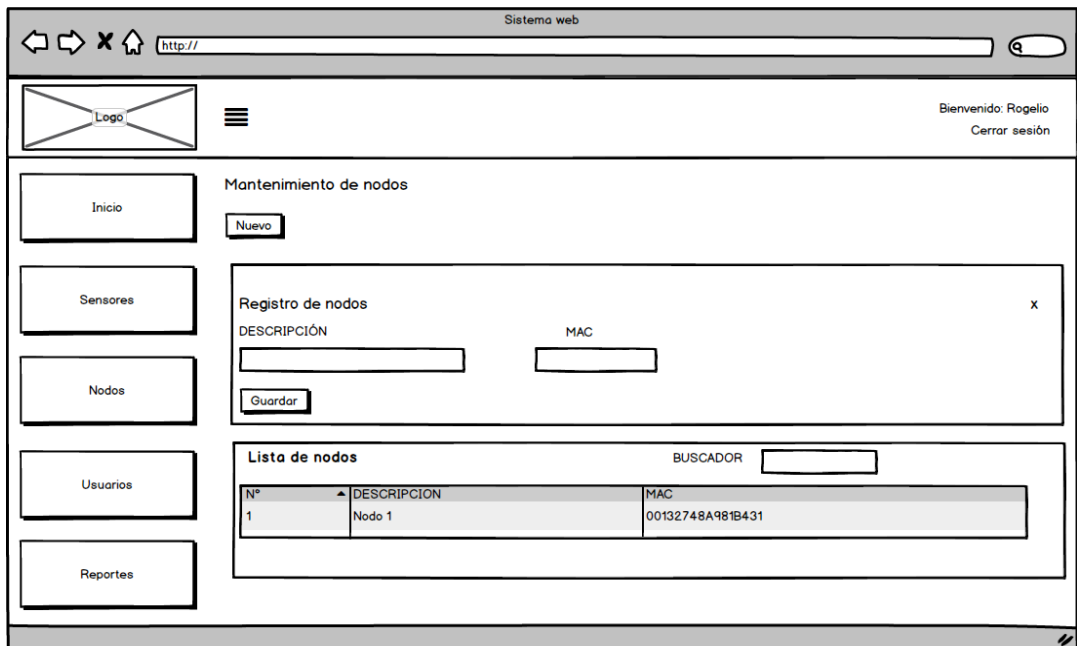


Figura 36. Interfaz de registro de nodos.
Fuente: Elaboración propia.

Por último, el sistema tiene la sección de reportes, que permite generar reportes por fechas. Para ello se ha diseñado una interfaz como se observa en la Figura 37.

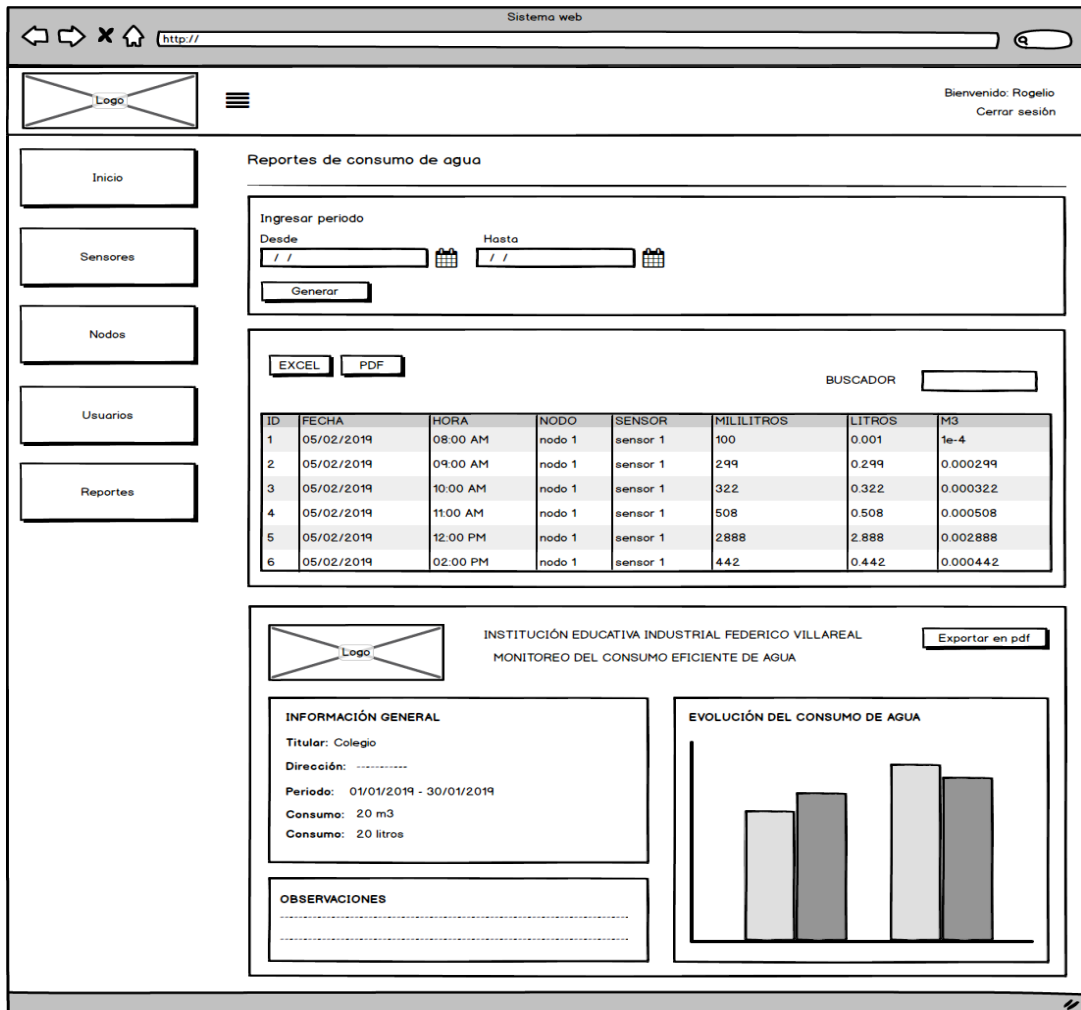


Figura 37. Interfaz de reportes.
Fuente: Elaboración propia.

4.2.4. Cuarta iteración

Se diseñó la topología física de la red inalámbrica según la ubicación de la llave principal de agua de la Institución Educativa Industrial Federico Villareal, como se muestra en la Figura 38. Donde se tiene 1 nodo sensor de flujo, 1 nodo coordinador y finalmente se tiene una computadora de escritorio como servidor para el almacenamiento de los datos ubicado en el aula de innovación tecnológica.

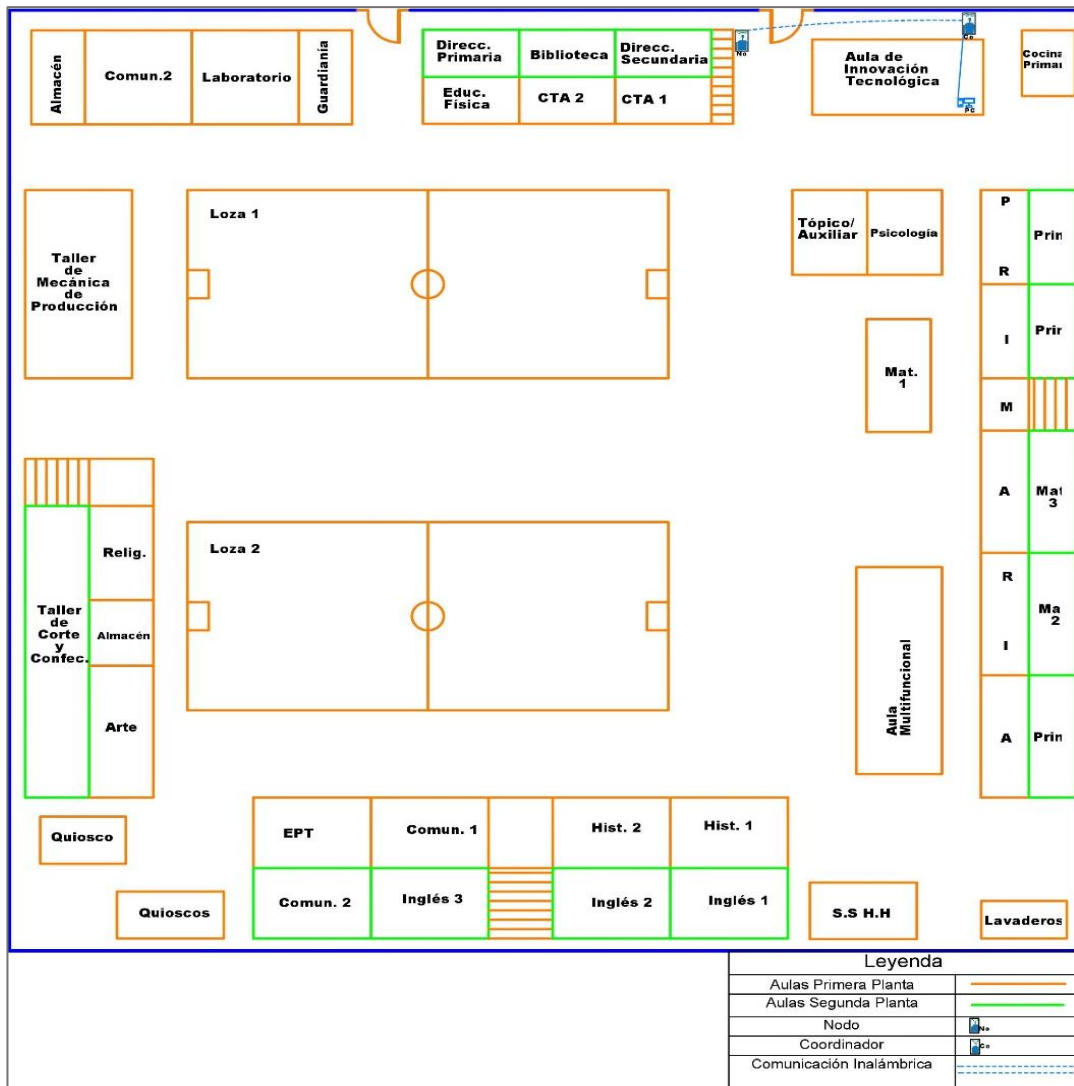


Figura 38. Topología Física de la red.
Fuente: Elaboración propia.

4.3. Desarrollo

En el orden de las fases definidas por la metodología de la Programación Extrema y siguiendo las iteraciones predefinidas en la fase de planificación, esta fase comprende la configuración de los dispositivos Xbee utilizando el software XCTU, codificación de tramas en el software Arduino, codificación del sistema web en el lenguaje php.

4.3.1. Primera iteración

Para la configuración de los módulos Xbee se utilizó el software XCTU versión 6.4.2 creado por la empresa Digi International, el software proporciona una interfaz gráfica para realizar la configuración de manera sencilla los módulos Xbee. También se empleó el software Arduino para la codificación de tramas API, que permite la comunicación del nodo coordinador con el nodo sensor.

a. Configuración del nodo coordinador

El nodo coordinador está configurado en modo API para que pueda transmitir una trama de petición (cabecera + datos + checksum) al nodo sensor que pertenecen al mismo canal de comunicación.

En la Tabla 15, se define los parámetros de configuración como el PAN ID que debe ser único para todos los nodos de la red, al igual el canal de comunicación y también habilitado el modo api para el nodo.

Además, el nodo coordinador se encarga de recibir las tramas enviadas por el nodo sensor y transmitir por el medio de cable Ethernet a un servidor que contiene la base de datos.

Tabla 15. Parámetros de configuración del nodo coordinador.

Modo API		
Parámetro	Descripción	Valor
ID	Identificador de la red PAN	10
SC	Canales de escaneo	7FFF
CE	Coordinador Habilitado	Enabled[1]
NI	Identificador de nodo	COORDINADOR
AP	API habilitado	API enabled[1]

Fuente: Elaboración propia.

El procedimiento de configuración para este nodo coordinador empieza conectado el Xbee en la tarjeta xbee explorer usb y esta última se conecta por cable usb al ordenador, como se muestra en la Figura 39.



Figura 39. Conexión del Xbee al ordenador.
Fuente: Elaboración propia.

Después de conectarlo se abre el software XCTU en el ordenador en el cual primero se actualizó el firmware del Xbee (ver Figura 40) y luego se registró los valores de los parámetros definidos anteriormente (ver Figura 41).

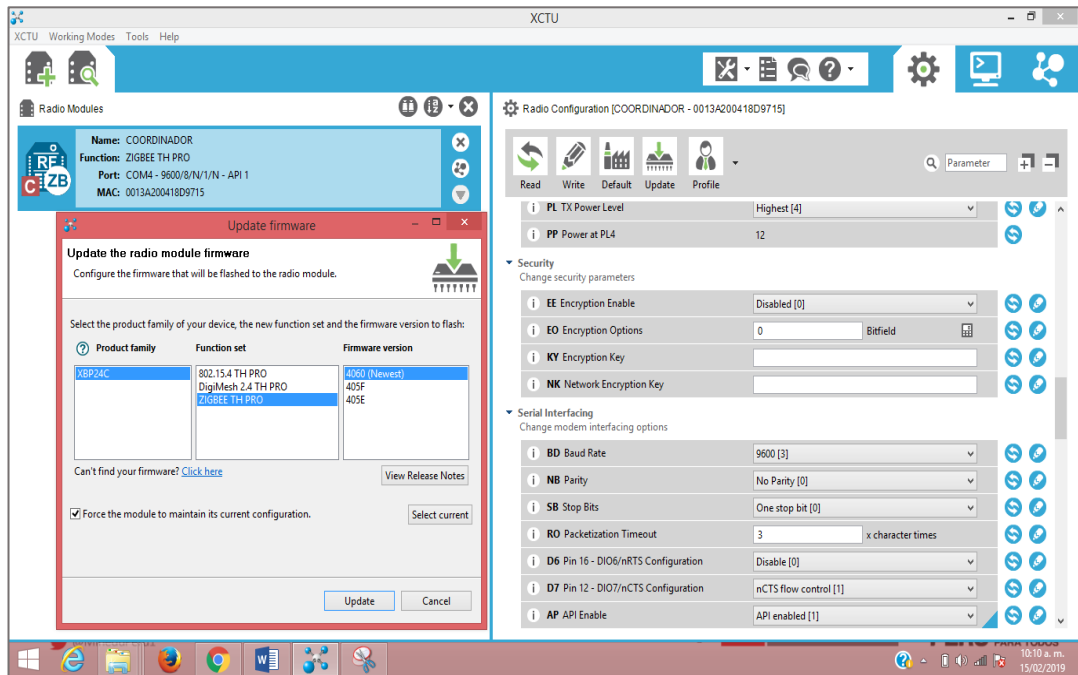


Figura 40. Actualización del firmware del nodo coordinador.
Fuente: Elaboración propia.

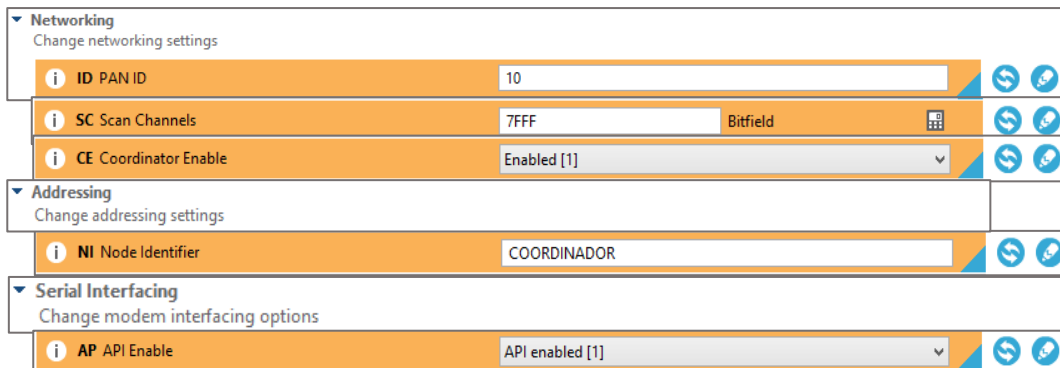


Figura 41. Registro de parámetros del nodo coordinador en el software XCTU.
Fuente: Elaboración propia.

b. Configuración del nodo sensor

Para el nodo sensor de la red, los parámetros de configuración se definen en la Tabla 16.

Tabla 16. Parámetros de configuración del nodo sensor.

Modo API		
Parámetro	Descripción	Valor
ID	Identificador de la red PAN	10
SC	Canales de escaneo	7FFF
CE	Coordinador Habilitado	Disabled[0]
NI	Identificador de nodo	Nodo 1
AP	API habilitado	API enabled[1]

Fuente: Elaboración propia.

En la Figura 42, se actualizó el mismo firmware del nodo coordinador también para el nodo sensor, con el propósito de que ambos nodos se comuniquen de manera eficiente.

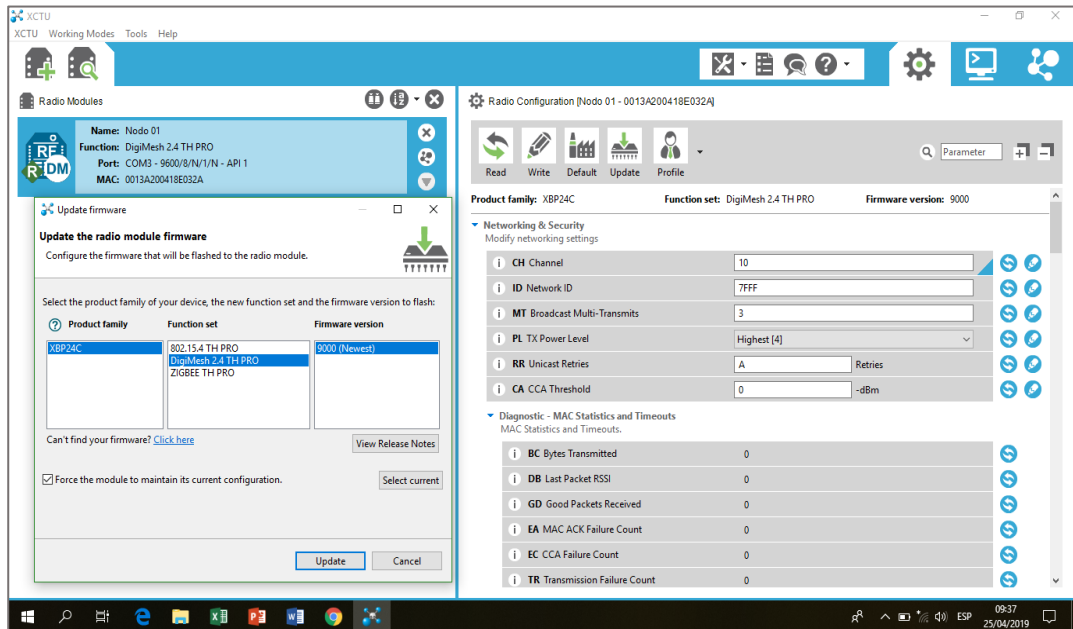


Figura 42. Actualización del firmware del nodo sensor.

Fuente: Elaboración propia.

Para el registro de los valores del nodo sensor se conecta el Xbee al puerto usb del ordenador, para que el software XCTU pueda reconocer al Xbee y permitir su configuración. Tal como se muestra en la Figura 43, donde se registró los valores.

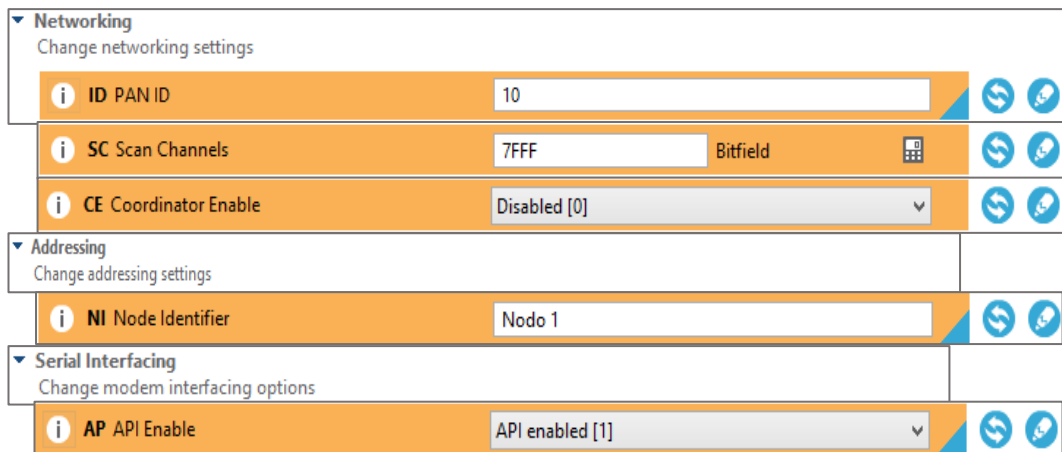


Figura 43. Registro de parámetros del nodo sensor en el software XCTU.
Fuente: Elaboración propia.

c. Prueba de comunicación

Después de configurar los módulos Xbee, se prueba la comunicación entre el nodo coordinador y el nodo sensor, siendo los resultados satisfactorios como se muestra en la Figura 44.

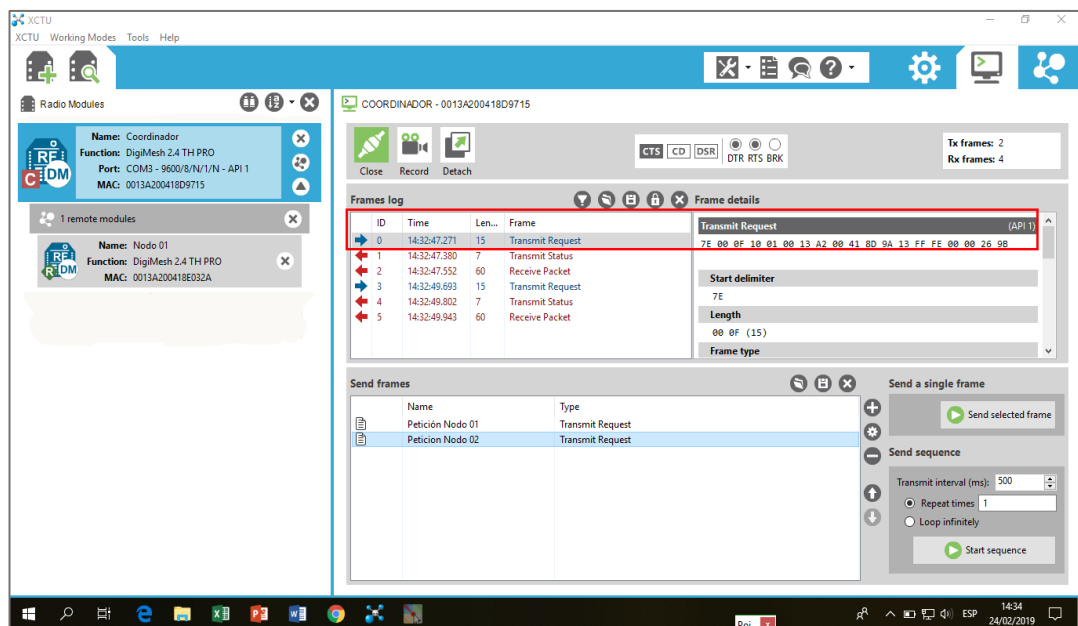


Figura 44: Prueba de comunicación entre el coordinador y el nodo sensor.
Fuente: Elaboración propia.

d. Creación de la trama api para el nodo coordinador

En el software XCTU se creó la trama específica de petición de datos al nodo sensor. De esta forma se estableció un protocolo común de comunicación entre el nodo coordinador con el nodo sensor. Cuando se trabaja en el modo API los datos deben ser empaquetados en un tipo de trama API, en este caso el tipo de trama que se utilizó es la solicitud de transmisión definido con el valor de 0x10 hexadecimal, la estructura de la trama en el software XCTU es como se muestra en

la Figura 45. Los datos que no son empaquetados en este tipo de trama simplemente serán ignorados.

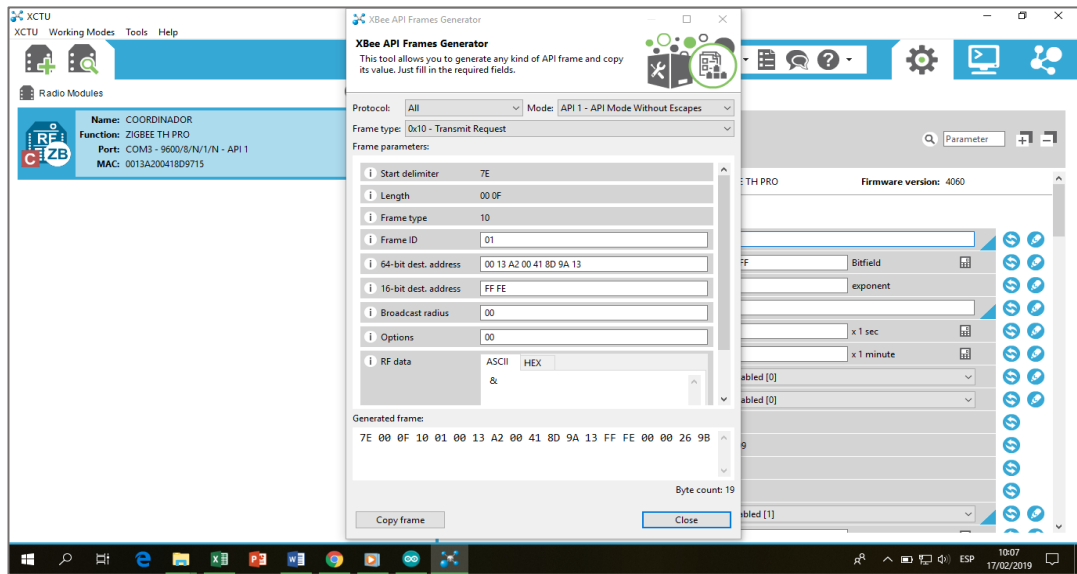


Figura 45. Generación de trama API en el nodo coordinador.
Fuente: Elaboración propia.

A partir de la estructura de la trama presentada en el software XCTU, se detalla las partes de la trama de petición de datos al nodo sensor como se muestra en la Tabla 17. La trama contiene un encabezado de 17 bytes, 1 byte de datos y 1 byte de comprobación de trama.

Tabla 17. Estructura de la trama de petición.

CABECERA 0x10 TRANSMIT REQUEST								DATOS	CHECKSUM
Inic io	Long itud	Tip o	Id	Direcci ón 64bits	Direcc ión 16bits	Broad cast radius	Opci ón	Data	Checksum
1 byte	2 bytes	1 byte	1 byte	8 bytes	2 bytes	1 byte	1 byte	1 byte	1 byte
7E	00 0F	10	01	direcci ón mac destino	FF FE	00	00	&	9B

Fuente: Elaboración propia.

El detalle de la estructura de la trama de petición de datos se describe a continuación:

- ✓ El valor hexadecimal 7E indica el inicio de trama.
- ✓ La longitud de la trama es el número de bytes desde el tipo de trama hasta la data, en este caso tiene el valor de hexadecimal 0F que indica 15 bytes.
- ✓ El valor hexadecimal del tipo de trama es 10.
- ✓ El valor hexadecimal del id de trama es 01.

- ✓ La dirección de 64 bits, es la dirección mac de destino de 8 bytes del nodo sensor.
- ✓ La dirección de 16 bits, es la dirección de red de 2 bytes. En este caso tiene el valor hexadecimal FF FE indica una dirección desconocida y es una transmisión de broadcast.
- ✓ El broadcast radius tiene el valor hexadecimal 00, con este valor el radio de transmisión se establece en el número máximo de saltos.
- ✓ El valor hexadecimal de la opción es 00.
- ✓ La parte de datos de la trama contiene el mensaje que se transmite al nodo sensor, en este caso se utiliza el símbolo de ampersand (&) como mensaje de petición de datos al nodo sensor.
- ✓ Finalmente, la trama se comprueba a través del checksum, que en este caso tiene el valor 9B en hexadecimal.

- **Código Arduino para la petición de datos**

El código Arduino de la trama de petición de datos al nodo sensor se muestra a continuación en el código 1, donde se establecen variables como por ejemplo la constante ZiseAPI de tipo byte que contiene la cantidad de bytes de la trama de petición, además se define un array Nodo_01[ZiseAPI] de tipo byte, en el cual se declara los bytes de la trama de petición que son 19 bytes. La declaración del array contiene los bytes de la trama de petición al nodo sensor.

Luego se inicializa el serial en el void setup y dentro del lazo void loop se realiza la petición de datos al nodo sensor a través de la sentencia if. Dentro de la sentencia if se recorre a través del for los bytes declarados en el array Nodo_01[ZiseAPI] e ir imprimiendo cada uno de los bytes en la salida del serial para su transmisión. Esta iteración de petición se realiza cada 13 segundos aproximadamente al nodo sensor.

```

sketch_jun01a §
#define ContSegundo 100000 //Contador que permite establecer un tiempo aproximado de un minuto
#define EnvioPeticion 3 //Peticion cada 3 segundos
unsigned long MyTimeSeg;
byte RealizarPeticion = 0;
const byte ZiseAPI =19; //Cantidad de bytes de la trama
//BYTES DE LA TRAMA DE PETICION PARA EL NODO SENSOR
byte Nodo_01[ZiseAPI] = {0x7E, 0x00, 0x0F, 0x10, 0x01, 0x00, 0x13, 0xA2, 0x00, 0x41,
                        0x8E, 0x03, 0x2A, 0xFF, 0xFE, 0x00, 0x00, 0x26, 0x1A };

void setup() {
  Serial.begin(9600); //la velocidad de transmision
}

void loop() {
  if (MyTimeSeg > ContSegundo){
    MyTimeSeg = 0;
    RealizarPeticion ++;
  }
  if (RealizarPeticion > EnvioPeticion){
    RealizarPeticion = 0;
    mySerial.println("Envio Trama de Petición");
    for (byte i = 0; i < ZiseAPI ; i ++ ){
      Serial.write((byte)Nodo_01[i]);
    }
  }
}
}

```

Código 1: Código para la trama de petición.
Fuente: Elaboración propia.

e. Creación de la trama api para el nodo sensor

Para que el nodo sensor pueda responder a la petición del coordinador, sea definido una estructura específica de trama API con una cabecera de tipo paquete recibido (0x90 hexadecimal), como se muestra en la Tabla 18. La trama tiene 15 bytes de cabecera, 48 bytes de datos y 1 byte de checksum.

Tabla 18. Estructura de la trama de envío de datos.

CABECERA 0x90 RECEIVE PACKET						DATOS	CHECKSUM
Inicio	Longitud	Tipo	Dirección 64bits	Dirección 16bits	Opción	Data	Checksum
1 byte	2 bytes	1 byte	8 bytes	2 bytes	1 byte	48 bytes	1 byte
7E	00 3C	90	dirección mac origen	FA A2	01	Mililitros	73

Fuente: Elaboración propia.

El detalle de la estructura de la trama API para el nodo sensor se describe a continuación:

- ✓ El valor hexadecimal del inicio de la trama es siempre 7E.
- ✓ La longitud de la trama es siempre de 2 bytes, en este caso tiene el valor de 3C en hexadecimal.
- ✓ El tipo de trama es 90 en hexadecimal, denominado paquete recibido según la hoja de datos del Xbee.

- ✓ La dirección de 64 bits hace referencia a la dirección mac del origen en este caso del nodo sensor.
- ✓ La dirección de 16 bits hace referencia a la dirección de origen de la red.
- ✓ El valor hexadecimal de la opción es 01, que significa admitido.
- ✓ En la parte de datos se ha definido 48 bytes de datos como se muestra en la Figura 46, estos datos son adjuntados a la cabecera de la trama API para su envío al nodo coordinador. Donde el símbolo ampersand (&) indica byte de arranque, los dos puntos son bytes separadores, los ceros representan las variables que leen los sensores, que para el caso del proyecto los ceros de color azul de la figura representan los mililitros que leyó el sensor de flujo. Siendo esta variable que recupera el coordinador para su registro en el servidor de la base de datos. Y finalmente el símbolo numeral (#) indica byte de parada.

La razón de la estructura de datos que se creó, es porque el sensor de flujo envía desde la mínima cantidad de mililitros que sería igual a 1000. Siendo este el caso de un mililitro convertido a binario sería 1111101000, donde este valor no está dentro de un byte. Por lo que la solución fue agrupar 8 bytes y la cantidad máxima de mililitros que enviará el sensor de flujo será de 99999999 considerando cuando se dé el caso.

&:0:00000000:00000000:00000000:00000000:000000:#

Figura 46. Datos en la estructura de la trama del nodo sensor.

Fuente: Elaboración propia.

- ✓ El último byte de la trama es el checksum, que comprueba la integridad de la trama antes de enviar.

Después de definir la estructura de la trama de envío de datos del sensor de flujo, se codifica en el software Arduino la misma estructura definida. El cual se muestra en el fragmento de código 2, donde se tiene declarado en variables de tipo byte las partes individuales de la trama. Siendo esta estructura de trama que toma el nodo sensor para enviar los mililitros hacia el coordinador.

```

const byte Trm_dat = 48; //cantidad de 48 bytes que se adjuntaran a la trama
byte Array_Trm_dat[Trm_dat+1]; //almacena en un array los 48 datos de la trama

//GENERACION DE TRAMA
const byte Dltr_init = 0x7E;//delimitador de inicio de la trama
const byte Long_Msb = 0x00; //Longitud de la trama con bit más significativos
const byte Long_Lsb = 0x3E; //Longitud de la trama con bit menos significativos
const byte Frame_Ty = 0x10; //tipo de trama
const byte Frame_ID = 0x01; //id de trama
const byte Address64[8] = {0x00, 0x13, 0xA2, 0x00, 0x41, 0x8D, 0x97,0x15}; //mac del coordinador
const byte Address16[2] = {0xFF, 0xFE}; //direccion broadcast de red
const byte BroadCast = 0x00; //establece el numero maximo de saltos
const byte Option = 0x00; //Opcion de comando AT
byte ByteChecksum=0; //comprobación de la trama

```

*Código 2: Código de trama para el nodo sensor.
Fuente: Elaboración propia.*

- **Código para la lectura del sensor de flujo**

La salida que emite el sensor de flujo son pulsos eléctricos, que forman una onda cuadrada cuya frecuencia es proporcional al caudal atravesado por el sensor. Dicha afirmación corresponde a la siguiente formula $Q(l * min) = \frac{f(Hz)}{k}$, que viene dada en la hoja de datos del sensor.

De la formula, el factor de conversión k viene de fábrica con valor de referencia de 7.5 para el caso del sensor de 1/2", que se usó en el trabajo de tesis. Sin embargo, este valor de referencia admite un error de precisión de +-10%. Para mejorar el error de precisión se realizó la calibración del sensor, y para ello sea considerado medir los pulsos para los volúmenes de 1 litro, 2 litros y 3 litros en un recipiente graduado para estos volúmenes, y los resultados de calibración se muestran en la siguiente Tabla 19.

Tabla 19: Resultados de calibración del sensor para el nodo 01

N° Muestras	Volumen					
	1 Litro		2 Litros		3 Litros	
	Pulsos	K	Pulsos	K	Pulsos	K
1	384	6.4	767	6.39	1151	6.39
2	418	6.96	823	6.85	1256	6.97
3	388	6.46	776	6.46	1165	6.47
4	379	6.31	790	6.58	1181	6.56
5	391	6.51	783	6.52	1174	6.52
6	413	6.88	817	6.80	1241	6.89
7	417	6.95	814	6.78	1261	7.00
8	419	6.98	819	6.82	1258	6.98
9	395	6.58	791	6.59	1187	6.59
10	392	6.53	785	6.54	1178	6.54
Promedio Parcial	399.6	6.66	796.5	6.63	1205.2	6.69
Constante k obtenida para el sensor	6.66					

Fuente: Elaboración propia.

Para poder obtener los resultados de la Tabla 19 se desarrolló el código 3 que se muestra a continuación, el cual hace posible la medición de los puntos que envía el sensor de flujo de agua. Luego para obtener el factor k de conversión se usó la siguiente fórmula $K = \frac{n^{\circ} \text{ Pulsos}}{\text{Volumen} * 60}$, finalmente se promedió los valores del factor de conversión k, este valor obtenido para la constante k que fue reemplazado más adelante en la siguiente fórmula $Q(l * min) = \frac{f(Hz)}{k}$ para calcular el caudal del agua.

```
volatile long NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 2; //Sensor conectado en el pin 2

//---Función que se ejecuta en interrupción-----
void ContarPulsos ()
{
  NumPulsos++; //incrementamos la variable de pulsos
}

void setup(){
  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0,ContarPulsos,RISING);//(Interrupción 0(Pin2),función,Flanco de subida)
  interrupts(); //Habilitamos las interrupciones
}

void loop (){
  //-----Enviamos por el puerto serie-----
  Serial.print ("Numero de Pulsos = ");
  Serial.println (NumPulsos);
  delay(100);
}
```

*Código 3: Código para obtener los pulsos del sensor.
Fuente: Elaboración propia.*

Para la medición del flujo de agua a través del sensor, se ha escrito el código 4 que se muestra a continuación. Donde se tiene la variable PinSensor como entrada del sensor de flujo a través del pin digital 2 del arduino uno, la variable Mili_litros de tipo float almacena los mililitros acumulados, también se tiene una función CuentaPulsos que devuelve los pulsos a través de la variable NumPulsos de tipo entero que guarda los pulsos recibidos durante el flujo de agua. También se ha definido una función ObtenerFrecuencia(), donde esta función retorna la frecuencia de los pulsos que varía según la velocidad del flujo de agua.

Luego de la declaración de las variables y de las funciones definidas, primero se realiza el cálculo del caudal de flujo en litros por minuto a través de la fórmula $Q(l * min) = \frac{f(Hz)}{k}$, donde se reemplaza el valor promedio de la constante k obtenido anteriormente en la calibración del sensor. Teniendo el caudal conocido se procede a calcular el volumen de agua, puesto que el caudal es la variación del volumen de agua con respecto al tiempo a través de la siguiente fórmula $V = V_0 + Q\Delta t$. Y

finalmente el volumen de agua obtenido en litros se convierte a mililitros, el cual se envía al nodo coordinador cada vez que este solicita al nodo sensor.

```
// VARIABLES DEL SENSOR DE FLUJO
int PinSensor = 2;           //Pin digital para el sensor de flujo YF-S201
float Litros;                //Variable que almacena el caudal (L/min)
float Mili_litros;          //Variable que almacena el número de mililitros acumulados
volatile int NumPulsos;     //Variable para la cantidad de pulsos recibidos
long dt=0; //variación de tiempo por cada bucle
long t0=0; //millis() del bucle anterior
//CALCULO DE VOLUMEN DEL AGUA
float frecuencia=ObtenerFrecuencia(); //obtenemos la frecuencia de los pulsos en Hz
float caudal_L_m=frecuencia/5.9;      //calculamos el caudal en L/m
dt=millis()-t0;                       //calculamos la variación de tiempo
t0=millis();
Litros=Litros+(caudal_L_m/60)*(dt/1000); //Litros(L)=caudal(L/s)*tiempo(s)
Mili_litros = Litros * 1000;

void CuentaPulsos(){
    NumPulsos++;           //Incrementa el número de pulsos
}
int ObtenerFrecuencia(){
    int frecuencia;
    NumPulsos = 0;        //Ponemos a 0 el número de pulsos
    interrupts();        //Habilitamos las interrupciones
    delay(1000);         //muestra de 1 segundo
    noInterrupts();      //Deshabilitamos las interrupciones
    frecuencia=NumPulsos; //Hz(pulsos por segundo)
    return frecuencia;
}
}
```

*Código 4: Código para medir el flujo de agua.
Fuente: Elaboración propia.*

○ Código para escuchar y leer la petición

Mediante la función de Arduino serialEvent el nodo sensor escucha y lee la petición del nodo coordinador quien envía una trama de petición que contiene el mensaje, en este caso el mensaje es el símbolo ampersand (&), serialEvent ocurre cada vez que llega un nuevo dato por el puerto 0 (Rx) del arduino uno. Esta función se ejecuta en cada iteración de la función void loop() si la función void loop demora en ejecutarse, también la función serialEvent demorará en leer los datos seriales. Esta función no está disponible para algunas tarjetas como el arduino explora, Leonardo y micro.

```
//FUNCION QUE ESCUCHA LA PETICIÓN DEL COORDINADOR A TRAVES DEL EVENTO SERIAL
void serialEvent(){
    while (Serial.available()){
        char VarChar = (char)Serial.read();
        Buffer += VarChar;
        if (VarChar == '&'){
            StringCompleta = true;
        }
    }
}
}
```

*Código 5: Código para la escucha de la petición del coordinador.
Fuente: Elaboración propia.*

El nodo sensor una vez que escucha y lee la petición del coordinador, presenta la trama al puerto serial mediante la función TrmApiTransmit(), dentro de esta función también se imprime los datos a través de la sentencia for. Antes de enviar se verifica la trama a través de la condición if (StringCompleta). Dentro de la condición se evalúa si el valor del checksum que comprueba si la trama es correcta. Si es correcto se envía la trama por el puerto serial 1 (Tx) del arduino uno, en seguida se limpia el buffer y las variables Litros, Mili_litros del sensor de flujo vuelven a cero.

```
//FUNCION QUE PRESENTA LA TRAMA EN EL PUERTO SERIAL
void TrmApiTransmit (void){
  // XBee remoto: API
  Serial.write((byte)Dltx_init); //byte de inicio
  Serial.write((byte)Long_Msb); //Largo MSB (siempre 0)
  Serial.write((byte)Long_Lsb); //byte de inicio
  Serial.write((byte)Frame_Ty); //es el tipo de mensaje para enviar comandos AT
  Serial.write((byte)Frame_ID); //Frame ID (no solicitamos respuesta)
  Serial.write((byte)Address64[0]); //Envía los 64 bit de la dirección mac del coordinador
  Serial.write((byte)Address64[1]);
  Serial.write((byte)Address64[2]);
  Serial.write((byte)Address64[3]);
  Serial.write((byte)Address64[4]);
  Serial.write((byte)Address64[5]);
  Serial.write((byte)Address64[6]);
  Serial.write((byte)Address64[7]);
  Serial.write((byte)Address16[0]);
  Serial.write((byte)Address16[1]);
  Serial.write((byte)BroadCast);
  Serial.write((byte)Option);
  //ENVIA LA INFORMACION DEL SENSOR DE FLUJO
  for (byte i = 0; i < Trm_dat ; i ++ ){ //48
    Serial.write((byte)Array_Trm_dat[i]); //49
  }
  Serial.write((byte)ByteChecksum); //Checksum
}
if (StringCompleta){
  ValorChecksum();
  TrmApiTransmit();
  Buffer = "";
  StringCompleta = false;
  Litros = 0;
  Mili_litros = 0;
}
```

*Código 6: Código para el envío de datos del nodo sensor.
Fuente: Elaboración propia.*

- **Desarrollo de código para la recepción de datos**

Cuando el nodo sensor responde a la petición del coordinador, envía una trama de respuesta conteniendo como dato los mililitros que registró el sensor de flujo hasta que le llega una petición. Por ende, el nodo coordinador debe recuperar la cantidad de mililitros que contiene la trama. Para esta operación de recuperación de los mililitros se ha desarrollado una función llamada serialEvent (), dentro de esta función hay una condición while que lee los datos cada vez que entran por el puerto serial, luego a través de la condición if se compara los datos que recibió con los caracteres ASCII que están dentro del rango de 35 hasta el 59 y lo almacena en la variable Buffer. El rango de caracteres ASCII contiene caracteres especiales y números como se puede observar en la Figura 47.

La comparación de los datos con los caracteres ASCII finaliza cuando en el dato recibido se encuentra el carácter numeral (#), el cual indica el carácter de parada de datos de la trama.

```
//FUNCION QUE RECUPERA LOS DATOS VALIDOS DEL NODO SENSOR
void serialEvent(){ //deteccion de serial por eventos
    while(Serial.available()){//lee la entrada del serial
        char VarChar = (char)Serial.read();
        if((VarChar>34)&&(VarChar<60)){
            Buffer += VarChar;
            StringCompleta = true;
            if(VarChar == '#'){
                band_trama = true;
                StringCompleta = true;
            }
        }
    }
}
```

Código 7: Código para la recuperación de los datos del nodo sensor.
Fuente: Elaboración propia.

Los mililitros que provienen como dato del nodo sensor están estructurados según el código ASCII de la Figura 47, donde a la hora de recuperar no se tiene inconvenientes.

Caracteres ASCII imprimibles								
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	\$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(72	48h	H	104	68h	h
41	29h)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	
61	3Dh	=	93	5Dh]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

Figura 47. Código ASCII.
Fuente: Elaboración propia.

f. Prueba de transmisión y recepción de datos

Para validar si la trama de petición que envía el nodo coordinador llega al nodo sensor. Se realizó la prueba utilizando el serial console del software XCTU, en donde el nodo coordinador se conecta al ordenador a través del xbee explorer usb. Mientras el nodo sensor está conectado al arduino a través del shield xbee. En el software XCTU se puede visualizar el nodo remoto que está dentro de la red de área personal establecida.

Para la prueba se generó una trama de petición para el nodo sensor en el software XCTU, donde la trama lleva como dirección destino la mac del nodo sensor. Luego se envía la trama presionando el botón Send Selected Frame en el software XCTU, como se observa en la Figura 48.

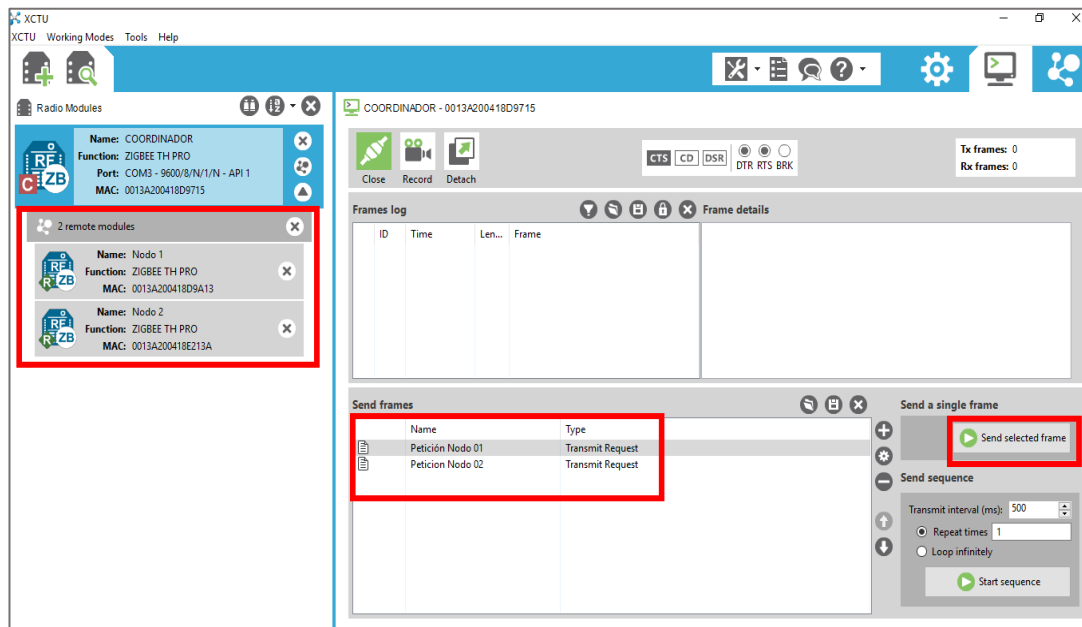


Figura 48: Prueba generar trama de petición.
Fuente: Elaboración propia.

Como se observa en el recuadro rojo de la Figura 49 la trama se envía. Y en el detalle de la trama se muestra la misma estructura de trama que se definió en la sección anterior.

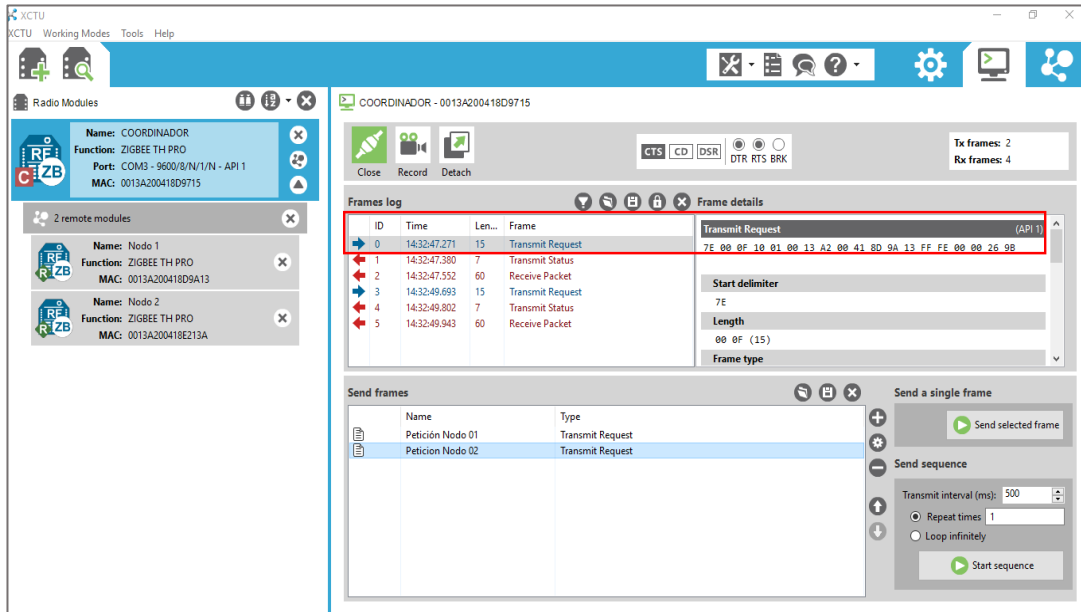


Figura 49: Prueba enviar trama de petición.
Fuente: Elaboración propia.

El nodo sensor responde de inmediato a la petición del coordinador enviando la trama de datos. Como se muestra en el recuadro rojo de la Figura 50, donde se tiene el detalle de la trama que es exactamente la misma estructura que se definió en el desarrollo del código para el nodo sensor.

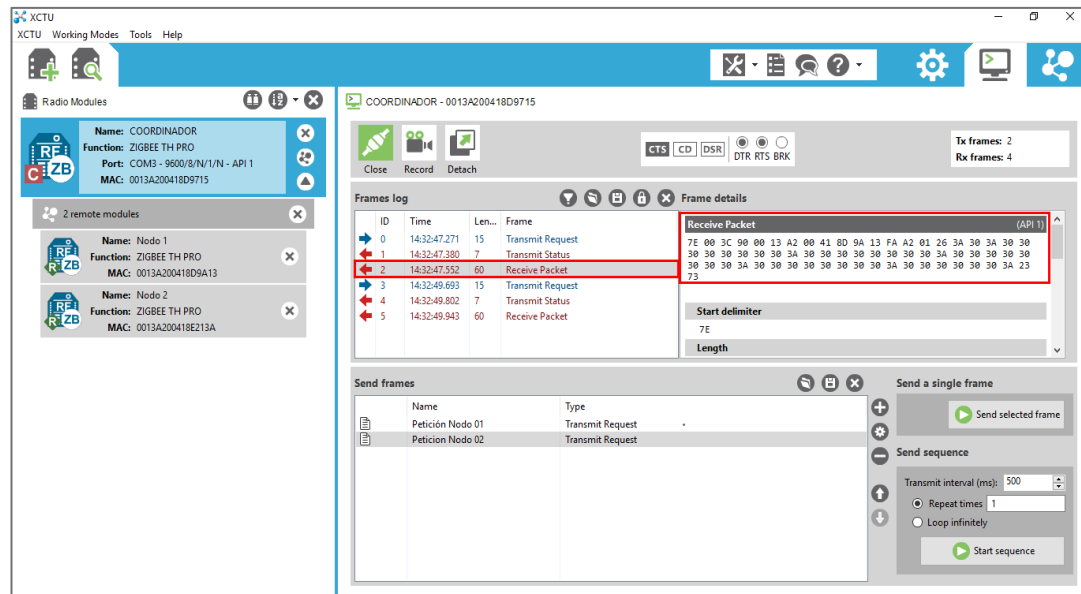


Figura 50: Prueba de respuesta de la trama de petición.
Fuente: Elaboración propia.

- **Código para él envío de datos al servidor**

Después que el nodo coordinador haya recuperado los datos de la trama procedente del nodo sensor agrega la mac del nodo a la trama. El coordinador se conecta con la base de datos que está almacenado en el servidor apache para registrar los datos en la tabla correspondiente. Para realizar esta conexión se

incluye la librería Ethernet.h al sketch del nodo coordinador, luego se inicializa la librería Ethernet y también se declaran variables que guardan las configuraciones de red como la mac de la shield Ethernet, puerta de enlace del modem, la máscara de subred, ip de la shield Ethernet, ip del servidor apache.

Luego se crea una función EnvioEthernetXbee (String MisDatos) que hace la conexión al servidor apache en el puerto 80. Dentro de la función la condición if pregunta si el ip del servidor apache y su puerto es mayor que cero. Si la condición es verdadera, el cliente Ethernet envía los datos en la variable "Vstr" por GET a un archivo php denominado datos.php. El código del archivo de php se detalla en el código 9 que se muestra a continuación.

```
#include <Ethernet.h>
#include <SPI.h>
//CONFIGURACIÓN DE CONEXION DE RED DEL SHIELD ETHERNET
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFF, 0xEE}; //Direccion mac del shield ethernet
IPAddress gateway(192, 168, 1, 2); //Puerta de enlace del modem
IPAddress subnet(255, 255, 255, 0); //Mascara del subred
IPAddress ip(192, 168, 1, 5); //ip del shield ethernet
byte server[] = {192,168,1,4}; //ip del servidor

Ethernet.begin(mac, ip); //Inicializamos el Ethernet Shield

//ENVIO DE DATOS AL SERVIDOR POR ETHERNET
void EnvioEthernetXbee (String MisDatos){
  if (client.connect(server, 80)>0){
    // Conexion con el servidor
    client.print("GET /agua/controlador/datos.php?Vstr="); // Enviamos los datos por GET
    client.print(MisDatos); //imprime los datos a enviar
    client.println(" HTTP/1.0");
    client.println("User-Agent: Arduino 1.0");
    client.println();
    mySerial.println("Conectado ET");
  }
  else{
    mySerial.println("Fallo en la conexion ET");
  }
  if (!client.connected()){
    //lcd.clear();lcd.setCursor(0, 0);lcd.print("Desconectado");
    mySerial.println("Disconnected! ET");
  }
  client.stop();
  client.flush();
}
```

*Código 8: Código para él envió de datos al servidor.
Fuente: Elaboración propia.*

El código 9 desarrollado en php, es el controlador que recupera los datos enviados por el nodo coordinador y mediante la función add_Datos() le pasa los parámetros mac, mililitros al modelo datos para que pueda grabar en la base de datos.

```

1  <?php
2  $VA_str = $_GET['Vstr'];
3  if($VA_str != ""){
4      include_once("../modelo/conexion.php");
5      include_once("../modelo/datos.php");
6      $objdata=new datos();
7      $Variables_str = explode(":", $VA_str);
8      $mac = $Variables_str[6];
9      $mililitros = $Variables_str[1];
10     $objdata->add_Datos($mac,$mililitros);
11 }
12 ?>

```

Código 9: Código del controlador para registrar datos en la base de datos.
Fuente: Elaboración propia.

g. Prueba de envío de datos al servidor de base de datos

Como se muestra en la Figura 51, la prueba del envío de datos desde el nodo coordinador al servidor de la base de datos se realizó utilizando el programa hércules, el cual es un programa hyperterminal que se conecta al puerto com del ordenador. La prueba realizada fue exitosa, porque el servidor retornar un mensaje “Conectado ET”, cada vez que graba un registro, en caso contrario muestra un “Desconectado ET”.

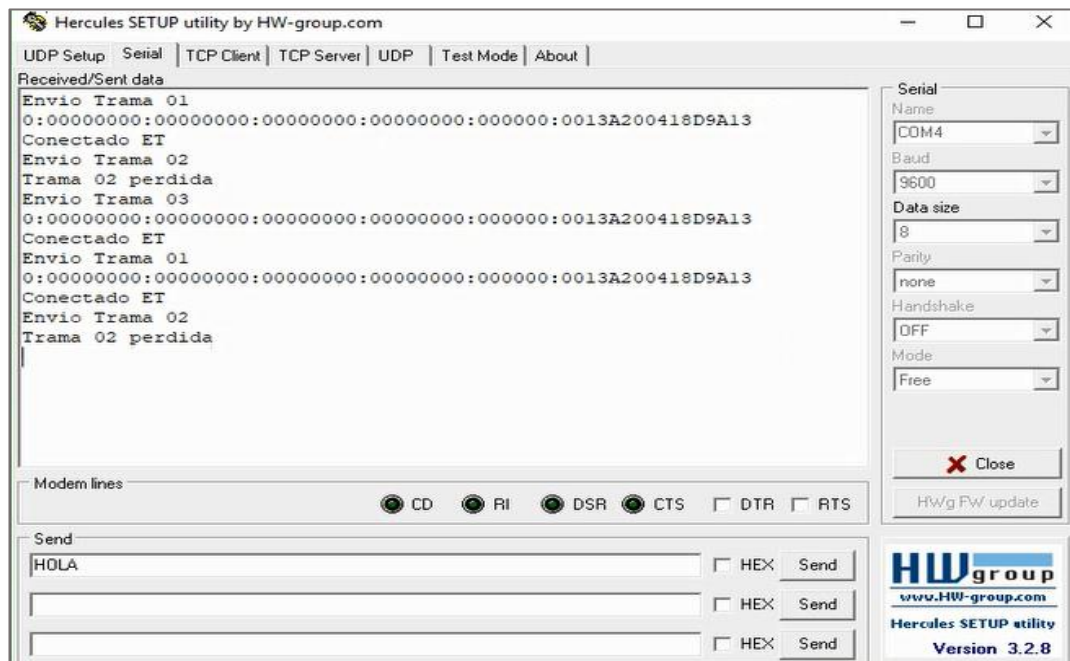


Figura 51: Prueba del envío de datos al servidor de base de datos.
Fuente: Elaboración propia.

4.3.2. Segunda iteración

Para la administración y gestión de la base de datos del sistema se instaló un servidor local XAMPP que incluye en su paquete de instalación el gestor de base de datos MySQL, el cual se utilizó en la implementación de la base de datos del sistema. La versión del software Xampp que se instaló fue 3.2.2.

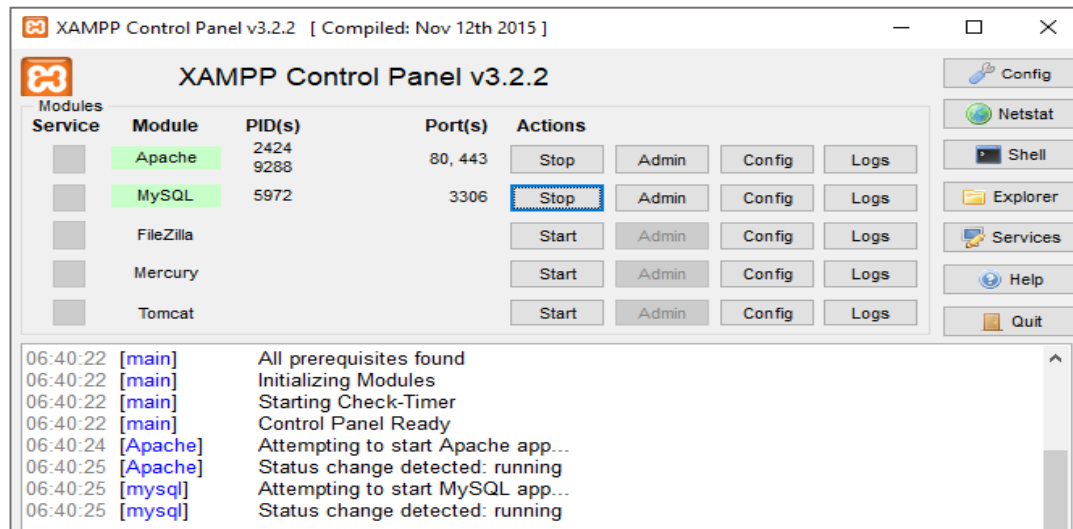


Figura 52. Interfaz del servidor local XAMPP
Fuente: Elaboración propia.

La implementación de la base de datos del sistema se realizó utilizando el gestor de base de datos MySQL, donde la implementación comprende la creación de tablas, las columnas, determinación de los tipos de datos, las restricciones (claves primarias, claves foráneas, etc.). La base de datos contiene 5 tablas, como se puede observar en la Figura 53, la funcionalidad de cada una de ellas es recibir y almacenar información del nodo sensor, a su vez es la fuente de información para el usuario sobre el consumo de agua.

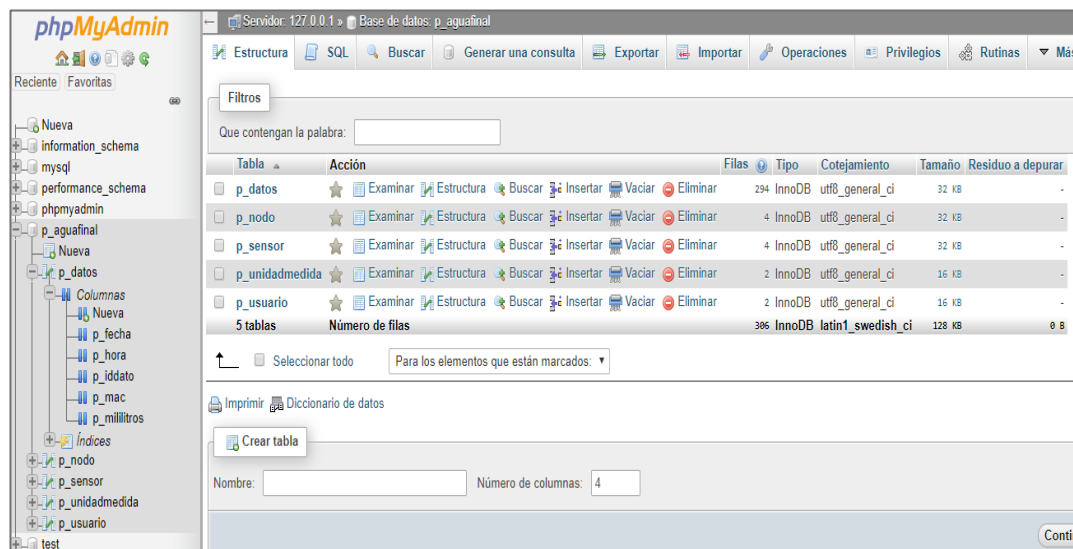
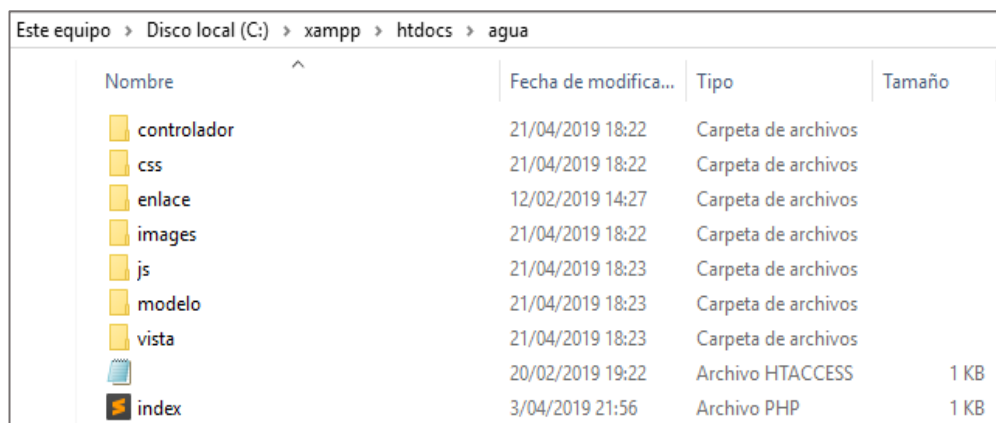


Figura 53. Base de datos con las respectivas tablas
Fuente: Elaboración propia.

4.3.3. Tercera iteración

Para el desarrollo del sistema web se utilizó el patrón Modelo-Vista-Controlador como arquitectura del sistema. Esta arquitectura facilitó el desarrollo de código en capas de forma ordenada con un ciclo de vida del sistema más adecuado, influyendo positivamente en la facilidad de mantenimiento y reutilización de código.

- ✓ **Modelo:** Es la capa encargada de trabajar con los datos, en esta capa se ha definido mecanismos para acceder a la información de la base de datos y también para registrar nuevos datos, actualizar su estado. Los datos fueron organizados según el sistema de gestión de base de datos MySQL.
- ✓ **Vista:** Es la capa intermediaria entre el usuario y el sistema. Esta capa contiene el código del sistema que permite la visualización de las interfaces graficas del usuario. El usuario interactúa con las distintas vistas del sistema para solicitar información o ejecutar acciones.
- ✓ **Controlador:** En esta capa se ha definido los códigos necesarios para responder a las acciones que solicitan los usuarios en el sistema, también es la capa intermediaria entre las vistas y los modelos. El controlador, es el encargado de recibir la información enviada por el nodo sensor y almacenar en la base de datos.



Nombre	Fecha de modifica...	Tipo	Tamaño
controlador	21/04/2019 18:22	Carpeta de archivos	
css	21/04/2019 18:22	Carpeta de archivos	
enlace	12/02/2019 14:27	Carpeta de archivos	
images	21/04/2019 18:22	Carpeta de archivos	
js	21/04/2019 18:23	Carpeta de archivos	
modelo	21/04/2019 18:23	Carpeta de archivos	
vista	21/04/2019 18:23	Carpeta de archivos	
index	3/04/2019 21:56	Archivo PHP	1 KB
HTACCESS	20/02/2019 19:22	Archivo HTACCESS	1 KB

Figura 54. Arquitectura MVC para el sistema web.

Fuente: Elaboración propia.

La codificación del sistema se realizó en el Lenguaje de Programación PHP, ya que este Lenguaje de Programación está diseñado especialmente para el desarrollo web de contenido dinámico. Y como herramienta de editor de código se utilizó el programa Sublime Text en su versión 3.2.1.

```
1 <?php
2 session_start();
3 if(isset($_SESSION["iulc"])){
4     if(isset($_REQUEST["page"])){
5         if($_REQUEST["page"]!=""){
6             $page=$_REQUEST["page"];
7             switch ($page) {
8                 case "login":
9                     include_once("./vista/index.php");
10                    break;
11                 case "index":
12                    include_once("./vista/index.php");
13                    break;
14                 case "reportes":
15                    include_once("./vista/reportes.php");
16                    break;
17                 case "close":
18                    include_once("./vista/close.php");
19                    break;
20                 case "usuarios":
21                    include_once("./vista/usuarios.php");
22                    break;
23                 case "sensores":
24                    include_once("./vista/sensores.php");
25                    break;
26                 case "nodos":
27                    include_once("./vista/nodos.php");
```

Figura 55. Interfaz de desarrollo Sublime Text.
Fuente: Elaboración propia.

A continuación, se muestra las secciones principales del sistema que fueron codificados a partir de la fase de diseño inicial.

Para acceder al sistema, se ha diseñado una página que se muestra previamente a la página principal si aún el usuario no ha iniciado sesión. Como se observa en la Figura 56, la página muestra un pequeño formulario para introducir usuario y clave.



I.E.I. Federico Villareal

Ingreso al sistema

Usuario

Clave

Ingresar

Figura 56. Página de inicio de sesión.
Fuente: Elaboración propia.

La página principal contiene cinco secciones las cuales son: sección de inicio, sensores, nodos, usuarios y reportes. A continuación, se describe las cinco secciones del sistema junto a sus formularios.

Después que el usuario haya iniciado sesión correctamente, el sistema redirigirá al usuario a la página principal. Donde esta página muestra los datos recibidos en una tabla en la que cada entrada representa la información del sensor y del nodo, también muestra el grafico de curvas que varía según el volumen de agua consumido en un tiempo determinado, como se aprecia en la Figura 57.

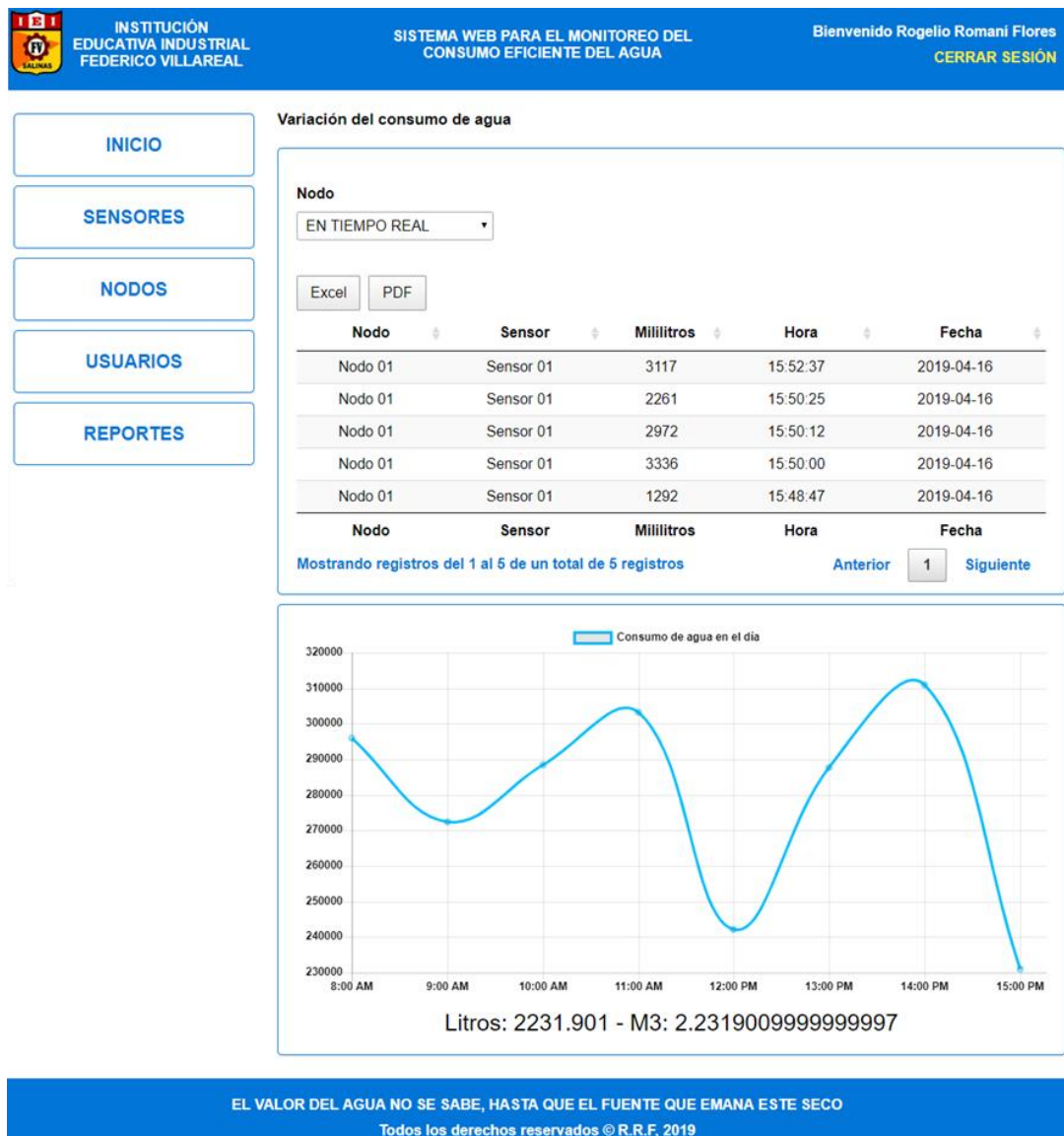


Figura 57. Sección de visualización del consumo de agua.
Fuente: Elaboración propia.

En el sistema existen elementos comunes en todas las secciones como son: las ventanas emergentes de información y los botones descriptivos que facilitan la interacción del usuario con el sistema.



Figura 58: Ventanas emergentes de eliminar y restaurar.
Fuente: Elaboración propia.

En la Figura 59, se muestra una captura del formulario diseñado para el registro y/o modificación de usuarios. Este formulario, contiene los campos nombre, apellidos, correo, rol y clave que permiten administrar usuarios del sistema.

Bienvenido Rogelio Romani Flores [CERRAR SESIÓN](#)

INICIO

SENSORES

NODOS

USUARIOS

REPORTES

Mantenimiento de usuarios

[Nuevo](#) [Limpiar](#)

Registro de usuarios X

Nombres Apellidos Correo

Rol Clave

[Guardar](#)

Buscar:

ID	Nombres	Apellidos	Correo	Rol	Opciones
1	Rogelio	Romani Flores	rromaflores@gmail.com	Administrador	

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

EL VALOR DEL AGUA NO SE SABE, HASTA QUE EL FUENTE QUE EMANA ESTE SECO

Todos los derechos reservados © R.R.F, 2019

Figura 59: Formulario de registro y/o modificación de usuarios.
Fuente: Elaboración propia.

El formulario para registrar y modificar sensores puede verse en la Figura 60. Los sensores registrados son asociados a un nodo sensor a través de su valor ID para que el sistema pueda identificar correctamente los datos provenientes desde un sensor determinado.

Los sensores, se listan en una tabla y cada registro dispone de botones para editar, deshabilitar/habilitar un registro.

INICIO

SENSORES

NODOS

USUARIOS

REPORTES

Mantenimiento de sensores

Registro de sensores X

Nombre: Descripción: Unidad de medida:

Buscar:

ID	Nombre	Descripción	Unidad	Opciones
1	Sensor 01	sensor de flujo	ml	

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior
1
Siguiente

*Figura 60: Formulario de registro y/o modificación de sensores.
 Fuente: Elaboración propia.*

A continuación, en la Figura 61 se muestra el formulario para registrar nuevos nodos en la base de datos o modificar de los ya existentes. En el campo mac se registra la dirección de 64 bits del dispositivo Xbee, la cual puede cambiar si se producen cambios de hardware. Luego se asocia un sensor ya registrado anteriormente. El valor de la mac que representa el id del nodo, el cual debe coincidir con la mac que trae de fábrica el dispositivo Xbee, para que el nodo coordinador y el sistema puedan interpretar correctamente los datos provenientes del nodo sensor y registrarlos en la base de datos.

Los nodos, se listan en una tabla y cada registro dispone de botones para editar, deshabilitar/habilitar un registro.

INICIO

SENSORES

NODOS

USUARIOS

REPORTES

Mantenimiento de nodos

Registro de nodos X

MAC Descripción Sensor

Buscar:

MAC	Descripción	Sensor	Opciones
0013A200418E032A	Nodo 01	Sensor 01	

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior Siguiente

*Figura 61: Formulario de registro y/o modificación de nodos.
 Fuente: Elaboración propia.*

Por último, en la Figura 62, se muestra una captura del reporte del consumo de agua generado para el periodo desde 01 hasta 15 de abril de 2019, donde el sistema muestra la tabla de los datos y el grafico correspondiente del nodo sensor.

- INICIO
- SENSORES
- NODOS
- USUARIOS
- REPORTES

Reporte de consumo de agua

Ingresar periodo

Desde: Hasta: Nodo:

Buscar:


Fecha	Hora	Nodo	Sensor	Mililitros	Litros	M3
2019-04-01	08:00:14	Nodo 01	Sensor 01	3425	3.425	0.003424999999999997
2019-04-01	08:00:27	Nodo 01	Sensor 01	3807	3.807	0.003807
2019-04-01	08:00:39	Nodo 01	Sensor 01	3520	3.52	0.00352
2019-04-01	08:01:20	Nodo 01	Sensor 01	4709	4.709	0.004709
2019-04-01	08:01:32	Nodo 01	Sensor 01	3523	3.523	0.003523
2019-04-01	08:01:45	Nodo 01	Sensor 01	4812	4.812	0.004812
2019-04-01	08:01:58	Nodo 01	Sensor 01	4802	4.802	0.004802
2019-04-01	08:02:10	Nodo 01	Sensor 01	3513	3.513	0.003513

Mostrando registros del 1 al 10 de un total de 8,374 registros

[Anterior](#)

...

[Siguiete](#)



INSTITUCION EDUCATIVA INDUSTRIAL
FEDERICO VILLAREAL

MONITOREO DE CONSUMO DE AGUA

Información general

Titular: INSTITUCIÓN EDUCATIVA INDUSTRIAL FEDERICO VILLAREAL

Dirección: Jirón Ayacucho S/N

Periodo: 2019-04-01 - 2019-04-15


Consumo (m3): 25.896843

Consumo (litros): 25896.843

Observaciones

.....

Evolución del consumo de agua del Nodo 01



Consumo de agua por semana

Figura 62: Sección de reportes del consumo de agua.
 Fuente: Elaboración propia.

4.3.4. Cuarta iteración

La elección de la topología, se basó según las necesidades del proyecto, la cual se fundamentaron en la implementación de 1 nodo sensor de flujo y un nodo coordinador. Dada esta distribución y el uso del protocolo DigiMesh, la topología que más se adaptó fue de punto a punto (ver Figura 63), ya que su funcionalidad se basa en:

- ✓ La transmisión de datos es en tiempo real y sin retardos, por lo que el nodo sensor no debe estar en modo Sleep o dormido, debido a que el monitoreo de flujo de agua es de manera constante.
- ✓ La utilización de un nodo Coordinador, facilita la captura de la información suministrada por el nodo sensor y la organización de toda la información, para registrarla en la base de datos y luego se visualiza a través de la interfaz gráfica del sistema web.

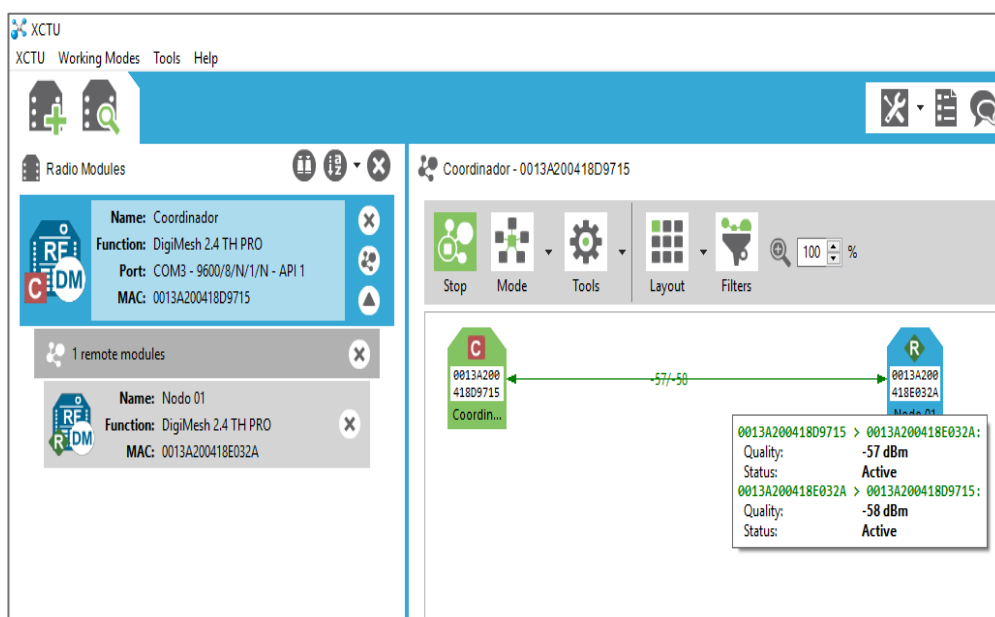


Figura 63. Topología lógica de la red.
Fuente: Elaboración propia.

a. Prueba de calidad del enlace entre los nodos

Mediante la herramienta Radio Range Test del software XCTU, se realizó la prueba de calidad del enlace entre el nodo coordinador con el nodo sensor, esta prueba garantiza la confiabilidad de la transmisión de mediciones del consumo de agua a través de la inalámbrica red. Como se muestra en la Figura 64, la prueba fue exitosa ya que los paquetes enviados desde el nodo sensor al nodo coordinador llegan sin pérdidas.

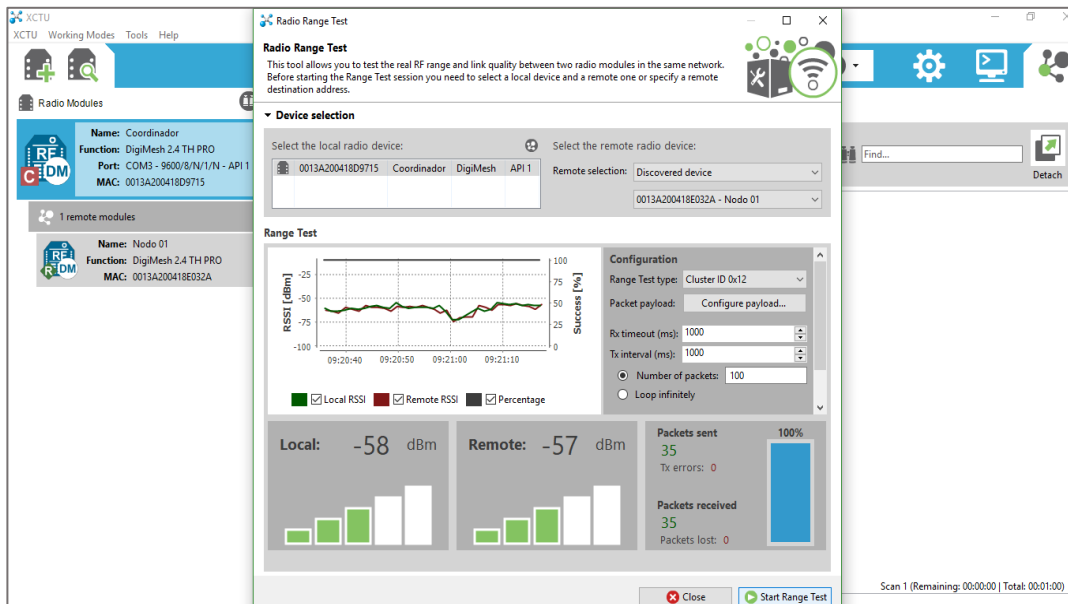


Figura 64: Prueba del enlace entre el nodo coordinador y el nodo sensor.
Fuente: Elaboración propia.

4.4. Pruebas

Esta es la última fase y la más importante de la gestión del desarrollo con la metodología XP, aplicado a la presente tesis de investigación. Esta fase comprende el proceso de verificación del correcto funcionamiento del prototipo para el monitoreo del consumo eficiente de agua desarrollado en la Institución Educativa Industrial Federico Villareal.

a. Especificación de pruebas de aceptación

Se realizó las pruebas de aceptación de cada historia del usuario para validar el correcto funcionamiento. Para la documentación según la perspectiva de la metodología XP, se utilizó la siguiente plantilla para realizar las respectivas pruebas de aceptación.

Tabla 20: Prueba de aceptación de medir el consumo de agua.

Prueba de aceptación
Caso de prueba: Medir el consumo de agua
N° de historia de usuario: 1
Descripción: Se realiza la medición del consumo de agua (mililitros).
Condiciones de ejecución: El nodo sensor realiza las mediciones cada vez que se abre un grifo de agua.
Resultado esperado: Se mide el consumo de agua.
Evaluación de la prueba: La prueba fue satisfactorio.

Fuente: Elaboración propia.

Tabla 21: Prueba de aceptación de peticionar datos de la medición.

Prueba de aceptación
Caso de prueba: Peticionar datos de la medición.
N° de historia de usuario: 1
Descripción: Se realiza la petición de la medición del consumo de agua (&, mac de nodo sensor).
Condiciones de ejecución: El nodo coordinador realiza la petición de las mediciones al nodo sensor.
Resultado esperado: El nodo sensor responde a la petición del nodo coordinador.
Evaluación de la prueba: La prueba fue satisfactorio como se muestra en la Figura 65.

Fuente: Elaboración propia.

Tabla 22: Prueba de aceptación de transmitir medición.

Prueba de aceptación
Caso de prueba: Transmitir medición.
N° de historia de usuario: 1
Descripción: Se realiza la transmisión de la medición del consumo de agua (mililitros).
Condiciones de ejecución: El nodo sensor realiza la transmisión de las mediciones al nodo coordinador cada vez que este le hace la petición.
Resultado esperado: El nodo transmite la medición.
Evaluación de la prueba: La prueba fue satisfactorio como se muestra en la Figura 65.

Fuente: Elaboración propia.

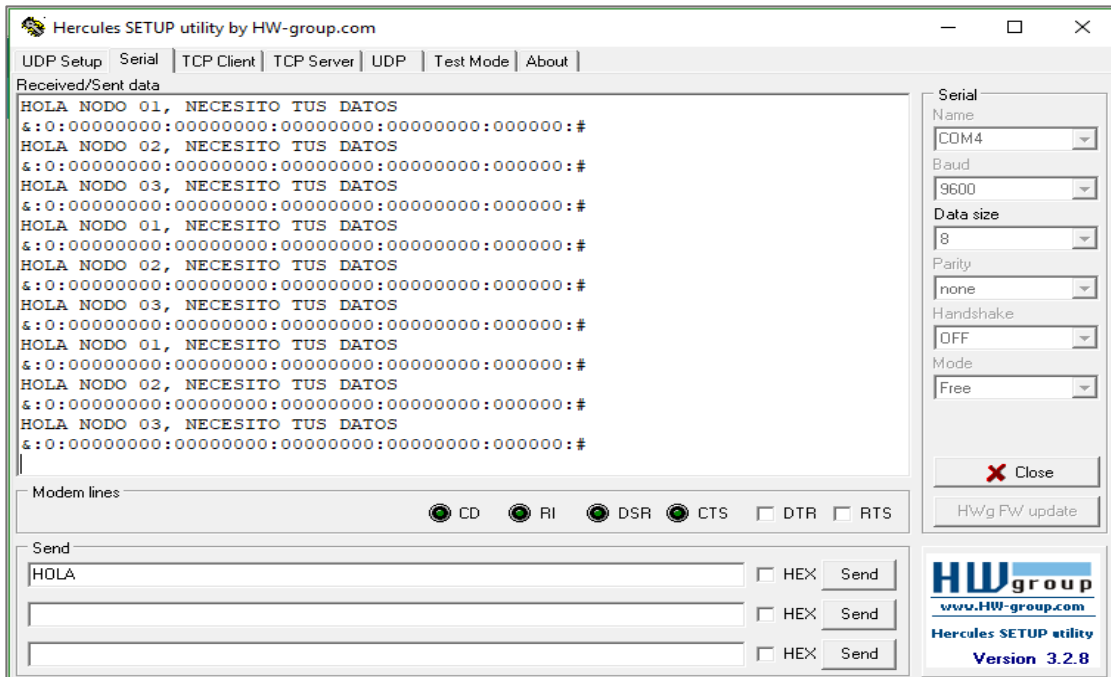


Figura 65: Prueba de petición y transmisión de datos.
Fuente: Elaboración propia.

Tabla 23: Prueba de aceptación de registrar medición en la base de datos.

Prueba de aceptación
Caso de prueba: Registrar medición en la base de datos.
Nº de historia de usuario: 1
Descripción: Se realiza el registro de la medición en la base de datos (mac del nodo sensor, mililitros).
Condiciones de ejecución: El nodo coordinador realiza el registro de las mediciones en la base de datos.
Resultado esperado: El registro se visualiza en el sistema web.
Evaluación de la prueba: La prueba fue satisfactorio como se muestra en la Figura 66.

Fuente: Elaboración propia.

IDDATO	MAC	SENSOR	MILILITROS	HORA	FECHA
1	0013A2004189AFB2	Sensor 01	913	16:42:54	2019-02-17
2	0013A2004189AFB2	Sensor 01	913	16:43:10	2019-02-17
3	0013A2004189AFB2	Sensor 01	913	16:43:11	2019-02-17
4	0013A2004189AFB2	Sensor 01	754	16:44:38	2019-02-17
5	0013B2004189AFB2	Sensor 02	754	16:45:10	2019-02-17
6	0013A200418D9A13	Sensor 03	1302	17:07:22	2019-02-17
7	0013A200418D9A13	Sensor 03	987	17:07:29	2019-02-17
8	0013A200418D9A13	Sensor 03	1149	17:07:37	2019-02-17
9	0013A200418D9A13	Sensor 03	982	17:07:44	2019-02-17
10	0013A200418D9A13	Sensor 03	987	17:07:52	2019-02-17
11	0013A200418D9A13	Sensor 03	1144	17:07:59	2019-02-17
12	0013A200418D9A13	Sensor 03	989	17:08:07	2019-02-17
13	0013A200418D9A13	Sensor 03	978	17:08:14	2019-02-17
14	0013A200418D9A13	Sensor 03	980	17:08:22	2019-02-17
15	0013A200418D9A13	Sensor 03	980	17:08:29	2019-02-17

Figura 66: Prueba de registro y almacenamiento de datos.
Fuente: Elaboración propia.

Tabla 24: Prueba de aceptación de administrar usuarios.

Prueba de aceptación
Caso de prueba: Administrar usuarios.
N° de historia de usuario: 4
<p>Descripción:</p> <p>Se realiza la administración de la información acerca de los usuarios que consiste en registrar, modificar, deshabilitar y habilitar (nombre, apellidos, correo, rol, clave).</p>
<p>Condiciones de ejecución:</p> <p>La administración de los usuarios corresponde ejecutar una de las acciones de registrar, modificar, deshabilitar y habilitar.</p>
<p>Entradas:</p> <p>El administrador accede al sistema/ Registrar usuario</p> <ul style="list-style-type: none"> • El administrador hace clic en el botón nuevo para registrar un nuevo usuario. • El administrador escribe en las cajas de texto nombre, apellidos, correo y clave. • El administrador hace clic en el botón guardar. <p>El administrador accede al sistema/ Modificar usuario</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono modificar usuario. • El administrador modifica los datos de las cajas de texto nombre, apellidos, correo y clave. • El administrador hace clic en el botón guardar. <p>El administrador accede al sistema/ Deshabilitar usuario</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono deshabilitar usuario. • El administrador hace clic en el botón ok para confirmar.

<p>El administrador accede al sistema/ Habilitar usuario</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono Habilitar usuario. • El administrador hace clic en el botón ok para confirmar.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se registra nuevo usuario en el sistema. • Se modifica el usuario en el sistema. • Se deshabilita el usuario del sistema. • Se habilita el usuario en el sistema. <p>El sistema muestra un mensaje de confirmación a cada una de las acciones realizadas.</p>
<p>Evaluación de la prueba:</p> <p>Buen trabajo. Datos registrados correctamente.</p> <p>Buen trabajo. Datos modificados correctamente.</p> <p>Buen trabajo. Registro deshabilitado.</p> <p>Buen trabajo. Registro habilitado.</p>

Fuente: Elaboración propia.

Tabla 25: Prueba de aceptación de administrar sensores.

Prueba de aceptación
Caso de prueba: Administrar sensores.
N° de historia de usuario: 4
<p>Descripción:</p> <p>Se realiza la administración de la información acerca de los sensores que consiste en registrar, modificar, deshabilitar y habilitar (nombre, descripción, unidad de medida).</p>
<p>Condiciones de ejecución:</p> <p>La administración de los sensores corresponde ejecutar una de las acciones de registrar, modificar, deshabilitar y habilitar.</p>
<p>Entradas:</p> <p>El administrador accede al sistema/ Registrar sensores</p> <ul style="list-style-type: none"> • El administrador hace clic en el botón nuevo para registrar un nuevo sensor. • El administrador escribe en las cajas de texto nombre, descripción y selecciona una unidad de medida. • El administrador hace clic en el botón guardar. <p>El administrador accede al sistema/ Modificar sensores</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono modificar sensor.

<ul style="list-style-type: none"> • El administrador modifica los datos de las cajas de texto nombre, descripción y selecciona una unidad de medida. • El administrador hace clic en el botón guardar. <p>El administrador accede al sistema/ Deshabilitar sensores</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono deshabilitar sensor. • El administrador hace clic en el botón ok para confirmar. <p>El administrador accede al sistema/ Habilitar sensores</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono Habilitar sensor. • El administrador hace clic en el botón ok para confirmar.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se registra nuevo sensor en el sistema. • Se modifica el sensor en el sistema. • Se deshabilita el sensor del sistema. • Se habilita el sensor en el sistema. <p>El sistema muestra un mensaje de confirmación a cada una de las acciones realizadas.</p>
<p>Evaluación de la prueba:</p> <p>Buen trabajo. Datos registrados correctamente.</p> <p>Buen trabajo. Datos modificados correctamente.</p> <p>Buen trabajo. Registro deshabilitado.</p> <p>Buen trabajo. Registro habilitado.</p>

Fuente: Elaboración propia.

Tabla 26: Prueba de aceptación de administrar nodos.

Prueba de aceptación
Caso de prueba: Administrar nodos.
Nº de historia de usuario: 4
<p>Descripción:</p> <p>Se realiza la administración de la información acerca de los nodos que consiste en registrar, modificar, deshabilitar y habilitar (mac, descripción, sensor).</p>
<p>Condiciones de ejecución:</p> <p>La administración de los nodos corresponde ejecutar una de las acciones de registrar, modificar, deshabilitar y habilitar.</p>
<p>Entradas:</p> <p>El administrador accede al sistema/ Registrar nodo</p> <ul style="list-style-type: none"> • El administrador hace clic en el botón nuevo para registrar un nuevo nodo. • El administrador escribe en las cajas de texto mac, descripción, y selecciona un sensor.

<ul style="list-style-type: none"> • El administrador hace clic en el botón guardar. <p>El administrador accede al sistema/ Modificar nodo</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono modificar nodo. • El administrador modifica los datos de las cajas de texto mac, descripción y selecciona un sensor. • El administrador hace clic en el botón guardar. <p>El administrador accede al sistema/ Deshabilitar nodo</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono deshabilitar nodo. • El administrador hace clic en el botón ok para confirmar. <p>El administrador accede al sistema/ Habilitar nodo</p> <ul style="list-style-type: none"> • El administrador hace clic en el icono Habilitar nodo. • El administrador hace clic en el botón ok para confirmar.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se registra nuevo nodo en el sistema. • Se modifica el nodo en el sistema. • Se deshabilita el nodo del sistema. • Se habilita el nodo en el sistema. <p>El sistema muestra un mensaje de confirmación a cada una de las acciones realizadas.</p>
<p>Evaluación de la prueba:</p> <p>Buen trabajo. Datos registrados correctamente.</p> <p>Buen trabajo. Datos modificados correctamente.</p> <p>Buen trabajo. Registro deshabilitado.</p> <p>Buen trabajo. Registro habilitado.</p>

Fuente: Elaboración propia.

4.5. Puesta en marcha del prototipo

Para la puesta en marcha del funcionamiento del prototipo se empezó con la instalación del sensor de flujo en la tubería principal de agua de la Institución Educativa, que a continuación se describen los detalles.



*Figura 67: Instalación del sensor de flujo.
Fuente: Elaboración propia.*

Una vez instalado el sensor se procedió con la conexión de los cables de alimentación (rojo, negro) del sensor a los pines (5v, gnd) del arduino y el cable amarillo del sensor al pin 2 digital del arduino. Para lo cual se empleó el cable Ethernet categoría 6 para alargar la conexión ya que el nodo está a una altura de 3 metros y el sensor está enterrado en la línea principal de la instalación de agua.



*Figura 68: Nodo 01 instalado.
Fuente: Elaboración propia.*

Luego se instaló el nodo coordinador, que es el responsable de formar la red inalámbrica con el nodo sensor y registrar las mediciones que le transmite en la base de datos. Esta instalación se muestra en la Figura 69.



*Figura 69: Nodo coordinador instalado.
Fuente: Elaboración propia.*

Para completar con la instalación se realizó la conexión del nodo coordinador con el modem de internet de la Institución Educativa mediante cable Ethernet y luego del modem se conectó con la computadora personal en el cual se instaló el servidor apache que incluye el servidor de base datos, tal como se muestra en la Figura 70.

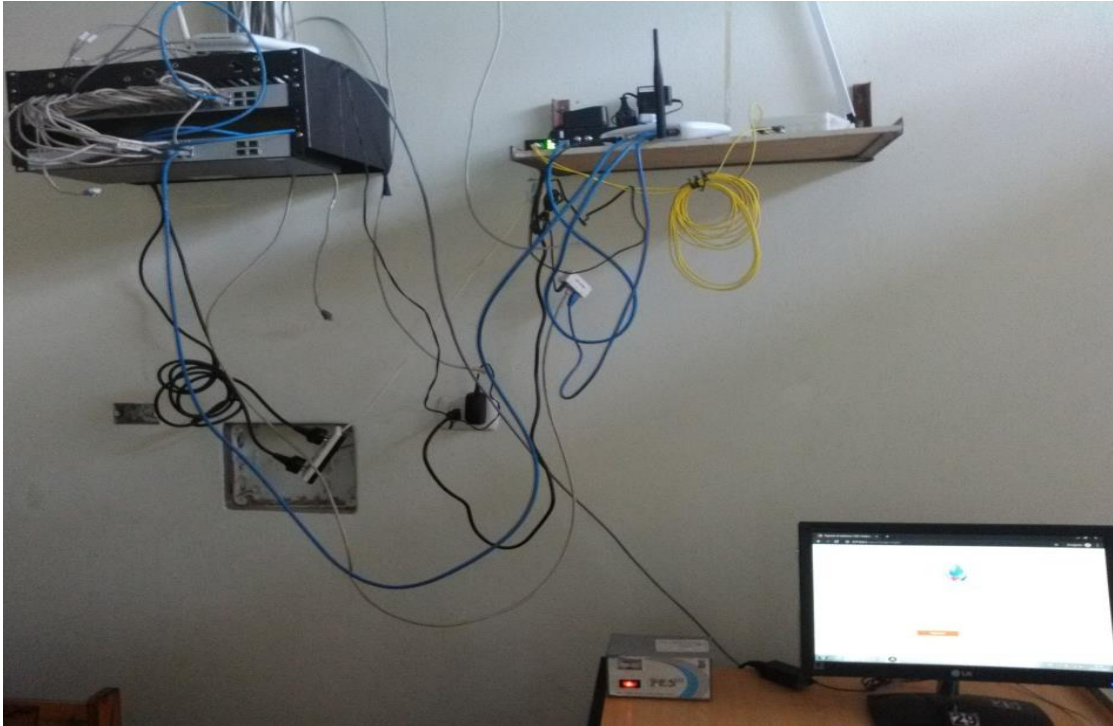


Figura 70: Conexión del nodo coordinador con el servidor de BD.
Fuente: Elaboración propia.

Como se observa en la Figura 71, se valida el correcto funcionamiento del prototipo desarrollado.

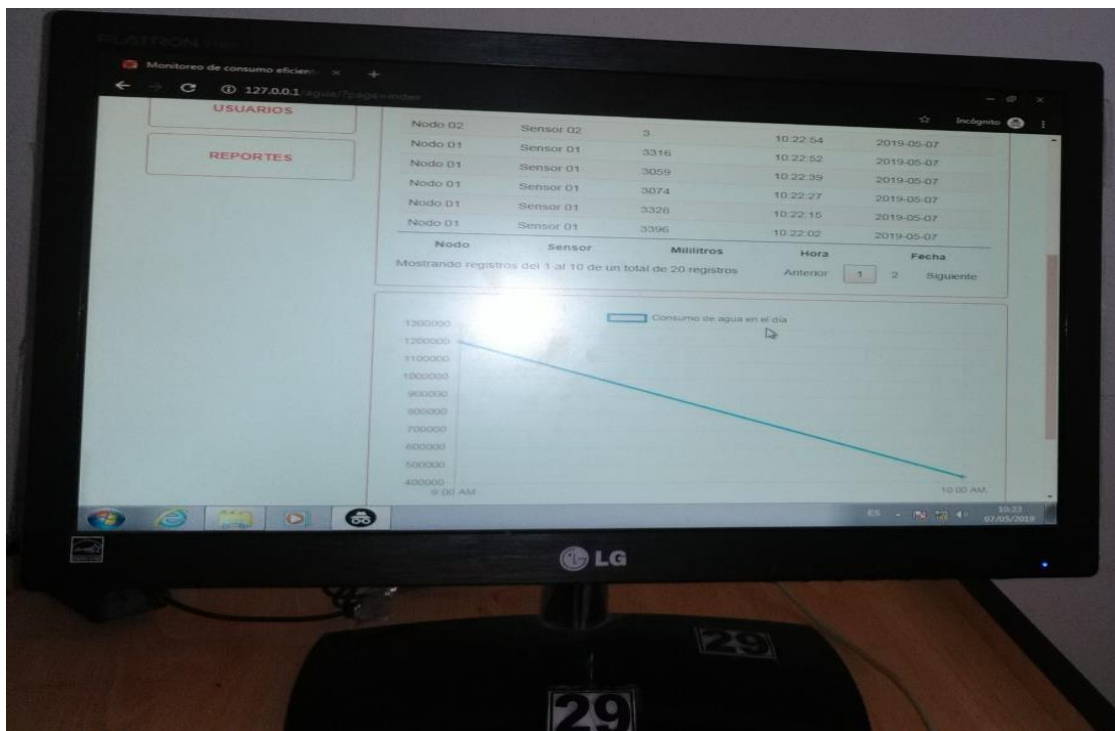


Figura 71: Visualización del monitoreo del consumo de agua en tiempo real.
Fuente: Elaboración propia.

CAPÍTULO V. RESULTADOS

En este capítulo se presenta los resultados obtenidos en el desarrollo de la investigación de tesis. En la Figura 72 se muestra el prototipo desarrollado, el cual consta de dos partes que son: modulo emisor y modulo receptor.

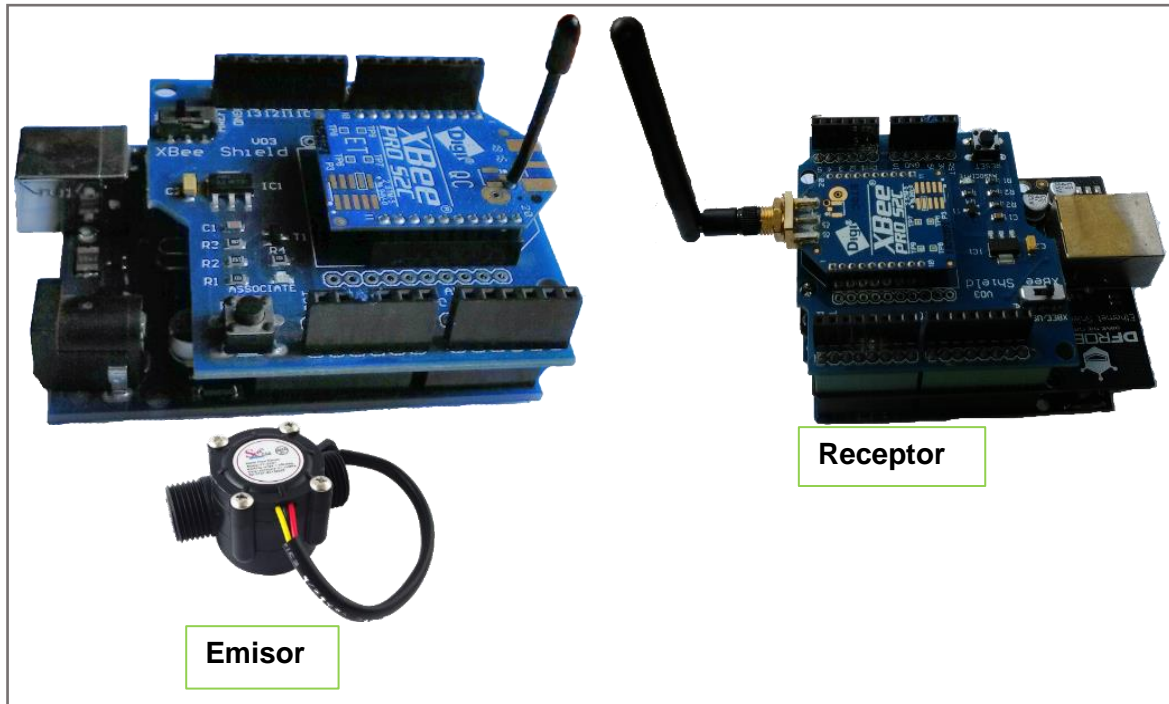


Figura 72: Prototipo desarrollado.
Fuente: Elaboración propia.

En la siguiente Figura 73 se muestra el desarrollo de la base de datos, que permitió el almacenamiento de la información del consumo de agua.

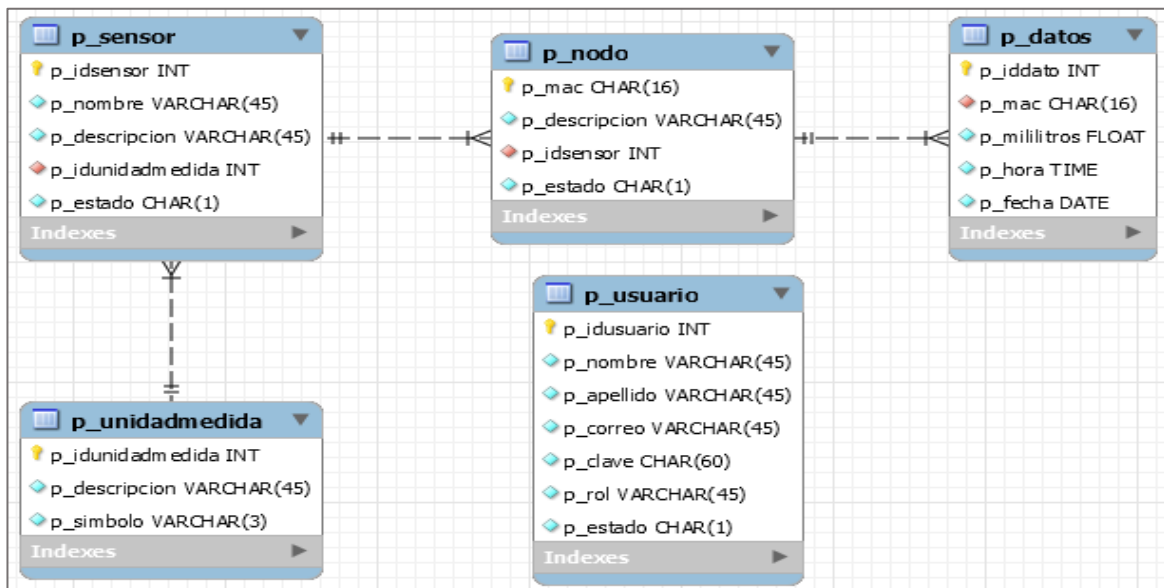


Figura 73: Base de datos desarrollado.
Fuente: Elaboración propia.

En la Figura 74 se muestra el desarrollo de la red inalámbrica ZigBee implementado para la transmisión de los datos desde el modulo emisor hasta el modulo receptor. La topología de la red es punto a punto.

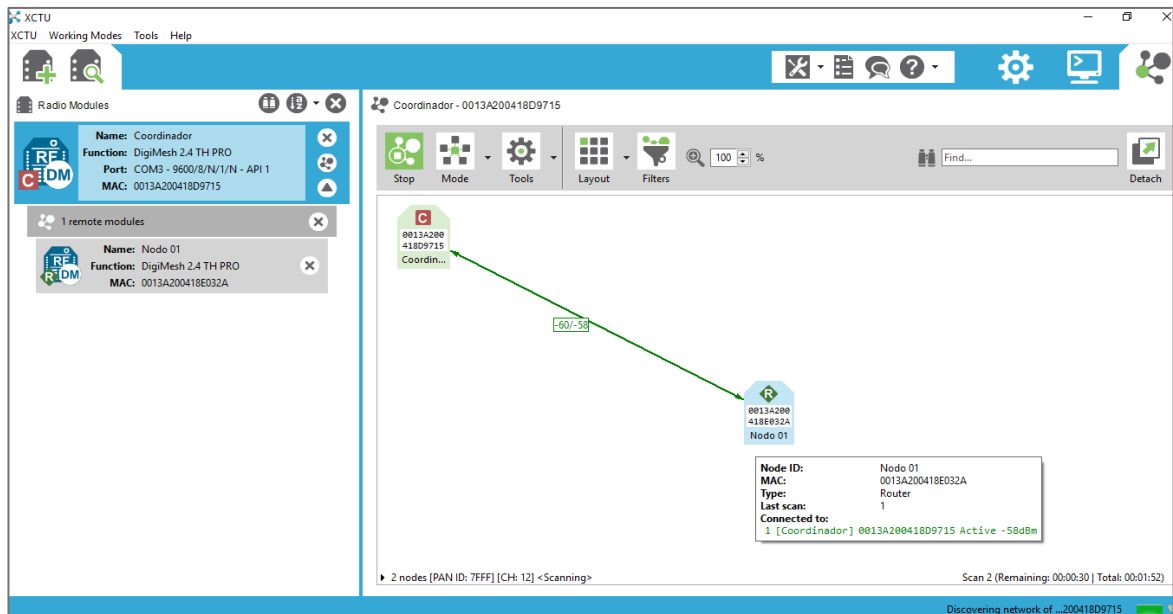


Figura 74: Red inalámbrica ZigBee implementado.
Fuente: Elaboración propia.

La siguiente Figura 75 muestra el sistema web desarrollado, que permitió el procesamiento y visualización de los datos del consumo de agua en la Institución Educativa Industrial Federico Villareal. El sistema tiene 5 módulos que permiten la administración de los usuarios, sensores, nodos y generar reportes del consumo de agua con total facilidad.



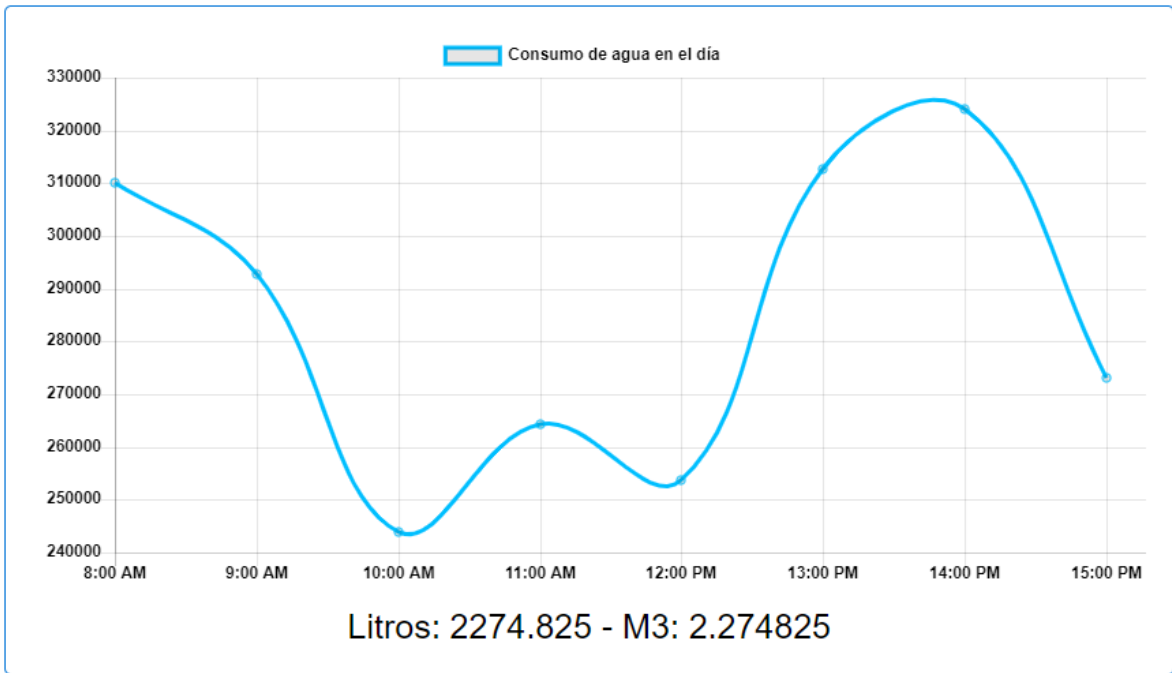
Figura 75: Sistema web local desarrollado.
Fuente: Elaboración propia.

Durante el tiempo de implantación del prototipo no se presentó mayor novedad en cuanto a pérdida de datos, el monitoreo constante del consumo de agua permitió conocer a detalle sobre la variación del consumo de agua que genera la comunidad educativa, gracias al almacenamiento de los datos y procesamiento a través del sistema web desarrollado. El resultado del consumo de agua que se muestra en la Figura 76 comprende al mes de abril del 2019, en el cuál el servidor almacenó exactamente 47.2327m³ datos del consumo de agua.

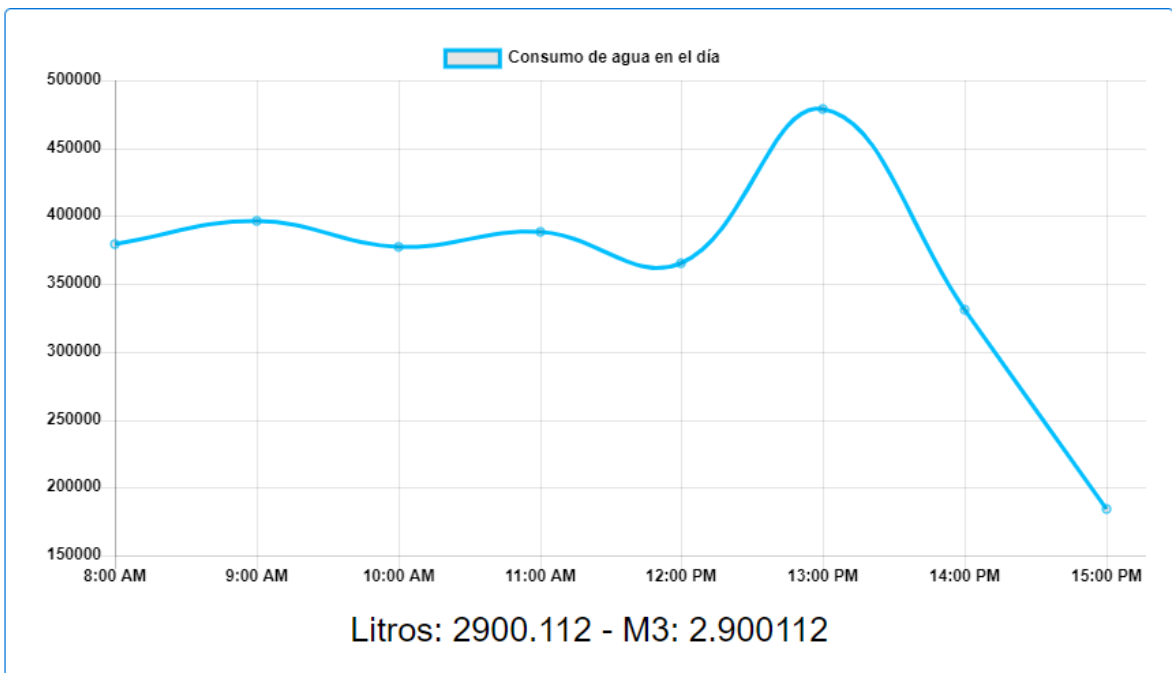


*Figura 76: Reporte del consumo de agua, abril 2019.
Fuente: Elaboración propia.*

A continuación, en las Figuras 77 y 78 se muestran los gráficos de la información diaria del consumo de agua correspondientes a las fechas de 05, y 30 de abril del 2019 respectivamente.



*Figura 77: Incidente 05 de Abril del 2019.
Fuente: Elaboración propia.*



*Figura 78: Incidente 30 de Abril del 2019.
Fuente: Elaboración propia.*

De acuerdo a los resultados obtenidos, se puede afirmar que el trabajo de tesis fue un éxito y cumple satisfactoriamente con lo requerido por lo mismo.

CAPÍTULO VI. DISCUSIÓN

En este capítulo se analiza los resultados del presente trabajo de tesis realizando una comparativa sobre los resultados de los trabajos antecesores en el tema de investigación. El desarrollo del prototipo, resultó ser una alternativa viable para el monitoreo del consumo eficiente de agua indistintamente del lugar donde se implemente, se manejó un intervalo de medición de 13 segundos aproximadamente. Sin embargo, el prototipo tiene la capacidad de hacer mediciones con intervalos de tiempos menores.

Para la transmisión de los datos desde el nodo sensor hasta el nodo coordinador se utilizó el módulo Xbee basado en el estándar ZigBee y se configuró la topología punto a punto, sin embargo, es posible crear redes inalámbricas de tipo malla y con esto incorporar la cantidad de sensores requeridas para la medición del consumo de agua, el cual permitiría monitorear todo un sistema de distribución de agua.

Según la investigación sobre el “Sistema remoto en red multipunto para las medidas de consumo de energía eléctrica domiciliaria” realizada por los autores Huamanyalli & Rua de la Universidad Nacional de Huancavelica. Los autores desarrollaron un módulo medidor remoto del consumo de energía eléctrica domiciliaria utilizando microcontrolador PIC16F628A. La solución contribuyó en la lectura de los datos del medidor de energía de manera remota y beneficio directamente a la empresa Electrocentro S.A. y a los usuarios. Los componentes utilizados para el desarrollo del módulo medidor no fueron los más actuales del mercado, pero los resultados fueron éxitos. En comparación con el presente trabajo de tesis desarrollado difiere en los materiales utilizados, puesto que el tema de investigación es del consumo de agua, mas no de energía eléctrica. Sin embargo, fue precedente para la génesis de la idea del tema de investigación sobre el monitoreo del consumo de agua en tiempo real.

En comparación con la investigación sobre “Implementación de un sistema de supervisión y monitorización del consumo de energía eléctrica y agua potable, utilizando redes Het-Net para la transmisión de datos; con la finalidad de obtener información oportuna para una eficiente facturación y disminuir perdidas en la dotación de los servicios”, desarrollado por los autores Arévalo & Rodríguez de la Universidad Politécnica Salesiana, es similar al presente trabajo de tesis desarrollado en cuanto al tema de monitorización de los servicios básicos como el agua y energía eléctrica. Sin embargo, las herramientas tecnológicas utilizadas son diferentes como el caso de GPRS que utilizaron los autores del proyecto mencionado para la transmisión de los datos de los medidores de agua y energía eléctrica hasta la estación central. El resultado que obtuvieron los autores con el desarrollo del

proyecto tuvo un impacto importante en la automatización a gran escala el proceso de medición remota del consumo de los servicios básicos de agua potable y energía eléctrica en la ciudad de Cuenca Ecuador. Por tal motivo la investigación de los autores fue precedente importante para el desarrollo del presente trabajo de tesis.

También la investigación sobre. “Diseño e implementación de un sistema de medición de consumo de energía eléctrica y agua potable remoto con interacción al usuario basado en el concepto internet de las cosas”, desarrollado por los autores Guacaneme & Pardo de la Universidad Distrital Francisco José de Caldas, fue antecedente importante para el desarrollo del presente trabajo de tesis. Porque utiliza el mismo sensor de flujo basado en efecto Hall que permite la medición del consumo de agua, sin embargo, difiere en las demás tecnologías utilizadas como el wi-fi para el caso de transmisión de datos. Los resultados que obtuvieron los autores fue beneficioso porque permite a un usuario monitorear su consumo de los servicios básicos como el agua y energía eléctrica desde cualquier lugar utilizando un dispositivo inteligente.

Según la investigación sobre. “Diseño e implementación del prototipo de un sistema domótico para la medición del consumo de agua potable a través de internet y correo electrónico”, desarrollado por Chuquimarca de la Universidad Nacional de Loja. El resultado fue bastante exitoso ya que permitió al usuario monitorear de manera local y remota su consumo de agua potable, ya que el prototipo está basado en el módulo Raspberry Pi que permite la conexión a internet. Es por ello que la investigación del autor fue precedente base para el desarrollo del presente trabajo de tesis.

Por otro lado, según la revista sobre el “Sistema de tele medición de consumo de agua potable para aplicaciones residenciales y comerciales” desarrollado por Cázarez, y otros. Es igual al presente proyecto en cuanto al uso del módulo Xbee que permite la transmisión inalámbrica de los datos del consumo de agua, en tal aspecto fue precedente indispensable para el desarrollo del trabajo de tesis. Los resultados que obtuvieron los autores fue satisfactorio ya que permitió la medición del consumo de agua en la zonas residenciales y comerciales empleando los requerimientos a menor costo.

Finalmente, en base a los resultados obtenidos en la presente tesis de investigación y en comparación con las investigaciones precedentes se comprueba que el desarrollo de esta solución tecnológica permite mejorar el monitoreo del servicio básico como el consumo de agua. Sin embargo, como toda investigación siempre es un precedente para desarrollar otra mejor.

CONCLUSIONES

- ✓ El primer objetivo específico está orientado al diseño del prototipo para el monitoreo del consumo de agua, para lo cual se ha adquirido los componentes necesarios y para la funcionalidad del prototipo se ha desarrollado código fuente en el software Arduino ya que este es el micro-controlador principal del prototipo. Con todos los requerimientos disponibles se logró el diseño y funcionalidad del prototipo que ayuda en la medición y monitoreo constante del consumo de agua en la Institución Educativa Industrial Federico Villareal. Además de la medición y monitoreo que realiza el prototipo incide indirectamente en el cuidado del recurso más importante para la vida que todas las personas están en la obligación de cuidar y hacer un uso más eficiente de este recurso.
- ✓ Con respecto al segundo objetivo específico, se logró desarrollar la base de datos para el almacenamiento de la medición que realiza el prototipo. Y sobre esta información almacenada se pueda realizar en trabajos futuros un análisis del consumo de agua que realiza la Institución Educativa Industrial Federico Villareal.
- ✓ Para el tercer objetivo específico se ha realizado un estudio sobre el medio por el cual transmitir las mediciones realizadas por el sensor hasta el almacenamiento en la base de datos. Teniendo en cuenta, el coste, su facilidad de integración con el micro-controlador Arduino uno, el medio elegido fue el módulo XBee de Digi. Mediante el uso de este medio se logró implementar la red inalámbrica punto a punto para la transmisión de las mediciones del consumo de agua.
- ✓ Para el último objetivo específico se ha desarrollado un sistema web bajo la arquitectura de software modelo-vista-controlador, para la visualización del consumo de agua y administración de los nodos, sensores con una interfaz intuitiva y amigable para el usuario que le permite interactuar de forma fácil con el sistema.

RECOMENDACIONES

- ✓ Recomiendo para trabajos futuros incluir al prototipo una batería como fuente de energía alternativa, en caso que ocurra cortes de energía comercial no se apague el prototipo y siga con las mediciones del consumo de agua. También recomiendo proteger el prototipo con material resistible a altas y bajas temperaturas para evitar el calentamiento con el sol y humedad cuando llueve. Esto con fin de mantener los circuitos de la placa y no tenga interferencias.
- ✓ Al esquema relacional de la base de datos es recomendable agregar una tabla que gestione las alertas en caso de algún evento inusual por ejemplo cuando hay flujo de agua en las horas de noche, el cual indicaría una fuga de agua en algún lugar de la distribución.
- ✓ Si se tiene la necesidad de cubrir grandes distancias entre el nodo coordinador y el nodo sensor se recomienda utilizar los módulos Xbee Pro 900HP S3B que tiene un alcance máximo de 14.5 km y también utilizar antenas de mayor potencia, para obtener una mejor estabilidad durante el proceso de comunicación. Al momento de alimentar el módulo Xbee, hacerlo mediante regulador y no directo desde una fuente de voltaje, debido a los picos presentados, que pueden afectar al módulo, procurando que sea siempre menor al máximo estipulado que es 3.3 V.
- ✓ En trabajos futuros con características de este proyecto, sería factible y beneficioso adquirir un hosting que aloje el sistema web, el cual permitiría el monitoreo del consumo de forma remota. Y además sería oportuno para las empresas suministradoras de agua realizar las mediciones de forma rápida de cada usuario sin emplear los recursos personales.

REFERENCIAS BIBLIOGRÁFICAS

- Agencia CTyS. (05 de 01 de 2014). La crisis del XXI, asignadas por la falta de agua potable. Recuperado el 11 de 07 de 2018, de <http://www.ctys.com.ar/index.php?idPage=20&idArticulo=2877>
- Alcaldía Mayor de Bogota D.C. (2017). Instrucciones Para el Control de Consumo de Agua, Energia, Papel y Combustible. Oficina Asesora de Planeción y la Dirección Administrativa. Recuperado el 13 de 08 de 2018, de <http://gaia.gobiernobogota.gov.co/sites/default/files/sig/instructivo/ple-pin-in005.pdf>
- Álvarez, P. M., & Toledo, G. M. (2011). Tipos de Medidores de flujo de agua domiciliarios. Coordinación de Desarrollo Profesional e Institucional Subcoordinación de Certificación de Personal, 6. Obtenido de <https://es.scribd.com/document/311725748/Cap-5-Tipos-de-Med-ANEAS-pdf>
- Arévalo, Q. G., & Rodríguez, B. J. (2016). Implementación de un sistema de supervisión y monitorización del consumo de energía eléctrica y agua potable, utilizando redes Het-Net para la transmisión de datos; con la finalidad de obtener información oportuna para una eficiente facturación y disminui. (Tesis). Universidad Politécnica Salesiana, Cuenca, Ecuador. Recuperado el 16 de 07 de 2018
- Argüello González, J. I. (2016). Interfaz Máquina-Máquina para el Control de Presión que Ejercen las Bombas de Distribución de Agua en Unidades Habitacionales. Tesis. Universidad Autónoma del Estado de México, Toluca, Mexico. Recuperado el 16 de 11 de 2018
- Autoridad Nacional del Agua (ANA). (2011). Gestión del Agua y Oportunidades para el Desarrollo Profesional. Recuperado el 05 de 08 de 2018
- Bautista, Q. J. (2012). Programción Extrema (XP). Universidad Unión Bolivariana, Bolivia.
- BBC. (19 de 03 de 2015). Por qué se está acabando el agua. Recuperado el 08 de 08 de 2018
- Berni, M. P., & De la Iglesia, D. G. (2010). Laboratorio de PHP y MySQL (Primera Edición ed.). Media Eureka. Recuperado el 08 de 08 de 2018
- Blum, J. (2014). Arduino a Fondo. Aprenda a Crear Fácilmente Gadgets, Gizmos, Robots y Mucho más. Anaya. Recuperado el 10 de 07 de 2018, de <http://www.libreriageli.com/informatica/403226-arduino-a-fondoaprenda-a-crear-facilmente-gadgets-gizmos-robots-y-mucho-mas.html>
- Bordignon, F., & Iglesias, A. A. (2015). Diseño y construcción de objetos interactivos digitales Experimentos con la plataforma Arduino (Primera Edición ed., Vol. 1). (E. Universitaria, Ed.) Argentina: UNIPE. Recuperado el 14 de 08 de 2018
- Buchenau, M., & Fulton Suri, J. (2000). Experience Prototyping. 11. Recuperado el 14 de 01 de 2019, de http://5a5f89b8e10a225a44accbed124c38c4f7a3066210c073e7d55.r9.cf1.rackcdn.com/files/pdfs/news/FultonSuriBuchenau-Experience_PrototypingACM_8-00.pdf

- Cázares, A. G., Rodríguez, B. A., Ramírez, B. M., Lugo, Z. S., Castillo, M. H., & Morales, L. H. (2016). Sistema de Telemedición de Consumo de Agua Potable para Aplicaciones Residenciales y Comerciales. 7. Recuperado el 18 de 07 de 2018
- Chuquimarca, S. E. (2014). Diseño e Implementación del Prototipo de un Sistema Domótico para la Medición del Consumo de Agua Potable a través de Internet y Correo Electrónico. Tesis. Universidad Nacional de Loja, Loja, Ecuador. doi:10.1017/S0010417500000463
- Congreso de la República. (07 de 01 de 2010). Ley de Recursos Hídricos [Ley N° 29338]. Lima, Perú. Recuperado el 05 de 08 de 2018
- Creus, S. A. (1997). Instrumentación Industrial (Sexta Edición ed.). (Alfaomega, Ed.) Barcelona, España. Recuperado el 19 de 07 de 2018
- Digi International Inc. (1985). Digi. Recuperado el 02 de 01 de 2019, de <https://www.digi.com>
- Eady, F. (2007). Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers. Amsterdam, Inglaterra.
- Faludi, R. (2010). A Practical Guide to the ZigBee Mesh Networking Protocol. Building Wireless Sensor Networks (Primera Edición ed.). (O. Reilly, Ed.) Sebastopol, Rusia: Jepson Brian. doi:10.1007/s13398-014-0173-7.
- Farahani, S. (2008). ZigBee Wireless Networks and Transceivers.
- Gómez, M. B. (1982). Manual Sobre Evaluación de Medidores Domiciliarios de Agua. Lima. Recuperado el 15 de 07 de 2018, de <http://bvsper.paho.org/bvsacd/scan2/012933/012933-08.pdf>
- Granger, B. G. (1965). Operaciones básicas de la Ingeniería Química. España: MARIN /S/A/. Recuperado el 20 de 07 de 2018
- Guacaneme, V. G., & Pardo, A. D. (2016). Diseño e Implementación de un Sistema de Medición de Consumo de Energía Eléctrica y Agua Potable Remoto con Interacción al Usuario Basado en el Concepto "Internet de las Cosas". Tesis. Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Recuperado el 20 de 07 de 2018
- Harper, G. E. (2004). Manual Práctico de Instalaciones Eléctricas (Segunda ed.). (Limusa, Ed.) Mexico. Recuperado el 19 de 06 de 2018
- Houde, S., & Hill, C. (1997). What do Prototypes Prototype? 16. Recuperado el 14 de 01 de 2019, de <https://pdfs.semanticscholar.org/30bc/6125fab9d9b2d5854223aeea7900a218f149.pdf>
- Huamanyalli, C. A., & Rua, F. J. (2014). Sistema Remoto en Red Multipunto para las Medidas de Consumo de Energía Eléctrica Domiciliaria. Tesis. Universidad Nacional de Huancavelica, Huancavelica, Perú. Recuperado el 20 de 08 de 2018

- Letelier, P., & Penadés, C. (15 de 01 de 2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 26. Recuperado el 17 de 10 de 2018, de <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Marqués, M. (2011). Bases de datos (Primera ed.).
- Marquez, A. (20 de 05 de 2014). Sensor de Flujo. Recuperado el 20 de 10 de 2018, de <https://hetpro-store.com/TUTORIALES/sensor-de-flujo/>
- Miñán, Y. P. (2017). Redes Ad hoc de sensores aplicadas a la medición de campos de viento. Recuperado el 02 de 08 de 2018
- Olking. (23 de Octubre de 2008). Tecnologías Inalambricas[Mi Primer Post]. Recuperado el 20 de 10 de 2018, de <http://www.taringa.net/posts/apuntes-y-monografias/1679519/Tecnologias-Inalambricas-Mi-Primer-Post.html>
- Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO). (2016). La Cultura del Agua. Lecciones de la América Indígena.
- Orjuela, A., & Rojas, M. (24 de 06 de 2008). Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. Avances en Sistemas e Informática, 5(2), 14. Recuperado el 17 de 10 de 2018, de <https://revistas.unal.edu.co/index.php/avances/article/view/10037/10567>
- Oyarce, A., Aguayo, P., & Martin, E. (2008). Guía del usuario Xee series 1. Recuperado el 19 de 09 de 2018, de http://www.hmangas.com/Electronica/Datasheets/Shield XBee Arduino/XBee-Guia_Usuario.pdf
- Pallás, A. R. (2003). Sensores y Acondicionadores de Señal (Cuarta ed.). (Marcombo, Ed.) Barcelona. Recuperado el 19 de 07 de 2018
- Pimentel, C. A., & Vislao, Q. R. (2014). Estudio de la Mejora en el Sistema de Lectura de Medidores Eléctricos de la División de Agua Potable y Energía Eléctrica del Proyecto Especial Chavimochic Mediante una Solución Inalámbrica. Tesis. Universidad Privada Antenor Orrego, Trujillo, Perú. Recuperado el 20 de 06 de 2018
- Rodríguez, G. J. (2012). Características del efecto Hall. Recuperado el 19 de 06 de 2018, de <https://electromagnetismo2012a.wikispaces.com/file/view/efecto+hall.pdf>
- Roman, M. (18 de 07 de 2010). Sensores de flujo y caudal (2ª parte). Recuperado el 20 de 08 de 2018, de <http://profesorroman.blogspot.pe/2010/07/sensores-de-flujo-y-caudal-2-parte.html>
- Sánchez, T. L., & Sánchez, T. A. (2004). Uso Eficiente Del Agua. Recuperado el 13 de 08 de 2018, de <http://www.bvsde.paho.org/bvsacg/fulltext/usoeficiente.pdf>
- Sevilla, J. C. (2015). Huella Hídrica del Perú Sector agropecuario. Recuperado el 05 de 08 de 2018
- Sommerville, I. (2005). Ingeniería del Software (Séptima ed.). (P. E. S.A, Ed.) Madrid, España. Recuperado el 13 de 08 de 2018

Superintendencia Nacional de Servicios de Saneamiento (SUNASS). (01 de 02 de 2016).
Superintendencia Nacional de Servicios de Saneamiento en acción. Recuperado el
05 de 08 de 2018, de <https://www.sunass.gob.pe/boletin2017/mayo/index.html>

Toledo, P. A. (2011). Gestión del Agua y Oportunidades para el Desarrollo Profesional.
Lima: Autoridad Nacional del Agua.

ANEXOS

Anexo1: Código desarrollado para el nodo coordinador

```
#include <SoftwareSerial.h>
#include <Ethernet.h>
#include <SPI.h>

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFF, 0xEE}; //Direccion mac del shield ethernet
IPAddress gateway(192, 168, 0, 1); //Puerta de enlace del modem
IPAddress subnet(255, 255, 255, 0); //Mascara del subred
IPAddress ip(192, 168, 0, 2); //ip del shield ethernet
byte server[] = {192,168,0,3}; //ip del servidor

EthernetClient client;
SoftwareSerial mySerial(2, 3); // RX, TX //para dar el seguimiento con el programa hercules

#define ContSegundo 100000
#define EnvioPeticion 3
#define TramasPerdidas 3
#define NunNodos 1
byte SelecNodos = 0;
byte RespuestaNodos[NunNodos];
boolean band_trama = false;
unsigned long MyTimeSeg;
byte RealizarPeticion = 0;
const byte ZiseAPI =19;
//BYTES DE LA TRAMA DE PETICION PARA EL NODO 01
byte Nodo_01[ZiseAPI] = {0x7E, 0x00, 0x0F, 0x10, 0x01, 0x00, 0x13, 0xA2, 0x00, 0x41,
                        0x8E, 0x03, 0x2A, 0xFF, 0xFE, 0x00, 0x00, 0x26, 0x1A };

//la mac se declara en el orden de la creacion de nodos
String NumMac[NunNodos] = {
    "0013A200418E032A"
};
//COMUNICACION SERIAL INTERRUPTIÓN
boolean StringCompleta = false;
String Buffer = "";
void EnvioEthernetXbee (String MisDatos);

void setup(){
    Serial.begin(9600); //la velocidad de transmision
    mySerial.begin(9600);
    Ethernet.begin(mac, ip); //Inicializamos el Ethernet Shield
    mySerial.println("Iniciando.....");
}

void loop(){
    if (MyTimeSeg > ContSegundo){
        MyTimeSeg = 0;
        RealizarPeticion ++;
    }
    if (RealizarPeticion > EnvioPeticion){
        RealizarPeticion = 0;
        SelecNodos ++;
        if (SelecNodos == 1){
            mySerial.println("Envio Trama 01");
            for (byte i = 0; i < ZiseAPI ; i ++ ){
                Serial.write((byte)Nodo_01[i]);
            }
        }
    }
}
```

```

    }
    RespuestaNodos[SelecNodos-1]++;
    if(RespuestaNodos[SelecNodos-1] > TramasPerdidas){
        mySerial.println("Trama 01 perdida");
    }
}
// Serial.print ("NumNodos: "); Serial.println (SelecNodos);
if (SelecNodos > NunNodos){SelecNodos = 0;}
}
if (StringCompleta){
    if(band_trama == true){
        RespuestaNodos[SelecNodos-1] = 0;
        int posicion = Buffer.indexOf('&');
        String StrEnvioET = Buffer.substring(posicion+2,posicion+46);
        StrEnvioET = StrEnvioET + NumMac[SelecNodos-1];
        mySerial.println(StrEnvioET);
        EnvioEthernetXbee (StrEnvioET);
        Buffer = "";
        band_trama = false;
    }
    StringCompleta = false;
}
MyTimeSeg ++;
}
//FUNCION QUE RECUPERA LOS DATOS VALIDOS DEL ESCLAVO
void serialEvent(){ //deteccion de serial por eventos
    while(Serial.available()){//lee la entrada del serial
        char VarChar = (char)Serial.read();
        if((VarChar>34)&&(VarChar<60)){
            Buffer += VarChar;
            StringCompleta = true;
            if(VarChar == '#'){
                band_trama = true;
                StringCompleta = true;
            }
        }
    }
}
}
//ENVIO DE DATOS AL SERVIDOR POR ETHERNET
void EnvioEthernetXbee (String MisDatos){
    if (client.connect(server, 80)>0){
        // Conexion con el servidor
        client.print("GET /agua/controlador/datos.php?Vstr="); // Enviamos los datos por GET
        client.print(MisDatos);//imprime los datos a enviar
        client.println(" HTTP/1.0");
        client.println("User-Agent: Arduino 1.0");
        client.println();
        mySerial.println("Conectado ET");
    }
    else{
        //lcd.clear();lcd.setCursor(0, 0);lcd.print("Fallo la conexion");
        mySerial.println("Fallo en la conexion ET");
    }
    if (!client.connected()){
        //lcd.clear();lcd.setCursor(0, 0);lcd.print("Desconectado");
        mySerial.println("Disconnected! ET");
    }
    client.stop();
    client.flush();
}
}

```

Anexo2: Código desarrollado para el nodo sensor

```
byte failed = 0;
String StrCheck;
String StrVar0;
String StrTramaSendSTR;
boolean StringCompleta = false;
String Buffer = "";
const byte Trm_dat = 48;
byte Array_Trm_dat[Trm_dat+1];
//GENERACION DE TRAMA
const byte Dltr_init = 0x7E;
const byte Long_Msb = 0x00;
const byte Long_Lsb = 0x3E;
const byte Frame_Ty = 0x10;
const byte Frame_ID = 0x01;
const byte Address64[8] = {0x00, 0x13, 0xA2, 0x00, 0x41, 0x8D, 0x97,0x15};
const byte Address16[2] = {0xFF, 0xFE};
const byte BroadCast = 0x00;
const byte Option = 0x00;
byte ByteChecksum=0;
int PinSensor = 2;
float Litros;
float Mili_litros;
volatile int NumPulsos;
long dt=0;
long t0=0;
void PrepararString (void);
void ValorCheckSUM (void);
void TrmApiTransmit (void);
void CuentaPulsos();


---


String floatToString( float,int=8,int=2,boolean=true);
void setup() {
  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0, CuentaPulsos, RISING);
  t0=millis();
  interrupts();
  PrepararString();
}
void loop() {
  float frecuencia=ObtenerFrecuecia();
  float caudal_L_m=frecuencia/5.35;
  dt=millis()-t0;
  t0=millis();
  Litros=Litros+(caudal_L_m/60)*(dt/1000);
  Mili_litros = Litros * 1000;
  PrepararString();
  if (StringCompleta){
    ValorCheckSUM();
    TrmApiTransmit();
    Buffer = "";
    StringCompleta = false;
    Litros = 0;
    Mili_litros = 0;
  }
}
void PrepararString (void){
  failed = 0;
```

```

    StrCheck = String(failed);
    float n = Mili_litros;
    StrVar0 = floatToString(n,8,0,true);
    StrTramaSendSTR="";
    StrTramaSendSTR = "s:" +StrCheck +":" + StrVar0 + ":00000000:00000000:00000000:000000:0#";
    StrTramaSendSTR.toCharArray(Array_Trm_dat, Trm_dat+1);
}
String floatToString( float n, int l, int d, boolean z){
    char c[l+1];
    String s;
    dtostrf(n,l,d,c);
    s=String(c);
    if(z){
        s.replace(" ", "0");
    }
    return s;
}
void CuentaPulsos(){
    NumPulsos++;
}
int ObtenerFrecuecia(){
int frecuencia;
    NumPulsos = 0;
    interrupts();
    delay(1000);
    noInterrupts();
    frecuencia=NumPulsos;
    return frecuencia;
}
void serialEvent(){
    while (Serial.available()){
        char VarChar = (char)Serial.read();
        Buffer += VarChar;
        if (VarChar == 's'){
            StringCompleta = true;
        }
    }
}
void TrmApiTransmit (void){
    Serial.write((byte)Dltr_init);
    Serial.write((byte)Long_Msb);
    Serial.write((byte)Long_Lsb);
    Serial.write((byte)Frame_Ty);
    Serial.write((byte)Frame_ID);
    Serial.write((byte)Address64[0]);
    Serial.write((byte)Address64[1]);
    Serial.write((byte)Address64[2]);
    Serial.write((byte)Address64[3]);
    Serial.write((byte)Address64[4]);
    Serial.write((byte)Address64[5]);
    Serial.write((byte)Address64[6]);
    Serial.write((byte)Address64[7]);
    Serial.write((byte)Address16[0]);
    Serial.write((byte)Address16[1]);
    Serial.write((byte)BroadCast);
    Serial.write((byte)Option);
}

```

```

for (byte i = 0; i < Trm_dat ; i ++ ){
    Serial.write((byte)Array_Trm_dat[i]);
}
Serial.write((byte)ByteChecksum);
}
void ValorChecksum (void){
    int ChkSUM = 0;
    for (byte i = 0; i < Trm_dat; i ++ ){
        ChkSUM += Array_Trm_dat[i];
    }
    ChkSUM = ChkSUM + Frame_Ty + Frame_ID + Address64[0] +
    Address64[1] + Address64[2] + Address64[3] + Address64[4] +
    Address64[5] + Address64[6] + Address64[7] + Address16[0] + Address16[1] + BroadCast + Option;
    byte Min_Lsb = (byte)ChkSUM;
    ByteChecksum = 0xFF - Min_Lsb;
}

```

Anexo3: Fotos de las actividades realizadas en la investigación



Foto1: Instalación del sensor de flujo al lado de la llave principal de agua.



Foto2: Calibración del sensor de flujo con la ayuda de un recipiente graduado.



Foto3: Conexión del sensor con el microcontrolador arduino y el radio Xbee



Foto4: Instalación del coordinador.



Foto5: Conexión del coordinador con el servidor web.



INICIO

SENSORES

NODOS

USUARIOS

REPORTES

Variación del consumo de agua

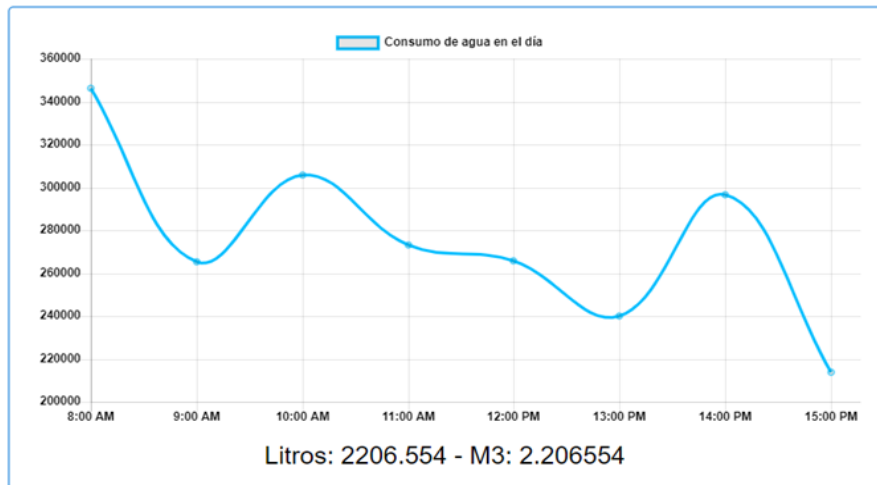
Nodo

EN TIEMPO REAL

Excel PDF

Nodo	Sensor	Mililitros	Hora	Fecha
Nodo 01	Sensor 01	3874	15:45:14	2019-04-08
Nodo 01	Sensor 01	3847	15:45:02	2019-04-08
Nodo 01	Sensor 01	4218	15:44:50	2019-04-08
Nodo 01	Sensor 01	2981	15:42:32	2019-04-08
Nodo 01	Sensor 01	2111	15:42:20	2019-04-08

Mostrando registros del 1 al 5 de un total de 5 registros Anterior 1 Siguiente



EL VALOR DEL AGUA NO SE SABE, HASTA QUE EL FUENTE QUE EMANA ESTE SECO

Todos los derechos reservados © R.R.F, 2019

Foto6: Monitoreo del consumo.

Anexo4: Constancia de ejecución del trabajo de tesis en la I.E.I Federico Villareal
CONSTANCIA DE EJECUCION DE PROYECTO DE INVESTIGACION

DIRECTOR DE LA INSTITUCIÓN EDUCATIVA INDUSTRIAL FEDERICO VILLAREAL.

El que suscribe, Lic. ULISES BULEJE OCAMPO. Otorga la presente constancia de ejecución del proyecto de investigación a:

Rogelio Romaní Flores, bachiller en Ingeniería de Sistemas, de la Universidad Nacional José María Arguedas, identificado con DNI n^{ro}. 47152776.

Quien ha realizado la ejecución de su proyecto de investigación titulado "DISEÑO DE UN PROTOTIPO PARA EL MONITOREO DEL CONSUMO EFICIENTE DE AGUA EN UNA INSTITUCIÓN EDUCATIVA" tendientes a la obtención del título profesional, bajo mi autorización como director de la Institución Educativa, durante un mes, desde el 01 de Abril al 30 de Abril del 2019.

La tesista Rogelio Romaní Flores realizó su ejecución del proyecto de investigación a completa satisfacción y mostro los resultados obtenidos.

Se otorga la presente constancia para fines que el interesado considere conveniente.

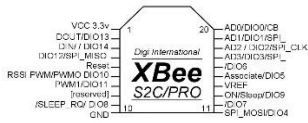
Andahuaylas, 01 de Junio del 2019.



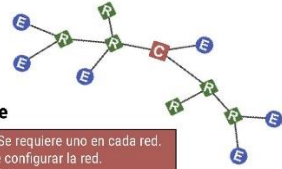
Ulises Buleje Ocampo
Lic. Ulises Buleje Ocampo
DNI: 31186963
C.M. 1031186963

Anexo5: Guía de referencia para xbee s2c pro

GUÍA DE REFERENCIA PARA XBEE S2C PRO



XBee es un microcontrolador fabricado por digi, el cual utiliza el protocolo Zigbee. XBee utiliza 3.3V y tienen un espaciado entre pines menor que el utilizado en un protoboard. Dado lo anterior es recomendable utilizar una tarjeta o kit para que sea más fácil su uso.



Roles XBee

- Coordinador:** Se requiere uno en cada red. Se encarga de configurar la red. No puede dormir.
- Router:** Pueden existir multiples en una red. Pueden redirigir los mensajes a otros routers o End Devices. No pueden dormir.
- End Device:** Pueden existir muchos, no pueden redirigir mensajes. Pueden dormir para ahorrar energía.

Modelo	Especificaciones	Pines I/O	Firmware
S2C	Voltaje de operación: 2.1 – 3.6V Corriente de operación: 33mA@3.3V Rango en interiores: 60 metros Rango en línea vista: 1200 metros Max valor lectura analógica: 1.2V	Entradas analógicas: 4 <i>Configurable como red mesh Red auto reparable</i>	Firmware: Zigbee, DigiMesh, 802.15.4 Velocidad de transmisión RF: 250kbps Frecuencia: 2.4GHz - 2.5GHz Temperatura operación: -40 a 85°C
S2C PRO	Voltaje de operación: 2.7 – 3.6V Corriente de operación: 31mA@3.3V Rango en interiores: 90 metros Rango en línea vista: 3200 metros Max valor lectura analógica: 1.2V	Entradas analógicas: 4 <i>Configurable como red mesh Red auto reparable</i>	Firmware: Zigbee, DigiMesh, 802.15.4 Velocidad de transmisión RF: 250kbps Frecuencia: 2.4GHz - 2.5GHz Temperatura operación: -40 a 85°C

Modos XBee
Transparente: Los dispositivos actúan como un reemplazo de cable serial. Cuando los datos RF son recibidos, el dispositivo envía los datos a través del puerto serie. Utilice la interfaz de modo de comando AT para configurar los parámetros del dispositivo.
Comando: Basada en tramas, amplía el nivel en que una aplicación host puede interactuar con las capacidades de red del dispositivo. Cuando esta en modo API, el dispositivo contiene todos los datos que entran y salen en marcos que definen operaciones o eventos dentro del dispositivo.

Setup XBee
 Conecta el Xbee a un adaptador TTL a Serial como un FTDI
 Utiliza el software gratuito X-CTU para configurar el módulo XBee
 Baud:9600 – FC: Hardware – Data Bits 8 – Parity: None – Stop Bits: 1

Ajustes Básicos
 PAN ID: es la red a la cual se conectará el módulo. Si es 0, el Xbee se asociará a cualquiera que esté disponible
 DH/DL: Es la dirección del módulo de destino. Se utiliza para enviar información a un Xbee en específico. Si se configura en 0 enviará datos solo al coordinador. Si se configura en 0x00000000FFFF hará un broadcast (envía a todos los módulos de la red)

Ajustes Pin
 Para poder trabajar con los pines como entradas/salidas en un Xbee, debe estar configurado en modo API.
 D0 - Configura el pin en 0 para comenzar a leer datos
 IR - realiza una lectura del pin cada XX milisegundos

Byte	Ejemplo	Descripción
0	0x7E	Byte de inicio - indica el comienzo del paquete de datos (frame)
1	0x00	Largo - Número de bytes (ChecksumByte# - 1 - 2)
2	0x10	
3	0x17	Tipo de mensaje - 0x17 significa que es solicitud de comando AT
4	0x01	Frame ID - secuencia del paquete
5	0x00	Dirección de destino de 64-bit (número de serie)
6	0x13	MSB es el byte 5, LSB es el byte 12
7	0xA2	
8	0x00	0x0000000000000000 = Coordinador
9	0x40	0x0000000000000000 = Broadcast
10	0x8B	
11	0x78	
12	0x4E	
13	0xFF	Dirección de la red de destino
14	0xFE	(configurable como 0xFFFF para enviar un bodcast)
15	0x02	Opción del comando remoto (configurable como 0x02 para aplicar los cambios)
16	0x44 (D)	Nombre del comando AT (Dos caracteres ASCII)
17	0x34 (Z)	
18	0x05	Parámetro del comando
19	0x25	Checksum

Conexión con Arduino:
 Arduino TX se conecta la RX de XBee (Data IN)
 Arduino RX se conecta al TX de XBee (Data Out)

Integración con Arduino:
 Los datos enviados utilizando Serial.print() saldrán por el puerto TX del Arduino, que estará conectado al RX del módulo XBee. Si XBee está en modo AT, se transmitirá inalmbricamente hacia el destino. Los datos recibidos en el módulo XBee serán enviados al puerto serial.

Ejemplo para Arduino: Lectura de un valor analógico utilizando modo API

```

// XBee remoto: AT, XBee base: API
if (Serial.available() >= 21) { // Nos aseguramos que ha llegado el mensaje completo
if (Serial.read() == 0x7E) { // 7E es el byte de inicio
for (int i = 1; i < 19; i++) { // descartamos los bytes hasta llegar los datos analógicos
byte discardByte = Serial.read();
}
int analogMSB = Serial.read(); // Lee el primer byte del dato analógico
analogLSB = Serial.read(); // Lee el segundo byte del dato analógico
int analogReading = analogLSB + (analogMSB * 256);
}
}
    
```

Ejemplo Arduino: Cambiar la configuración de un pin en un XBee remoto

```

// XBee remoto: AT, XBee base: API
Serial.write(0x7E); // byte de inicio
Serial.write(byte0x0); // Largo MSB (siempre 0)
Serial.write(0x10); // Largo LSB
Serial.write(0x17); // 0x17 es el tipo de mensaje para enviar comandos AT
Serial.write(byte0x0); // Frame ID (no solicitamos respuesta)
Serial.write(byte00); // Envía los 64 bit de la dirección de destino
Serial.write(byte00); // (Enviando 0x0000000000000000 broadcast)
Serial.write(byte00);
Serial.write(byte00);
Serial.write(byte00);
Serial.write(byte00);
Serial.write(byte00);
Serial.write(0xFF);
Serial.write(0xFF);
Serial.write(0xFF); // Red de destino
Serial.write(0xFE); // (enviar 0xFFFF si es desconocida)
Serial.write(0x02); // configurar 0x02 para aplicar los cambios
Serial.write('D'); // Comando AT : D1
Serial.write('1');
Serial.write(0x05); // Configura D1 para ser 5 (Digital Out HIGH)
long chexsum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '1' + 0x05;
Serial.write(0xFF - (chexsum & 0xFF)); // Checksum
    
```

Byte	Ejemplo	Descripción
0	0x7E	Byte de inicio - indica el comienzo del paquete de datos (frame)
1	0x00	
2	0x14	
3	0x92	Tipo de frame 0x92 indica que es un muestreo de las entradas del Xbee
4	0x00	Dirección de origen de 64-bit (número de serie)
5	0x13	MSB es el byte 4, LSB es el byte 11
6	0xA2	
7	0x00	
8	0x40	
9	0x8B	
10	0x78	
11	0x4E	
12	0xA4	Dirección de 16-bit de la red de origen
13	0x02	
14	0x01	Opciones de recepción: 01 = Packet acknowledged 02 = Broadcast packet
15	0x01	Número de muestras. Siempre debe ser 1 dadas las limitaciones de Xbee
16	0x00	Máscara para el canal digital, indica que pines estan configurados como DIO
17	0x30	
18	0x01	Máscara para el canal analógico, indica cuales pines están configurados como ADC
19	0x00	Lectura de los canales digitales. Estos dos bytes contienen los estados de los pines configurados como DIO
20	0x20	
21	0x02	Lectura del canal analógico.
22	0x0C	Cada canal entrega 2 bytes con el resultado de la lectura del ADC
23	0x20	Checksum (0xFF - la suma de todos los bytes desde el byte 3 a hasta el último)

Modo Sleep
 End Device puede dormir para ahorrar energía. Un End Device que solo despierta cada 5 minutos para enviar datos puede solo estar despierto por 6 segundos en un día.
 SM - 4 = Cyclic Sleep
 SP - Sleep time (hasta 28 segundos)
 SN - Número de ciclos sleep
 ST - Tiempo que permanecerá despierto

Pin I/O Opciones
 0 - Disabled
 1 - N/A
 2 - ADC
 3 - Digital IN
 4 - Digital OUT, LOW
 5 - Digital OUT, HIGH

Máscara para canal digital
 Primer Byte
 n/a n/a n/a D12 D11 D10 n/a n/a
 Segundo Byte
 D7 D6 D5 D4 D3 D2 D1 D0
 Ejemplo:
 0x00 0x13 = 0000 0000 0000 1101
 Pins D3, D2 y D0

Máscara para canal analógico
 (volt) n/a n/a n/a A3 A2 A1 A0
 Ejemplo:
 0x05 = 0000 0101 = Pin A2 and A0