



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**  
**CENTRO UNIVERSITARIO UAEM ATLACOMULCO**



**“SISTEMA AUTOMÁTICO DE ABASTECIMIENTO, TANDEO Y  
DETECCIÓN DE FUGAS EN UNA RED DE AGUA POTABLE A LARGA  
DISTANCIA MEDIANTE TECNOLOGÍA ZIGBEE”**

**T E S I S**

Que para obtener el Título de:

**INGENIERO EN COMPUTACIÓN**

Presenta:

**JUAN CARLOS PÉREZ SOLÍS**

Director de Tesis:

**DR. EN C. I. E. EVERARDO EFRÉN GRANDA GUTIÉRREZ**

Atlacomulco, México; Febrero del 2019

## RESUMEN

El abastecimiento de agua se ha convertido en un gran problema en los últimos años, no solo en las grandes urbes, sino también en las regiones rurales. El gasto excesivo de este recurso, la presencia de fugas, y la contaminación de ríos, lagos y mantos freáticos, ponen el riesgo el acceso al agua para todas las personas. Actualmente se encuentran en marcha diversos proyectos e investigaciones para mitigar los efectos adversos de este fenómeno.

En este trabajo, se presenta la implementación de un sistema de llenado automático, tandeo (distribución de agua, alternativamente o por tandas (Real Academia Española, 2014) y detección de fugas para depósitos elevados de agua potable de la localidad de Palmillas, San Felipe del Progreso, en el Estado de México, ya que los dos depósitos con que dicha comunidad cuenta se localizan alejados de la bomba de abastecimiento con distancias de 1.55 km y 2.03 km, respectivamente.

La lejanía de los depósitos complica la supervisión del llenado de estos por las personas encargadas, ya que es frecuente que se desborden cuando llegan a su máxima capacidad, ocasionando un mal uso de la bomba de abastecimiento, desperdiciando agua y energía eléctrica. Con la aplicación del sistema que se propone en este documento, potencialmente se podrían ahorrar costos en cuanto al personal que se le está pagando, así como ahorro de energía eléctrica y, sobre todo, evitar el desperdicio de agua potable.

El sistema diseñado está basado en la plataforma Arduino y módulos comunicados mediante la tecnología ZigBee, de tal forma que permite realizar el llenado automático de los depósitos, además de detectar fugas mediante sensores ultrasónicos que detectan el nivel de agua en los depósitos. También permite control sobre el tandeo de agua mediante válvulas que se activan de manera automática con la ayuda de módulos ZigBee y motores engranados con capacidad de abrir válvulas de cuatro pulgadas. Finalmente se incorporan celdas fotovoltaicas en los elementos remotos del sistema, ya que en algunas zonas de esta comunidad aún no se cuenta con energía eléctrica. El funcionamiento del sistema propuesto se validó mediante el uso de un modelo funcional a escala, para evaluar su posible implementación física.

**Palabras clave:** protocolo ZigBee, sistema de abastecimiento de agua, automatización.

## **ABSTRACT**

Water supply has become a big problem over the past few years, not only in big cities but also in rural regions. The excessive spending of this resource, the occurrence of leakages and the pollution of rivers, lakes and phreatic levels, put at risk the access to water for the people. Nowadays, different projects and researches to mitigate the adverse effects of this phenomena, are in progress.

In this work, it is presented the implementation of a system for automatic filling, valve opening and leakage detection for elevated tanks of drinkable water from the town of Palmillas, San Felipe del Progreso, Sate of Mexico. The two available tanks for water distribution are located at distances of 1.55 and 2.03 km, respectively, from the supply pump.

Remoteness of water tanks complicates the supervision of their filling for the responsible people, and the overflow of the tanks is frequent due to the unattended filling when they reach their maximum capacity, thus causing pump overuse, as well as water and energy waste. With the application of the system proposed in this document, could potentially save costs in terms of the personnel who are being paid, as well as saving electricity and avoiding the waste of drinking water.

The designed system is based on Arduino platform and communication modules with ZigBee technology. It allows the automatic filling of water tanks, as well as the leakage detection by means of ultrasonic sensors which detect the water level inside the tanks. Also, it affords the possibility of control in water distribution by automatic actuation of valves remotely activated by ZigBee modules and geared motors to open 4-inches valves. Finally, photovoltaic cells are incorporated in remote elements of the systems, because some zones of the community do not have yet electric energy. The functioning of the system was validated through a functional model, to evaluate its possible physical implementation.

**Keywords:** ZigBee protocol, water distribution system, automation.

# ÍNDICE

DEDICATORIAS .....	i
AGRADECIMIENTOS .....	ii
RESUMEN.....	iii
ABSTRACT.....	iv
ÍNDICE .....	v
ÍNDICE DE TABLAS .....	vii
ÍNDICE DE FIGURAS.....	viii
1 INTRODUCCIÓN.....	11
2 PLANTEAMIENTO DEL PROBLEMA .....	12
2.1 Definición del problema.....	12
2.2 Objetivos de investigación .....	13
2.2.1 Objetivo General:.....	13
2.2.2 Objetivos Específicos:.....	13
2.3 Preguntas de investigación .....	14
2.4 Justificación.....	14
2.5 Impactos .....	15
3 META DE INGENIERÍA.....	16
4 ESTADO DEL ARTE .....	17
4.1 Trabajos relacionados con el cuidado del agua .....	17
4.2 Sistemas.....	21
4.3 Señales.....	22
4.4 Medidores de niveles en líquidos .....	23
4.5 Arquitectura Arduino .....	26
4.6 Protocolo de comunicación inalámbrica ZigBee.....	28

4.7	Celdas fotovoltaicas .....	34
5	METODOLOGÍA .....	36
5.1	Requerimientos de diseño .....	36
5.2	Diseño e implementación de la circuitería y lógica.....	37
5.3	Interfaz.....	55
5.4	Experimentación para la validación del sistema .....	61
6	ESCALAMIENTO AL SISTEMA REAL .....	66
7	RESULTADOS Y DISCUSIÓN .....	73
	CONCLUSIONES .....	82
8	Bibliografía .....	84
	ANEXOS .....	88

## ÍNDICE DE TABLAS

Tabla 4.1 Diferencia entre señal Analógica y señal Digital.....	23
Tabla 5.1. Sensores y Actuadores. ....	37
Tabla 6.1 Características de la bomba MS125-40 .....	67
Tabla 6.2 Características de módulos XBee ZigBee.....	67
Tabla 6.3 Características del sensor JSN-SR04T.....	69
Tabla 6.4 Características Motor a pasos Nema 23 .....	70
Tabla 6.5 Características de controlador TB6600.....	70
Tabla 6.6 Características celdas fotovoltaicas. ....	71

# ÍNDICE DE FIGURAS

Figura 4.1. Sistema de control automático. Fuente: (Hernández Gaviño, 2010).....	22
Figura 4.2 Sensor Ultrasónico HC-SR04 Fuente: (ETOOLS, 2016).....	24
Figura 4.3. Operación del sensor ultrasónico. (Corona Ramírez, et al., 2014). ....	25
Figura 4.4. Tarjeta de adquisición de datos Arduino UNO Fuente: (Arduino, 2018).....	27
Figura 4.5. Módulo XBee Fuente: (XBee.cl, 2018).....	29
Figura 4.6. Tipos de antena del módulo XBee.....	30
Figura 4.7. Correspondencia del Buffer Fuente: (Oyarce, 2008).....	33
• Figura 4.8. Modo de operación del módulo XBee Fuente: (XBee Comunicaciones inalámbricas, 2014).....	34
Figura 5.1. Diagrama de flujo de la Metodología de investigación. ....	38
Figura 5.2. Obtención de distancias mediante el sensor. ....	39
Figura 5.3 Conversión a porcentaje. ....	39
Figura 5.4. Visualización de porcentajes .....	40
Figura 5.5. Conexión sensor (HC-SR04) con Arduino. ....	40
Figura 5.6. Especificación de estado de la bomba según los niveles del tanque. ....	41
Figura 5.7. Suministro de agua .....	41
Figura 5.8. Programación de encendido de bomba según los niveles.....	42
Figura 5.9. Proceso detección de fugas. ....	42
Figura 5.10. Diagrama de flujo de las válvulas.....	43
Figura 5.11. Asignación de puerto (COM). ....	44
Figura 5.12. Botón de búsqueda de los módulos. ....	44
Figura 5.13. Selección de dispositivos encontrados.....	45
Figura 5.14. Parámetros del XBee. ....	46
Figura 5.15. Dispositivos encontrados con sus especificaciones.....	46
Figura 5.16. Divisor de voltaje 5 V a 3.3 V .....	48
Figura 5.17. Configuración para el módulo Coordinador. ....	48
Figura 5.18. Configuración para módulo End-Device. ....	49
Figura 5.19. Comunicación entre el Coordinador y el Deposito1.....	49
Figura 5.20. Comunicación Coordinador y el Deposito2. ....	50
Figura 5.21. Declaración de pines para entrada de un nuevo serial. ....	51

Figura 5.22. Encendido de la bomba según los niveles de agua.....	51
Figura 5.23 Detección de dispositivos Bluetooth. ....	52
Figura 5.24 Conexión Bluetooth.....	52
Figura 5.25 Información insertada en los Labels de la interfaz. ....	52
Figura 5.26. Conexión del circuito de los tanques.....	53
Figura 5.27. Conexión del circuito del Coordinador.....	54
Figura 5.28. Diagrama de interacción del usuario y sistema. ....	56
Figura 5.29. Diagrama de la interfaz grafica.....	57
Figura 5.30. Instalación de la aplicación móvil. ....	57
Figura 5.31. Conexión Bluetooth.....	58
Figura 5.32. Interfaz gráfica móvil. ....	59
Figura 5.33. Funcionamiento del sistema modo Automatico. ....	60
Figura 5.34. Funcionamiento del sistema modo manual. ....	61
Figura 5.35. Nivel del contenedor 1 a 96%.....	62
Figura 5.36. Nivel del contenedor 2 a 50%.....	62
Figura 5.37. Niveles de los tanques. ....	62
Figura 5.38. Conversión de distancias a porcentaje.....	63
Figura 5.39. Detección de fugas sin ninguna perdida. ....	64
Figura 5.40. Fuga no significativa. ....	64
Figura 5.41. Detección de fugas notoria. ....	65
Figura 5.42. Realización de tandeo.....	65
Figura 6.1. Protocolo en malla.....	68
Figura 6.2. Diseño del sistema real de agua potable de la comunidad Palmillas, San Felipe del Progreso, en el Estado de México.....	72
Figura 7.1. Comunicación entre los dispositivos. ....	73
Figura 7.2. Información enviada por el microcontrolador Arduino.....	74
Figura 7.3. Recepción de datos del Coordinador.....	74
Figura 7.4. Interfaz gráfica móvil. ....	75
Figura 7.5. Alerta de fuga.....	76
Figura 7.6. Fuga pequeña no relevante. ....	76
Figura 7.7. Tandeo modo automático. ....	77

Figura 7.8. Intensidad de señal a 10 metros.....	78
Figura 7.9. Intensidad de señal a 30 metros.....	78
Figura 7.10. Intensidad de señal a 50 metros.....	79
Figura 7.11. Intensidad de señal a 80 metros.....	80
Figura 7.12. Intensidad de señal a 90 metros.....	80
Figura 7.13 Gastos generados mensualmente con y sin el sistema.....	81

# 1 INTRODUCCIÓN

El desperdicio de agua en todo el mundo es muy común debido, entre otras cosas, a los descuidos de la humanidad; pero ahora, gracias a la tecnología, se puede ayudar en la solución de este problema, creando aplicaciones móviles que ayudan al cuidado del vital líquido. Estos sistemas han tratado el problema del abastecimiento, y usualmente necesitan de internet para ser operados (Rashid, et al., 2017).

Con el constante avance que tiene la tecnología y especialmente el uso de sensores colocados en las tuberías de agua ha sido posible verificar fugas, y si las hay informar a los usuarios de lo que está sucediendo mediante dispositivos móviles para solucionar lo más pronto posible este problema (Gama Moreno, et al., 2016).

En lo que se refiere al control de llenado de depósitos de agua, en algunos trabajos, se reporta que se han implementado sensores ultrasónicos en tanques de almacenamiento de agua potable para evitar el desbordamiento del agua mediante la activación y desactivación de la bomba, reduciendo el desperdicio de agua y proporcionando cierto control del sistema mediante LabVIEW y Arduino (Shrenika, et al., 2017).

En poblaciones semiurbanas, es común que el llenado de depósitos, así como el tandeo del líquido a diferentes sectores de la comunidad, sea llevado a cabo manualmente, por personal operario que recorre grandes distancias para abrir o cerrar válvulas, y para activar o apagar la bomba de suministro principal en el pozo, por lo que frecuentemente ocurren desbordamientos o fugas que tardan mucho en detectarse, ocasionando el desperdicio de agua y el aprovechamiento inadecuado de recursos humanos.

En este trabajo, se presenta el desarrollo de un prototipo que pueda detectar automáticamente cuando los depósitos de agua necesiten ser llenados con la bomba de abastecimiento, así como la activación de válvulas en días específicos para repartir el agua en todos los barrios que tiene la comunidad Palmillas, San Felipe del Progreso, en el Estado de México. El sistema hace uso de tecnología Zigbee, con módulos RF Digi XBee-PRO S2C (Digi, 2018), que tienen un alcance de hasta 3200 m; también se hará uso de la plataforma Arduino para lograr dicha comunicación, y de este modo tener un mejor control del agua, evitando el desperdicio de tan valioso recurso.

## **2 PLANTEAMIENTO DEL PROBLEMA**

El cuidado del agua es muy importante en la actualidad, debido a la escasez de esta y el incremento en su demanda. Es una actividad que genera interés, debido a la escasez del agua en el mundo se ha visto la necesidad de cuidar el agua de todas las maneras posibles para evitar el desperdicio de este recurso.

En la localidad de Palmillas, San Felipe del Progreso, en el Estado de México, este recurso se bombea a largas distancias y es muy complicado saber cuándo debe detenerse su suministro, por lo que los tanques se desbordan con frecuencia, provocando desperdicio del líquido (se ha estimado en hasta 4,800 L en cada llenado, tomando como base el tiempo que dura el bombeo de agua). Además, los tanques no tienen un sistema que les permita saber si hay fugas en ellos, generando dudas sobre el estado de dichos contenedores.

La aplicación de la tecnología como herramienta de ayuda para resolver este problema es el objetivo de esta investigación, aprovechando las competencias obtenidas en el trayecto de la licenciatura de Ingeniería en Computación, específicamente abordando conocimientos sobre programación y electrónica, para lograr la comunicación de la bomba de abastecimiento y los depósitos que se encuentran muy alejados para poder tener un mejor aprovechamiento del agua, identificar fugas y tener un mejor control del suministro de agua hacia todos los barrios de dicha comunidad. Los métodos tendrán bases en el desarrollo de técnicas de comunicación con módulos RF Digi XBee-PRO S2C, gestionados mediante Arduino.

### **2.1 Definición del problema**

En la comunidad antes mencionada, se tiene una red de suministro de agua conformada por dos depósitos con capacidad de 172,800 L cada uno, que se encuentran a una distancia de 1.55 km y 2.03 km, respectivamente, de la bomba de abastecimiento. Para su operación se necesitan cuatro personas, que cobran un sueldo de \$ 1500.00 pesos semanales y se encargan de realizar el llenado y suministro de agua; en ocasiones se desperdicia agua por errores humanos atribuibles de los encargados (suministro a horas no adecuadas, mal uso de la bomba de abastecimiento) o por no tener un control preciso del recurso (no se tiene

certeza del nivel del agua, las fugas se identifican hasta que ya es muy notorio el desperdicio).

Actualmente se tiene una mala distribución de agua, ya que se abren las válvulas durante muchas horas para algunos sectores de la población, lo que permite que las personas tengan agua excesivamente, ocasionando el desinterés del ahorro de dicho recurso, y fomentando el desperdicio, mientras que otras quedan desatendidas. Además, los tanques no cuentan con diagnóstico de fugas, lo que lleva a que los encargados no sepan si hay desperdicio de agua; esto se agrava también puesto que en estos lugares no se cuenta con energía eléctrica, lo que no ha permitido implementar sistemas para el cuidado del recurso hídrico.

## **2.2 Objetivos de investigación**

### **2.2.1 Objetivo General:**

Diseñar un prototipo para el control de abastecimiento de agua potable en un sistema de bombeo y tanques elevados, que permita su operación automática, el monitoreo y diagnóstico de fugas y la distribución por tandeo mediante válvulas motorizadas. Este prototipo requiere además la alimentación mediante celdas fotovoltaicas.

### **2.2.2 Objetivos Específicos:**

Diseñar un sistema para monitorear el nivel del agua de los depósitos, basado en el uso de sensores, módulos Arduino y XBee.

Desarrollar un método que permita abrir las válvulas de agua automáticamente, con la ayuda de motores.

Obtener información sobre los tanques para verificar que no tengan fugas, con uso de sensores ultrasónicos como detectores de nivel.

Calcular la cantidad de paneles solares requeridos para lograr la comunicación de los dispositivos y ahorrar energía eléctrica.

Crear una aplicación móvil para el monitoreo del sistema automático de abastecimiento, tandeo y detección de fugas.

Simular la operación del sistema diseñado mediante el uso de un modelo funcional como una representación del sistema real.

### **2.3 Preguntas de investigación**

¿Cuáles son las herramientas tecnológicas disponibles hoy en día para monitorear los niveles de recursos hídricos (sensores, dispositivos, algoritmos, aplicaciones, etc.)?

¿Cuáles son las técnicas computacionales y materiales más adecuados para el desarrollo de un sistema de monitoreo de nivel en tanques de agua y tandeo del recurso?

De acuerdo con el funcionamiento del sistema de distribución de agua que opera en la comunidad de Palmillas, ¿qué acciones se pueden llevar a cabo para mejorar su funcionamiento mediante el uso de herramientas computacionales, tanto de software como de hardware?

### **2.4 Justificación**

El recurso hídrico es muy importante para la humanidad; debido a la escasez de este recurso, el hombre ve la necesidad de implementar sistemas que nos ayuden a su cuidado. El acceso al agua potable es, además, un derecho humano.

Durante los últimos años, en la localidad de Palmillas, San Felipe del Progreso, se han desperdiciado más de 4800 litros de agua en cada llenado de los depósitos, ya que los responsables de llevar a cabo el bombeo y repartición del agua deben realizar grandes recorridos para activar o desactivar los elementos de la red de distribución, lo que hace ineficiente su trabajo.

Esta comunidad ha tenido este problema ya que sus depósitos se construyeron en cerros, los cuales están alejados de la bomba de abastecimiento a una distancia de 1.55 km a 2.03 km de distancia; esto ha llevado a que los encargados del pozo de agua activen la bomba durante un tiempo estimado por su experiencia, sin saber si el depósito ha alcanzado la máxima capacidad, produciendo derrames y desperdicio de agua.

En esta localidad, por el momento, no se sufre de escasez de agua, pero esto no es motivo para que se desperdicie tan valioso recurso. La presente investigación se justifica en la importancia de la prevención y adecuada administración del agua en dicha localidad.

El prototipo que aquí se presenta potencialmente permitirá evitar el desperdicio del recurso hídrico, detectar fugas y tener un mejor control del suministro de dicho recurso, ya que con su implementación se podrían reducir algunos de los gastos que se realizan, como como la optimización de los recursos humanos (operarios) y el uso de la luz eléctrica.

## **2.5 Impactos**

### **Científico:**

Optimización de los métodos ya establecidos en cuanto al cuidado del agua potable, dirigido al cuidado automático del agua a largas distancias, con la ayuda de XBee y Arduino.

### **Tecnológico:**

Establecer las bases para el desarrollo de un sistema que nos permita tener un mejor control del agua potable ya sea abastecimiento de los tanques o suministro del recurso para los habitantes de Palmillas, San Felipe del Progreso, en el Estado de México

### **3 META DE INGENIERÍA**

Mediante un sistema para el monitoreo de nivel en dos depósitos de almacenamiento de agua potable, que haga uso de la tecnología ZigBee para la transmisión de datos en una red de comunicaciones y paneles solares para la alimentación de energía en los nodos, se podrá controlar a distancia, de manera autónoma, el llenado de dichos depósitos, así como la activación de válvulas para el tandeo en la distribución del líquido en diferentes zonas de la comunidad, además de un mecanismo para la detección de fugas durante los periodos de reposo. Para validar el sistema se hará uso de un modelo funcional, que considere las características de un sistema real.

## **4 ESTADO DEL ARTE**

El cuidado del recurso hídrico ha sido un gran reto para la humanidad durante los últimos cincuenta años: se han creado varios sistemas para su cuidado utilizando diferente tipos de dispositivos electrónicos, tales como: implementación de sensores ultrasónicos (Gama Moreno, et al., 2016), sensores infrarrojos (Rashid, et al., 2017), entre otros. Estos sistemas tienen un objetivo en común: cuidar el agua de forma automática y evitar su desperdicio, ya que es un recurso muy importante para todo el mundo.

Atender los problemas sobre el desperdicio de agua ha sido un gran reto para la ciencia y un buen tema de interés para todo el mundo; para esto se han desarrollado algunos sistemas para llevar el control de dicho recurso. Por ejemplo, algunos pueden manipular el abastecimiento de agua para uso doméstico desde dispositivos móviles, y de esta manera saber cuándo los tanques están llenos para detener el abastecimiento (Gama Moreno, et al., 2016).

### **4.1 Trabajos relacionados con el cuidado del agua**

Con respecto al tema de investigación, se han desarrollado diversos trabajos a nivel mundial, pero en México no se han atendido como una prioridad. El tener tanques con sensores ultrasónicos que puedan informar en los móviles el llenado de los tanques son proyectos de gran utilidad, ya que el suministro de agua se ha convertido en un gran problema en las ciudades, así como en instalaciones de agricultura debido a la sobrepoblación, el cambio climático, las infraestructuras antiguas de agua y la contaminación (Gama Moreno, et al., 2016).

En la mayoría de los sistemas para el cuidado del agua que se han identificado en el estado del arte, se emplean tarjetas de adquisición de datos Arduino, ya que es ideal para proyectos que requieren que la placa se comporte (actúe) como un dispositivo de interfaz USB humano (Arduino, 2018)

Humayun Rashid y colaboradores (Rashid, et al., 2017), en el trabajo “Sistema de automatización del hogar para pacientes con discapacidad física” ayudan a personas con paraplejia, a lograr el llenado de tanques de manera automática desde dispositivos móviles

sin la necesidad de estar al pendiente de dicha operación. Mediante este proyecto se redujo el desperdicio de agua, usando el agua del grifo y sistema de tanque de agua automático. Se implementó un sistema renovable mediante el panel de carga solar y controlador de carga solar para accionar el sistema completo y reducir el consumo de energía de la energía convencional.

En el trabajo mencionado anteriormente se llenan los tanques automáticamente mediante el control del sistema a partir de una aplicación móvil. Sin embargo, este sistema no podría trabajar a distancias mayores a 20 m además de no contar de una detección de fugas. El sistema que se propone en este documento tiene un alcance de comunicación en distancias mayores a 1 km, debido al uso de la tecnología ZigBee, además de que se incorpora un mecanismo para identificar fugas a partir del nivel de agua en los tanques; también tiene la capacidad de abrir válvulas de agua, para poder llevar acabo el tandeo del líquido a diferentes zonas de distribución. El diseño de este sistema pretende satisfacer las necesidades de una red de agua potable (abastecimiento, tandeo y detección de fugas), aunque la validación de este se realiza en un modelo funcional que representa el sistema real.

Por otra parte, se ha identificado que en varios sistemas de monitoreo se hace uso de LabVIEW, que es un software de ingeniería diseñado para aplicaciones que requieren pruebas, medidas y control con acceso rápido a información de datos y hardware (National Instruments, s.f.). Este software permite una programación gráfica de algoritmos para la adquisición de datos por medio de una computadora de escritorio o portátil.

En el artículo, "Sistema de monitoreo de nivel de agua sin contacto implementado con LabVIEW y Arduino" (Shrenika, et al., 2017), se realiza un sistema de control del nivel de agua sin contacto utilizando LabVIEW y Arduino. El nivel del agua del depósito se mide por medio de un sensor ultrasónico, por lo que el nivel de agua presente en el tanque se conoce. A partir de una aplicación desarrollada en LabVIEW, se toma una lectura del sensor, que envía los datos a través de Arduino. En la interfaz programada se aprecia cuando la bomba se ilumina (indicando la activación de esta) si el nivel del agua en el depósito superior es bajo y la bomba se desconecta cuando el nivel del agua está completamente lleno en el depósito superior. La placa Arduino tiene la tarea de leer la

altura y enviar los informes de la profundidad del agua del tanque, para su posterior visualización en el panel frontal de LabVIEW.

La desventaja del uso de software de licencia propietaria, como LabVIEW tiene el inconveniente del costo asociado a dicha licencia, por lo que también se han desarrollado sistemas para el cuidado del agua tratando de ser lo más económicos que se puedan con la utilización de hardware libre Arduino, junto con recursos software de código de Java (para monitorear los datos), App Inventor2 (para crear aplicación en Android), MySQL Server (para guardar la información en una base de datos), entre otros, lo que ha favorecido la reducción de los costos. Inicialmente, se estudian los sensores y detectores actualmente disponibles en el mercado, para poder tomar una decisión adecuada a la aplicación particular de cada proyecto (Guerrero Cruz & Games Segales, 2017).

El agua se ha cuidado de muchas formas, así que se implementan sistemas de varios tipos como lo son los sistemas de riego automáticamente para lograr su dicho cuidado (Calle Zambrano & Gaibor Vistin, 2017), estos sistemas de una u otra manera cuidan el agua solo regando las plantas cuando la tierra no tienen la humedad adecuada, pero de igual manera utilizan tarjetas de adquisición de datos Arduino para su elaboración.

Se han desarrollado sistemas para el control del nivel y flujo del agua mediante elementos electrónicos, tales como: sensores, bombas, variadores de frecuencia, pantallas HMI (Interfaz Hombre Máquina), módulos de comunicación, conversor digital analógico, PLC's (Controlador Lógico Programable), que son utilizados en la industria. De esta forma se ha evitado el desbordamiento de líquidos en tanques, y como consecuencia de esto, se evitan costos asociados al desperdicio de fluidos. Además, se ha demostrado que el realizar un proceso de llenado de forma manual, sin la supervisión adecuada, presenta problemas en los productos por exceso de líquidos (Bastida Perero, 2017).

Las fugas de agua afectan mucho en los costos, la salud y el medio ambiente, ya que al desperdiciar agua se gasta también una gran cantidad de energía. Las fugas pueden afectar la calidad del agua contaminándose por las impurezas de las fugas. También dependen del sistema de mantenimiento de las redes de distribución de agua, se han propuesto soluciones mediante un análisis de datos de control de fugas, en las redes de distribución

de agua no se ha podido reducir el desperdicio de agua debido a las fugas. Hay básicamente tres métodos de gestión de fugas (Gupta, et al., 2014).

1. Método de evaluación: verificar el estado de los tanques y las tuberías.
2. Método de detección: la pérdida de agua es notoria, se detecta el área dañada.
3. Método de control: reparación de las fugas detectadas, evitando pérdidas de agua actual y futura.

La detección de una fuga debería detectarse lo más pronto posible, para procurar la solución a la misma y evitar que el recurso hídrico se desperdicie de manera innecesaria (Soubam, et al., 2017).

Se han utilizado sistemas que monitorean diversos parámetros en procesos que utilizan líquidos, mediante la identificación de las condiciones de proceso de la planta. En estos casos, se resuelve un problema de índole matemático para la variable “nivel” en un tanque de agua, al cual se brinda una mejor exactitud del modelo empleando técnicas de identificación y control robusto, mejorando la dinámica del proceso ante perturbaciones (Añez & Scarano, 2014).

Como se ha comentado anteriormente, para hacer frente a la problemática de la distancia que existe entre el punto de bombeo, los depósitos de almacenamiento y las válvulas de distribución del agua, se propone el uso de herramientas de bajo consumo energético y baja tasa de respuesta, pero gran alcance, para poder realizar el enlace entre las localizaciones físicas de estos lugares. Para ello, se propone el uso de la tecnología ZigBee, como parte central del proceso de comunicación.

ZigBee es un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en redes inalámbricas de área personal. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de las baterías. (Aprendiendo Arduino, 2016)

En el artículo, “Parámetros de configuración en módulos XBee-PRO® S2B ZB para medición de variables ambientales”, se presenta información sobre la configuración de la comunicación inalámbrica mediante los protocolos del Instituto de Ingenieros Eléctricos

y Electrónicos IEEE 802.4.15 y ZigBee®. Haciendo uso de los módulos de radiofrecuencia XBee-PRO® S2B ZB de Digi International Inc. En dicho trabajo se muestra la fase de diseño del enlace inalámbrico utilizando la topología estrella; también se realiza la caracterización de los parámetros para configurar los dispositivos de radiofrecuencia que permiten la comunicación para la medición de los parámetros de algunas variables ambientales mediante el acoplamiento de diversos sensores utilizando el XBee *Shield* para la placa del microcontrolador Arduino Mega 2560. Se forman cuatro nodos, cada uno de los cuales permite medir cuatro variables: temperatura ambiental, humedad relativa, precipitación y radiación solar global. A su vez, el coordinador y una tarjeta Arduino Ethernet permiten visualizar los datos en tiempo real mediante un aplicativo en la red de área local (Vera Romero, et al., 2015).

## **4.2 Sistemas**

Un sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado, y que no está necesariamente limitado a los sistemas físicos; el concepto de sistema se puede aplicar a fenómenos abstractos y dinámicos, como los que se encuentran en la economía. Por tanto, la palabra sistema debe interpretarse en un sentido amplio que comprenda sistemas físicos, biológicos, económicos u otros (Ogata, 2010). Los sistemas reciben datos, energía o materia del ambiente (entrada) y proveen información, energía o materia (salida). (Franquet Bernis, 2013).

### **Sistemas analógicos**

Contienen dispositivos que manipulan cantidades físicas representadas en formas analógicas, es decir que las cantidades varían sobre un intervalo continuo de valores. Por ejemplo, en un receptor de radio la amplitud de la señal de salida para una bocina puede tener cualquier valor entre cero y su límite máximo (Tocci & Widmer, 2003).

## Sistemas digitales

Son una combinación de dispositivos diseñados para manipular cantidades físicas o información que estén representadas en forma digital; es decir, que solo puedan tomar valores discretos. La mayoría de las veces, estos dispositivos son electrónicos, pero también pueden ser mecánicos, magnéticos o neumáticos. Algunos de los sistemas digitales más conocidos incluyen las computadoras y calculadoras (Tocci & Widmer, 2003).

## Sistema de control automático

Un sistema de control automático es una interconexión de elementos que forman una configuración denominada sistema, de tal manera que el arreglo resultante es capaz de controlarse por sí mismo. Un sistema o componente del sistema susceptible de ser controlado, al cual se le aplica una señal  $r(t)$  a manera de entrada para obtener una respuesta o salida  $y(t)$ . (Hernández Gaviño, 2010). Este tipo de sistemas puede representarse mediante el esquema de la Figura 4.1.

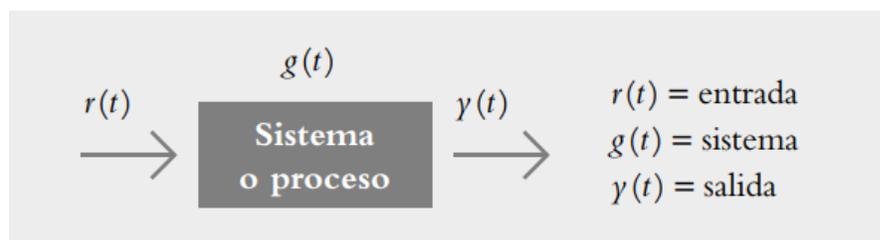


Figura 4.1. Sistema de control automático. Fuente: (Hernández Gaviño, 2010)

## 4.3 Señales

Las señales pueden describir una amplia variedad de fenómenos físicos. Las señales se representan matemáticamente como funciones de una o más variables independientes. (Oppenheim & Willski, 2014). Las señales se pueden clasificar en analógicas o digitales. La diferencia principalmente es su amplitud y tiempo, como se ilustra en la Tabla 4.1.

*Tabla 4.1 Diferencia entre señal Analógica y señal Digital*

	Analógica	Digital
Amplitud	Continua	Discreta
Tiempo	Continua	Discreta
Notación	$x(t)$	$x[n]$

Las señales analógicas usualmente son variables que pueden presentarse en forma de corriente, tensión o una carga eléctrica. Varían en forma continua entre un límite inferior y un límite superior; cuando estos límites coinciden con los límites que admite un determinado dispositivo, se dice que la señal está normalizada. La ventaja de trabajar con señales normalizadas es que se aprovecha mejor la relación señal/ruido del dispositivo (Miyara, 2004).

Por otro lado, las señales digitales son variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo transmitiendo información según un código previamente acordado. Cada nivel eléctrico representa uno de dos símbolos: 0 o 1, Verdadero o Falso. Los niveles específicos dependen del tipo de dispositivos utilizado (Miyara, 2004).

#### **4.4 Medidores de niveles en líquidos**

Este tipo de dispositivos trabajan midiendo directamente la altura del líquido sobre una línea de referencia, mediante la presión hidrostática, el desplazamiento producido por el líquido contenido en un tanque de proceso o bien por las características eléctricas del líquido (QuimiNet.com, 2010). Los instrumentos que utilizan las características eléctricas del líquido son (Creus, 2010):

1. Medidor resistivo/conductivo
2. Medidor capacitivo
3. Medidor ultrasónico
4. Medidor de radar o microondas

5. Medidor de radiación
6. Medidor de láser

El sensor ultrasónico HC-SR04 que se muestra en la Figura 4.2, es útil para medir distancias y detectar obstáculos. Su funcionamiento consiste en enviar una señal ultrasónica (inaudible para el ser humano) y calcular el tiempo que demoró en ir y venir hasta el obstáculo más cercano que detectó. La respuesta del dispositivo es una señal proporcional a esta distancia. Generalmente están conformados por dos cilindros puestos uno al lado del otro; uno de ellos es quien emite la señal ultrasónica, mientras que el otro es quien la recibe (ETOOLS, 2016).

La descripción de los pines es la siguiente:

1. VCC. Voltaje de alimentación del sensor de 5 V.
2. Trigger. Señal de entrada que habilita una medición del sensor.
3. Echo. Señal de salida que emite un pulso en alto con una duración correspondiente al tiempo que le toma a la señal ultrasónica salir del TX y llegar al RX (Corona Ramírez, et al., 2014).



*Figura 4.2 Sensor Ultrasónico HC-SR04 Fuente: (ETOOLS, 2016).*

Un sensor ultrasónico puede usarse en distintos modos de operación, de acuerdo con el tipo de medición que se desea obtener. En el siguiente diagrama de la Figura 4.3 se listan los posibles modos de colocación (Corona Ramírez, et al., 2014).

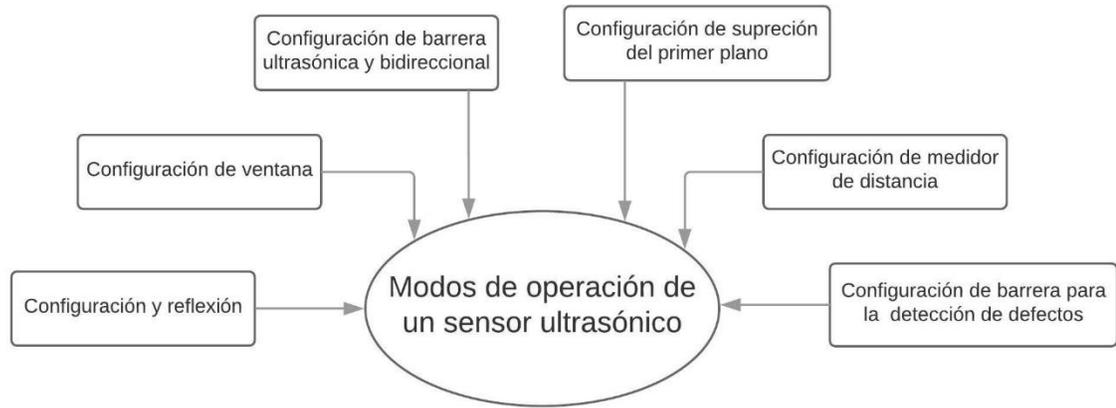


Figura 4.3. Operación del sensor ultrasónico. (Corona Ramírez, et al., 2014).

Para realizar el cálculo de la distancia mediante este sensor, se debe considerar que la señal es transmitida como una onda de sonido, de tal manera que se considera lo siguiente: (Corona Ramírez, et al., 2014)

La expresión (1) muestra la velocidad y el respectivo despeje para la obtención de la distancia.

$$v = \frac{d}{t}; \text{ por tanto, } d = v * t \quad (1)$$

$d$ =Distancia

$t$ =Tiempo

La expresión (2) muestra la sustitución de la velocidad del sonido a través del aire

$$d = 343 \frac{m}{s} * t \quad (2)$$

Sin embargo, como  $d$  corresponde a la distancia recorrida en un tiempo, que va del TX al objeto y del objeto al RX,  $d$  es el doble de la distancia entre el sensor y el obstáculo.

La expresión (2) se divide en dos obteniendo como resultado la expresión (3).

$$d = 171.5 \frac{m}{s} * t \quad (3)$$

Si la distancia se requiere en centímetros y el tiempo se cuenta en microsegundos se realizan las conversiones obteniendo como resultado la expresión (5).

$$d = 171.5 \frac{m}{s} * t = 17150 \frac{cm}{s} * t(\mu s) \times 10^{-6} \quad (4)$$

$$d = 0.01715 * t(\mu s) cm \quad (5)$$

## 4.5 Arquitectura Arduino

Arduino es una plataforma de electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activar un motor, encender un LED y publicar algo en línea (Arduino, 2018).

Arduino simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas sobre otros sistemas:

1. Las placas Arduino son relativamente económicas comparadas con otras plataformas microcontroladoras. Algunas versiones de Arduino pueden ser ensambladas sin requerir equipos especiales, e incluso los módulos de Arduino preensamblados suelen ser económicos.
2. El software de Arduino se ejecuta en sistemas operativos Windows, Mac OS X y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
3. El entorno de programación de Arduino es fácil de usar, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo.
4. El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, para entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.

5. Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168. Los planos para los módulos están publicados bajo licencia, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo (Enríquez Herrador, 2009).

Existen otros componentes que pueden realizar las acciones que se requieren para este proyecto pero se eligió la tarjeta de adquisición Arduino, ya que contiene todos sus componentes necesarios para ser programada, sin embargo el PIC es más económico pero requiere de componentes adicionales (PICKit, cristal) para su programación y funcionamiento, en comparación con Raspberry Pi esta es una computadora funcional y mucho más cara que Arduino, para este proyecto los dos pueden realizar el trabajo pero por la economía se ha elegido esta tarjeta de adquisición.

La tarjeta de desarrollo Arduino UNO se muestra en Figura 4.4. , es una placa de microcontrolador basada en ATmega328P. Tiene 14 terminales de entrada / salida digital, de los cuales 6 se pueden usar como salidas PWM (modulación de ancho de pulso), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, 6 pines que componen el ICSP (Programación serial en circuito) y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador (Arduino, 2018).



Figura 4.4. Tarjeta de adquisición de datos Arduino UNO Fuente: (Arduino, 2018)

El microcontrolador en la placa Arduino y Genuino AVR tiene 1 kB de memoria EEPROM. En este tipo de memoria, los datos se guardan y no se pierden cuando la placa se apaga (como un disco duro pequeño). Para lograr esto Arduino incluye la librería EEPROM que viene junto con Arduino IDE. Consta básicamente de dos métodos:

read()

write()

El método read() se lee los valores que se vayan guardando en determinada posición de memoria mientras que con el método write() se guarda la información en las posiciones de memoria. (García González, 2013).

#### **4.6 Protocolo de comunicación inalámbrica ZigBee**

ZigBee se apoya sobre IEEE 802.15.4, para la comunicación entre dispositivos cercanos, es decir, el acceso al medio y el intercambio de mensajes por este. Define un nivel de *routing* que permite que los dispositivos con funcionalidades de *router* puedan transportar mensajes a otro destinatario extendiendo el alcance total de la red, a su vez, existen (*end-point*) que permiten comunicación de diversas aplicaciones mediante un identificador de circuito (Sergio R, 2009).

ZigBee es un estándar que define un conjunto de protocolos de comunicación de baja velocidad de datos, los dispositivos inalámbricos basados en ZigBee operan a una frecuencia de 868 MHz, 915 MHz y 2.4 GHz, la velocidad de datos máxima es de 250 Kbps. Esta tecnología está dirigida principalmente a aplicaciones que requiera bajo consumo de energía y bajo costo (Farahani, 2008).

La red ZigBee ha sido utilizada para la medición de variables físicas tales como temperatura, humedad, gases entre otros. La implementación de comunicación inalámbrica ha solucionado muchas necesidades tecnológicas (Álvarez Gutiérrez, 2016).

Los módulos XBee son dispositivos para la comunicación en redes ZigBee, tienen soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan el protocolo de red llamado IEEE

802.15.4 para crear redes (punto a multipunto), o para redes (punto a punto). Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible. (XBee.cl, 2018)

El módulo XBee está compuesto con los siguientes pines como se muestra en la Figura 4.5. .

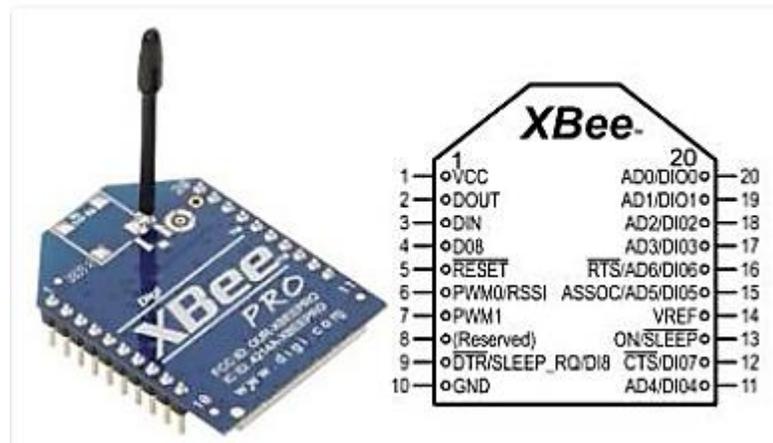


Figura 4.5. Módulo XBee Fuente: (XBee.cl, 2018)

Estos dispositivos suelen a tener diferentes tipos de antenas como se muestran en la Figura 4.6, que se mencionan continuación.

1. Antena Chip, que permite la recepción y envío de datos con una frecuencia de 2.4 GHz.
2. Antena de látigo es un pequeño cable que sobresale del módulo.
3. Antena U. FL cuenta con un conector pequeño para conectar tu propia antena.
4. Antena RPSMA cuenta con un conector más grande para conectar tu propia antena y de esta manera poder tener una mejor recepción y envío de datos.
5. Esquema básico de conexión y funcionamiento del módulo de comunicación inalámbrica XBee. Este sistema no debe superar los 3.3 V, de lo contrario se puede dañar (Álvarez Gutiérrez, 2016).

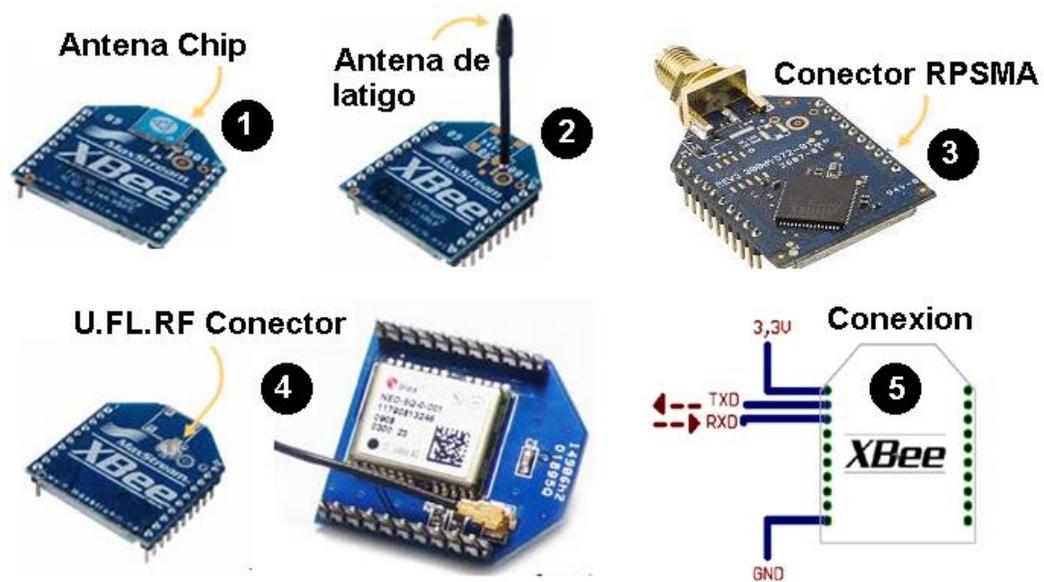


Figura 4.6. Tipos de antena del módulo XBee

En redes ZigBee se pueden configurar los dispositivos en tres tipos de nodos: *Coordinator*, *Router* y *End-devices* (coordinador, ruteador y dispositivos terminales).

Para el nodo coordinador de una red, es un *router* especial, que además de las funciones propias de un *router* tiene a su cargo iniciar y mantener una red ZigBee, el coordinador mide la energía en cada uno de los canales y elige el más silencioso. (Sergio R, 2009)

Por otro lado, el nodo *router* actúa como *buffer* para otros, y pueden ser coordinadores de un pequeño grupo. Los *end-devices* siempre entregan los mensajes a su coordinador (sea un *router* o el Coordinador), quien a su vez almacenan los mensajes hasta que estos despierten y lo contacten para la entrega. Los *routers* son los encargados de derivar las tramas hacia otros *routers* para que puedan llegar a su destino final (Sergio R, 2009).

Los dispositivos que tienen una funcionalidad reducida son los que están en nodo *End-devices*, los cuales tienen permitido dormir periódicamente. Para esto, se asocian a un *router*, que cumple la funcionalidad de un coordinador en una red al cual reportan periódicamente. Estos dispositivos carecen de funcionalidades de *routing* y siempre entregan sus mensajes a su coordinador. (Sergio R, 2009)

ZigBee permite tener interconectados los módulos en diferentes topologías, una de ellas es estrella, permite la comunicación entre los nodos y el controlador central que hemos llamado (PAN coordinador) todas las redes en estrella funcionan independientemente, gracias a que cada una usa un identificador (PAN id) distinto. (Maestre Torreblanca, 2015)

En topología malla los *routers* y coordinadores son los encargados de identificar la mejor ruta de los mensajes enviados, en caso de no tener comunicación, se entregan los mensajes al *router* más cercano quien a su vez repetirá dicha tarea sucesivamente, hasta que el mensaje llegue al destinatario, los dispositivos configurados como *end-devices* siempre entregan los mensajes a su padre, quien a su vez, si estos duermen periódicamente, almacenan los mensajes hasta que despierten, lo contacten y realicen su entrega. (Sergio R, 2009)

Para la topología en árbol los dispositivos se conectan con su padre que a su vez tienen descendientes, la configuración tiene ventaja de permitir llegar a cualquier destino mediante el pasaje sucesivo de mensajes padre-hijo/hijo-padre. El coordinador establece parámetros de asignación, y cada *router* tiene una capacidad permitida de hijos, de los cuales un número determinado de *routers* y el resto deben de ser *end-devices*. (Sergio R, 2009)

La tecnología ZigBee es muy eficiente permitiendo facilidad en la comunicación entre los dispositivos proporcionando las siguientes ventajas.

1. Ideal para conexiones punto a punto y punto a multipunto
2. Diseñado para el direccionamiento de información y el refrescamiento de la red.
3. Opera en la banda libre a 2.4 GHz para conexiones inalámbricas.
4. Óptimo para redes de baja tasa de transferencia de datos.
5. Reduce tiempos de espera en el envío y recepción de paquetes de datos.
6. Bajo ciclo de trabajo.
7. Proporciona larga duración de la batería.
8. Soporte para múltiples topologías de red: Estática, dinámica, estrella y malla.
9. Soporta hasta 65.000 nodos en una red.
10. Son económicos en comparación con otras redes inalámbricas semejantes de la familia (EcuRed, 2011)

La comunicación entre XBee y el PC se realiza mediante el USB Explorer, esta tarjeta permite conectar los dispositivos mediante un puerto USB. Permitiendo configurar cada uno de los parámetros de comunicación de los módulos XBee, como son: velocidad de transmisión de datos, paridad, direcciones, entre otros (Álvarez Gutiérrez, 2016).

Para la comunicación entre dos módulos XBee, uno trabaja como emisor y otro de receptor independiente de la comunicación bidireccional. En la comunicación es necesario saber cuál de los dos dispositivos es el encargado de establecer la comunicación con el PC y cuál es el encargado de recibir la información. Por ejemplo, realizar lecturas de sensores o transmitir una señal de activación a un relé o a una electroválvula. (Álvarez Gutiérrez, 2016)

La comunicación entre el módulo XBee y la tarjeta de adquisición de datos Arduino, se realiza a través de los pines que comparten estos dos (Rx y Tx). Esta interacción permite el envío y la recepción de caracteres o datos que cumplirán con una función específica de acuerdo con cada caso. (Álvarez Gutiérrez, 2016)

Estos dispositivos operan en diferentes modos según sea su configuración como se muestra en la Figura 4.8.

En el modo Recibir/Transmitir la información transmitida puede ser Directa o Indirecta, en el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita (Oyarce, 2008).

En modo de sueño, el módulo RF (Radio frecuencia) entra en un modo de bajo consumo de energía, la configuración de los ciclos de sueño se realiza principalmente con el comando SM. Para poder entrar en modo de sueño, se debe cumplir una de las siguientes condiciones:

1. Sleep\_RQ (pin 9) está en alto y el módulo está en pin modo de reposo (SM= 1,2 o 5)
2. El módulo está en reposo (no hay transmisión ni recepción de datos) por la cantidad de tiempo definido por ST (Tiempo antes de dormir). [ST sólo está activado cuando SM=4,5] (Oyarce, 2008).

En el modo comando se permite ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la del destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es necesario utilizar el Hyperterminal de Windows, el programa X-CTU o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma. (Oyarce, 2008)

Para el modo transparente todo lo ingresa por el pin 3 (entrada), es guardado en el buffer de entrada y luego transmitido y todo lo que ingresa como paquete RF, es guardado en el buffer de salida y luego enviado por el pin 2 (salida). El modo Transparente viene por defecto en los módulos XBee. Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control. También se usa para reemplazar alguna conexión serial por cable, ya que es la configuración más sencilla posible y no requiere una mayor configuración. (Oyarce, 2008)

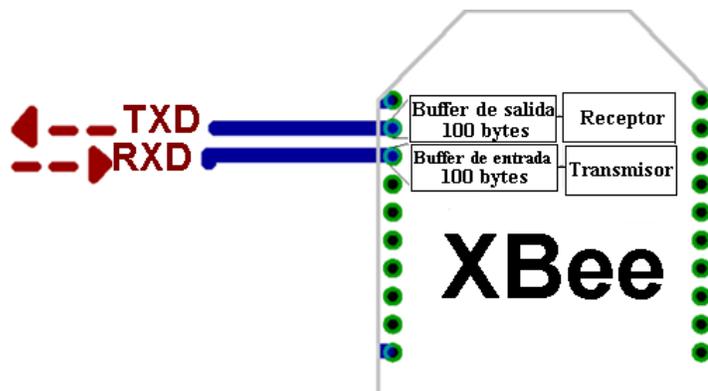


Figura 4.7. Correspondencia del Buffer Fuente: (Oyarce, 2008)

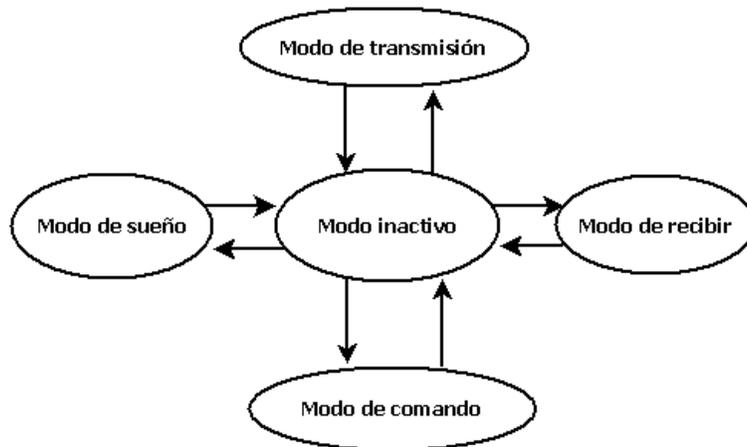
En modo de operación API suele ser el más complejo, pero permite el uso de *frames* con cabeceras que aseguran la entrega de los datos, al estilo TCP. Extiende el nivel en el cual la aplicación del cliente puede interactuar con las capacidades de red del módulo. Cuando el módulo XBee se encuentra en este modo, toda la información que entra y sale, es empaquetada en *frames*, que definen operaciones y eventos dentro del módulo. (Oyarce, 2008)

Así, un *frames* de transmisión de Información (información recibida por el pin 3 o DIN) incluye:

1. *Frame* de información RF transmitida.
2. *Frame* de comandos (equivalente a comandos AT).

Mientras que un *Frame* de Recepción de Información incluye:

1. *Frame* de información RF recibida.
2. Comando de respuesta.
3. Notificaciones de eventos como Restablecer, Asociar, Desasociar, etc.



• *Figura 4.8. Modo de operación del módulo XBee Fuente: (XBee Comunicaciones inalámbricas, 2014)*

## 4.7 Celdas fotovoltaicas

Una celda fotovoltaica produce electricidad gracias a la luz solar que incide sobre ellos. Es decir, absorbe los fotones de la luz para liberar electrones que se pueden usar como corriente eléctrica. Los paneles solares no son más que varias celdas trabajando de forma conjunta para generar un mayor potencial eléctrico. Algunos paneles cuentan con 36 o más celdas (Iluminet, 2016).

Las celdas fotovoltaicas pueden generar energía para ser utilizada en lugares donde no se cuenta con acceso a la red de energía eléctrica. Además de esto, otro de los objetivos del

uso de las celdas solares es el ahorro de energía eléctrica utilizando recursos naturales (Panel solar, 2018).

Las celdas fotovoltaicas han demostrado ser útiles debido a su capacidad para generar energía eléctrica a partir de la luz solar, por lo que se han implementado en varios sistemas. Por ejemplo, se utilizaron en un sistema que realiza el control de lámparas de led de alta potencia para el alumbrado público. Este sistema funciona con una batería de gel que se carga con un panel solar logrando el funcionamiento en su punto de máxima potencia mediante un algoritmo MPPT (Seguidor de Punto de Máxima Potencia), y el voltaje aceptable de la batería y el control de corriente para hacer que la carga y descarga apropiadas no reduzcan su vida útil (A. Bauernfeind, et al., 2014).

## **5 METODOLOGÍA**

En este apartado se describe la metodología utilizada para el desarrollo de la investigación, la secuencia de procesos que se realizaron para la elaboración del sistema, la experimentación para verificar el funcionamiento mediante un modelo funcional, así como requerimientos que se establecieron para su desarrollo.

### **5.1 Requerimientos de diseño**

El software que se elige para la elaboración del sistema es Arduino IDE ya que es gratuito y de código abierto, junto con la herramienta App Inventor que permite realizar aplicaciones móviles Android, y se utilizó para la creación de la interfaz gráfica; su diseño ayuda al administrador a verificar el estado de los tanques y las válvulas, así como su llenado de forma automática y manual.

Para la visualización de datos y activaciones del sistema se ocupó un dispositivo móvil con las siguientes características.

1. Sistema operativo Android 4.3.
2. Procesador QuadCore a 1.2GHz.
3. Memoria interna 4GB.
4. Memoria RAM 1GB.
5. Conexiones inalámbricas USB, Wifi, Bluetooth.

El sistema real se plantea usar módulos RF Digi XBee-PRO S2C (Digi, 2018), que tienen un alcance de 3200m para poder enviar y recibir información (niveles de agua, fugas, información de las válvulas), permitiendo llenar dos tanques de 172,800 L cada uno, que a su vez están alejados de la bomba de abastecimiento con distancias aproximadas de 1.55 km y 2.03 km. Para el modelo funcional se hace uso de módulos Xbee con alcance de 100m.

La implementación del protocolo ZigBee ofrece la opción de tener interconectados los módulos XBee, ya que cuenta con diferentes topologías de configuración y permite que se alcancen distancias más extensas para el envío de información.

## 5.2 Diseño e implementación de la circuitería y lógica

El desarrollo de la investigación se efectuó a partir del diagrama de flujo de la Figura 5.1, donde se presentan los pasos generales para la elección de materiales y procedimiento para el funcionamiento del modelo funcional.

Para el control de los niveles de líquido del sistema se necesitan monitorear las diferentes variables detalladas en la Tabla 5.1, donde también se especifican los elementos a utilizar.

*Tabla 5.1. Sensores y Actuadores.*

Sensores		
Variable a medir	Sensor	Tipo de señal
Nivel del tanque 1	Sensor ultrasónico HC-SR04	Digital
Nivel del tanque 2	Sensor ultrasónico HC-SR04	Digital
Actuadores		
Dispositivo	Actuador	Tipo de señal
Bomba	Bomba	Analógica
Válvula de llenado Tanque 1	Válvula	Analógica
Válvula de llenado Tanque 1	Válvula	Analógica
Válvula de Tandeo 1	Válvula	Analógica
Válvula de Tandeo 2	Válvula	Analógica

La programación de Arduino y el sensor ultrasónico (HC-SR04), permiten la adquisición de la distancia entre el sensor y el agua, la implementación del sistema real se recomienda el uso del sensor JSN-SR04T, ya que es resistente al agua y lo hace ideal para utilizarse en proyectos de exteriores; dicho sensor se conecta en el pin d7 donde saldrán los pulsos y el d6 recibirá la información de la distancia (Figura 5.2). Cada tanque cuenta con un sensor en la parte superior, permitiendo obtener los niveles del agua, dicha conexión entre estos dispositivos se muestra en la Figura 5.5.

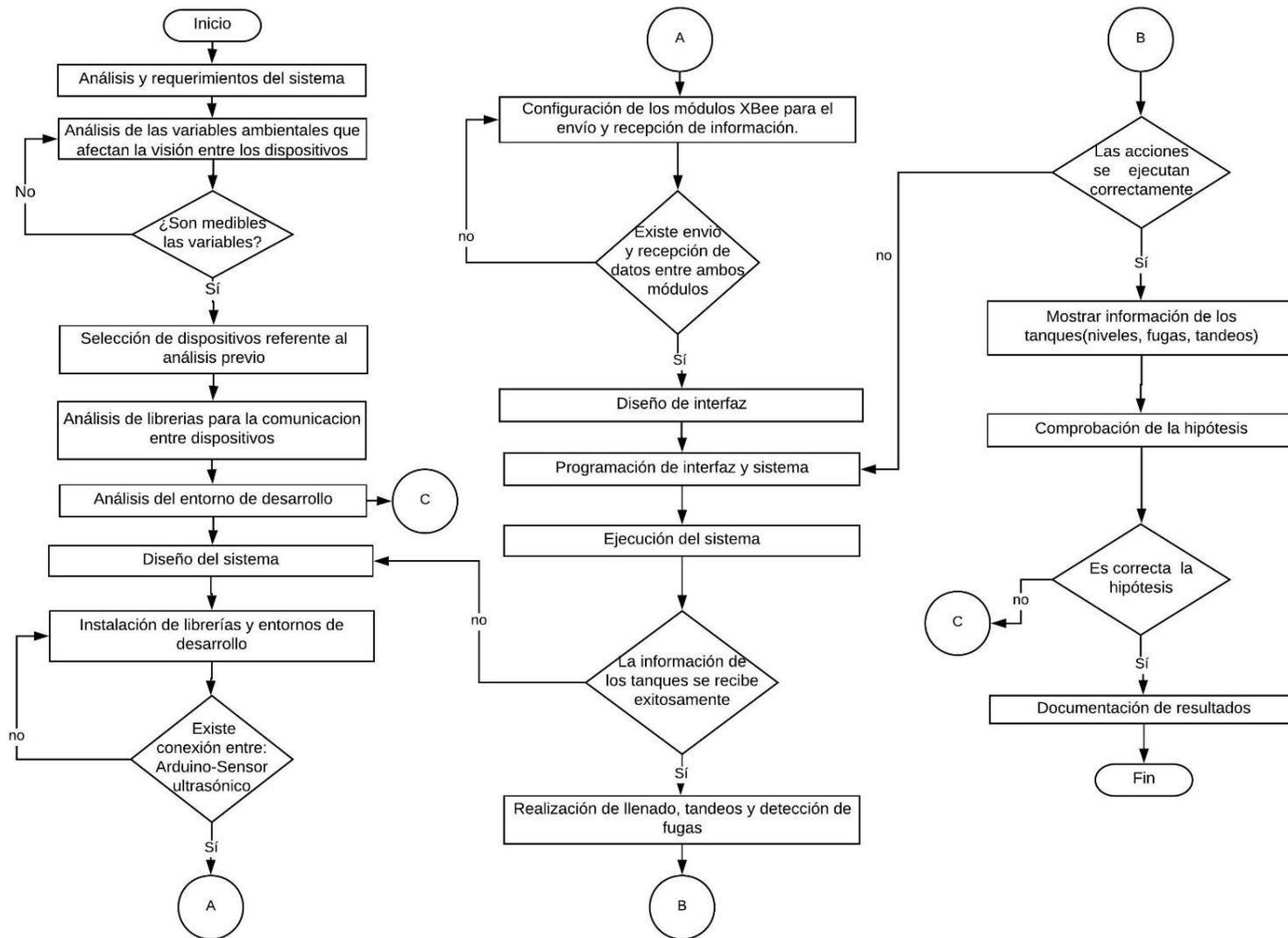


Figura 5.1. Diagrama de flujo de la Metodología de investigación.

```

#define trigPin 7 //se define el nombre del pin 7
#define echoPin 6 //se define el nombre del pin 6
long tiempo;
long distancia;
digitalWrite (trigPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
delayMicroseconds (4);
digitalWrite (trigPin, HIGH); // genera el pulso de trigger por 10ms
delayMicroseconds (10);
digitalWrite (trigPin, LOW);
tiempo = pulseIn(echoPin, HIGH); //Medimos el tiempo entre pulsos, en microsegundos
distancia = tiempo*0.01715; // Distancia entre el sensor con el nivel del agua.
}

```

*Figura 5.2. Obtención de distancias mediante el sensor.*

La tarjeta de adquisición Arduino permite obtener los valores obtenidos del sensor ultrasónico para esto se deben de declarar los pines de entrada y salida de señal, estos pines mandan la información de los pulsos que genera el sensor, con la ayuda de variables se obtienen los pulsos en microsegundos para posteriormente realizar la conversión de distancias a porcentajes donde se toman 3cm de distancia para tener un 100% y 33 cm de distancia para tener un 0% como se muestra en la Figura 5.3

```

long operacion1;//Variable para obtener la distancia en porcentaje
long operacion2;//Variable para reducir el porcentaje según el 100%
operacion1=((distancia-3) *(10/3)) //distancia menos el nivel máximo por la división de
100/30 |
operacion2= (100) -(operacion1) //resta del 100% con el valor obtenido de la operación 1

```

*Figura 5.3 Conversión a porcentaje.*

La conversión a porcentajes permite al administrador entender y visualizar los niveles del tanque como se muestra en la Figura 5.4.

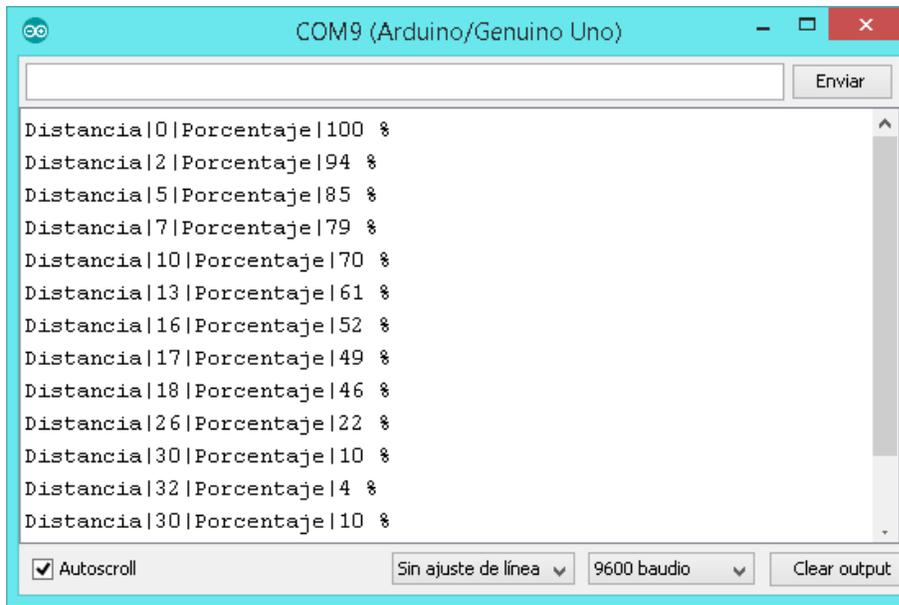


Figura 5.4. Visualización de porcentajes

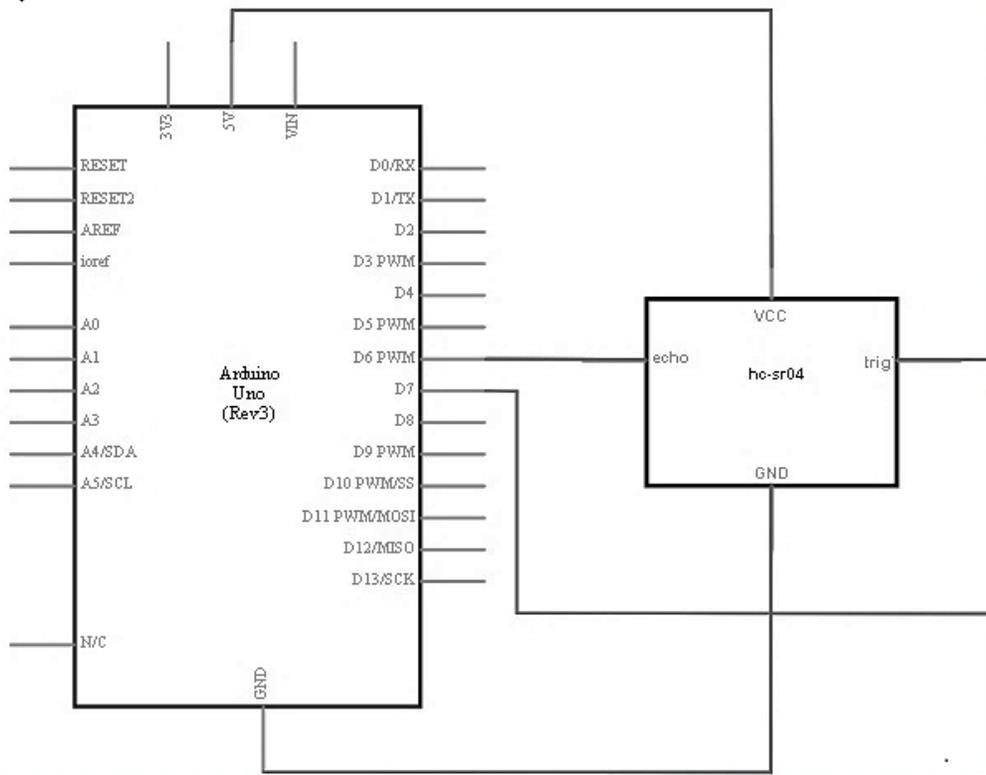


Figura 5.5. Conexión sensor (HC-SR04) con Arduino.

El sistema en modo automático permite activar la bomba cuando los niveles de los tanques son menor al 100% (Figura 5.6) reduciendo el desperdicio de agua por desbordamiento y ahorro de energía ya que a estos depósitos actualmente se les bombea agua 10 minutos de más, generando una pérdida de 4800 L por cada suministro; estos depósitos solo necesitan 6 horas para llenarlos si están completamente vacíos sabiendo que la bomba suministra 8 L por segundo (Figura 5.7), aumentando la vida útil de la bomba de abastecimiento dicha programación de Arduino que permitirá realizar dichos llenados según los niveles recibidos se muestra en la Figura 5.8.

```

void llenados () { //función de los llenados
lecturasensor (); //se obtienen las distancias del sensor con el nivel del agua
if (Serial.available()) { //Si el puerto serie está disponible
valor = Serial.read(); //se leen los datos de entrada mediante el serial
if (valor=="H"){ //si la entrada en el serial es H se cancela el modo automático
asm ("jmp 0x0000"); //reset
}}
if(distancia >3 && distancia <=33){ //El suministro de agua se realiza mientras la
distancia sea mayor a 3 cm y menor o igual a 33 cm.
Serial.print ("1"); //Se manda mediante el serial el número del tanque en el que se están
realizando las acciones.
Serial.print ("");
Serial.print ("P1"); //Se manda la letra P mediante el serial que identifica que la bomba
debe de prenderse.

```

Figura 5.6. Especificación de estado de la bomba según los niveles del tanque.



Figura 5.7. Suministro de agua

```

if (XbeeSerial.available()) { //si hay datos por el serial 2,3 Modulo XBee
incomingByte = XbeeSerial.read(); //guardamos los datos en la variable
Serial.print((char)incomingByte); //Se reciben los datos del módulo XBee
if (incomingByte == 'P1') { //si el valor entrante es P1 enciende la bomba
digitalWrite(valvula2, LOW); //válvula 2 se sierra
digitalWrite(bomba, LOW); //bomba encendida
}
if (incomingByte == 'L') { //si el valor entrante es L la bomba se apaga
digitalWrite (bomba, HIGH); //bomba apagada
digitalWrite (valvula2, HIGH); //Se abre la válvula
}
}

```

Figura 5.8. Programación de encendido de bomba según los niveles.

Con Arduino y el sensor ultrasónico se detectan fugas, tomando muestras en tiempos específicos cuando el agua este en completo reposo, para el sistema real se tomarán de 8 a 10 muestras durante 12 horas; esto para poder detectar fugas grandes y poder avisar al encargado lo más pronto posible, el proceso se lleva a cabo como se muestra en el diagrama de flujo de la Figura 5.9.

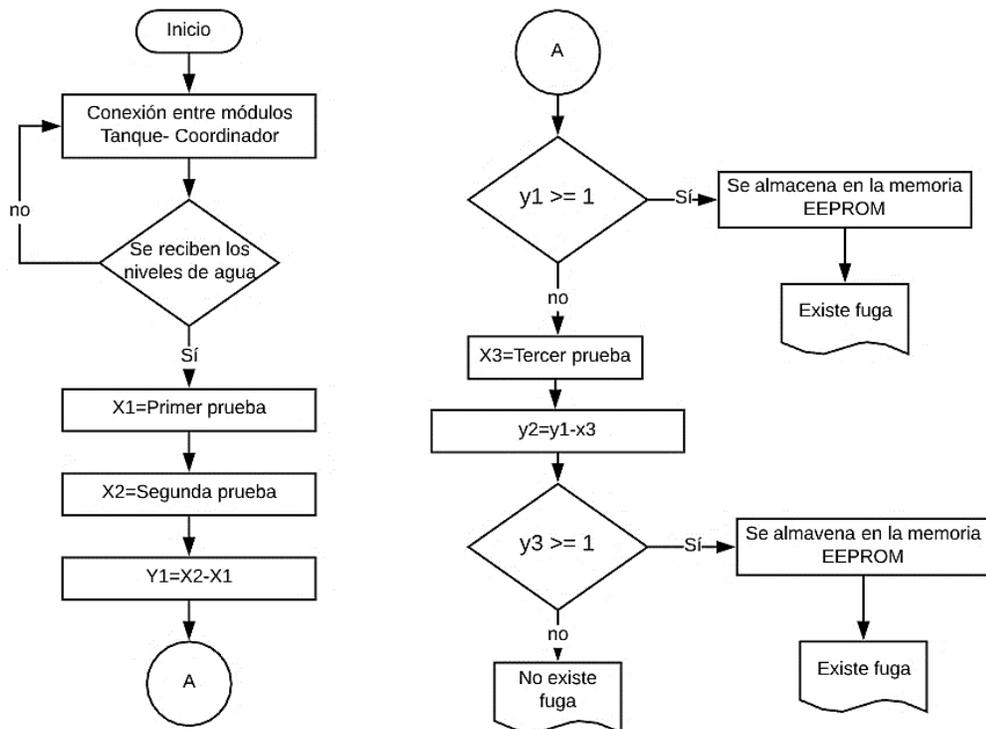
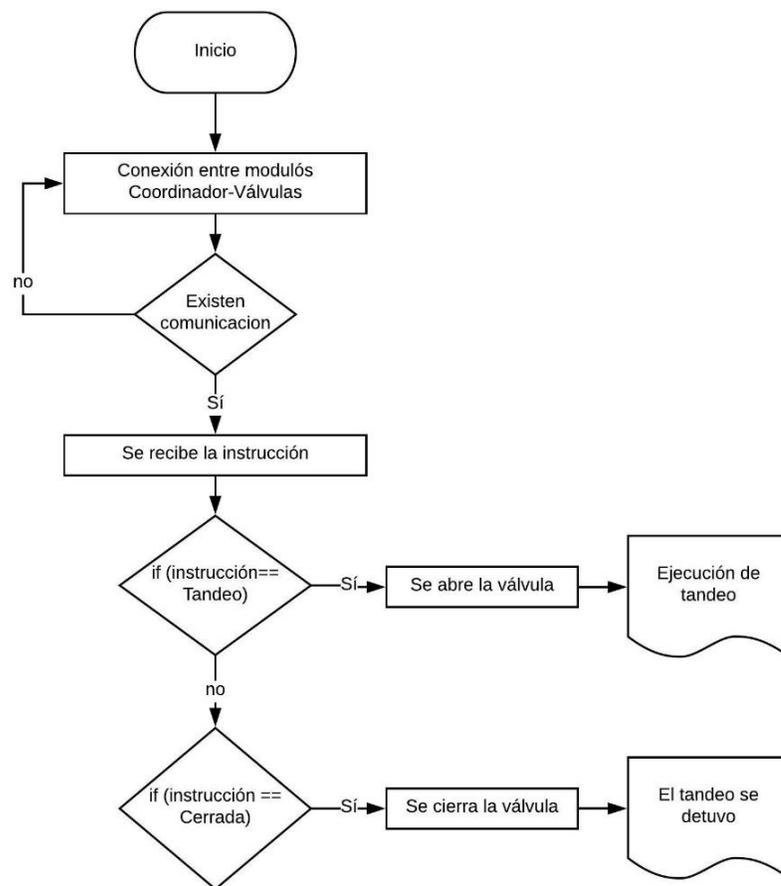


Figura 5.9. Proceso detección de fugas.

La distribución de agua se llevará a cabo mediante válvulas monitorizadas que se abrirán automática o manualmente dependiendo la acción que ordene el administrador, efectuándose durante dos horas; siendo tiempo suficiente para satisfacer a 800 habitantes. Hoy en día el tandeo se realiza abriendo la válvula esperando a que el tanque este vacío, la distribución actual de agua se realiza durante 3:00 horas aproximadamente. Estas válvulas funcionan como se muestra en el diagrama de flujo de la Figura 5.10.

El modelo funcional simula la abertura de las válvulas con leds, pero para su implementación real se ocuparán, motores a pasos, con un torque de 14 kg/cm; estos motores estarán engranados para poder realizar las aberturas y cierres de manera eficaz ya que deberán abrir válvulas de 4 pulgadas controlado con Arduino y un controlador CNC Nema Tb6600.



*Figura 5.10. Diagrama de flujo de las válvulas.*

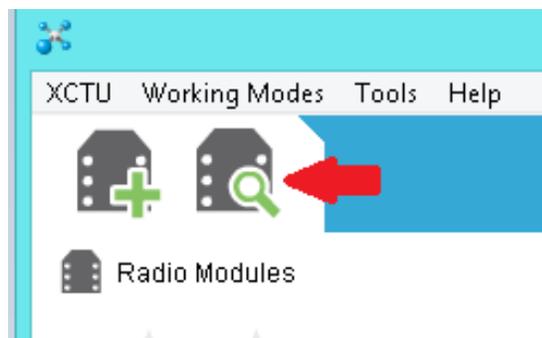
Los niveles de los tanques, así como las acciones que se están realizando ya sea tandeo o detección de fugas, esta información es enviada a la bomba de abastecimiento haciendo uso de los módulos de radiofrecuencia XBee que permiten enviar y recibir la información. La configuración de los módulos XBee en topología malla se efectuó mediante el software XCTU que proporciona Digi Internacional Inc., ya que este es el fabricante de dichos módulos.

Para que la computadora reconozca los módulos y se pueda realizar dichas configuraciones debe de instalarse el controlador DIGI\_USB\_RF\_Drivers, que permite asignar a los XBee un puerto de comunicaciones (COM) mediante el adaptador XBee Explorer y el puerto USB del ordenador como se muestra en la Figura 5.11.



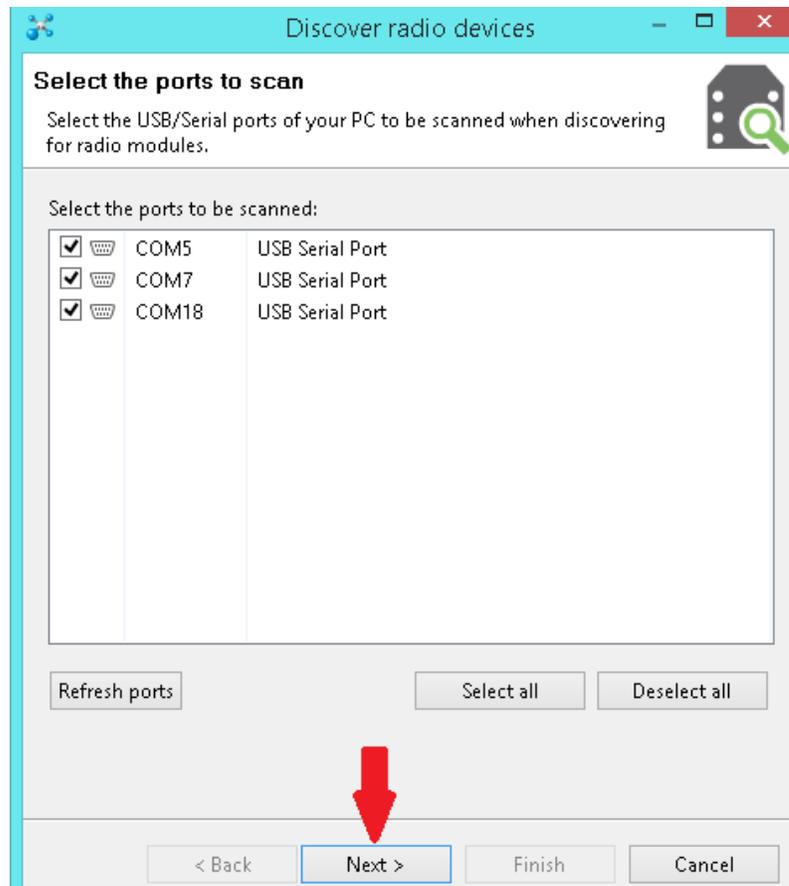
*Figura 5.11. Asignación de puerto (COM).*

Una vez que el ordenador ha identificado el módulo se ejecuta el software XCTU. Se realiza el escaneo de los puertos de comunicación para visualizar los dispositivos disponibles que se pueden configurar, en la barra superior del software en el costado izquierdo se muestra el botón para realizar dicha búsqueda.



*Figura 5.12. Botón de búsqueda de los módulos.*

Después de realizar la búsqueda de los módulos el software nos muestra los dispositivos disponibles en los puertos de comunicación, se procede a seleccionar los dispositivos a configurar, clic en “Next”.



*Figura 5.13. Selección de dispositivos encontrados.*

Al seleccionar los dispositivos se configuran los parámetros como paridad, velocidad en baudios (bits por segundo), número de bits, bits de parada y el control de flujo. Para el caso de los módulos que son de fábrica los parámetros se dejan por defecto como se muestra en la Figura 5.14.

El software muestra los dispositivos conectados como se exhibe en la (Figura 5.15); para este proyecto se utiliza un Coordinador que recibe la información de los tanques y dos *End-Device* los cuales envían información al coordinador el estado y acciones de estos.

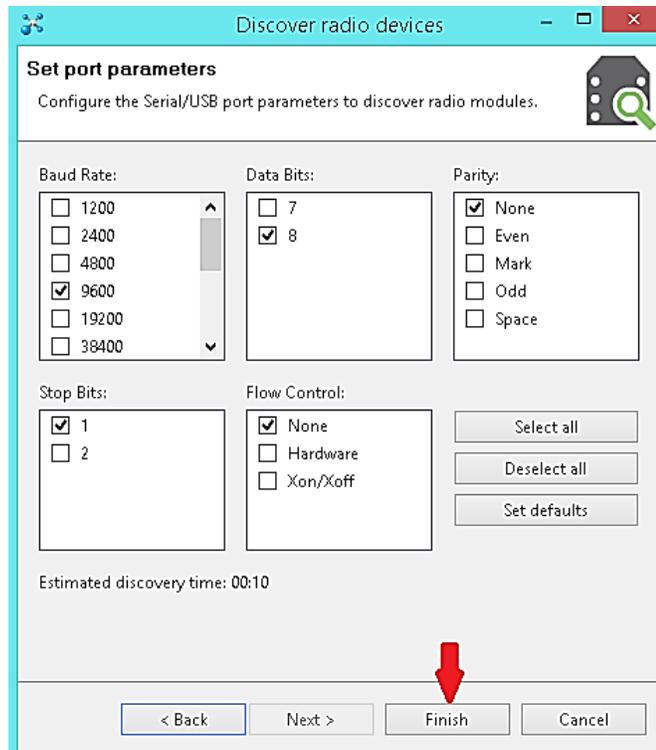


Figura 5.14. Parámetros del XBee.

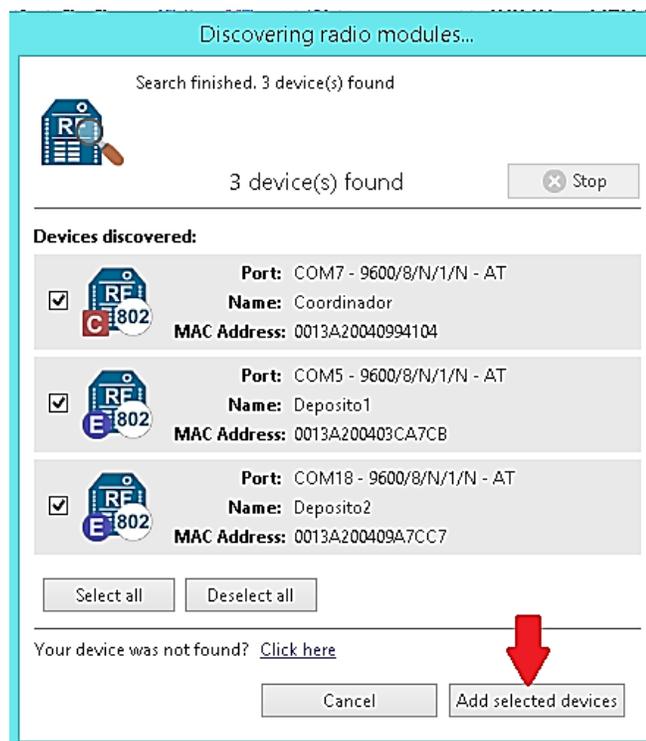


Figura 5.15. Dispositivos encontrados con sus especificaciones.

Estos dispositivos cuentan con el modo de suspensión pin-controlado que permite dormir los módulos (modo de bajo consumo de energía) cuando no estén realizando ningún envío y recepción de datos de lo contrario se despertará para realizar dichas comunicaciones. La configuración para este trabajo es  $SM = 1$ . En este modo, el módulo XBee va a dormir cuando el Sleep\_RQ (pin 9) se afirma en alto conectándolo a 3,3 volts. Para este trabajo se convirtió la salida del Pin usando un divisor de voltaje como se muestra en la Figura 5.16.

Para despertar el módulo es necesario poner (pin 9) en bajo de esta manera el módulo empezará a trabajar. Los módulos XBee nos permiten saber el estado mediante el (pin 13) ya que pasa en nivel alto cuando el módulo está despierto y baja cuando está durmiendo para este trabajo se ha colocado un led para poder observar el estado de dicho modulo.

Arduino se encarga de dormir los módulos ya que sus pines se activan con 5 V se hace uso del pin D8 ya que cuando este pin se encuentre en alto el módulo dormirá y si está en bajo el módulo despertara para realizar la transferencia de la información

Una vez seleccionados los dispositivos se pueden configurar, para que cumplan con la función que se les establezca para el Coordinador se habilita la dirección CE (Coordinador habilitado) esta opción pone al dispositivo como Coordinador. Para que el dispositivo esté configurado en red malla (*mesh*) se configuran las siguientes direcciones ID (PAN ID)=3332, DH (Dirección de destino alta)=0, DL (Dirección de destino baja)=FFFF y BD (Velocidad de datos de la interfaz)=9600[3]; para la configuración modo *sleep* SM(Modo de sueño)=[1] una vez efectuados los cambios, se programan los parámetros en el módulo con el uso del botón de escritura como se muestra en la Figura 5.17.

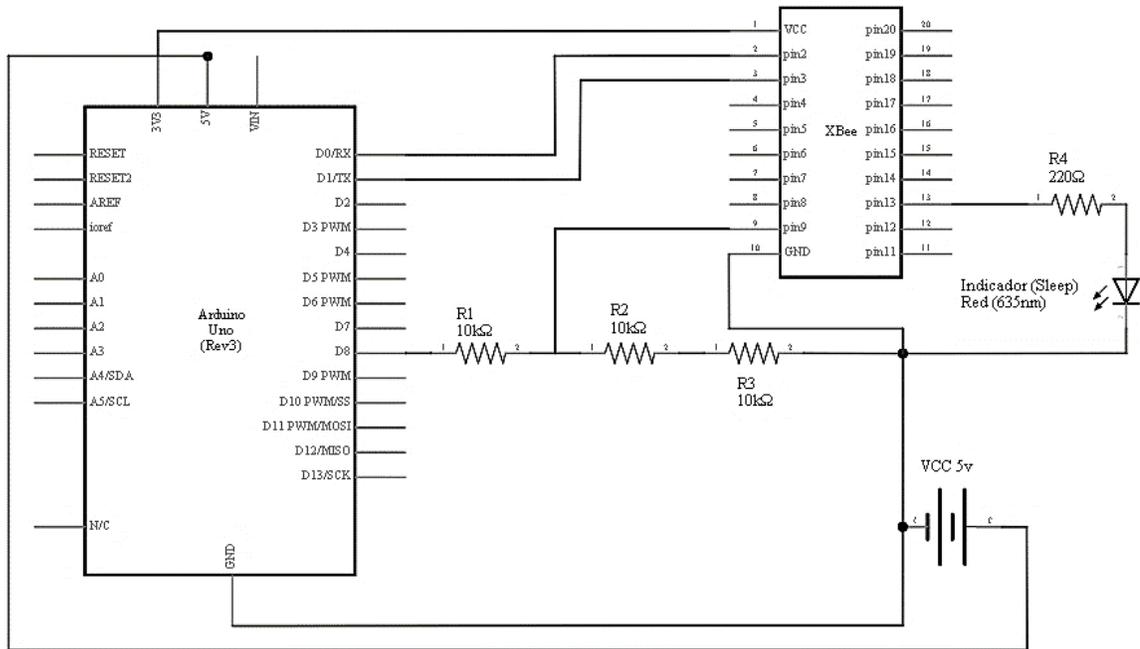


Figura 5.16. Divisor de voltaje 5 V a 3.3 V

Name	Function	Port	MAC
Coordinador	XBEE PRO 802.15.4	COM7 - 9600/8/N/1/N - AT	0013A20040994104
Deposito1	XBEE PRO 802.15.4	COM5 - 9600/8/N/1/N - AT	0013A20040994104
Deposito2	XBEE PRO 802.15.4	COM18 - 9600/8/N/1/N - AT	0013A20040994104

Parameter	Value
ID PAN ID	3332
DH Destination Address High	0
DL Destination Address Low	FFFF
SM Sleep Mode	Pin Hibernate [1]
SH Serial Number High	13A200
SL Serial Number Low	40994104
CE Coordinator Enable	Coordinator [1]
BD Interface Data Rate	9600 [3]

Figura 5.17. Configuración para el módulo Coordinador.

Para la configuración de los módulos *End-Device* se sigue el mismo procedimiento antes realizado, con la diferencia que en estos módulos la dirección CE (Coordinador habilitado) estará configurada con la opción *End-Device* [0].

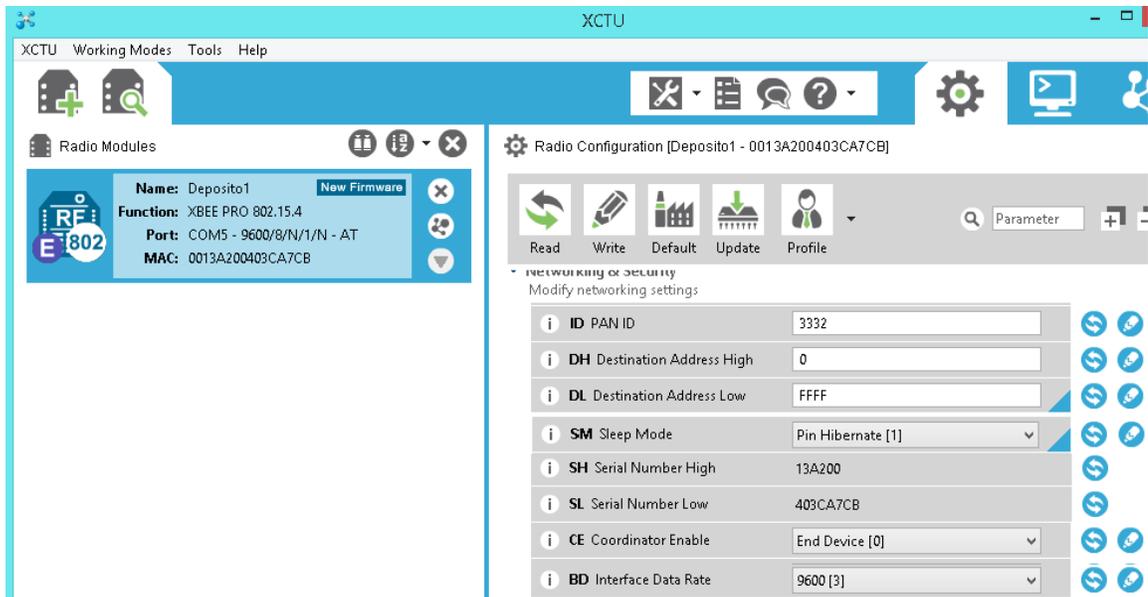


Figura 5.18. Configuración para módulo End-Device.

Realizadas las configuraciones necesarias en los módulos XBee se realizan pruebas de comunicación entre los módulos para verificar que las configuraciones sean correctas.

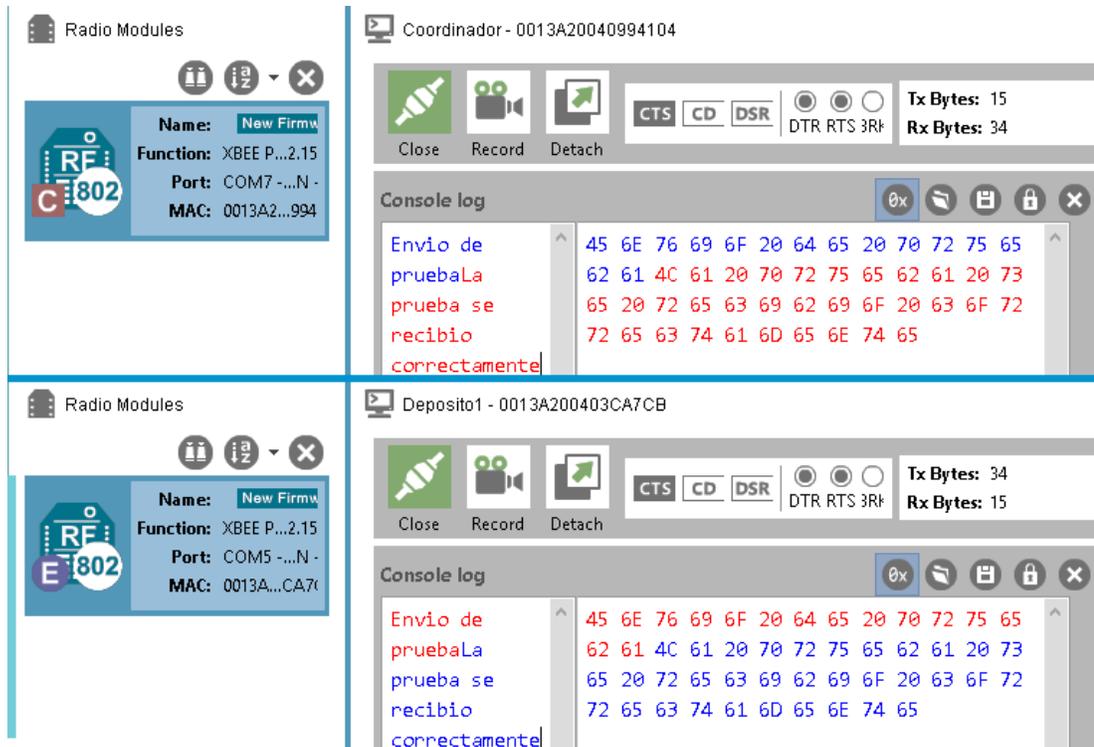


Figura 5.19. Comunicación entre el Coordinador y el Deposito1.

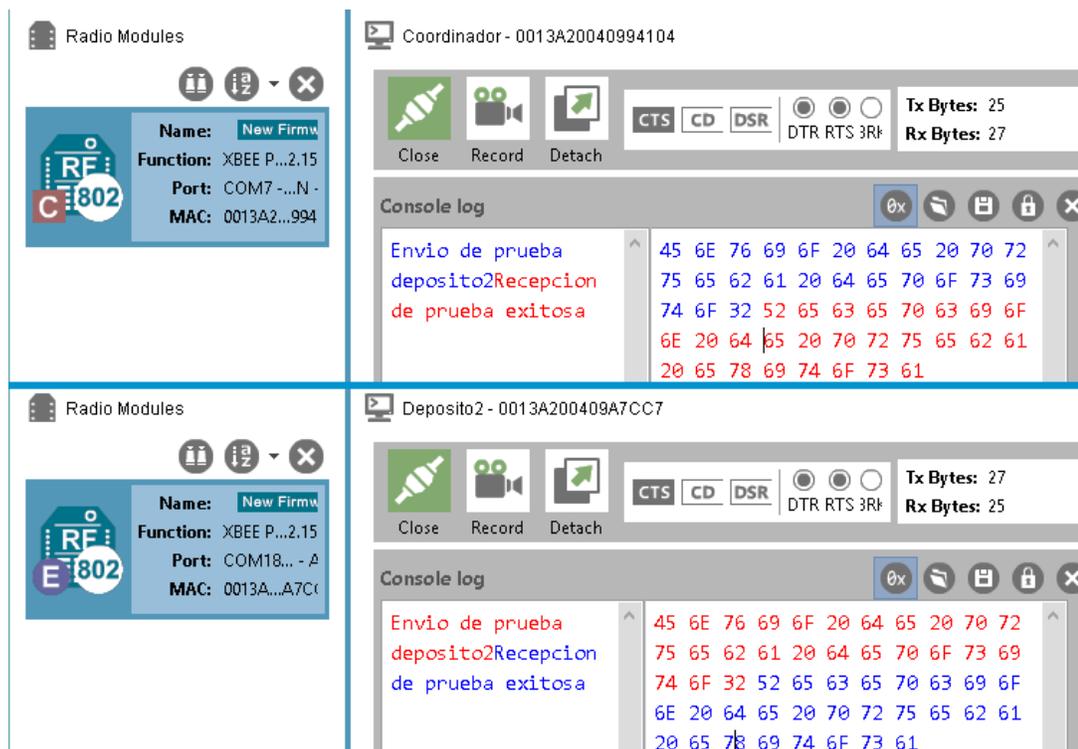


Figura 5.20. Comunicación Coordinador y el Deposito2.

Configurados los módulos, se hace uso de la librería <SoftwareSerial.h>, para que los datos recibidos por el módulo XBee puedan ser obtenidos por Arduino. La comunicación inalámbrica establecida anteriormente del coordinador obtiene información de los tanques en tiempo real. El módulo XBee introduce los datos por el puerto serie declarado en los pines (2,3) del Arduino (Figura 5.21); en estos entran los datos que se reciben de los tanques, permitiendo activar la bomba cuando es requerido y desactivarla cuando los depósitos están a su máxima capacidad. Para este proyecto la activación de la bomba se realiza mediante un Módulo Relevador de 5V que se activa estando en bajo y se desactiva estando en alto; se hace uso del pin 13 de Arduino que permite activar la bomba poniendo el pin en bajo y apagándola en alto, bajo el diagrama de la Figura 5.27, en tanto que el comportamiento de la bomba se muestra en el diagrama de flujo de la Figura 5.22.

```
#include <SoftwareSerial.h> //librería de los seriales
SoftwareSerial XbeeSerial(3, 2); //entrada serial en los pines 2,3
```

Figura 5.21. Declaración de pines para entrada de un nuevo serial.

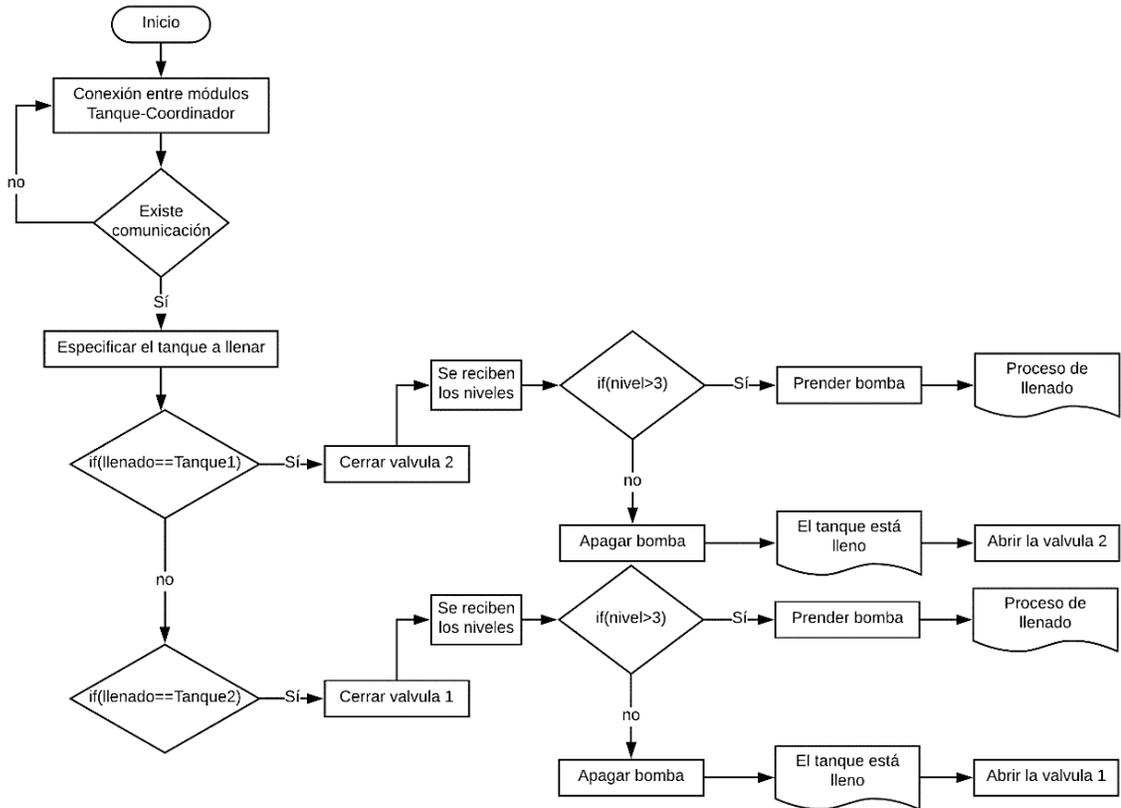


Figura 5.22. Encendido de la bomba según los niveles de agua

A partir del puerto serie que, por defecto, se encuentra habilitado en Arduino (pines 0,1), se envían los datos vía Bluetooth, para que se puedan visualizar en el dispositivo móvil y a su vez se puedan activar las acciones de manejo del sistema (manual o automático), dicha conexión se muestra en la Figura 5.26 y la Figura 5.27. Esto permite al administrador saber los estados de los tanques sin necesidad de estar cerca de ellos, a su vez permitiendo activar el sistema para que trabaje por sí mismo o realizar las activaciones manualmente.

Para visualizar los datos en el dispositivo móvil se realiza la programación para la detección del módulo Bluetooth, que es el medio para recibir la información, como se

presenta en la Figura 5.23, posteriormente realizar la comunicación con el dispositivo que se ha configurado, como se muestra en la Figura 5.24.



Figura 5.23 Detección de dispositivos Bluetooth.

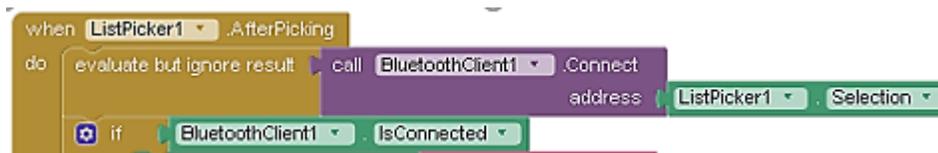


Figura 5.24 Conexión Bluetooth.

Conectado el dispositivo Bluetooth se puede visualizar la información en el dispositivo móvil, permitiendo al usuario realizar acciones con mayor facilidad, la programación para mostrar la información de los tanques se muestra en la Figura 5.25 donde se recibe la información y se acomodan en los “Labels” que se han insertado en la interfaz, mostrando la información adecuadamente de cada tanque según su posición.

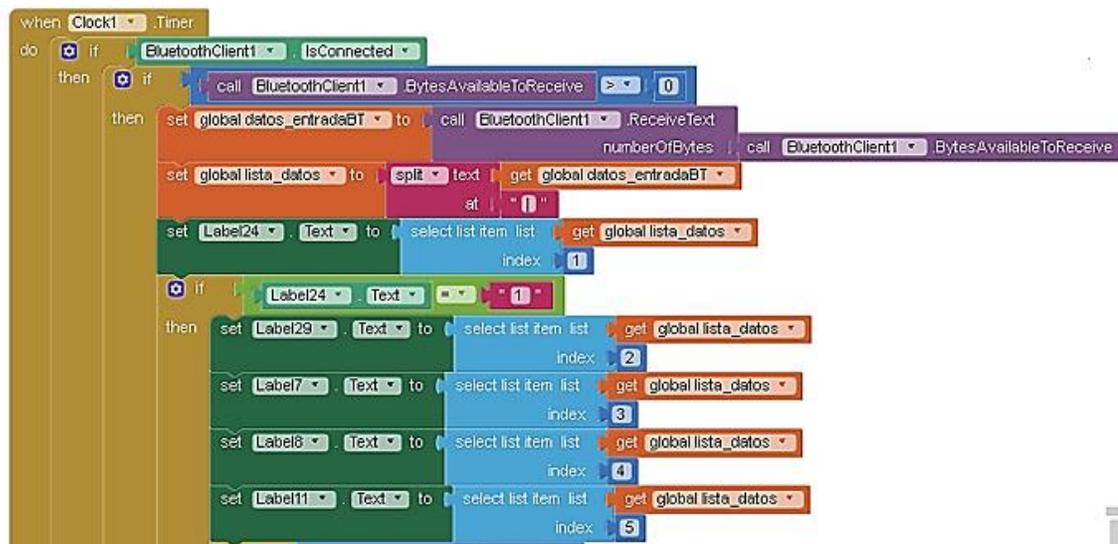


Figura 5.25 Información insertada en los Labels de la interfaz.

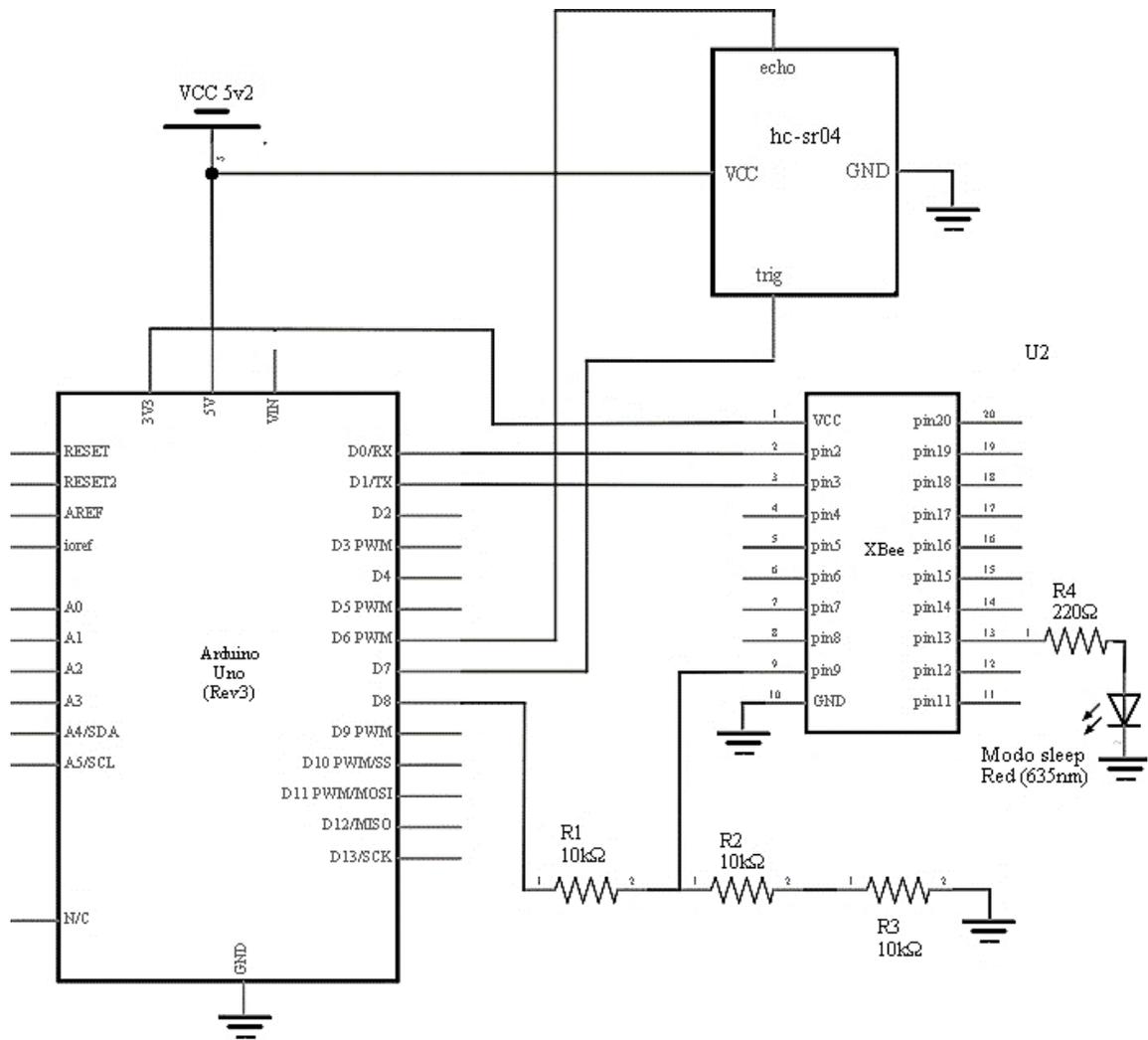


Figura 5.26. Conexión del circuito de los tanques.

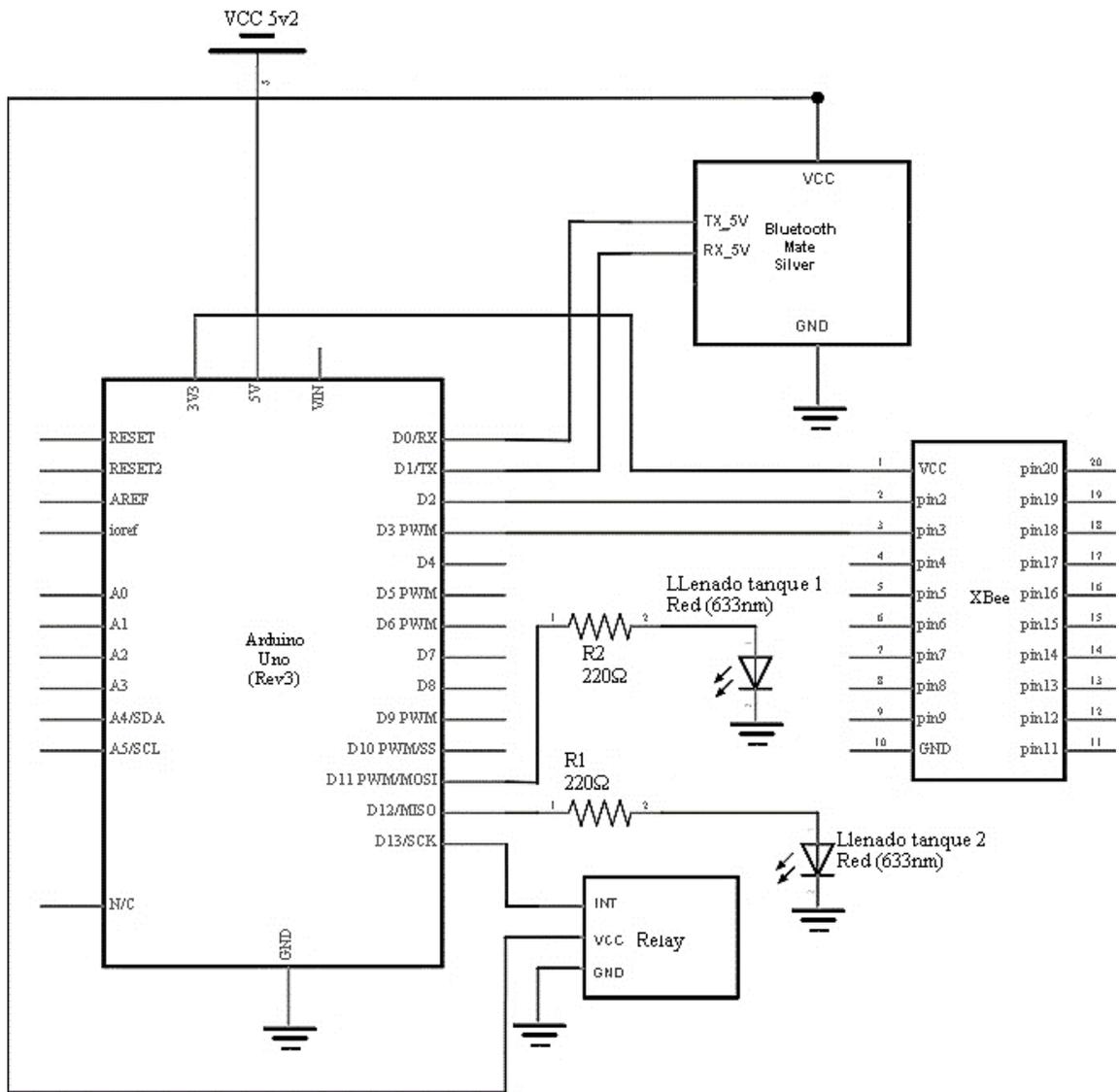


Figura 5.27. Conexión del circuito del Coordinador.

### **5.3 Interfaz**

Para que el usuario pueda comprobar el funcionamiento y la efectividad del sistema, se ha desarrollado una interfaz gráfica de usuario usando el entorno de desarrollo App Inventor, permitiendo al administrador a verificar el estado de los tanques y las válvulas así su cómo su llenado de forma automática y manual. La interfaz se divide en dos partes principales: “Interfaz de conexión” e “Interfaz de inicio”, como se indica en la Figura 5.28.

El control automático permite llenar los tanques, tandeo y detección de fugas; una vez activado este servicio, el administrador no necesariamente debe de estar presente para la verificación de dichos procesos, teniendo en cuenta que debe de regresar para poder desactivar este modo de control.

Para el control manual el administrador debe elegir y controlar el llenado de los tanques uno a la vez de forma presencial, para evitar el desbordamiento del agua. Este procedimiento opera prácticamente bajo la modalidad actual de operación manual, pero con la ventaja de que se puede realizar a distancia, desde la misma aplicación.

En el diagrama flujo de la (Figura 5.29) se muestra el funcionamiento de la interfaz gráfica del sistema móvil.

Para el uso de la interfaz gráfica debe instalarse aplicación en un dispositivo que cuente con el sistema Android y la conexión inalámbrica Bluetooth, esta aplicación solo se proporciona al administrador, la instalación se muestra en la Figura 5.30.

La primera interfaz con la que el usuario interactúa solo se puede realizar la conexión estando activado solo el botón (Conectar) que tiene la función de realizar la conexión del dispositivo con el Bluetooth, para posteriormente obtener los datos de los tanques y poder operar el sistema, como se muestra en la Figura 5.31.

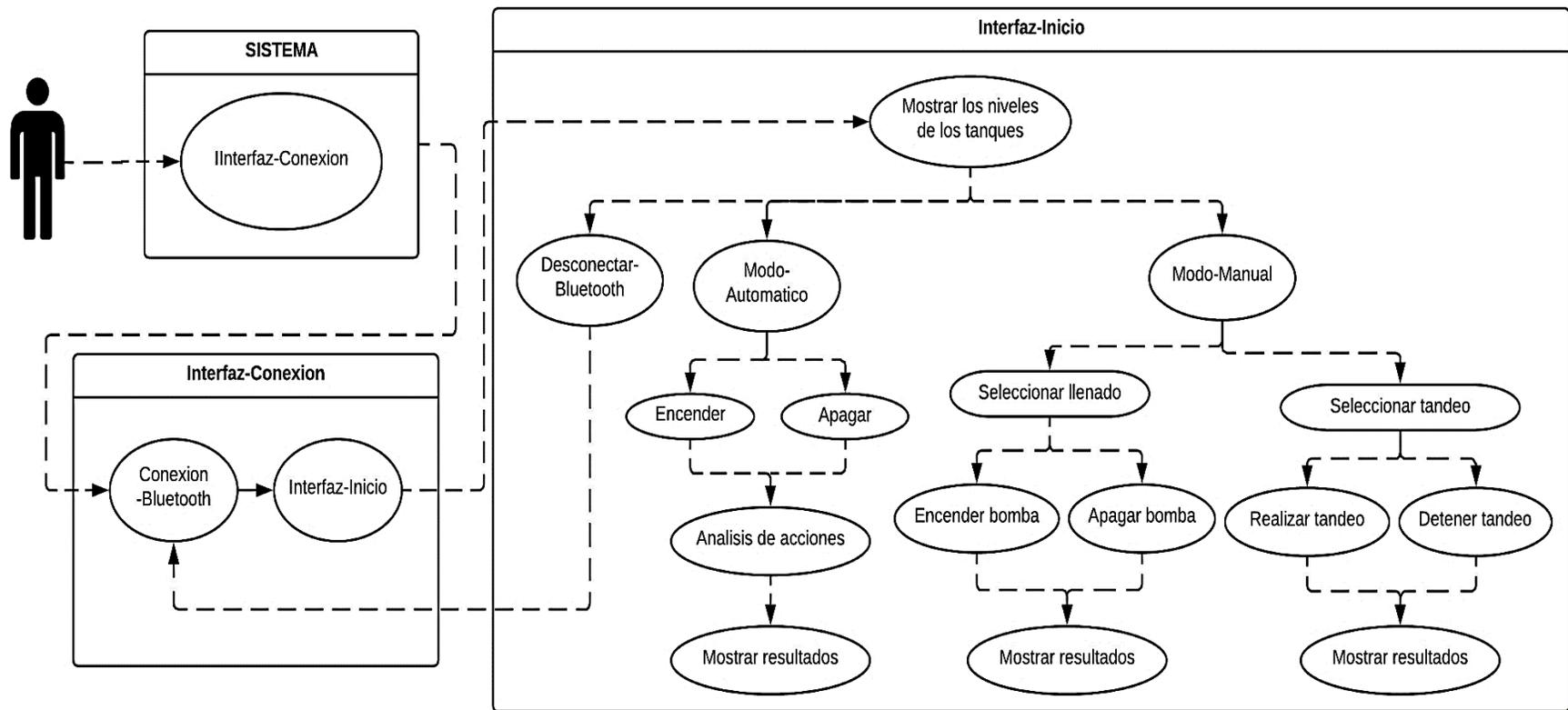


Figura 5.28. Diagrama de interacción del usuario y sistema.

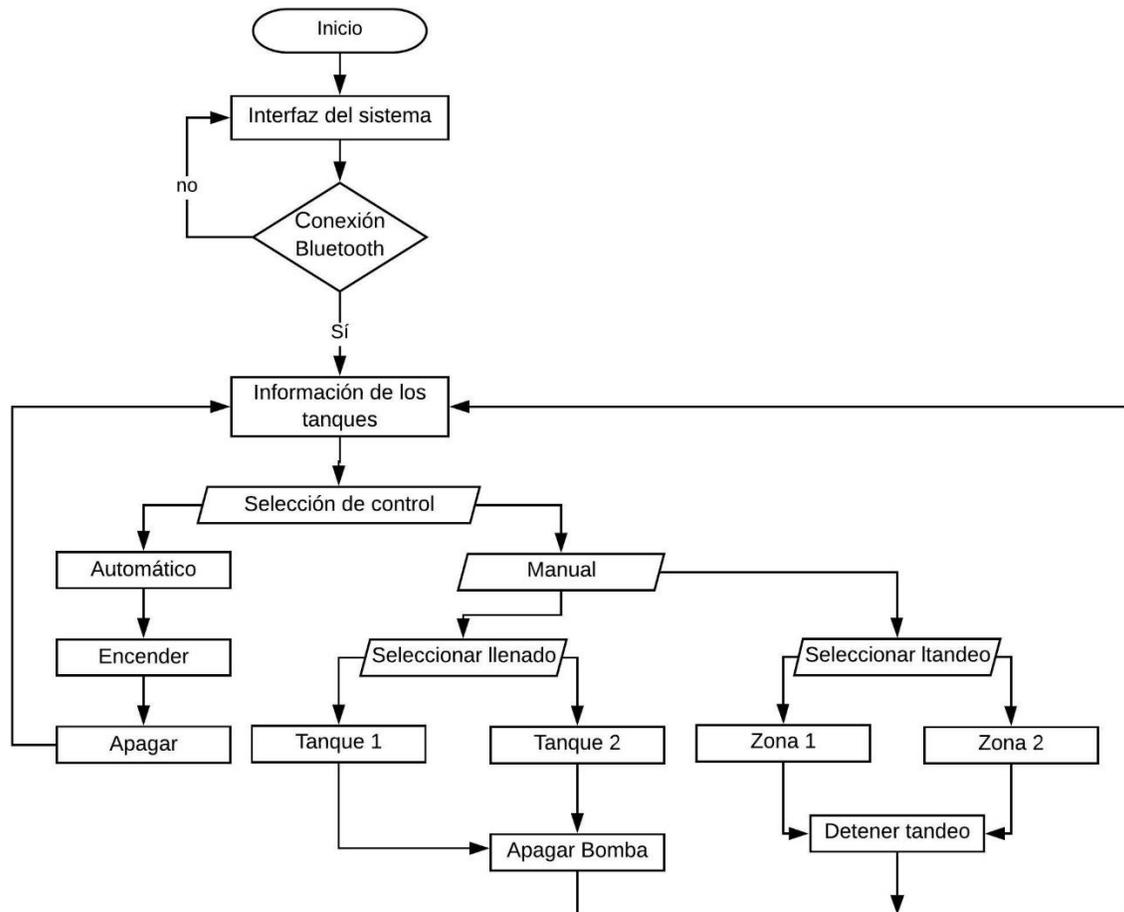


Figura 5.29. Diagrama de la interfaz grafica

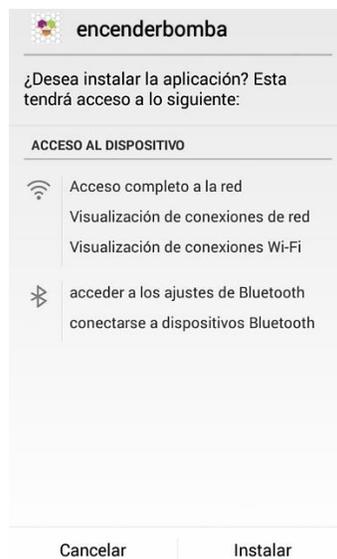


Figura 5.30. Instalación de la aplicación móvil.



*Figura 5.31. Conexión Bluetooth.*

Al realizar la conexión vía Bluetooth la interfaz tiene la apariencia como se muestra en la (Figura 5.32), donde se activan los botones para poder operar el sistema ya que consta con 9 botones con diferentes acciones, que se mencionan a continuación.

1. Botón Desconectar: esta opción desvincula el dispositivo móvil con el dispositivo Bluetooth guardando los datos de la aplicación.
2. Botón Encender: esta opción permite la activación del modo automático permitiendo que el sistema realice las acciones (Abastecimiento, Tandeo, Detección de fugas), por sí mismo sin necesidad de estar presencialmente.
3. Botón Apagar: Desactiva el modo automático permitiendo al usuario realizar las acciones manualmente.
4. Botón Llenar Tanque 1: Activa la bomba y las válvulas monitorizadas para que el agua que se bombea llegue al depósito 1, al activar este botón se obtienen los niveles del tanque si existe comunicación entre los módulos.
5. Botón Llenar Tanque 2: Activa la bomba y las válvulas monitorizadas para llenar el depósito 2.

6. Botón Apagar Bomba: Apaga la bomba cuando los tanques se están llenando manualmente.
7. Botón Tandeo Zona 1: Se realiza el tandeo manualmente en la zona 1.
8. Botón Tandeo Zona 2: Se realiza el tandeo manualmente en la Zona 2
9. Botón Detener tandeo: Se detiene el tandeo que se estaba realizando.
10. Información Tanque 1: se muestran los niveles del depósito 1 así como la información de las zonas donde se está llevando a cabo el tandeo e informa si existen fugas en el depósito.
11. Información Tanque 2: obtienen la misma información mencionada en el punto 10 pero del depósito 2.



*Figura 5.32. Interfaz gráfica móvil.*

Un ejemplo del funcionamiento automático de la interfaz se muestra en la (Figura 5.33), donde se están obteniendo los niveles de los depósitos, permitiendo al coordinador apagar la bomba cuando el nivel se encuentra al 100%, de lo contrario suministre el agua al tanque para llenarlo completamente. Los tanques se llenarán una vez al día para evitar el calentamiento de la bomba y su uso excesivo. El tandeo se activa durante un tiempo específico suministrando agua a una de las zonas, este a su vez se puede activar manualmente. Para la detección de fugas se toman muestras y si ocurre una pérdida, se emite una alerta que puede ver el administrador a cualquier hora del día para que pueda tomar acciones.

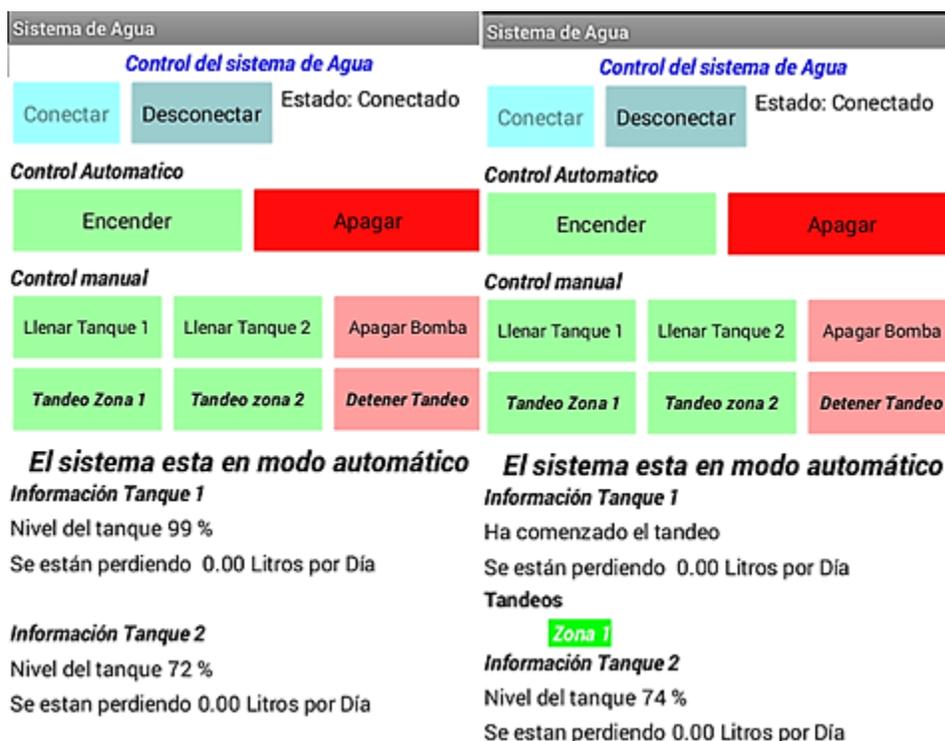


Figura 5.33. Funcionamiento del sistema modo Automatico.

Por otro lado, en modo manual se pueden activar los llenados y tandeos cualquier hora del día, siempre y cuando el administrador esté pendiente para detener dichas acciones; la interfaz permite accionar un llenado y un tandeo al mismo tiempo, si así lo requiere el administrador. Un ejemplo de la interfaz gráfica en modo manual se muestra en la Figura 5.34.

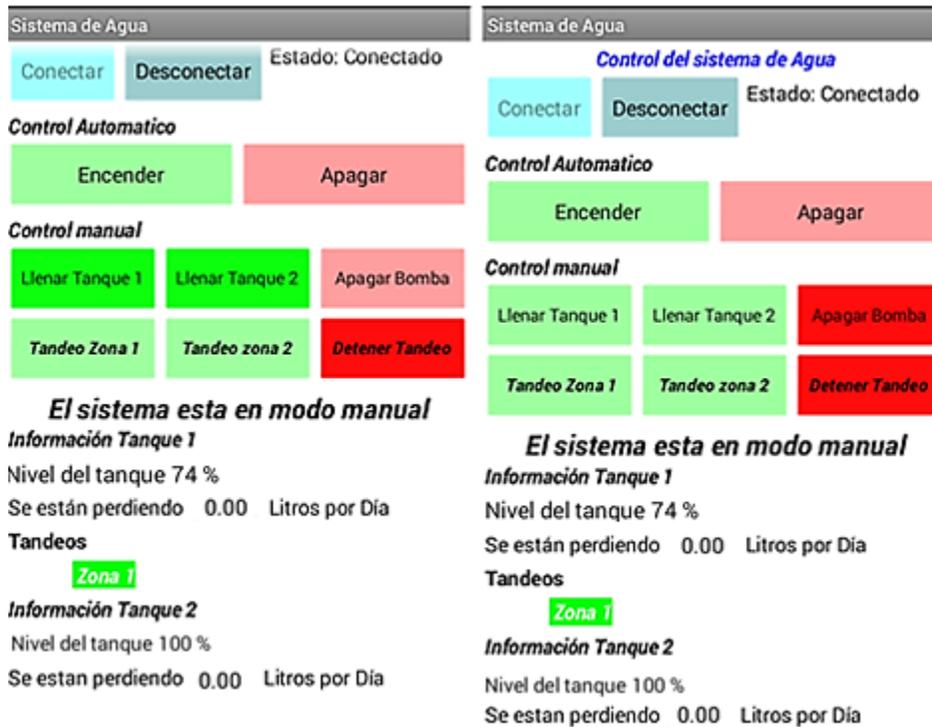


Figura 5.34. Funcionamiento del sistema modo manual.

## 5.4 Experimentación para la validación del sistema

Para realizar la validación de la operación de llenado de los tanques se utilizaron dos contenedores de agua con capacidad de 17.17 L, simulando la capacidad de los contenedores originales, suponiendo que 1L equivale a 10,064L. Dichos contenedores permiten realizar las pruebas para verificar los diferentes procesos que realiza el sistema: llenado, tandeo y detección de fugas. De esta forma, saber la efectividad del sistema mediante un modelo funcional, antes de su implementación real.

En la Figura 5.35 y Figura 5.36, se muestran los contenedores de agua, donde se realizaron respectivas pruebas de llenados, tandeos y detección de fugas. En cada contenedor se suministró diferente cantidad de agua para diferenciar los porcentajes y mostrar la efectividad del modelo funcional.

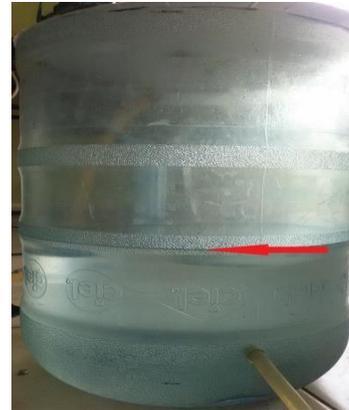


Figura 5.35. Nivel del contenedor 1 a 96%    Figura 5.36. Nivel del contenedor 2 a 50%

Cada contenedor se muestra con diferente porcentaje de llenado según la distancia del sensor con el agua, el contenedor 1 se muestra a un 96% de llenado en comparación con el contenedor 2 cuenta con un 50% de llenado. Para obtener un 100% de llenado, la distancia del sensor con el agua debe ser 3 cm.

Sistema de Agua

**Control del sistema de Agua**

Conectar
Desconectar
Estado: Conectado

**Control Automatico**

Encender
Apagar

**Control manual**

Llenar Tanque 1
Llenar Tanque 2
Apagar Bomba

Tandeo Zona 1
Tandeo zona 2
Detener Tandeo

**El sistema esta en modo automático**

**Información Tanque 1**

Ha comenzado el tandeo

Se están perdiendo 0.00 Litros por Día

**Tandeos**

Zona 1

**Información Tanque 2**

Nivel del tanque 74 %

Se estan perdiendo 0.00 Litros por Día

Figura 5.37. Niveles de los tanques.

Para poder conocer los valores en porcentajes, las distancias se convierten con la ayuda de la expresión 6 dicha grafica se muestra en la Figura 5.38.



$$Porcentaje = (100 - \left(\frac{100}{33} * (Distancia - 3)\right)). \tag{6}$$



Figura 5.38. Conversión de distancias a porcentaje

La detección de fugas se muestra con pruebas realizadas en diferentes tiempos, para el modelo funcional se da una espera de un minuto en comparación del sistema real que se tomaran cada hora. Esto para saber cuándo existen grandes fugas, saberlo de la manera más

rápida, a su vez con las cuatro pruebas como pérdidas se realiza el análisis para ver la pérdida entre estas cuatro y saber si necesita atención, de lo contrario no se alertaría al encargado.

La alerta de fugas no se activa cuando la pérdida de agua no es continua o no se tiene una pérdida grande, en el modelo funcional se toman cuatro pruebas para verificar el desperdicio. Se optó por tomar pruebas en determinados tiempos debido a los asentamientos de los suelos. En la (Figura 5.39) se muestran las pruebas sin tener ninguna pérdida significando que el tanque se encuentra en perfectas condiciones y no necesita ser verificado.

```
|Nivel del tanque|97 %|23.16|0|
1||Deteccion de fugas estado de pruebas:||0.00|0|
1||Se estan perdiendo :||0.00|0|
```

*Figura 5.39. Detección de fugas sin ninguna perdida.*

Los niveles de agua pueden ser diferentes sin tener alguna fuga ya que existen movimientos inesperados, como la propia turbulencia del agua, como se aprecia en la Figura 5.40 , donde se muestra solo una prueba de perdida, pero es por el movimiento del agua que se puede tener regularmente, sin embargo, como las demás pruebas no existen pérdidas no se notifica como fuga.

```
|Nivel del tanque|100 %|0.00|0|
1||Deteccion de fugas estado de pruebas:||0.00|0|
1||Deteccion de fugas estado de pruebas:||0.55|0|
1||Deteccion de fugas estado de pruebas:||0.00|0|
1||Deteccion de fugas estado de pruebas:||0.00|0|
1||Se estan perdiendo :||0.00|0|
```

*Figura 5.40. Fuga no significativa.*

La alerta de fugas se activa cuando existe una fuga demasiado grande o las cuatro pruebas cuentan con pérdidas continuas; en la Figura 5.41 se muestran las cuatro pruebas con pérdidas significando que el tanque tiene un problema y necesita atención urgentemente, en este caso

el desperdicio de dicho tanque es durante las 24 horas ya que en cada prueba se desperdicia una gran cantidad de agua.

```

1| |Deteccion de fugas estado de pruebas:| |0.55|0|
1| |Deteccion de fugas estado de pruebas:| |1.10|0|
1| |Deteccion de fugas estado de pruebas:| |1.65|0|
1| |Deteccion de fugas estado de pruebas:| |0.55|0|
1| |Se estan perdiendo :| |23.16|0|

```

*Figura 5.41. Detección de fugas notoria.*

La acción de tandeos (ya sea en modo manual o automático), se realizan desde la interfaz gráfica permitiendo al encargado no estar presente físicamente para abrir la válvula; en el modelo funcional se especifica una abertura de válvula con un led que ayuda a saber cuándo se está realizando dicho tandeo, como se muestra en la Figura 5.42. , donde se ejecuta el tandeo de la zona uno y en el tanque se muestra el led encendido notificando que el tandeo se está realizando.



*Figura 5.42. Realización de tandeo.*

## 6 ESCALAMIENTO AL SISTEMA REAL

La implementación del sistema real, potencialmente permitirá a la comunidad de Palmillas, San Felipe del Progreso, reducir gastos del personal que se le paga actualmente, así como llevar un control adecuado en suministro y repartición de agua, para la implementación del sistema se hará uso de los siguientes componentes.

- 2 sensores ultrasónicos (JSN-SR04T).
- 11 reguladores a 5V.
- 11 microcontroladores Arduino Leonardo.
- 10 paneles solares con producción estimada de 5.5KWh/día ya que los motores solo necesitan 2.2 KW.
- 10 inversores de corriente de 2500 Watts con salida 110V.
- 11 módulos RF Digi XBee-PRO S2C ZigBee.
- Driver de motor a pasos 2 fases (TB6600)
- Motor a pasos Nema 23 (JK57HS112-3004-03)

Esta comunidad cuenta con dos tanques con capacidad de 172,800 L dicho suministro se efectuará con la bomba que se cuenta actualmente realizando un llenado diario para evitar sobrecalentamientos, teniendo tiempo suficiente para que el pozo profundo logre recuperar el agua que se ha perdido por dicho suministro, asegurando tener la cantidad de agua suficiente para poder llenar el otro tanque. El encendido de la bomba se llevará acabo de manera automática con la ayuda de un Motor a pasos Nema 23 (JK57HS112-3004-03) a un determinado torque de rotación requerida para subir y bajar la pastilla con la que se le suministra energía actualmente. Dicha característica de la bomba se muestra en la siguiente Tabla 6.1.

La distancia entre los tanques y la bomba de abastecimiento abarcan de 1.55 Km a los 2.03 Km dicha comunicación entre los dispositivos se realiza mediante módulos XBee-PRO ZigBee S2C, estos dispositivos logran realizar dicha comunicación hasta 3200 metros de distancia al aire libre, permitiendo el envío y recepción de datos con mucha efectividad dichos módulos compartirán la información necesaria para poder efectuar las acciones que se necesiten durante la activación del sistema (tandeos, detección de fugas y llenados). Dichas especificaciones se muestran en la Tabla 6.2.

Tabla 6.1 Características de la bomba MS125-40

Bomba de agua Pozos profundos	
Tipo	MS125-40
Serial	NR 06009605
KW (kilowatt)	30
HP (Caballos de fuerza)	40
Hz (Corriente que cambia su polaridad N. veces por segundo)	60
RPM (Revoluciones por minuto)	3475
Voltaje	220
F.L. AMP (Amperaje de carga nominal (motor en marcha normal))	106
COSF	0.87
S.F. (Pérdida de carga unitaria)	1.15
S.F. AMP (Pérdida de Amperaje unitario)	120,0

Tabla 6.2 Características de módulos XBee ZigBee.

Especificación	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Distancia Interior / urbano	Hasta 60 m	Hasta 90 m	Hasta 60 m
Línea de RF al aire libre (rango de visión)	Hasta 1200 m	Hasta 3200 m	Hasta 1200 m
Potencia de transmisión salida (máximo)	6.3 mW (+8 dBm), modo boost 3.1 mW (+5 dBm), modo normal La potencia máxima del canal 26 es de +3 dBm.	-63 mW (+18	-6.3 mW (+8 dBm) canal -26 potencia máxima de +1
Velocidad de datos	250,000 b / s	250,000 b / s	250,000 b / s

Sensibilidad del Receptor	102 dBm, modo boost 100 dBm, modo normal	101 dBm	102 dBm, modo boost 100 dBm, modo normal
---------------------------	---	---------	---

Para lograr una comunicación eficaz entre estos dispositivos se hace uso de la tecnología ZigBee configurando estos dispositivos con el protocolo malla, gracias a este protocolo los módulos trabajan como *routers* repitiendo la información con los demás dispositivos alcanzando distancias más alejadas de los que estos pueden soportar, permitiendo la realización de los tandeos en prácticamente cualquier parte de la comunidad.

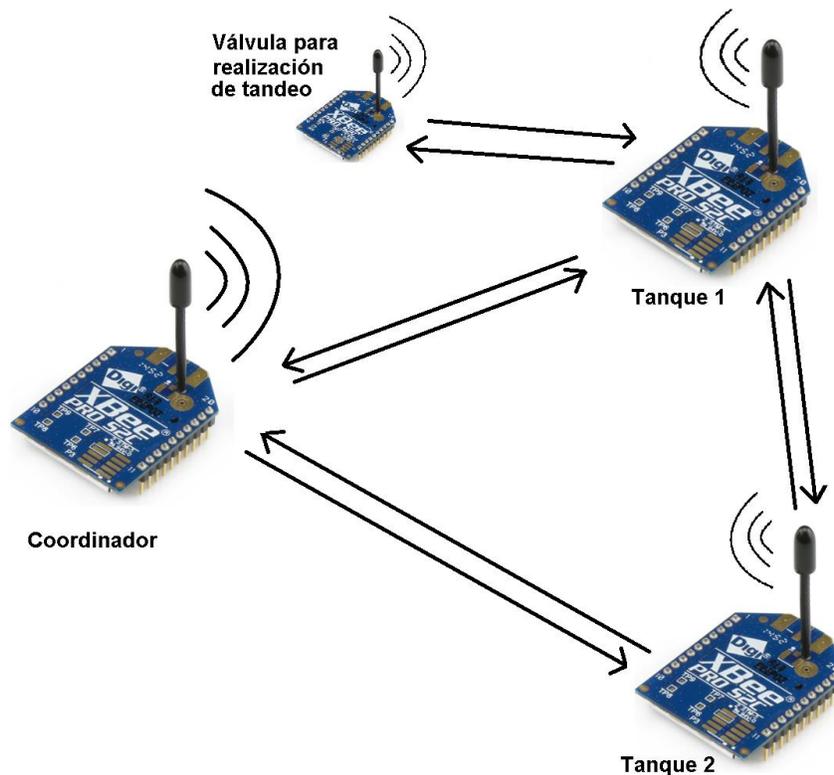


Figura 6.1. Protocolo en malla

La obtención de niveles se realiza mediante el sensor ultrasónico JSN-SR04T, este dispositivo es ideal ya que es resistente a la humedad y es viable para trabajos en exteriores, lo que permite trabajar con mucha efectividad, para el sistema real solo se necesitan 270 cm

de distancia para poder tener el nivel mínimo de agua, este dispositivo puede detectar los niveles del líquido a distancias no superiores a 450 cm superando la distancia que normalmente necesitan los tanques de agua, dichas características se muestran en la Tabla 6.3, con la ayuda de Arduino y el sensor ultrasónico se pueden realizar las acciones correspondientes ya que el sensor obtendrá los niveles y Arduino hace la comparación de los niveles para decidir cuándo inicia o para dicho suministro, a su vez detectar cuando hay pérdidas de agua muy notorias para tener una solución inmediata si se requiere.

*Tabla 6.3 Características del sensor JSN-SR04T*

<b>Sensor ultrasónico JSN-SR04T</b>	
Especificaciones técnicas	<ul style="list-style-type: none"> <li>• Modelo: JSN-SR04(FZ1655)</li> <li>• Voltaje de Operación: 5V DC</li> <li>• Corriente de trabajo: 30mA</li> <li>• Rango de detección: Hasta 450cm</li> <li>• Precisión: puede variar entre los 3mm o 0.3cm</li> <li>• Frecuencia de emisión acústica: 40KHz</li> <li>• Duración mínima del pulso de disparo (nivel TTL): 10 <math>\mu</math>S.</li> <li>• Tiempo entre una medida y el inicio de otro 20 mS.</li> <li>• Ángulo de detección: menor a 50°</li> <li>• A prueba de agua</li> <li>• Diámetro: 22mm</li> <li>• Longitud: 17mm</li> <li>• Temperatura de trabajo: -10°C hasta 70°C</li> </ul>
Pines de conexión	<ul style="list-style-type: none"> <li>• VCC (+5V DC)</li> <li>• TRIG (Disparo del ultrasonido)</li> <li>• ECHO (Recepción del ultrasonido)</li> <li>• GND (0V)</li> </ul>

Los tandeos para las zonas de la localidad mencionada se efectuarán mediante válvulas adaptadas a motores bipolares a pasos Nema 23 con un torque de 14 kg/cm; estos motores estarán engranados para poder realizar las aberturas y cierres de manera eficaz ya que deberán abrir válvulas de 4 pulgadas. Las características del motor se muestran la Tabla 6.4, y para lograr la activación de estos motores desde Arduino, se hace uso del controlador TB6600

*Tabla 6.4 Características Motor a pasos Nema 23*

<b>Características Motor a pasos Nema 23</b>	
Ángulo de paso	1,8 grados
Pasos	200 por vuelta
Fases	4
Corriente	2A/fase
Diámetro de eje:	6.35mm
Par de retención	14kg / cm
Tamaño	NEMA 23

*Tabla 6.5 Características de controlador TB6600*

<b>Características del controlador TB6600</b>	
Corriente de entrada / salida	0 a 5A / 0.5 a 4.0A
Señal de control	3.3 a 24V
Potencia (MAX)	160 W
Paso micro	1, 2 / A, 2 / B, 4, 8, 16, 32
Temperatura	-10 a 45 °C
Humedad	Sin condensación
Peso	0,2 kg
Dimensiones	96 * 71 * 37 mm

La implementación de celdas fotovoltaicas en los tanques y válvulas para el sistema son necesarias ya que en partes de esta localidad no se cuenta con energía eléctrica, estas celdas tendrán pilas recargables permitiendo el funcionamiento de los dispositivos las 24 horas del día.

*Tabla 6.6 Características celdas fotovoltaicas.*

<b>Características de las celdas fotovoltaicas</b>	
Módulos solares	3
Capacidad (CD)	765 WP
Baterías solares	4
Producción diaria	5.5Kwh/día

En la Figura 6.2, se muestra el diseño del sistema real para la comunidad Palmillas, San Felipe del Progreso, en el Estado de México, donde se muestra la distribución de las válvulas y los depósitos según el funcionamiento que realizara con la implementación del sistema real.

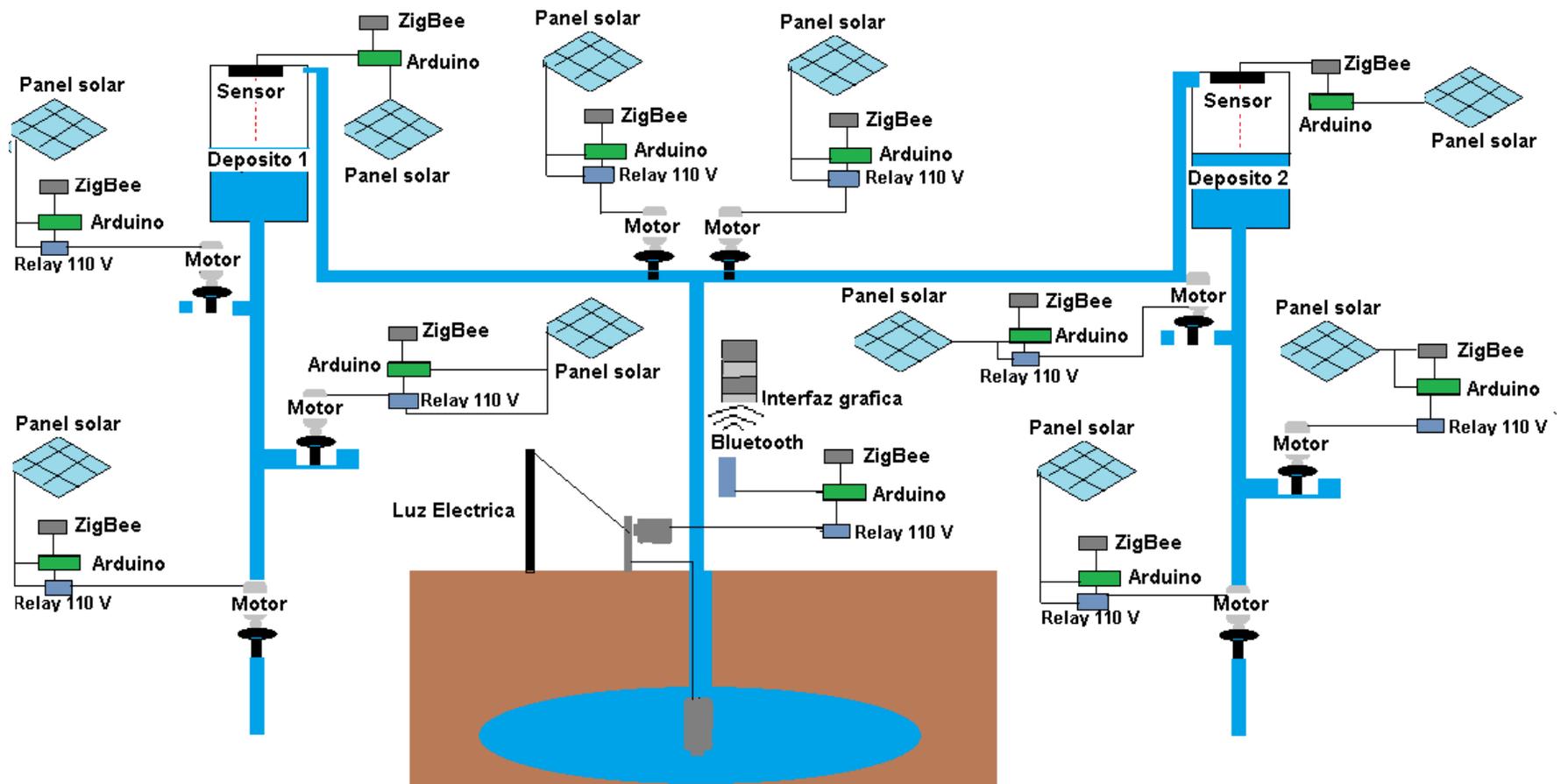
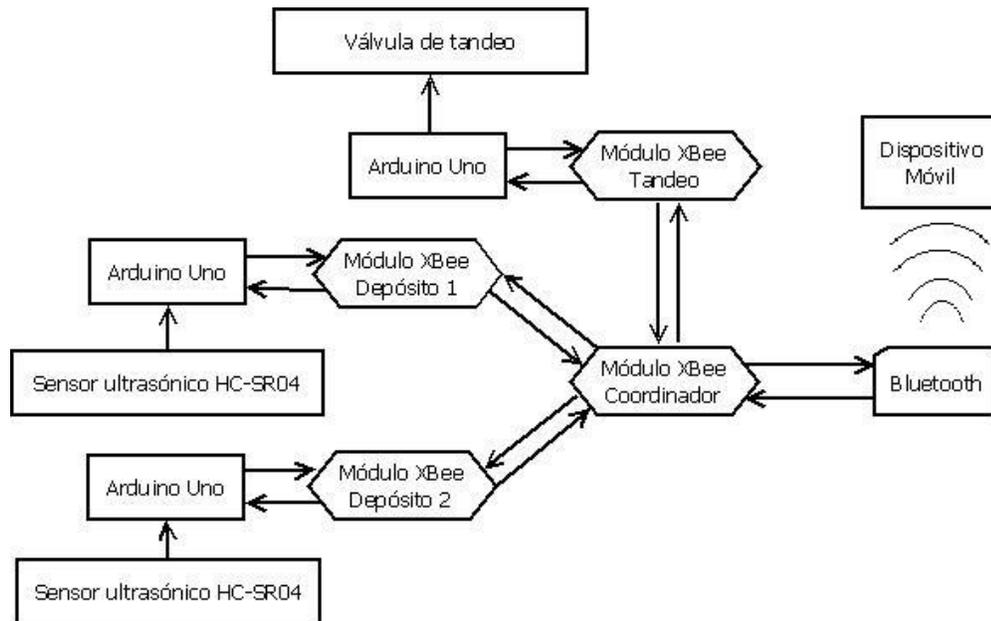


Figura 6.2. Diseño del sistema real de agua potable de la comunidad Palmillas, San Felipe del Progreso, en el Estado de México

## 7 RESULTADOS Y DISCUSIÓN

Los valores obtenidos, así como las acciones que se realizan en los depósitos se envían de forma remota a través de radiofrecuencia (RF) utilizando módulos XBee como se muestra en la Figura 7.1.



*Figura 7.1. Comunicación entre los dispositivos.*

Al realizar las pruebas pertinentes sobre el funcionamiento, tales como mediciones del nivel de agua, estado de las válvulas para los tandeos, así como la detección de fugas, dicha información es obtenida y estructurada con la ayuda de Arduino como se muestra en la Figura 7.2, para posteriormente enviar los datos mediante los módulos XBee, dicha información está estructurada de la siguiente manera (Tanque, Acción, Nivel, Perdida, Tandeo), para posteriormente mostrarse en el dispositivo móvil, en tiempo real.

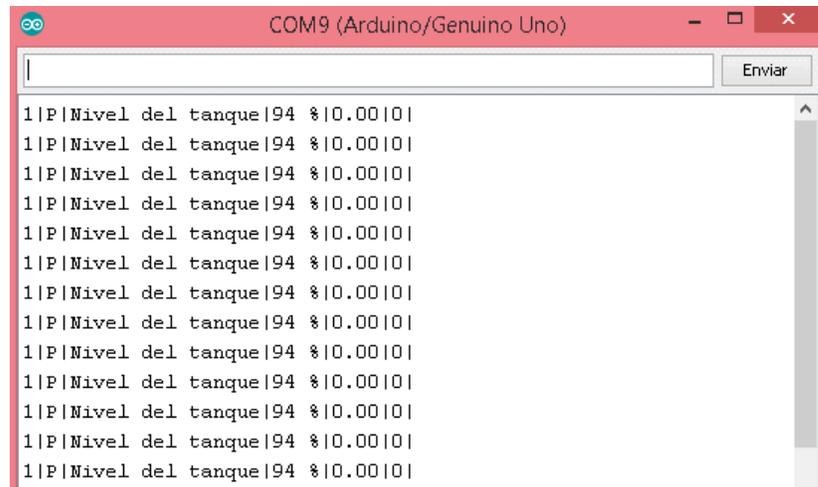


Figura 7.2. Información enviada por el microcontrolador Arduino.

Mediante los módulos XBee se envía la información y se verifica dicha información con la ayuda del programa XCTU, mediante el puerto de comunicaciones se visualiza la información que se recibe y envía como se muestra en la Figura 7.3.

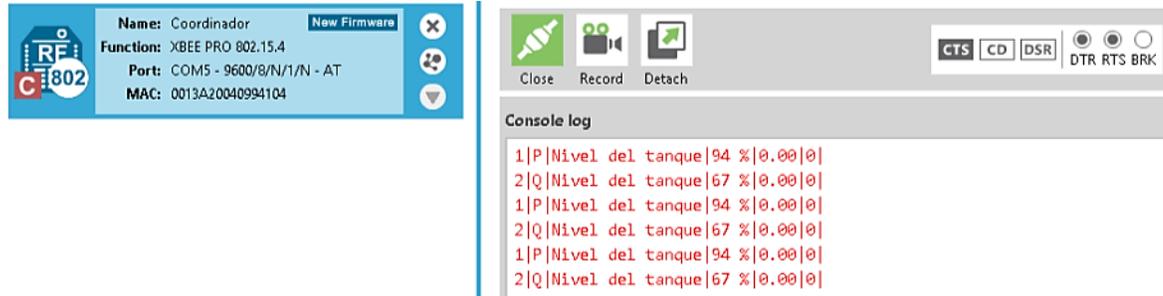


Figura 7.3. Recepción de datos del Coordinador.

El uso de la interfaz gráfica permite a las personas interactuar con el sistema, permitiendo activar o desactivar las funciones que sean requeridas desde cualquier parte de la estación de bombeo, ya que la interfaz se vincula vía Bluetooth permitiendo visualizar los niveles y acciones que se realizan durante el día; dicha interfaz se muestra en la Figura 7.4.



*Figura 7.4. Interfaz gráfica móvil.*

Mediante el uso de la interfaz se realizan actividades automáticamente permitiendo el cuidado del recurso hídrico, así como la reducción de gastos. Además, la implementación del protocolo ZigBee permite tener interconectados los dispositivos, en red malla configurando los dispositivos como *routers* alcanzando la distancia suficiente para el envío de información a distancias de más de 1 km.

La detección de fugas se realiza mediante pruebas en determinados intervalos de tiempo, permitiendo diferenciar la magnitud de la fuga; esto para poder informar al responsable de dicho problema y solucionarlo lo más pronto posible. El modelo funcional toma cuatro muestras, al finalizar las mediciones, si existe una pérdida demasiado grande se informa

de inmediato, de lo contrario se procede a finalizar las cuatro pruebas, si todas las pruebas tienen perdida se alerta al encargado informándole los litros de perdida estimados durante el día, en la Figura 7.5 se muestra activada la alerta, con una pérdida de 19.86 Litros.

```
|1|L|Nivel del tanque|100 %|0.00|0|
|Nivel del tanque|100 %|0.00|0|
1||Deteccion de fugas estado de pruebas:||0.55|0|
1||Deteccion de fugas estado de pruebas:||1.10|0|
1||Deteccion de fugas estado de pruebas:||1.10|0|
1||Deteccion de fugas estado de pruebas:||0.55|0|
1||Se estan perdiendo :||19.86|0|
```

### ***El sistema esta en modo automático*** ***Información Tanque 1***

Ha comenzado el tandeo

Se están perdiendo 19.86 Litros por Día **Alerta**

*Figura 7.5. Alerta de fuga*

Mediante mediciones continuas, el sistema permite diferenciar entre la existencia de fugas y movimientos inesperados del agua, donde solo se tiene una perdida como se muestra en la Figura 7.6, donde no se enciende la alerta ya que ha identificado como movimiento de agua debido a turbulencia.

```
|1|L|Nivel del tanque|100 %|0.00|0|
|Nivel del tanque|100 %|0.00|0|
1||Deteccion de fugas estado de pruebas:||0.00|0|
1||Deteccion de fugas estado de pruebas:||0.00|0|
1||Deteccion de fugas estado de pruebas:||1.10|0|
1||Deteccion de fugas estado de pruebas:||0.00|0|
1||Se estan perdiendo :||0.00|0|
```

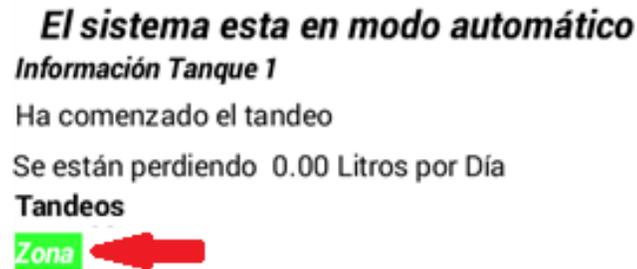
### ***El sistema esta en modo automático*** ***Información Tanque 1***

Nivel del tanque 100 %

Se están perdiendo 0.00 Litros por Día

*Figura 7.6. Fuga pequeña no relevante.*

Los tandeos son representados mediante un led, estos son controlados desde el dispositivo móvil, permitiendo activarlos automática o manualmente según lo desee el responsable, en la Figura 7.7 se muestra en la aplicación el respectivo tanqueo poniéndose en color verde.



*Figura 7.7. Tandeo modo automático.*

Los módulos XBee permiten buena comunicación con las distancias que manejan, en el modelo funcional, donde se usan módulos con un alcance de 100 m mostrando efectividad en 80 m y pérdida de datos a los 90 m al aire libre, los módulos reciben y envían la información de manera correcta, permitiendo adecuadamente la realización de los procesos que se realizan, así como la obtención de los niveles.

Se realizaron pruebas de alcance entre los módulos con la ayuda del software XCTU que nos permite visualizar el indicador de fuerza de la señal recibida (RSSI) generalmente la escala se expresa dentro de valores negativos; cuanto más negativo, mayor pérdida de señal. La intensidad de señal es medida en dBm (decibelio-miliwatt), que es una unidad de medida utilizada principalmente en telecomunicaciones para expresar la potencia absoluta mediante una relación.

La evaluación de distancias entre coordinador-depósito son importantes para verificar el alcance entre estos dispositivos y validar la información enviada, se realiza la primera prueba a 10 metros de distancia descartando pérdida de información enviando 34 paquetes y recibiendo los mismos con una intensidad de -68 dBm (Figura 7.8).

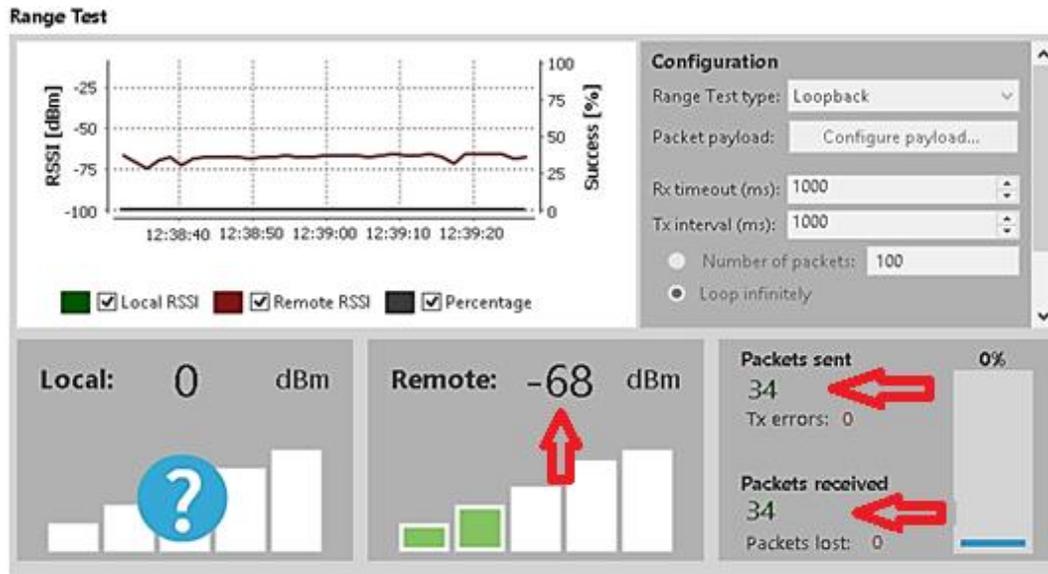


Figura 7.8. Intensidad de señal a 10 metros.

Los dispositivos a 30 metros de distancia son eficientes permitiendo operar sin ningún problema, emitiendo y recibiendo la información con gran efectividad ya que se tiene una señal con fuerza de -78 dBm (Figura 7.9).

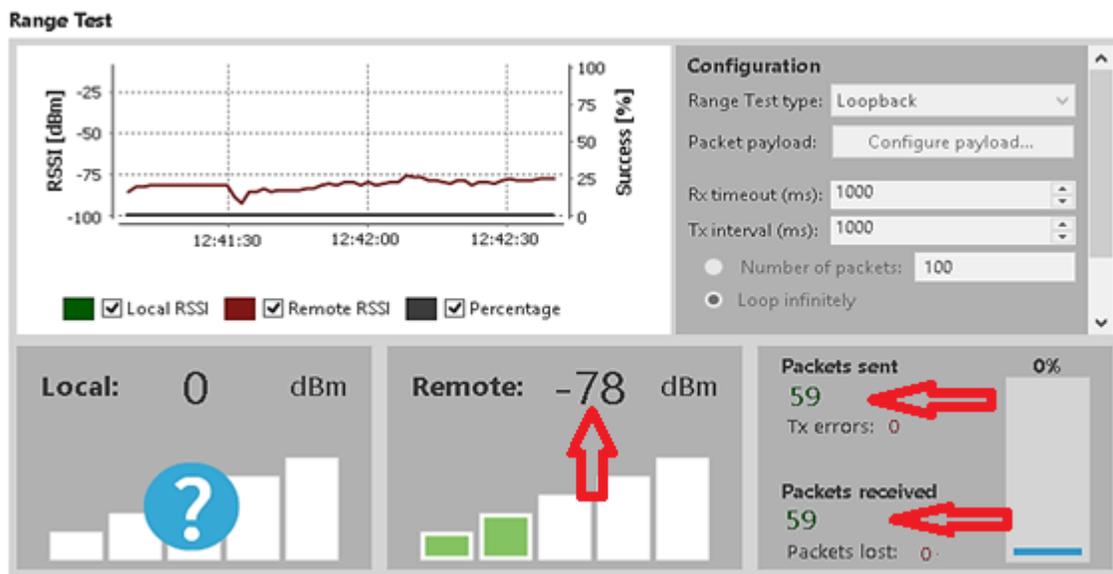


Figura 7.9. Intensidad de señal a 30 metros.

Durante la evaluación a 50 metros de distancia, los dispositivos tienen buena emisión y recepción de datos, descartando pérdidas de información al enviar y recibir los mismos paquetes, con una intensidad de -85 dBm, la importancia de esta evaluación es tener los dispositivos lo más alejado sin que se pierda la información (Figura 7.10).

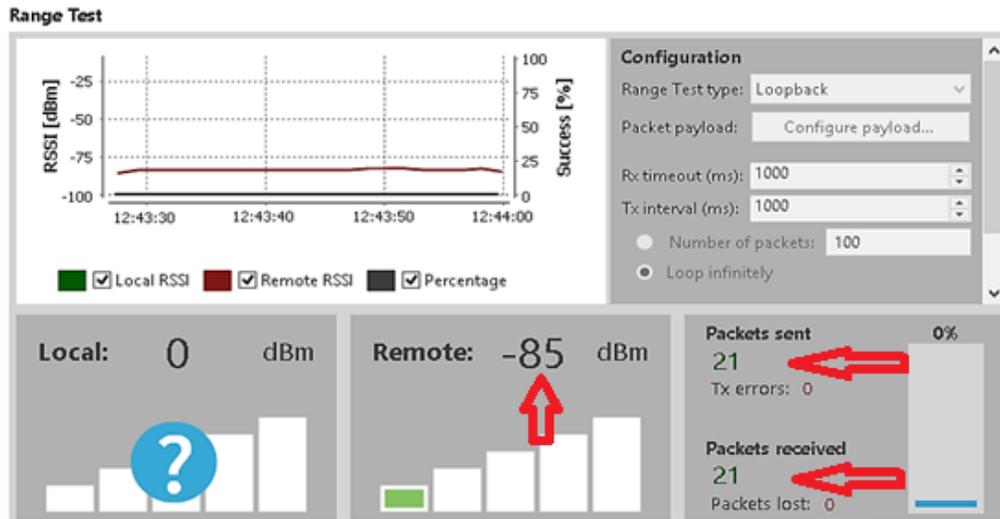


Figura 7.10. Intensidad de señal a 50 metros.

A 80 metros de distancia la intensidad de la señal es -92 dBm (Figura 7.11), los dispositivos se comportan adecuadamente permitiendo efectividad durante el envío de datos, siendo la distancia que se deben de colocar dichos dispositivos, ya que, a partir de los 87 metros, la información suele perderse. Colocar los dispositivos a más de 87 metros de distancia no es recomendable ya que la señal entre más negativa es, mayor pérdida de información con un valor de -95 dBm, donde se encontró que se pierden 6 paquetes de los 40 enviados, provocando que el sistema no opere de la mejor manera, lo que puede generar errores por la pérdida de información durante su operación (Figura 7.12).

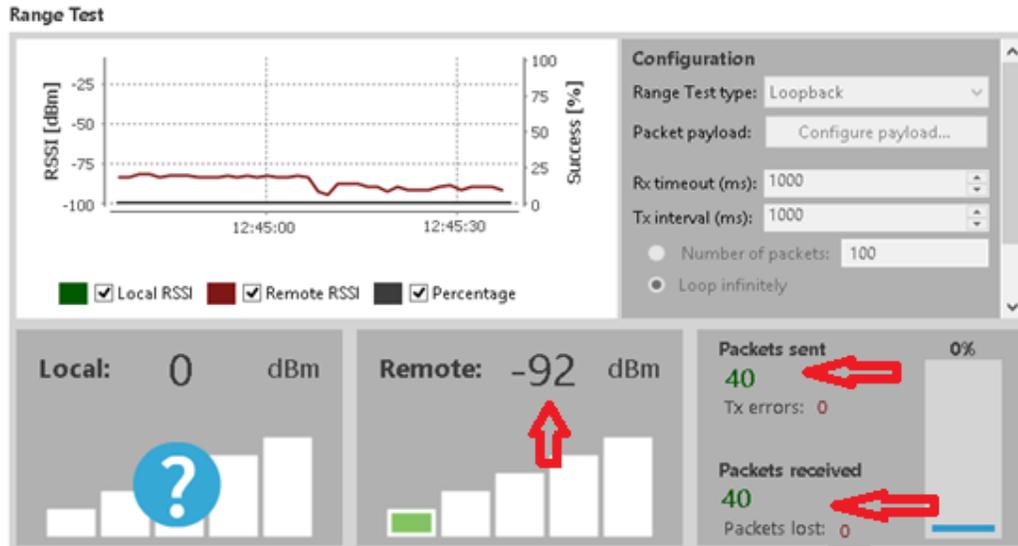


Figura 7.11. Intensidad de señal a 80 metros.

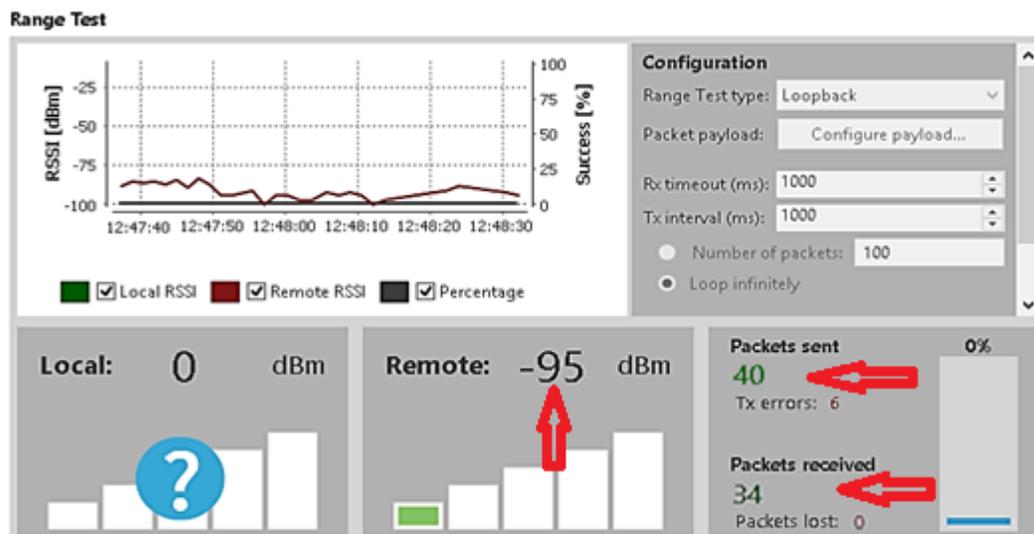


Figura 7.12. Intensidad de señal a 90 metros.

Como un comentario final, es importante mencionar que la implementación del sistema permitiría a las personas de la comunidad a reducir gastos y ahorrar energía eléctrica. Actualmente se solventan gastos mensuales de \$67,440 pesos pagando energía eléctrica y personal que labora entre 843 usuarios que pagan agua, cubriendo un gasto de \$80 pesos por persona, con la implementación del sistema los usuarios pagaran \$52 pesos,

solventando solamente el gasto de energía con un costo de \$43,440 pesos dicha información se muestra en la Figura 7.13.

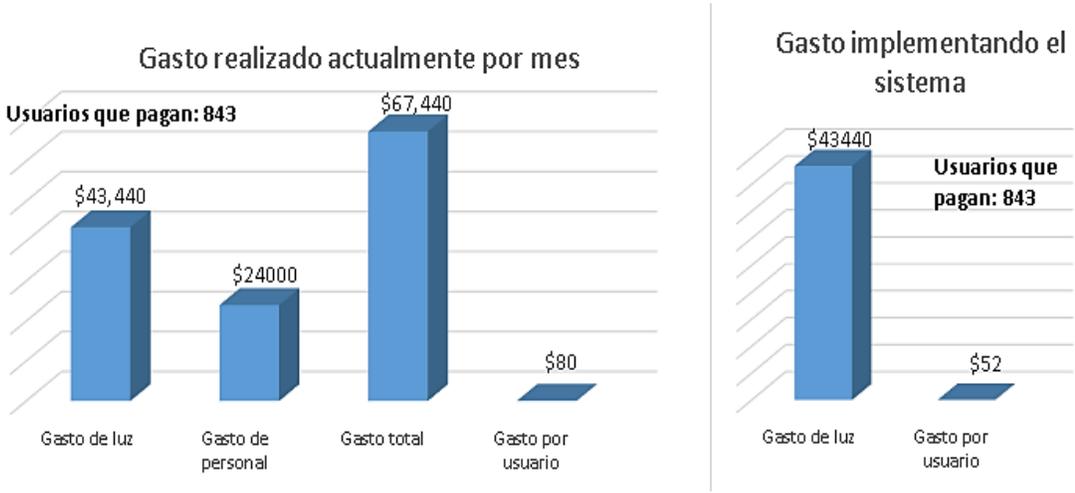


Figura 7.13 Gastos generados mensualmente con y sin el sistema

## CONCLUSIONES

El desperdicio de agua es muy frecuente, ya sea por la acción deliberada del hombre, o por descuidos, fugas o falta de capacidad tecnológica en su manejo; esto sucede tanto en comunidades rurales como en las grandes ciudades y es un problema muy preocupante para la humanidad, debido a la escasez y a la contaminación de este vital líquido.

El uso de tecnologías de comunicación inalámbrica y sistemas computarizados para la automatización de tareas, han sido de mucha utilidad para el control del llenado de tanques elevados de agua potable, ayudando a evitar desperdicios innecesarios, como los que se generan actualmente en la comunidad de Palmillas, donde se estima la pérdida de aproximadamente 4,800 L en cada llenado de sus depósitos de abastecimiento, sin contabilizar las pérdidas ocasionadas por fugas que son detectadas hasta que son plenamente identificables a simple vista.

En este documento se ha presentado el diseño e implementación a escala de un sistema para automatizar el llenado y tandeo de agua, así como para la detección de fugas en tanques. Se hace el uso de la tecnología de transmisión de datos basada en el protocolo ZigBee, que permite la comunicación inalámbrica en distancias de hasta 3,200 m, con bajo consumo energético (120 mA para transmitir y 31 mA para recibir), lo que representa una solución aceptable para las comunidades rurales, donde existen problemas de rápido acceso a los tanques de almacenamiento, falta de supervisión y carencia de energía eléctrica.

La implementación de la tecnología inalámbrica ZigBee es útil para la interconexión de todos los dispositivos en una red de comunicaciones, permitiendo que todos los elementos conectados trabajen como *routers*, lo que admite alcanzar distancias más alejadas de lo que cada dispositivo individual soporta. Esto permite que los diferentes componentes del sistema trabajen de manera coordinada: sistema central de activación de la bomba, sensores de nivel en los tanques de almacenamiento de agua y actuadores de válvulas para sincronizar la distribución de agua en diferentes zonas. Los elementos más alejados se alimentan mediante paneles solares fotovoltaicos, eliminando así la restricción de acceso a la red eléctrica.

El sistema permite el funcionamiento automático de la bomba de llenado de los depósitos mediante el monitoreo constante de los niveles de agua en cada uno de ellos, por lo que se evita el desbordamiento de los tanques, eliminando prácticamente la posibilidad de desperdicio por esta causa.

Para reducir la necesidad de supervisión visual constante, se ha diseñado y programado una aplicación móvil para teléfonos celulares, que permite el monitoreo de los niveles de los tanques y la activación manual de la bomba y válvulas de distribución de agua. Con esto el operador del pozo donde se ubica la bomba de agua, puede acceder a la información más importante del sistema desde una sola ubicación, permitiendo incluso la recepción de alertas por posibles fugas, con lo que se podría actuar de manera inmediata sobre las mismas para minimizar el desperdicio.

Arduino resulta ser una herramienta de desarrollo efectiva para la comunicación con ZigBee, además de ser una tecnología de acceso abierto. En este proyecto se hizo uso de esta plataforma para realizar la recolección de datos desde los sensores ultrasónicos, así como la conversión de los datos analógicos a digital para ser enviados por los módulos de comunicación.

Si bien la validación de la solución propuesta se realiza en un modelo funcional, que representa a escala el sistema real, se ha realizado un ejercicio de diseño para especificar los componentes del sistema, por lo que como trabajo a futuro se propone la estimación de los costos de la implementación física, el financiamiento y la puesta en marcha en la comunidad antes mencionada.

## 8 Bibliografía

A. Bauernfeind, O., O. Zarratea, D. & Botterón, F., 2014. *Control con PSoC de LEDs de Alta Potencia con Paneles Solares para Iluminación Vial*. Argentina, s.n.

Álvarez Gutiérrez, E. L., 2016. Red ZigBee para la Medición de Variables Físicas con Interfaz en Arduino-MATLAB. *Investigación Innovación Ingeniería*, 3(2), pp. 50-64.

Añez, G. & Scarano, M., 2014. Modelado experimental del nivel de un tanque de agua en una planta piloto. *Télématique*, 14(1), pp. 58-78.

Aprendiendo Arduino, 2016. *Definición del protocolo ZigBee*. [En línea]

Available at: <https://aprendiendoarduino.wordpress.com/2016/11/16/zigbeexbee/>  
[Último acceso: 02 Mayo 2018].

Arduino, 2018. *Microcontrolador Arduino*. [En línea]

Available at: <https://www.arduino.cc/>  
[Último acceso: 16 08 2018].

Bastida Perero, Z. M., 2017. *Análisis y diseño de un sistema de control utilizando PLC para el proceso de Fluido y nivel de llenado de tanques*, Tesis de licenciatura en *Ingeniería Electrónica y Telecomunicaciones*. Ecuador: Universidad Estatal Península de Santa Elena.

Calle Zambrano, F. A. & Gaibor Vistin, J. X., 2017. *Automatización de un sistema de riego con monitoreo local usando una touch y control remoto inalámbrico vía GSM (Arduino open source) para el mejoramiento dentro del campo Agrícola en ASOFRUT de la ciudad de Ambato*. Tesis de licenciatura de Ingeniero Industrial, Ecuador: Escuela Superior Politécnica de Chimborazo.

Corona Ramírez, L. G., Abarca Jiménez, G. S. & Mares Carreño, J., 2014. *Sensores y Actuadores Aplicaciones con Arduino*. Primera ed. México: Patria.

Creus, A., 2010. *Instrumentación industrial*. Octava ed. Mexico: Alfaomega.

Digi, 2018. *Módulo RF Zigbee*. [En línea]

Available at: <https://www.digi.com/products/xbee-rf-solutions/2-4-ghz-modules/xbee-zigbee>

[Último acceso: 13 07 2018].

EcuRed, 2011. *Protocolo de comunicación inalámbrica ZigBee*. [En línea]

Available at: <https://www.ecured.cu/ZigBee>

[Último acceso: 16 08 2018].

Enríquez Herrador, R., 2009. *Guía de Usuario de Arduino*. s.l.:s.n.

ETOOLS, 2016. *Funcionamiento sensor ultrasonico HC-SR04*. [En línea]

Available at: <http://www.electrontools.com/Home/WP/2016/04/01/como-funciona-el-sensor-ultrasonico-hc-sr04/>

[Último acceso: 16 08 2018].

Farahani, S., 2008. *ZigBee Wireless Networks and Transceivers*. NEW YORK: Elsevier.

Franquet Bernis, J. M., 2013. *Ecuaciones diferenciales ordinarias y en diferencias finitas : curso práctico*. Primera ed. España: UNED-Tortosa.

Gama Moreno, L. A. y otros, 2016. *A Design of a Water Tanks Monitoring System Based on Mobile Devices*. Cuernavaca, s.n.

García González, A., 2013. <http://panamahitek.com>. [En línea]

Available at: <http://panamahitek.com/guardando-datos-en-arduino-utilizando-la-memoria-eeeprom/>

[Último acceso: 24 08 2018].

Guerrero Cruz, J. L. & Games Segales, D., 2017. *Sistema de monitoreo del nivel del agua en los tanques, para empresas avícolas usando la arquitectura JAVA J2EE y plataforma de prototipos Electronica-Arduino, Tesis de licenciatura en Ingeniero en Sistemas*. Perú: Universidad Privada de TACNA.

Gupta, S. P., Mahalwar, A. & Udaykumar, P., 2014. *Analysis of different techniques for locating leaks in pipes in water distribution system using WSN*. Ghaziabad, s.n.

Hernández Gaviño, R., 2010. *Introducción a los sistemas de control*. Primera ed. México: Pearson Educación.

Iluminet, 2016. *Funcionamiento de paneles fotovoltaicos*. [En línea]  
Available at: <https://www.iluminet.com/funcionamiento-paneles-fotovoltaicos-energia-solar/>  
[Último acceso: 16 08 2018].

Maestre Torreblanca, J. M., 2015. *Domótica para ingenieros*. España: Paraninfo.

Miyara, F., 2004. *CONVERSORES D/A y A/D*. Segunda ed. Rosario, Argentina: Universidad Nacional de Rosario.

National Instruments, s.f. *Definición de LabVIEW*. [En línea]  
Available at: <http://www.ni.com/es-mx/shop/labview.html>  
[Último acceso: 03 Mayo 2018].

Ogata, K., 2010. *Ingeniería de control moderna*. Quinta ed. Madrid: PEARSON.

Oppenheim, A. V. & Willski, A. S., 2014. *Señales y sistemas*. Segunda ed. s.l.:Pearson Prentice Hall.

Oyarce, A., 2008. *Guía del Usuario XBee Series 1*. Santiago, Chile: Ingeniería MCI LTDA.

Panel solar, 2018. *Sistema fotovoltaico*. [En línea]  
Available at: <https://grupoindustronic.com/paneles-solares/>  
[Último acceso: 03 Mayo 2018].

QuimiNet.com, 2010. *Medidores de nivel liquido*. [En línea]  
Available at: <https://www.quiminet.com/articulos/los-medidores-de-nivel-en-liquidos-43573.htm>  
[Último acceso: 15 08 2018].

Rashid, H. y otros, 2017. *A new design approach of home automation system for patients with physical disability to reduce water wastage and power consumption using renewable energy*. Dhaka, s.n.

Real Academia Espanola, 2014. <http://dle.rae.es>. [En línea]

Available at: <http://dle.rae.es/?id=Z4X1Jr8>

[Último acceso: 31 Julio 2018].

Sergio R, C., 2009. *Equisbi: desarrollo de aplicaciones con comunicación remota basadas en módulos ZigBee y 802.15.4*. Primera ed. Buenos Aires: s.n.

Shrenika, R. M. y otros, 2017. *Non-contact Water Level Monitoring System Implemented Using LabVIEW and Arduino*. Bangalore, s.n.

Soubam, S., Agrawal, M. & Naik, V., 2017. *Using an Arduino and a smartwatch to measure liquid consumed from any container*. Bengaluru, s.n.

Tocci, R. & Widmer, N. S., 2003. *Sistemas Digitales Principios y aplicaciones*. Octava ed. México: Pearson Educación.

Vera Romero, C. A., Barbosa Jaimes, J. E. & Pabón González, D. C., 2015. Parámetros de configuración en módulos XBEE-PRO® S2B ZB para medición de variables ambientales. *Tecnura*, 19(45), pp. 141-157.

XBee Comunicaciones inalambricas, 2014. *Modos de operacion de los Modulos XBee*.

[En línea]

Available at: <http://modulosxbee.blogspot.com/2014/01/modos-de-operacion.html>

[Último acceso: 16 08 2018].

XBee.cl, 2018. *Comunicacion inalambrica Modulo XBee*. [En línea]

Available at: <http://xbec.cl/que-es-xbee/>

[Último acceso: 16 08 2018].

## ANEXOS

### Acciones que realizan los dispositivos colocados en los tanques

#### Calculo de distancia.

Para el caso del método de la distancia, se hace uso de la Ec 7 para trabajar la distancia en cm.

$$d = 0.01715 * t (\mu\text{s})\text{cm} \quad \text{Ec 7}$$

Dónde:

d→ Distancia entre el sensor y nivel del agua.

0.01715→ velocidad del sonido a través del aire en ( $\mu\text{s}$ ).

t→ Tiempo de retardo de la señal.

El código fue el siguiente:

```
#define trigPin 7 //se define el nombre del pin 7
#define echoPin 6 //se define el nombre del pin 6

long tiempo;
long distancia;
void setup() {
  Serial.begin (9600); // inicializa el puerto serial a 9600 baudios
  pinMode (trigPin, OUTPUT); // define el pin 7 como salida (triger)
  pinMode (echoPin, INPUT); // define el pin 6 como entrada (echo)
}
void lecturasensor(){
  digitalWrite (trigPin, LOW); //para generar un pulso limpio ponemos a LOW 4us
  delayMicroseconds (4);
  digitalWrite (trigPin, HIGH); // genera el pulso de triger por 10ms
  delayMicroseconds (10);
  digitalWrite (trigPin, LOW);
```

```

tiempo = pulseIn(echoPin, HIGH); //Medimos el tiempo entre pulsos, en microsegundos
distancia = tiempo*0.01715; // Distancia entre el sensor con el nivel del agua.
}

```

### Niveles de los tanques.

Los niveles de los tanques se mandan una vez iniciado el sistema tomando tres centímetros de distancia del sensor como el 100%.

Para obtener la distancia en porcentajes se realizaron dos ecuaciones:

En la primera ecuación se reducen tres centímetros, ya que es la distancia donde llega nuestro 100% de llenado, y se multiplica por la división del 100% entre los 30 cm que se tomaran para los porcentajes donde tres centímetros representan el 100% y 33 cm el 0%.

$$\text{operacion1} = ((\text{distancia}-3) * (10/3)); \quad \text{Ec 8}$$

En la segunda ecuación se toma el 100% y se resta el valor obtenido de la Ec 8, permitiendo obtener los porcentajes con gran exactitud como se muestra en la Ec 9.

$$\text{operacion2} = (100) - (\text{operacion1}); \quad \text{Ec 9}$$

El código fue el siguiente:

```

void datos(){
long operacion1;//Variable para obtener la distancia en porcentaje
long operacion2;//Variable para reducir el porcentaje según el 100%
operacion1=((distancia-3)*(10/3));//distancia menos el nivel máximo por la división de
100/30
operacion2=(100)-(operacion1);//resta del 100% con el valor obtenido de la operación 1
EEPROM.get(direccion, calc1 );//Inicializa el valor guardado en la memoria EEPROM
Serial.print ("|");
Serial.print("Nivel del tanque");

```

```

Serial.print ("|");
Serial.print (operacion2);
Serial.print("%");
Serial.print ("|");
Serial.print(calc1,2);//Muestra el valor guardado en la EEPROM con 2 decimales
Serial.print ("|");
Serial.print ('0');
Serial.println ("|");
}

```

### **Entrada al modo automático y acciones manuales**

Con la ayuda de Arduino, se realizan acciones manuales y automáticas, cuando el sistema entra en modo automático procede a realizar los llenados, tandeos y detección de fugas por sí mismo sin necesidad de estar al pendiente.

Con las acciones manuales este dispositivo realiza los tandeos recibiendo las letras que manda el coordinador que especifican el inicio del tanqueo y la finalización de dicha acción.

El código fue el siguiente:

```

void loop() {
if(Serial.available()){ // Verificamos si el puerto serie está disponible
valor = Serial.read(); //Leemos los valores que introducen en el serial
if(valor=='A'){ //Si el valor entrante es A el sistema entra en modo automático e inician
los llenados
llenados(); //la función para realizar los llenados de los tanques
}
if(valor=='T1'){ //Si se recibe T1 en el serial se inicializa el tanqueo
digitalWrite(tandeezonas, HIGH);
}
if(valor=='D'){ //Si se recibe D en el puerto serial se apaga el tanqueo
digitalWrite(tandeezonas, LOW);
}
}
}

```

```
}}
```

### Llenados automáticos

El llenado automático se realiza mediante la comparación de distancias, ya que el suministro de agua se realiza mientras las distancias entre el sensor y el nivel del agua se encuentre mayor a 3 cm y menor o igual a 33 cm.

```
if(distancia >3 && distancia <=33 ){
```

El suministro de agua se detiene cuando la distancia es menor o igual a 3cm

```
if(distancia <=3){
```

Para salir del modo automático Arduino debe de recibir mediante el puerto serial la letra H el cual reinicia el código permitiendo realizar nuevamente las acciones de inicio.

```
if (valor=='H'){
```

```
asm("jmp 0x0000");
```

El código fue el siguiente:

```
void llenados(){ //función de los llenados
```

```
lecturasensor(); //se obtienen las distancias del sensor con el nivel del agua
```

```
if (Serial.available()){ //Si el puerto serie está disponible
```

```
valor = Serial.read(); //se leen los datos de entrada mediante el serial
```

```
if (valor=='H'){ //si la entrada en el serial es H se cancela el modo automático
```

```
asm("jmp 0x0000"); //reset
```

```
}}
```

```
if(distancia >3 && distancia <=33 ){ //El suministro del agua se realiza mientras la distancia sea mayor a 3 cm y menor o igual a 33 cm.
```

```
Serial.print ('1'); //Se manda mediante el serial el número del tanque en el que se están realizando las acciones.
```

```
Serial.print ("");
```

```
Serial.print ('P1'); //Se manda la letra P mediante el serial que identifica que la bomba debe de prenderse.
```

```

datos();// Se mandan los datos sobre el estado del tanque.
}
if(distancia <=3){ //El suministro de agua se detiene cuando la distancia es menor o igual
a 3 cm.
Serial.print ('1'); //Se manda mediante el serial el número del tanque en el que se están
realizando las acciones.
Serial.print ("|");
Serial.print ('L'); //Se manda la letra L que es el identificador con el que se apaga la bomba
datos(); //Se mandan los datos sobre el estado del tanque.
delay(3000);
digitalWrite(sleep, LOW); //Pin que despierta a los módulos XBee para mandar
informacion
delay(30000); //Retardo para esperar que el modulo encienda completamente
digitalWrite(sleep, HIGH); //Pin que duerme el modulo para aumentar la durabilidad de
las baterías
delay(1500);
fugas(); //Cuando el sistema determina que el tanque está completamente lleno, procede
a esperar un determinado tiempo para posteriormente realizar la detección de fugas.
}
delay(5000); //retardo para realiza nuevamente la función
llenados(); //la función se repite hasta que el tanque se llene completamente
}

```

### **Detección de fugas**

La detección de fugas se realiza mediante pruebas realizadas en determinados tiempos

Con la ayuda de la función `lecturasensor()` se obtienen las distancias correspondientes a las pruebas.

```

void fugas(){
//Primer nivel de prueba
lecturasensor(); //Función que manda los niveles del agua.

```

```
datos(); // Información del tanque
```

```
float distancia1 = distancia; //se toma el primer nivel cuando el agua está en completo reposo.
```

```
delay(20000);
```

Las pruebas se toman en diferentes tiempos, Si existe una fuga grande se notifica al encargado informándole la cantidad en Litros de agua que se está perdiendo.

```
lecturasensor();
```

```
float distancia2 = distancia; se toma la segunda prueba para compararla con la primera
```

```
float r2=distancia2-distancia1; //diferencia entre las distancias
```

```
if(r2 >2){
```

```
float aguamili=551.54715*r2; //Operación para calcular la perdida (pi* radio(13.25) al cuadrado)*comparacion
```

```
float calc2 = (aguamili*.001); // Conversión de mililitros a Litros
```

```
informarfugas();
```

Si la comparación entre las pruebas es mayor a 2 cm se notifica al encargado sin tomar las demás pruebas de lo contrario se siguen tomando las demás, al finalizar con las pruebas se realiza un análisis donde:

- Si todas las pruebas resultan con pérdidas bajas se notifica al encargado para su reparación guardando el dato en la memoria (EEPROM) de Arduino para mostrar esta información en el dispositivo todo el tiempo hasta que se realice la siguiente detección de fugas, mostrando el aproximado de pérdida por día según las pruebas, dejando en conciencia al encargado si requiere reparación no.
- Si existe solo una pérdida insignificante no se notifica como fuga ya que se puede tratar de un asentamiento del suelo.

El código fue el siguiente:

```
float distancia1 = distancia; //se toma el primer nivel cuando el agua está en completo reposo.
```

```
delay(20000);
```

```
lecturasensor();
```

```

float distancia2 = distancia; se toma la segunda prueba para compararla con la primera
float r2=distancia2-distancia1; //diferencia entre las distancias
if(r2 >2){
float aguamili=551.54715*r2; //Operación para calcular la perdida (pi* radio(13.25) al
cuadrado)*comparacion
float calc2 = (aguamili*.001); // Conversión de mililitros a Litros
informarfugas();
Serial.print(calc2);
Serial.print ("|");
Serial.print ('0');
Serial.println ("|");
}
else{
delay(20000);
lecturasensor();
float distancia3 = distancia;// prueba 2
float r3=distancia3-distancia2;
if(r3 >2){
float aguamili=551.54715*r3;//(pi* radio 13.25 al cuadrado)*comparación
float calc3 = (aguamili*.001);
informarfugas();
Serial.print(calc3);
Serial.print ("|");
Serial.print ('0');
Serial.println ("|");
}
else{

delay(20000);
lecturasensor();
float distancia4 = distancia;// prueba 2

```

```

float r4=distancia4-distancia3;
if(r4>2){
float aguamili=551.54715*r4;//(pi* radio 13.25 al cuadrado)*comparacion
float calc4 = (aguamili*.001);
informarfugas();
Serial.print(calc4);
Serial.print ("|");
Serial.print ('0');
Serial.println ("|");
}
else{

delay(20000);
lecturasensor();
float distancia5 = distancia;// prueba 2
float r5=distancia5-distancia4;
if(r5>2){
float aguamili=551.54715*r5;//(pi* radio 13.25 al cuadrado)*comparacion
float calc5= (aguamili*.001);
informarfugas();
Serial.print(calc5);
Serial.print ("|");
Serial.print ('0');
Serial.println ("|");
}
else{

if(distancia >=0){
float suma=r2+r3+r4+r5;
float aguamili=551.54715*suma;//(pi* radio 13.25 al cuadrado)*comparacion

```

```

float calc1= (aguamili*.001)*(horas del día); //perdida de agua según las horas que
duraron las pruebas
Serial.print ('1');
Serial.print ("|");
Serial.print ("");
Serial.print ("|");
Serial.print ("Se estan perdiendo :");
Serial.print ("|");
Serial.print ("");
Serial.print ("");
Serial.print ("|");
if(r2>0 && r3>0 && r4>0 && r5>0 ){
Serial.print(calc1);
Serial.print ("|");
Serial.print ('0');
Serial.println ("|");
int direccionInicio = 0; //declaro una variable con la dirección de inicio
EEPROM.put(direccionInicio, calc1); //escribe la variable calc1 en la dirección definida
(EEPROM)
}
else{
float calc2=0;
Serial.print(calc2);
Serial.print ("|");
Serial.print ('0');
Serial.println ("|");
int direccionInicio = 0; //declaro una variable con la dirección de inicio
EEPROM.put(direccionInicio, calc2); //escribe la variable calc1 en la dirección definida
(EEPROM)
}
}
delay(3000);

```

```

digitalWrite(sleep, LOW);
delay(10000);//esperamos 10 segundo
digitalWrite(sleep, HIGH);
delay(1500);
tandeo();
}}
}}
}

```

### Realización del Tandeo

Los tandeos actualmente se especifican por un led el cual nos muestra cuando se esta realizando el tandeo y cuando para dicha acción.

El código fue el siguiente:

```

void tandeo(){
EEPROM.get( direccion, calc1 ); //lee una variable float de la posición indicada de
memoria
Serial.print ('1');
Serial.print ("|");
Serial.print ('T1');
Serial.print ("|");
Serial.print("Ha comenzado el tandeo"); //Mensaje de inicialización del tandeo
Serial.print ("|");
Serial.print ("");
Serial.print ("|");
Serial.print(calc1,2); //coloca el valor guardado EEPROM con 2 decimales
Serial.print ("|");
Serial.println("Zona 1");
Serial.print ("|");
delay(3000);
digitalWrite(sleep, LOW); //pin controlado para dormir el módulo XBee

```

```

digitalWrite(tandeezonas, HIGH);
delay(30000);
digitalWrite(sleep, HIGH);
delay(1500);
digitalWrite(tandeezonas, LOW);
llenados();
}

```

### Acciones que realizan los dispositivos colocados en el coordinador

El coordinador tiene la función de recibir todas las acciones que se realizan desde el dispositivo móvil con la ayuda del Bluetooth.

Desde el dispositivo móvil se realizan las acciones, y el coordinador se encarga de reenviar la información a los tanques para realizar las acciones correspondientes. En este dispositivo es necesario utilizar dos entradas seriales ya que se recibirán y enviarán datos mediante los módulos XBee y Bluetooth.

En este Arduino está conectado el relay para activar la bomba, así como los leds para especificar que válvula que estará abierta y cerrada para el respectivo llenado, de esta manera evitar que se sobrecaliente la bomba de abastecimiento y se llenen los dos tanques al mismo tiempo.

```

#include <SoftwareSerial.h> //librería de los seriales
SoftwareSerial XbeeSerial(3, 2); //entrada serial en los pines 2,3
int bomba = 13; //pin de salida para la bomba
int valvula1 = 11; //pin de salida para activación de válvula para el llenado del tanque 1
int valvula2 = 12; //pin de salida para activación de válvula para el llenado del tanque 2
float incomingByte; //variable de entrada del serial conectado al módulo Xbee
char valor; //Variable para indicar que llega una orden desde el dispositivo móvil

void setup() {
pinMode(bomba, OUTPUT); //pin de salida

```

```

pinMode(valvula1, OUTPUT); //pin de salida para especificar acciones de las válvulas
abierto-cerrado.
pinMode(valvula2, OUTPUT); //pin de salida para especificar acciones de las válvulas
abierto-cerrado.
Serial.begin(9600); // inicializa el puerto serial a 9600 baudios
XBeeSerial.begin(9600); // inicializa el puerto serial del XBee a 9600 baudios
digitalWrite(bomba, HIGH); //pin para mantener la bomba apagada
digitalWrite(valvula1, HIGH); //las válvulas de entrada abiertas
digitalWrite(valvula2, HIGH); //las válvulas de entrada abiertas
}
void loop() { Al inicio de la conexión se muestran los niveles de los tanques y se reciben
las acciones a realizar.
niveles();
menu(); //función menu recibe las acciones que se desean realizar desde el dispositivo
móvil
}
void menu(){
if (Serial.available()){ //Si el puerto serie (Bluetooth) está disponible
valor = Serial.read(); //Guardamos los datos entrantes en la variable valor.
if (valor == 'A'){ //Si el dato que llega es una A esta en modo automático
XBeeSerial.print('A'); //se manda la especificación a los tanques en el modo que deben de
estar mediante la letra A
automatico (); //Entra en modo automático
}
if (valor == 'M'){ //llenado manual del tanque 1
digitalWrite(valvula2, LOW); //La valvula2 se cierra permitiendo solo el flujo en la
valvula 1.
digitalWrite(bomba, LOW); //Se enciende la bomba para abastecer el tanque de forma
manual
manual(); //Entramos en modo manual.
}
}

```

```

if (valor == 'N'){ //llenado manual del tanque 2
digitalWrite(valvula1, LOW); //La valvula1 se cierra permitiendo solo el flujo en la
valvula 2.
digitalWrite(bomba, LOW); //Se enciende la bomba para abastecer el tanque de forma
manual
manual();
}
if (valor == 'T1'){ //Se recibe la letra T1 se reenvía por el serial del módulo XBee para
iniciar el tandeo en una de las zonas del Tanque1
XbeeSerial.print("T1");
}
if (valor == 'D'){ //Se envía la letra P para detener el tandeo
XbeeSerial.print("D");
}
if (valor == 'T2'){ // Se recibe la letra T2 se reenvía por el serial del módulo XBee para
iniciar el tandeo en una de las zonas del Tanque2
XbeeSerial.print("T2");
}
if (valor == 'H'){ //Si el dato que llega es una H se reenvía por el serial del módulo XBee
para finalizar las acciones que se están realizando en caso de que sea necesario.
XbeeSerial.print("H");
}}
}

void automatico(){ //Las acciones de llenado inician en modo automático recibiendo las
variables desde los tanques para inicializar el suministro o apagarlo.
//llenar tanque 1
if (XbeeSerial.available()) { //si hay datos por el serial 2,3 Modulo XBee
incomingByte = XbeeSerial.read(); //guardamos los datos en la variable
Serial.print((char)incomingByte); //Se reciben los datos del módulo XBee
if (incomingByte == 'P1') { //si el valor entrante es P enciende la bomba
digitalWrite(valvula2, LOW); //válvula 2 se sierra

```



```

}
if (incomingByte == 'L') { // Si el valor entrante es L se apaga la bomba
digitalWrite(bomba, HIGH); //Bomba apagada
digitalWrite(valvula1, HIGH); //válvula abierta
}
if (incomingByte == 'T2') { //Cuando se realizan los tandeos del tanque 2 empieza el
llenado del tanque 1
automatico(); //Función para llenar el tanque 1
}}
if (Serial.available()) { //lee los datos entrantes por el serial del Bluetooth
valor = Serial.read(); //guardamos los datos en variable
if (valor == 'H'){ //Si el dato que llega es una H se detiene la acción que se está realizando
XbeeSerial.print('H');
digitalWrite(bomba, HIGH); //Se apaga la bomba
delay(500);
asm("jmp 0x0000"); //reset
}
}
llenartanque2();
}

```

Cuando se inicializan las acciones manuales es necesario pararlas, se declaró que cuando se reciba la letra P se detiene cualquier acción manual que se esté realizando.

```

void manual(){
if (Serial.available()) //Si el puerto serie (Bluetooth) está disponible
{
valor = Serial.read(); //Lee el dato de entrada en el módulo Bluetooth
if (valor == 'P') //Si el dato que llega es una P se apaga la bomba y se reinicia el programa
{
digitalWrite(bomba, HIGH); // Se apaga la bomba
delay(500);
asm("jmp 0x0000"); //reseteamos el programa
}
}
}

```

```
}}
```

niveles(); //Si los módulos XBee reciben Información se mostraran los niveles de los tanques.

manual(); //Se recarga la función manual hasta que se detenga

```
}
```

Los niveles se reciben mediante el serial donde se conecta el módulo XBee, estos niveles se recibirán siempre y cuando exista comunicación entre dichos dispositivos, para mantener informado al encargado sobre los niveles de ambos tanques.

```
void niveles(){
```

```
if (XbeeSerial.available()) { //si hay datos por el serial 2,3
```

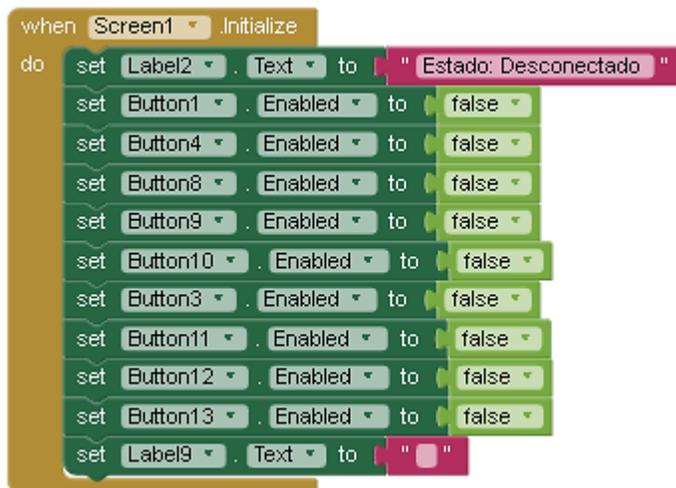
```
incomingByte = XbeeSerial.read(); //guardamos los datos en la variable
```

```
Serial.print((char)incomingByte); //Mostramos los niveles de los tanques mediante el serial para mostrarlos en el dispositivo movil.
```

```
}}
```

## Interfaz del dispositivo móvil.

Pantalla de inicio de la aplicación con el estado de los botones activados y desactivados.



Inicializamos la detección de dispositivos Bluetooth.



Se realiza la conexión y se activan los botones que se pueden utilizar después de la conexión, también se realiza el guardado de las acciones de los botones para que cuando se vuelva a reconectar el sistema permanezcan los botones activados y desactivados que se dejaron activos la última vez.



Los botones se programan para desactivar botones que no se usaran durante ese proceso y mandar las letras que especifican las acciones que se deben de realizar.

```
when Button4 .Click
do
  call BluetoothClient1 .SendText
  text " A "
  set Label9 .Text to " El sistema esta en modo automático "
  call TinyDB1 .StoreValue
  tag " Label9 "
  valueToStore " El sistema esta en modo automático "
  set Button4 .Enabled to false
  call TinyDB1 .StoreValue
  tag " Boton4 "
  valueToStore false
  set Button8 .Enabled to true
  call TinyDB1 .StoreValue
  tag " Boton8 "
  valueToStore true
  set Button9 .Enabled to false
  call TinyDB1 .StoreValue
  tag " Boton9 "
  valueToStore false
  set Button10 .Enabled to false
  call TinyDB1 .StoreValue
  tag " Boton10 "
  valueToStore false
  set Button11 .Enabled to false
  call TinyDB1 .StoreValue
  tag " Boton11 "
  valueToStore false
  set Button13 .Enabled to false
  call TinyDB1 .StoreValue
  tag " Boton13 "
  valueToStore false
```

Realizamos la recepción de los datos que entran por el serial para mostrar la información de los tanques

```
initialize global lista_datos to create empty list
initialize global datos_entradaBT to ""
```

La información se acomoda según sea el tanque permitiendo tener la información de los tanques en diferente posición.

```
when Clock1 .Timer
do
  if BluetoothClient1 .IsConnected
  then
    if call BluetoothClient1 .BytesAvailableToReceive > 0
    then
      set global datos_entradaBT to call BluetoothClient1 .ReceiveText
      numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
      set global lista_datos to split text get global datos_entradaBT
      at " "
      set Label24 .Text to select list item list get global lista_datos
      index 1
      if Label24 .Text = "1"
      then
        set Label29 .Text to select list item list get global lista_datos
        index 2
        set Label7 .Text to select list item list get global lista_datos
        index 3
        set Label6 .Text to select list item list get global lista_datos
        index 4
        set Label11 .Text to select list item list get global lista_datos
        index 5
        if Label11 .Text > 5
        then
          set Label22 .BackgroundColor to red
        else
          set Label22 .BackgroundColor to white
        set Label23 .Text to select list item list get global lista_datos
        index 6
        if Label23 .Text = 0
        then
          set Label23 .BackgroundColor to white
        else
          set Label23 .BackgroundColor to green
      else if Label24 .Text = "2"
      then
        set Label14 .Text to select list item list get global lista_datos
        index 2
        set Label14 .Text to select list item list get global lista_datos
        index 3
        set Label15 .Text to select list item list get global lista_datos
        index 4
        set Label17 .Text to select list item list get global lista_datos
        index 5
        if Label17 .Text > 5
        then
          set Label27 .BackgroundColor to red
        else
          set Label27 .BackgroundColor to white
        set Label28 .Text to select list item list get global lista_datos
        index 6
        if Label28 .Text = 0
        then
          set Label28 .BackgroundColor to white
        else
          set Label28 .BackgroundColor to green
```

Desconexión del dispositivo móvil con Bluetooth.

```
when Button1 .Click
do
  call BluetoothClient1 .Disconnect
  set Label2 .Text to " Estado: Desconectado "
  set ListPicker1 .Enabled to true
```