UJAT
Universidad Juárez Autónoma de Tabasco
"Estudio en la duda. Acción en la fe"

DAIS 11111000011
División Académica de Informática y Sistemas

Universidad Juárez Autónoma de Tabasco

Tesis de Maestría

Prototipo de alerta temprana para desborde de ríos

Que presenta

Ernesto Rafael León Cornelio

Para obtener el grado de

Maestro en Ciencias de la Computación

Director

Dr. Miguel Antonio Wister Ovando

Cuerpo Académico

Sistemas Distribuidos

Línea de Generación y Aplicación del Conocimiento

Sistemas Ubicuos

Cunduacán, Tabasco, México Noviembre 2019

UJAT
Universidad Juárez Autónoma de Tabasco
"Estudio en la duda. Acción en la fe"

DAIS 11111000011
División Académica de Informática y Sistemas

Universidad Juárez Autónoma de Tabasco

Tesis de Maestría

Prototipo de alerta temprana para desborde de ríos

Que presenta

Ernesto Rafael León Cornelio

Para obtener el grado de

Maestro en Ciencias de la Computación

Director

Dr. Miguel Antonio Wister Ovando

Jurado:

Dr. José Adán Hernández Nolasco	Presidente
Dr. José Hernández Torruco	Secretario
Dr. Oscar Alberto Chávez Bosquez	Vocal

Cuerpo Académico

Sistemas Distribuidos

Línea de Generación y Aplicación del Conocimiento

Sistemas Ubicuos

Cunduacán, Tabasco, México Noviembre 2019

CARTA DE AUTORIZACIÓN

El que suscribe, autoriza por medio del presente escrito a la Universidad Juárez Autónoma de Tabasco para que utilice tanto física como digitalmente la Tesis de **Maestría "Prototipo de Alerta Temprana para Desborde de Ríos"**, de la cual soy autor y titular de los Derechos de Autor.

La finalidad del uso por parte de la Universidad Juárez Autónoma de Tabasco de la tesis antes mencionada, será (única y exclusivamente para difusión, educación, sin fines de lucro; autorización que se hace de manera enunciativa más no limitativa para subirla a la Red Abierta de Biblioteca Digitales (RABID) y a cualquier otra Red Académica con las que la Universidad tenga relación institucional.

Por lo antes mencionado, libero a la Universidad Juárez Autónoma de Tabasco de cualquier reclamación legal que pudiera ejercer respecto al uso y manipulación de la Tesis mencionada y para los fines estipulados en este documento.

Se firma la presente autorización en la Ciudad de Villahermosa, Tabasco a los 31 días del mes de octubre del año 2019.

AUTORIZO

C. Ernesto Rafael León Cornelio

Oficio No. 3273/2018/DAIS/D
28 de noviembre de 2018

Dr. Miguel Antonio Wister Ovando
Profesor-Investigador
Presente

De acuerdo al artículo 46 fracción III del Reglamento General de Estudios de Posgrado Vigente, de la Universidad Juárez Autónoma de Tabasco, me permito informarle a Usted, que ha sido asignado director del trabajo de tesis titulado **"PROTOTIPO DE ALERTA TEMPRANA PARA DESBORDE DE RÍOS"**, a realizar por el **C. Ernesto Rafael León Cornelio**, para obtener el grado de Maestro en Ciencias de la Computación.

Sin otro particular, aprovecho la ocasión para enviarle un afectuoso saludo.

Atentamente

UNIVERSIDAD JUAREZ AUTONOMA DE TABASCO
DIVISION ACADEMICA DE INFORMATICA Y SISTEMAS

MTE. Óscar Alberto González González
Director

C.c.p. MASI. Arturo Corona Ferreira.-Encargado del Despacho de la Coordinación de Posgrado.
Archivo.
Consecutivo.

Carretera Cunduacán-Jalpa Km. 1, Colonia Esmeralda, C.P. 86690. Cunduacán, Tabasco, México.
E-mail: direccion.dais@ujat.mx
Teléfonos: (993) 358 1500 ext. 6727; (914) 336 0616; Fax: (914) 336 0870

UNIVERSIDAD JUÁREZ AUTÓNOMA DE TABASCO

"ESTUDIO EN LA DUDA. ACCIÓN EN LA FE"

DIVISIÓN ACADÉMICA DE INFORMÁTICA Y SISTEMAS

Cunduacán Tabasco 31 Octubre 2019

En la Universidad Juárez Autónoma de Tabasco, de acuerdo al Reglamento de Estudios de Posgrado vigente, se revisó el trabajo de investigación titulado **"PROTOTIPO DE ALERTA TEMPRANA PARA DESBORDE DE RÍOS"**, realizado por el **C. Ernesto Rafael León Cornelio**, para obtener el Grado de Maestro en Ciencias de la Computación bajo la modalidad de Tesis.

Los integrantes del jurado, después de revisar el trabajo, lo declararon aceptado. Firmando la presente a los 31 del mes de octubre de 2019.

Dr. José Adán Hernández Nolasco

Dr. José Hernández Torruco

Dr. Oscar Alberto Chávez Bosquez

Carretera Cunduacán-Jalpa Km. 1, Colonia Esmeralda, C.P. 86690. Cunduacán, Tabasco, México.
E-mail: direccion.dais@ujat.mx
Teléfonos: (993) 358 1500 ext. 6727; (914) 336 0616; Fax: (914) 336 0870

Oficio No.2530/19/DAIS/D
31 de octubre de 2019

C. Ernesto Rafael León Cornelio
Matrícula 172H13002

En virtud de que cumple satisfactoriamente los requisitos establecidos en el Reglamento General de Estudio de Posgrado vigente en la Universidad, informo a Usted que se autoriza la impresión del trabajo recepcional **"PROTOTIPO DE ALERTA TEMPRANA PARA DESBORDE DE RÍOS"**, para presentar examen y obtener el Grado de Maestro en Administración de Tecnologías de la Información bajo la modalidad de Tesis.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente

UNIVERSIDAD JUAREZ AUTONOMA DE TABASCO
DIVISION ACADEMICA DE INFORMATICA Y SISTEMAS

MTE. Oscar Alberto González González
Director

C.c.p. MASI Arturo Corona Ferreira. - Encargado del Despacho de la Coordinación de Posgrado.
Archivo.
Consecutivo.

Carretera Cunduacán-Jalpa Km. 1, Colonia Esmeralda, C.P. 86690. Cunduacán, Tabasco, México.
E-mail: direccion.dais@ujat.mx
Teléfonos: (993) 358 1500 ext. 6727; (914) 336 0616; Fax: (914) 336 0870

Índice general

Índice de figuras

Índice de tablas

Índice de fragmentos de códigos de programación

Resumen

En el presente trabajo se realizó un prototipo de sistema de alerta temprana para desborde de ríos, en el que se implementó Hardware y Software libre para lograr la transferencia de datos, donde se emplearon dos protocolos de comunicación. El protocolo clave de este desarrollo es LoRa, ya que tiene gran cobertura para transmitir en lugares con poca extensión territorial de servicios de telecomunicación; y el protocolo MQTT fue utilizado para la emisión de los datos a un sistema informático, cuyo objetivo en este desarrollo es, enviar alertas con los datos obtenidos por los sensores a una red social.

Mediante la integración del Hardware y software de estándares libres y empleando el lenguaje C++, un poco de Java Script y la herramienta de programación Node-RED, se obtuvo un sistema automatizado que capta las variaciones de nivel de agua de un río mediante un nodo sensor, el cual transmite la información en tiempo real al nodo receptor, quien al mismo tiempo envía la información a una Raspberry pi, en consecuencia se encarga de registrar los datos en una base de datos y publicarlos en Twitter.

Capítulo 1

Generalidades

1.1. Introducción

Hoy en día la computación tiene mucha utilidad, es difícil imaginarse algún área o disciplina en donde la informática no se aplique, desde actividades del ámbito administrativo, educación, transporte, seguridad hasta en el campo de la medicina[8].

La computación está compuesta por dos partes opuestas software y hardware. El software es el entorno virtual (sistemas operativos, programas informáticos, aplicaciones, protocolos de comunicación), tiene la finalidad de ser la interfaz entre el usuario y el hardware, en consecuencia, el hardware es el otro aspecto de la computación, de manera que hace referencia a las partes mecánicas y electrónicas de un computador[32].

Una rama de la ciencia de la computación es la computación ubicua, de manera que este concepto trata de cómo la computación está presente en el entorno de las personas en objetos cotidianos, objetos cotidianos como por ejemplo; un teléfono inteligente o un reloj inteligente, están recabando datos como temperatura, ubicación o frecuencia cardíaca por mencionar algunos, estos datos pueden ser transmitidos, debido a lo antes mencionado, el cómputo ubicuo es empleado por el Internet de las cosas, ya que este concepto trata de la interacción que tienen los objetos cotidianos con Internet, así como también un termostato inteligente puede avisar al usuario la temperatura a través de su teléfono inteligente, teniendo esto en cuenta en nuestro entorno hay variables de tipo ambiental que se pueden medir por medio de redes de sensores, estos datos recabados se pueden usar para alertar de ciertas

variables a las personas o para advertir de un posible desastre natural[46].

Los desastres naturales son acontecimientos catastróficos que pueden ocurrir en cualquier parte del mundo. Se presentan de diferentes maneras como inundaciones, terremoto, tormentas, etc., que derivan de los fenómenos naturales como lluvia, viento y temperatura. El organismo de la Agencia Europea de Medio Ambiente (AEMA) determina a los desastres naturales como transformaciones violentas, que provocan la pérdida de vidas humanas y bienes materiales. Una catástrofe natural es una fase que sucede habitualmente a lo largo de la vida de la humanidad, por lo que la sociedad en gran mayoría es afectada por destrucciones de sus bienes [17].

En años pasados, la población se hallaba en constante desprotección, tanto que no podían adivinar si pasaría alguna catástrofe. No obstante, es sustancial prevenir a los habitantes, hallar una forma de informar a los demás, disminuir las pérdidas materiales y aumentar en conveniencia de las personas la evasión de los desastres naturales. El crecimiento de la humanidad y la transformación de ramas como la ciencia y la tecnología, favorecen el desarrollo de sistemas de seguridad y alarma, empleados en ubicaciones de alto riesgo.

En el siglo XX, los sistemas de alerta temprana se desarrollaron rápidamente debido a la primera y la segunda guerra mundial. Pasaron a tener carácter masivo y más centralizado, cubriendo un importante número de poblaciones, ciudades y proporcionando la oportunidad de informar a tantos habitantes como fuera posible[42].

El presente trabajo se enfoca en el desarrollo de un prototipo de medición del nivel de agua en los cuerpos acuíferos con el propósito de captar los cambios, registrar los datos obtenidos en una base de datos y publicar en una red social las variaciones de nivel del agua. El motivo de publicar los datos en una red social es porque la mayoría de las personas alrededor del mundo tienen la posibilidad de acceder a ellas, ya que no tiene restricciones de género, etnicidad, nacionalidad, ni clase social.

De tal manera que cualquier individuo podrá acceder a la información publicada en una cuenta específica para el proyecto en cuestión. Las redes sociales se han vuelto muy importantes para los internautas, no sólo en importancia y funcionalidades, sino también en tiempo y uso que les dedican a ellas tanto personas como empresas.

Por tanto, aunque las personas y empresas suelen pensar de forma inmediata solo en tres plataformas principales como: “Facebook, Twitter e Instagram”, así pues, existen cientos en todo el mundo, y con millones de usuarios,

que pueden ser beneficiosas para las empresas según sus necesidades y ámbitos de acción [15]. Debido a lo antes mencionado se puede sacar provecho de la tendencia de las personas a utilizar las redes sociales, para de esa manera poder informales de ciertos acontecimientos, en este caso de alertarles de posibles desbordamientos de agua o informarles acerca del estado en el que se encuentra algún río que le interese al usuario conocer.

La plataforma que se utilizará para este trabajo es Twitter, ya que es una red social dedicada a la mensajería y cuenta con una cantidad considerable de usuarios alrededor del mundo, con unos 319 millones de usuarios registrados en esta red social a la fecha.

En el Capítulo 1 explica el problema a resolver y plantea los objetivos de la investigación, incluyendo la hipótesis, justificación y metodología.

En el Capítulo 2 se expone el estado del arte de que se permitió adquirir el conocimiento necesario para saber la manera más idónea de desarrollar este proyecto, se introducen conceptos relacionados con el "Internet de las cosas", la comunicación con LoRa y redes de sensores lo cual son los conceptos primordiales para este desarrollo y también en este apartado se expone las tecnologías utilizadas para concretar el prototipo que se ubican en el marco tecnológico.

En el Capítulo 3 se redactaron los pasos, configuraciones y el cómo se vincula cada una de las partes de este prototipo.

En el Capítulo 4 se describen las pruebas realizadas y los resultados obtenidos.

Finalmente, las Conclusiones, Contribuciones y Trabajos futuros plasman las contribuciones y resultados esperados con esta investigación.

1.2. Planteamiento del problema

1.2.1. Definición del problema

Los desastres naturales se dan en cualquier parte del mundo, cuando ocurre un acontecimiento de esa índole es muy importante a las personas, previo a que el éste ocurra, en especial a quienes les afectará de manera directa. Lamentablemente en los tiempos actuales, muchos medios de comunicación han dejado de ser utilizados habitualmente como son la radio, la televisión o el periódico, debido a que las nuevas generaciones se informan por medio de redes sociales o páginas de Internet, ya que la información de un suceso se

difunde de manera más rápida. También los medios de comunicación se están adaptando al cambio de lo físico a lo digital, es una manera fácil de llegar al público y menos costosa en cuestión de manufactura. Aunado a lo anterior, la información que publican incluso en un noticiero puede variar en cuestión de segundos, puede darse el caso de manipular información y reportar datos erróneos. Por lo que el problema detectado es cómo transmitir en tiempo real a las personas de manera inmediata la información que presenta un río, algunas desventajas que se pueden presentar al momento de monitorear algún río radica en que se pueden encontrar en zonas donde no haya acceso a Internet, por ello una comunicación Wi-Fi o una comunicación vía telefónica no se podría lograr y por medio de *bluetooth* sería muy costoso ya que se necesitarían demasiados nodos para poder lograr la transmisión de datos.

1.3. Delimitaciones de la investigación

1.3.1. Alcances

1. El prototipo desarrollado es una red de sensores, que mide el nivel de agua y transfiera la información obtenida a un sistema informático, de tal forma que este sistema informático emita las alertas en Twitter por medio de tweets con el dato obtenido por la red de sensores.

2. Se probó en un entorno controlado (laboratorio).

1.3.2. Limitaciones

1. El sensor empleado para el prototipo tiene una cadencia de sensado desde 20 cm hasta 3 metros.

2. Las tarjetas TTGO LoRa transmite a una distancia máxima de 700 m.

3. Para realizar las pruebas de este prototipo se empleó un *smartphone*, el cual funciona como un Access point, puesto que para transmitir los tweets se necesita conectividad a Internet y esto genera gasto de datos.

4. Se emplea la API de Twitter, la cual tiene restricciones[1].

[1]La API no entrega el 100 % de todos los tweets ni da acceso a el firehose para más información: https://flows.nodered.org/node/node-red-node-twitter.

5. Probar el prototipo en un entorno real seria costoso y dificil, por este motivo solo se probara en un laboratorio.

1.4. Hipótesis

Es posible comunicar/alertar a los usuarios de una red social, el nivel de un cuerpo de agua mediante la interacción de una red de sensores y un sistema informático, con una efectividad de entrega de mensajes del 80 %.

1.5. Objetivo general

Realizar un prototipo de alerta temprana (ubicuo) que informe el nivel de agua de un río a través de la publicación de una red social (Twitter).

1.6. Objetivos específicos

1. Desarrollar una red de sensores capaz de medir los cambios de nivel de agua.
2. Integrar un protocolo de comunicación que sea idóneo para el desarrollo del prototipo.
3. Establecer comunicación con Twitter y publicar las alertas.

1.6.1. Justificación

Tabasco sufre de fuertes lluvias en distintas temporadas del año, esto ocasiona que los niveles de agua en los ríos asciendan y lleguen a desbordarse, lo que ha provocado afectaciones en el estado y sobre todo en los habitantes de zonas bajas, quienes no logran obtener información en tiempo real para conocer la situación que presentan los ríos. Los datos se obtienen a través de una red de sensores que consta de un nodo (nodo sensor), ubicado a la orilla del cuerpo de agua para conocer el nivel del agua. Estos datos se transmiten vía radio frecuencia a otro nodo (nodo receptor), ubicado en un lugar con acceso a Internet para compartir los datos obtenidos con una computadora de placa única encargada de registrar, mostrar y emitir una notificación/alerta por medio de una red social a los usuarios. Dado lo anterior, se justifica un

sistema de alerta temprana que informe a la población de las perturbaciones que tenga un río, en tiempo real y de una manera prácticamente inmediata. La notificación/alerta se envía mediante una red social[2], ya que en la época actual la mayoría de las personas son usuarios de por lo menos una red social y cuentan con un dispositivo móvil con conexión a Internet. Además, se utilizará un protocolo de radio frecuencia llamado LoRa, para el envío de datos de manera remota porque no es posible disponer de Internet o comunicación vía telefónica en zonas en donde se encuentran los ríos.

1.7. Metodología

Este proyecto es de tipo cuantitativo, ya que se obtendrán datos de nivel de agua por medio de una red de sensores, los cuales procesará un microcontrolador para posteriormente transmitir los datos a una computadora y así ser publicados en una red social y almacenados en una base de datos.

1. Investigación bibliográfica: se realiza una investigación bibliográfica acerca de trabajos previos que sean similares, como resultado de la búsqueda bibliográfica se obtiene un conocimiento de cómo se han podido integrar una red de sensores con sistema informático.

2. Diseñar una red de sensores: llevar acabo los arreglos pertinentes de la red de sensores para su buen funcionamiento, es ideal para que los datos publicados sean correctos.

3. Obtención del nivel de agua: este dato se obtendrá mediante la implementación de un nodo sensor el cual forme parte de la red de sensores.

4. Transferencia de datos: los datos obtenidos en el cuerpo de agua sensado, necesitan ser transferidos a una computadora para su posterior publicación en una red social. Los protocolos de comunicación implementados son LoRa y MQTT.

5. Pruebas: se realiza en un entorno controlado, por medio de un contenedor de agua, del cual se sacará y se agregará agua para ver como varía su nivel, se observará si la red de sensores adquiere los datos correctos,

[2]Hay 3,2 mil millones de usuarios de redes sociales en todo el mundo, y este número continúa creciendo. Esto equivale aproximadamente al 42 % de la población actual [24].

sí envía los datos al sistema informático para que registre y publique los datos obtenidos por la red de sensores en Twitter.

6. Ajustes: Los resultados obtenidos en las pruebas y las salidas de los sensores, permitirán observar si es necesario realizar ajustes en el prototipo.

Capítulo 2

Marco teórico

2.1. Conceptos y teorías fundamentales de la investigación

En este capítulo se presentan muchos conceptos relacionados con el desarrollo del prototipo de sistema de alerta temprana para desborde de ríos, por lo que se dará una sencilla y breve descripción de cada uno de ellos.

2.1.1. Prototipo

Es el primer intento de desarrollo de un ejemplar, del que se recolectan conceptos como diseño, soporte y tecnología, para posteriores versiones mejoradas [12].

2.1.2. Alerta temprana

Es una herramienta que permite la reducción de pérdida material y de vidas ante un desastre natural [14].

2.1.3. Sensor

Instrumento que tiene la capacidad de percibir las magnitudes físicas y químicas, el mismo las interpreta en magnitudes eléctricas [50].

2.1.4. Red de sensores inalámbricas

Es una red de microcontroladores (nodos) que integran sensores, que realizan una tarea específica. Las redes de sensores inalámbricas están formadas por un grupo de nodos sensores los cuales tienen el atributo de percibir ciertas sensaciones del mundo real. Este sistema emite señales por medio de un grupo de nodos para que por medio de ellas se envíe y se reciba información [1].

2.1.5. Protocolo de comunicación

Es un conjunto de normas que permiten que los dispositivos electrónicos puedan transferir y recibir datos (información) entre ellos [48].

2.1.6. Comunicación LoRa

Es una tecnología que funciona en la banda Industrial, *Scientific and Medical* (ISM). La asignación de frecuencias y los requisitos reglamentarios para ISM varían por región. En la región de México se utiliza la 915 MHz, la asignada para el continente americano. Esta tecnología es funcional para proyectos que requieran la transmisión de datos en distancias lejanas con un rango de transmisión de más 15 km entre nodo [31].

2.1.7. Access point

Es un punto de acceso inalámbrico, por sus siglas en inglés: *Wireless Access Point* (WAP). Este dispositivo vincula dispositivos de comunicación inalámbricos para formar una red inalámbrica; es decir, aquellos dispositivos móviles como lo son las computadoras, tabletas y celulares. Normalmente un WAP también puede conectarse a una red cableada, y puede transmitir datos entre los dispositivos conectados a la red cableada y los dispositivos inalámbricos [43].

2.1.8. Computadora de placa única

Es una computadora que está embebida en una placa reducida, se le conoce como *Mini Single Board Computer*. Esta placa nos da las prestaciones que tiene una computadora de bajo costo en un tamaño pequeño y con un

costo accesible. La computadora mini se ha introducido versátilmente en la satisfacción de las personas que utilizan el medio electrónico para una resolución eficaz sin requerir un sitio amplio [47].

2.1.9. Internet de las cosas

El "Internet de las cosas", "Internet de los objetos" o en ingles *internet of Things (IoT)* , este concepto equivale a la alteración del vínculo con las cosas, a la conexión de piezas comunes con Internet. La esencia está en los instrumentos que transforman a los objetos, en objetos inteligentes, estos para que puedan funcionar a través del Internet, ofrecen a los usuarios de objetos inteligentes la oportunidad de recabar datos del mundo real [40].

2.1.10. API

Application Programming Interface (Interfaz de Programación de Aplicaciones) es un conjunto de servicios que es proporcionada por la biblioteca de programación, puede ser utilizada para mantener una comunicación entre Softwares [51].

2.1.11. Base de datos

Es un conjunto de datos relacionados que están agrupados y ordenados por un programa, el cual otorga un fácil acceso a los datos almacenados en ella [45].

2.1.12. Redes sociales

Son páginas de Internet que están conformadas por grupos de personas (usuarios) con intereses y actividades en común, tales como amistad, parentesco y trabajo que permiten el contacto entre ellos. De manera que se puedan comunicar e intercambiar información [33]. La estructura que presentan las distintas redes sociales permiten la interacción de los usuarios a través de la tecnología, es una intercomunicación social no física por medio de las páginas web.

2.2. Literatura relacionada

A continuación, se presentan trabajos anteriores relacionados y similares al propuesto.

En trabajos recientes acerca de Sistemas de Alerta Temprana (S.A.T.), los autores plantean arquitecturas de captura y procesamiento de variables físicas mediante placas de Hardware libre, dado que son versátiles. En las placas de Hardware libre se pueden desarrollar entornos interactivos como ver los datos captados, los cuales se podrán observar por medio de dispositivos electrónicos que estén en el rango de cobertura. Estos datos se transmiten a dispositivos capaces de mostrar las variaciones mediante algún protocolo de comunicación. El protocolo de comunicación Zigbee[1] se puede implementar ya que tiene un alcance de 1 km de distancia por cada nodo como lo expone [11]. Presenta una arquitectura basada en una red de sensores, comunicada a un servidor que transmite los datos a los dispositivos que están en el rango de cobertura. La arquitectura tiene una particularidad que es el obtener datos de precipitaciones atmosféricas.

Por otra parte, se puede realizar un sistema de alerta temprana con herramientas informáticas y simulaciones para corroborar si funcionan, como lo presenta [16], ellos adquieren y procesan los datos mediante simulaciones en tiempo real. Consideran 4 componentes principales: modelos de simulación, sistema de información geográfica, adquisición y supervisión de datos en tiempo (real y diferido) y una base de datos histórica actualizada. Su sistema interactúa con dos modelos de simulación: El HEC-HMS[2] que en este caso simula los procesos de lluvia y la respuesta hidrológica y el HEC-RAS[3] el cual realiza un estudio para saber qué tanto es el riesgo de sufrir una inundación en cierta área geográfica. La adquisición y supervisión de datos en tiempo (real y diferido) es realizada por medio de un sistema SCADA[4] en el caso de tiempo real ya que mediante sistema SCADA se puede visualizar, controlar y recopilar datos del proceso que se esté llevando a cabo. Por otro lado, del tiempo diferido se introdujeron valores aleatorios de manera manual al igual

[1]Es un protocolo de comunicación, que se utiliza para la transición de datos por medio de radiofrecuencia, basado en la norma IEEE 802.15.4.

[2]Es un simulador de procesos de lluvia-escurrimiento en sistemas dendríticos de cuencas.

[3]Es un Software de simulación que permite realizar estudios de Modelización de flujo en régimen permanente, Modelización de flujo en régimen no permanente, Modelización del trasporte de sedimentos, Análisis de calidad de aguas.

[4]SCADA: por sus siglas en ingles de supervisión, control y adquisición de datos.

que se emplearon registradores de variables.

En [18] exponen como medir el nivel del agua a estanques, de tal manera da la percepción de que, con cierto instrumento, en este caso un medidor ultrasónico, se ha podido realizar este tipo de mediciones. Su propuesta de tecnología lo conforma una unidad remota instalada en el sitio de interés y la unidad local dentro de un laboratorio cercano, ambas desarrolladas con el microcontrolador STAMP BS2p. Hoy en día se cuenta con tecnología más sofisticada que puede realizar la medición de igual o mejor forma, pero este trabajo hace ver que ya se han implementado sistemas de medición de niveles de agua con Hardware antiguo.

Una herramienta que se utilizará en el desarrollo del prototipo que se propone, es una computadora de placa única, específicamente la *Raspberry pi 3* . En [19] se demuestra que la Raspberry pi es muy versátil, su propuesta de monitoreo de ambiente mide qué tan limpio está el aire, variables del ambiente como temperatura, humedad, luminosidad, detecta la presencia de terremotos y las variables las obtiene mediante sensores. En cuanto al Hardware que emplean, en primer lugar, usan una Raspberry pi (no especifican cual) en cuanto, a los sensores, usan el TMP36. Este es un sensor capaz de detectar la temperatura de un ambiente, el DHT22 es un sensor que mide la humedad y aunado a ello también la temperatura relativa, una fotorresistencia el cual se encarga de obtener qué tanta luz hay presente en determinada área, mq7 es un sensor electroquímico que expuestos a cierto gas entrega el dato de qué tan contaminado está el aire y utilizan LDT0-028K el cual mide las vibraciones y así obtienen la información si hay o no un terremoto.

El protocolo de comunicación elegido para la elaboración de este prototipo es LoRa, debido que es un protocolo de comunicación inalámbrico que tiene un rango considerable de unos 15 km. En [37] presentan un desarrollo que integra este tipo de comunicación, y el objetivo de ellos fue diseñar una arquitectura para el control y monitoreo de la electricidad de los edificios. Pusieron a prueba el módulo RN2903 este es un módulo de 915MHz certificado, basado en tecnología inalámbrica LoRa, lo prueban a diferentes distancias: 1 m, 5m, 10 m, 20 m y 30 m, y cada una de estas pruebas enviaron 10 paquetes de datos y en ninguna de las distancias hubo pérdida de información. Pero si mencionan que utilizaron en otros entornos, tanto pueden afectar la transmisión a como les puede beneficiar dependiendo del área de transmisión y recepción de estos dispositivos.

Gracias al trabajo realizado por [6], se observa el potencial que tiene el protocolo de comunicación LoRa, debido a que realiza una serie de pruebas

para determinar el rango de pérdida de los paquetes de datos. Lleva a cabo sus pruebas en ambientes forestales a orilla de ríos diferentes entornos: urbano, semi urbano y rural y describe con detalle las características de lugar. Se llegó a la conclusión que el entorno le puede afectar o beneficiar dependiendo de las características del lugar y en sus resultados declara que no obtuvo una transferencia de datos a distancias mayores de 1600 metros.

Twitter será la red social que se utilizará para reportar los datos obtenidos por la arquitectura de S.A.T., tal y como fue utilizada por [9], donde utilizan esta red social para una propuesta orientada al área de seguridad. Su arquitectura se basa en notificar al propietario de una residencia, la presencia de un visitante mediante Twitter, de igual manera el usuario tiene control de la puerta así que puede abrirle la puerta o negarle el paso mediante un dispositivo, en este caso un Smartphone. Su arquitectura se basa en un sensor Pir, para detectar la presencia de un visitante; una cámara Wi-Fi, encargada de tomar una foto del individuo y una Raspberry pi, por la cual son recibidos los datos y publicados en twitter. Para que la *Raspberry pi* sea capaz de publicar en la red social es necesario instalar un programa llamado Twython, en el que la puerta tiene una cerradura eléctrica que puede controlarse mediante un *smartphone*.

En la propuesta de [39] se expone un sistema de monitoreo enfocado a inundaciones. Se utilizan redes de sensores distribuidos topográficamente a lo largo de un cuerpo de agua en un río, los sensores que se implementan en esta propuesta no son especificados, pero se menciona que está arquitectura mide variables tales como: nivel del agua del río, lluvia, datos de velocidad del viento y presión del aire. Una vez obtenidas estas variables se transmiten por medio de un Arduino a una*Raspberry pi*, esta comunicación es mediante el protocolo Xbee, una vez que obtienen los datos la Raspberry los almacena en una base de datos y se puede visualizar en una interfaz que presenta cada variable por un punto (sensor). Estos puntos pueden obtener diferentes colores dependiendo el rango de las variables; el color rojo significa que una variable específica está presente pero muy poco, el color azul significa que está un poco más presente, pero no tanto y el color gris es que está muy presente la variable en cierto momento.

En este artículo se prueba la comunicación LoRa en un espacio cerrado. En [29] se menciona que el interés de realizar pruebas a este protocolo de comunicación es debido a que está siendo muy llamativo para las empresas por sus características, las cuales son muy favorables en cuestión de rangos de transmisión. Su red está conformada por un puerto de enlace, un servidor,

y un dispositivo final. En sus experimentos cuenta que existe un "delay" entre transmisión, ya que el dispositivo de transmisión no puede seguir mandando paquetes de datos hasta que termine la recepción de los mismos y este tiempo es más o menos de 273 ms que es inducido por el enlace serial. De igual manera, en su conclusión menciona que mediante los experimentos, se observó que la velocidad de transmisión es demasiado rápida y se pueden perder paquetes de datos.

El interés de realizar el trabajo de investigación parte de [49], es continuación del desarrollo con cambios en el Hardware y en la red social que publica las alertas. Este artículo expone una arquitectura conformada por un Arduino y un sensor ultrasónico. La función del Arduino es el procesamiento de las variables obtenidas del nivel del agua, esto es posible debido a que es una plataforma electrónica de recursos libres, dicha plataforma obtiene los datos mediante el sensor ultrasónico Parallax PING. Este sensor es capaz de emitir ráfagas de sonido y cuando el sonido rebota en algún objeto (el agua), se conoce la distancia y ubicación del sensor, independientemente de la distancia en la que se encuentre el agua. Una vez que se obtienen los datos requeridos se publican en la red social Facebook mediante un Software que diseñaron en PHP y que es capaz de publicar.

2.3. Marco tecnológico

En este apartado se mencionan el Hardware y Software utilizado para el desarrollo de este prototipo.

2.3.1. Hardware

Sensor ultrasónico jsn-sr04t-2.0

Es un sensor que emite ondas ultrasónicas para determinar la distancia que hay de un objeto con respecto al sensor, la manera que obtiene la distancia es mediante la emisión de pulsos de sonidos (trig), se mide cuánto tiempo tardó en retornar el sonido emitido (echo) y el tiempo se obtiene mediante la diferencia de tiempo entre el (trig) y (echo), otra característica que tiene este sensor es que es resistente al agua [27], este sensor se ha empleado en trabajos como en [26] donde es utilizado para medir el nivel de agua, también describen muy bien el funcionamiento que tiene este sensor, en [10] el sensor

ultrasónico es empleado para obtener la dimensión de paquetes en cuestión de altura.

TTGO LoRa32

Es una tarjeta programable que tiene embebido un chip ESP32. Dispone de una pantalla oled y un módulo de radio el cual implementa el protocolo de comunicación LoRa [7], es utilizada por [36] y [2], quienes la implementan para la transferencia de datos hacia un *gateway* para subir los datos a los servidores de TheThingsNetwork.org, los datos se transmiten por medio de tecnología LoRa en los dos casos.

Raspberry pi

Es una computadora de placa única, que cuenta con los componentes de computadora convencional, esta idea surgió de la necesidad de utilizar Hardware libre y de bajo costo en las escuelas y de esa manera incentivar la informática a los estudiantes. En la actualidad se pueden encontrar en distintas fuentes proyectos que se han llevado a cabo empleando alguna de las versiones de la Raspberry pi ya que la placa cuenta con una gran comunidad [13], es empleada para el desarrollo tecnológico por su versatilidad y, en [28] es ocupada para guardar datos del estado del suelo, los cuales son trasmitidos desde los nodos a la Raspberry pi mediante MQTT y los autores de [35] un sistema de alerta de incendios, por medio de procesamiento de imágenes utilizando la Raspberry pi.

2.3.2. Software

IDE Arduino

El entorno de desarrollo integrado *(Integrated Development Environment)* de Arduino, es un programa que integra herramientas de programación como: un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, en este entorno de programación se pueden utilizar lenguajes como c y c++ [4].

Raspbian

Es un sistema operativo para raspberry pi (se halla perfecto para su hardware) se fundamenta en la partición de GNU/Linux nombrada debian[41].

C++

Es un lenguaje de programación planteado por Bjarne Straustrop, su ingenio fue para expandir el lenguaje de programación C, de tal manera que contará con mecanismos que accedieran a la utilización de objetos[25].

Node Red

Es una herramienta de programación, se utiliza para relacionar: Hardware, API y servicios en línea. La forma de programación de ésta herramienta está basada en flujos mediante nodos en los que se ejecutan funciones dependiendo de lo que el programador desee [30], en [22] es empleado para implementar un flujo de dato para procesar la información captada por el sensor DHT11 y en [34] es utilizado para comunicar tarjetas programables con dispositivos inteligentes, ellos proponen la complementación de Alexa[5] para activar microcontroladores mediante Node Red.

InfluxDB

Es una base de datos de estándares libres, capaz de proporcionar consultas y escrituras de los datos mediante un lenguaje llamado InfluxQl[21].

Chronograf

Es el tablero que permite ver por medio de gráficos los datos almacenados en InfluxDB[20].

Twitter

Es una red social en la cual los usuarios pueden publicar mensajes cortos llamados tweets, estos tweets están limitados a 140 caracteres, los usuarios de Twitter pueden enviar, recibir y seguir los tweets mediante diferentes dispositivos como *smartphone*, computadoras de escritorio, laptops, tablets.

[5]Es un asistente virtual desarrollado por Amazon.

La API está abierta y disponible para desarrolladores de aplicaciones [38], esta api tiene variantes y se pueden emplear como en [9] que es enfocado a la seguridad.

MQTT

Es un protocolo de comunicación de usuarios con fundamento en difusión e inscripción a los designados tópicos. El protocolo MQTT actúa sobe TCP/IP o encima de distintos protocolos de red con base bidireccional y sin extravío de datos[44], este protocolo se utilizó en [3] para transferir los datos de los nodos sensores hacia una raspberry pi, la cual almacena los datos y los muestra en un dashboard mediante Node-Red.

Capítulo 3

Diseño arquitectónico

En este capítulo se muestra de manera gráfica la arquitectura del prototipo, por medio de dos diagramas, el primero 3.1 muestra de manera simbólica los elementos involucrados en el proyecto y se explica el trayecto del dato de nivel de agua a través del prototipo, el segundo 3.2 es un diagrama esquemático de los elementos electrónicos del proyecto y de cómo se enlazan entre ellos, también se explica cómo está conformada la arquitectura y cómo fueron configurados los elementos de Hardware (microcontroladores, sensores y computadora de placa única) y Software (API, interfaces y aplicaciones).

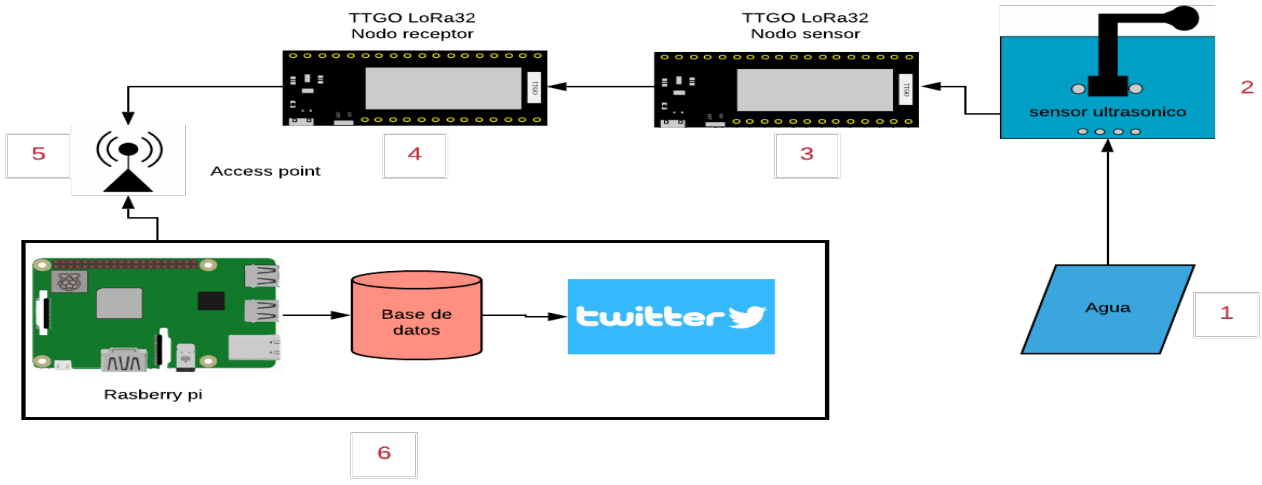

Figura 3.1. Diagrama de la arquitectura.

En la siguiente lista se describe a los elementos de la figura 3.1 cómo están involucrados cada uno de los componentes.

1. La variable de entrada es el nivel del agua de algún río, pero las pruebas para el prototipo se realizaron en un contenedor de agua.

2. El nivel es medido por el sensor ultrasónico jsn-sr04t-2.0, el cual tiene la particularidad que es a prueba de agua, este se ubicaría al borde del río.

3. El sensor mencionado está conectado a una tarjeta programable TTGO LoRa32, esta funciona como un nodo sensor y envía los datos obtenidos a un receptor por medio del protocolo de comunicación LoRa.

4. El nodo receptor es otra tarjeta TTGO LoRa32, de manera que recibe los datos provenientes del nodo sensor y transfiere los datos a la computadora de placa única (Raspberry pi).

5. El Access point hace posible la vinculación del nodo receptor y de la Raspberry pi.

6. La Raspberry pi funciona como sistema informático por lo que obtiene los datos mediante el Access point, registra la información en una base de datos y emite una alerta en la red social (Twitter).

En la figura 3.2 se muestra el diagrama esquemático de la arquitectura propuesta del prototipo. Tomando en cuenta el diagrama de derecha a izquierda se puede apreciar lo siguiente.

1. El sensor ultrasónico está conectado a la tarjeta TTGO LoRa32, el ping GND del sensor está conectado al pin GND de la tarjeta (línea negra), el pin de vcc (voltaje de corriente directa) del sensor está conectado al pin de 5 volts de la tarjeta (línea rojo), el pin de Trig y de Echo del sensor ultrasónido se conecta a conveniencia del programador, en este caso se selecciona el pin 33 de la tarjeta para el pin trig (línea verde), y el pin 32 de la tarjeta para el pin echo del sensor (línea azul), la conjunción del sensor con la tarjeta forman el nodo sensor.

2. El nodo sensor transmite los datos obtenidos hacia el nodo receptor[1] por medio del protocolo de comunicación LoRa.

[1]Tarjeta TTGO LoRa32 configurada para recibir los datos del nodo sensor.

3. El nodo receptor a su vez está vinculado con la Raspberry pi mediante el Acces point, vía Wi-Fi.

4. La Raspberry pi recibe los datos provenientes del nodo receptor para posteriormente guardarlos en una base de datos y publicarlos en Twitter de forma automática.

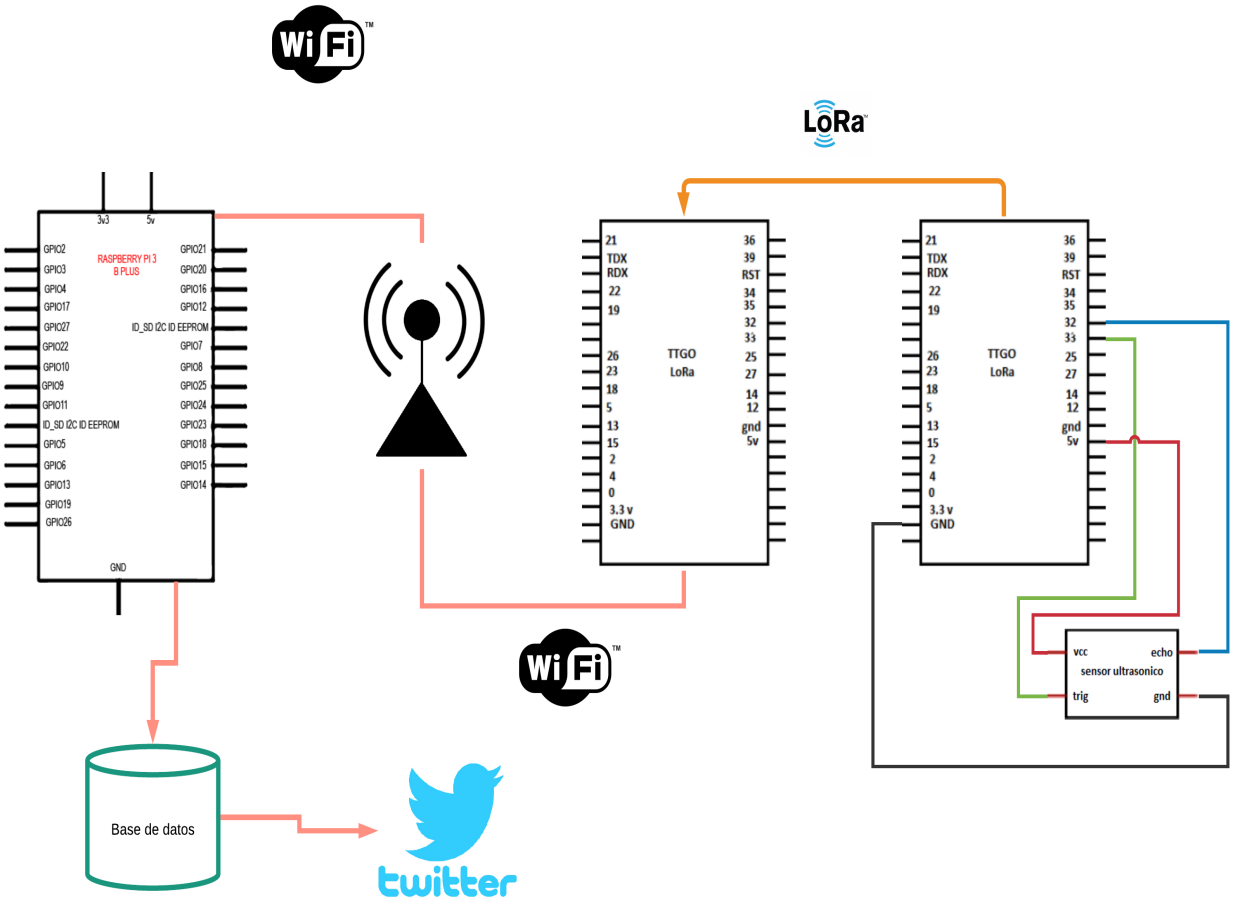

Figura 3.2. Diagrama esquemático.

3.1. Nodo sensor

El nodo sensor está conformado por una tarjeta TTGO LoRa32 y un sensor ultrasónico jsn-sr04t-2.0, el encargado de tomar la medición del nivel de agua, para su posterior transmisión.

3.1.1. Configuración

Para realizar el denominado nodo sensor se realizan las conexiones de los pines como: pin de tierra del sensor (se conecta en cualquier pin de tierra de la tarjeta TTGO LoRa32), el pin de 5 volts del sensor va conectado en el pin de 5 volts de la tarjeta TTGO LoRa32, el pin de Trig del sensor se configura el pin 33 de la tarjeta y el pin de Echo del sensor va conectado en el pin 32 de la tarjeta (configurado en la programación).

En la figura 3.3 se puede apreciar de manera gráfica las conexiones realizadas entre el sensor ultrasónico y la tarjeta TTGO LoRa32.

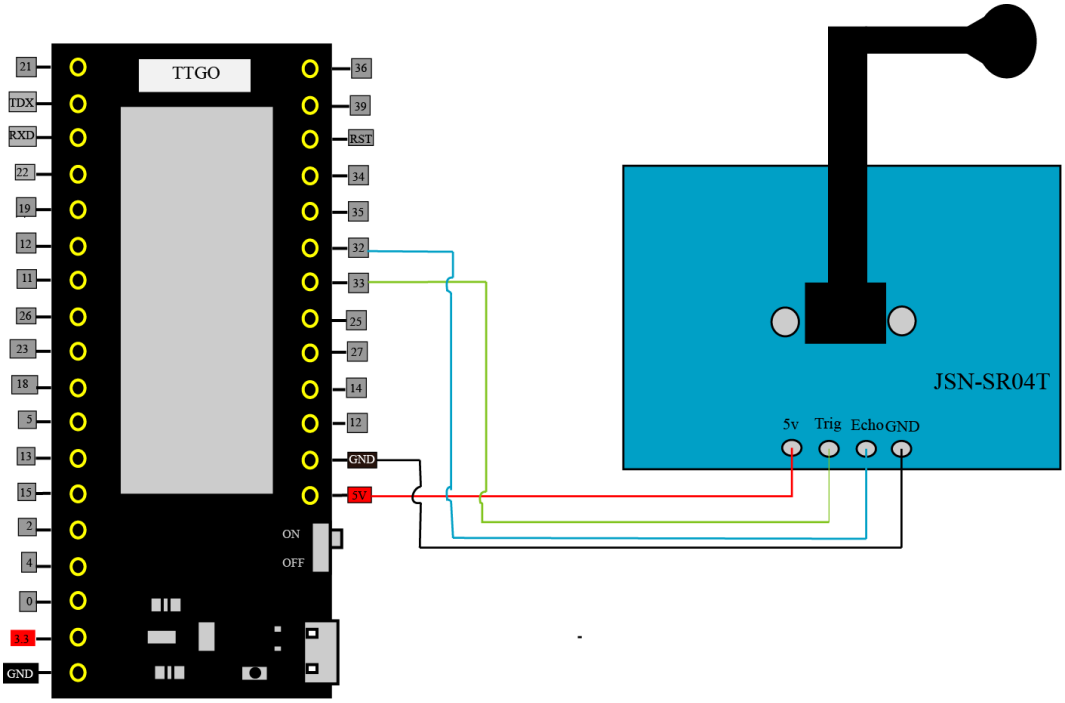

Figura 3.3. Diagrama de conexiones del nodo sensor.

- EL pin de GND (TIERRA)[2] del sensor ultrasónico está conectado al pin GND de la tarjeta TTGO LoRa, se puede apreciar la conexión en el diagrama mediante la línea negra.
- El pin Echo[3] del sensor va conectado al pin 32 de la tarjeta, estos pines se puede ver con la línea azul que los entrelaza.
- El pin Trig[4] del sensor está conectado al pin 33 de la tarjeta, ésta conexión se puede ver con la línea verde en el diagrama.
- El pin de 5 volts del sensor está conectado a la salida de 5 volts[5] de la tarjeta, la unión de estos pines se representa mediante la línea roja.

En la programación de la tarjeta TTGO LoRa32, se utiliza el entorno de programación de Arduino[6] y se establecen los parámetros para el nivel de medición del agua y configuración del sensor ultrasónico de manera óptima dependiendo la necesidad. Para programar la tarjeta se debe descargar el Core (núcleo) de la tarjeta TTGO LoRa32 en la IDE de Arduino.

Los núcleos son necesarios para que los nuevos microcontroladores sean compatibles con el software Arduino (IDE), por ello se descarga e instala núcleo de terceros ya que la TTGO LoRa32 es independiente de Arduino.

Para descargar e instalar en la IDE de Arduino el módulo ESP32 el cual incorpora la tarjeta TTGO LoRa32, se realizaron los siguientes pasos:

Para empezar, se necesita tener instalada la última versión del entorno de desarrollo de Arduino, se puede descargar de la página oficial de Arduino https://www.arduino.cc/en/Main/Software.

Para facilitar la descarga de los archivos que se requieren para que la tarjeta TTGO sea compatible con el entorno de desarrollo de Arduino, un gestor de descarga llamado GIT es necesario, una vez instalado se debe iniciar GIT GUI, una vez en el gestor seleccionar la opción Clone Existing Repository, la opción a elegir se muestra en la figura 3.4.

[2]Significa tierra y sirve como protección de una posible sobrecarga (0 volts).

[3]La función del pin Echo se explican en el apartado de Hardware en la sección llamada Sensor ultrasónico jsn-sr04-2.0.

[4]La función del pin Trig se explican en el apartado de Hardware en la sección llamada Sensor ultrasónico jsn-sr04-2.0.

[5]El pin de salida de 5 volts de la tarjeta, sirve para suministrar los 5 volts que requiere el sensor para funcionar.

[6]Es una plataforma de hardware (tarjetas programables) y software libe (IDE Arduino)[5]

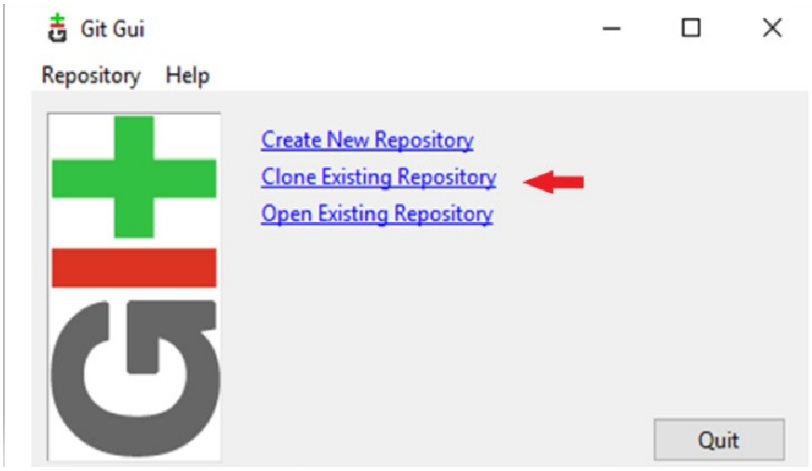

Figura 3.4. Gestor de descarga (Recuperado de Git Gui).

Una vez seleccionada la opción de *Clone Existing Repository*, se abre una ventana con dos opciones y un campo de búsqueda. La primera opción es la de *Source Location* (fuente para descargar los archivos requeridos) y la segunda es *Target Directory* (dirección en la que se desea copiar los archivos).

La fuente de los archivos es la siguiente: https://github.com/espressif/arduino-esp32 y los archivos descargados de esta fuente se guardaran en la siguiente dirección dentro de la carpeta de la IDE de Arduino: C:\Users \(USUARIO) \Documents \Arduino \hardware \espressif \esp32, una vez llenado los campos de búsqueda se da clic en Clone, como se puede ver en la figura 3.5.

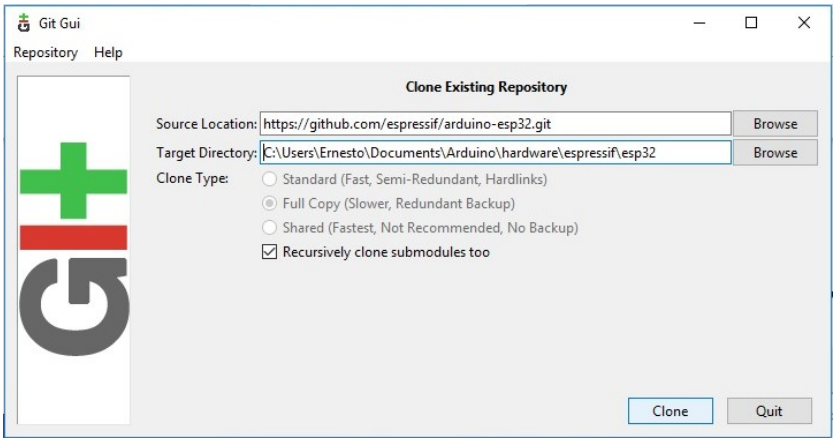

Figura 3.5. Git Gui, Source Location y Target Directory (Recuperado de Git Gui).

Una vez finalizado lo anterior hay que dirigirse a la siguiente dirección: C:\Users \(USUARIO) \Documents \Arduino \hardware \espressif \tools, donde se ubica un gestor llamado get ya que este permitirá la descarga de las bibliotecas del esp32, figura 3.6

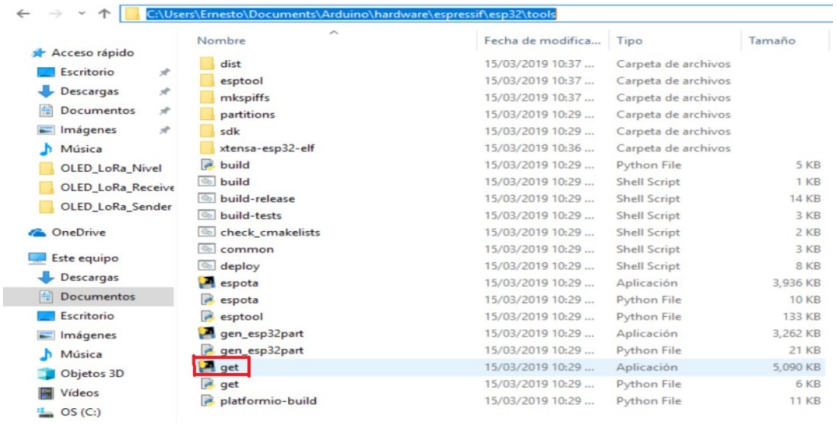

Figura 3.6. Dirección del gestor get (Recuperado de windows 10).

Después de haber hecho exitosamente lo anterior, se podrá emplear la IDE de Arduino para programar el esp32, para configurar la placa en la IDE de Arduino se tienen que seguir unos sencillos pasos que son los siguientes: primero conectar en esp32 a la computadora, una vez que la computadora haya reconocido el esp32, se ejecuta la IDE de Arduino donde se selecciona la tarjeta en la pestaña de herramientas como se muestra en la siguiente figura 3.7.

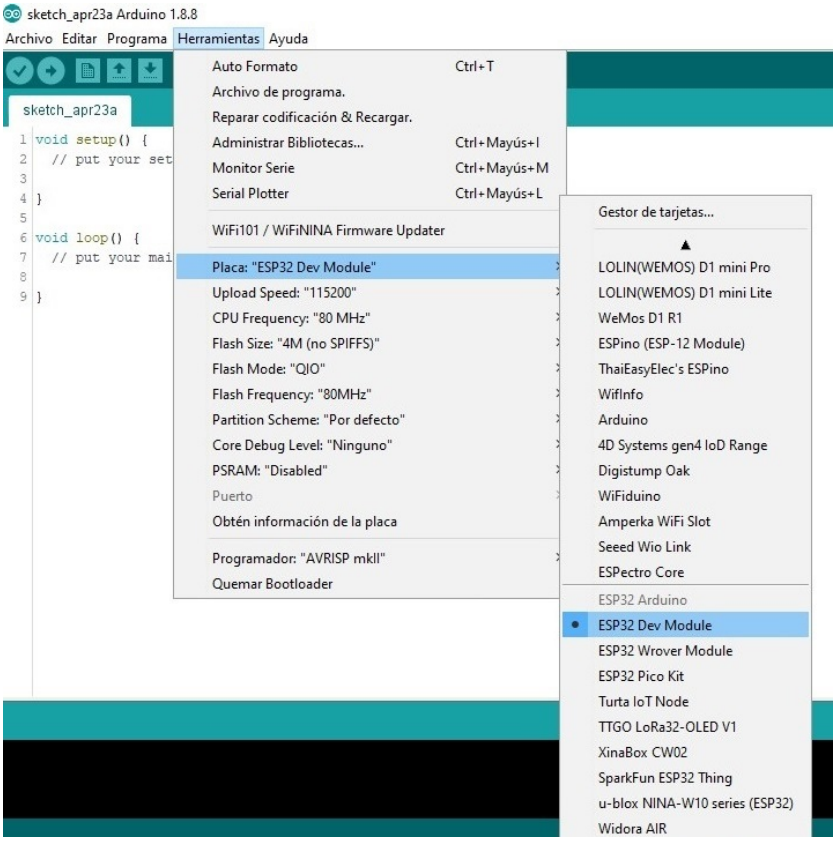

Figura 3.7. Selección de tarjeta (Recuperado de Arduino IDE).

3.1.2. Programación

Como ya se mencionó, se emplea el entorno de desarrollo de Arduino para programar el nodo sensor. El código de programación del proyecto, toma como base el código proporcionado por el proveedor de la tarjeta TTGO LoRa32, dicho código servía para corroborar que la tarjeta transmite de manera óptima mediante el protocolo LoRa. El código fue modificado agregándole líneas para la obtención del nivel del agua, configuración del sensor y de igual forma para agregarle un nivel de seguridad.

En la programación del nodo sensor se utilizan diferentes librerías que permiten un correcto funcionamiento del display que tiene integrado la tarjeta de la trasmisión LoRa y para encriptar los mensajes trasmitidos del nodo sensor hacia el nodo receptor.

En el fragmento 1 se muestra el listado de librerías necesarias. En línea 1 a la 5 son las librerías que incluía el pseudocódigo, código proporcionado por el distribuidor, la línea 6 fue agregada para proporcionar la funcionalidad de encriptación en el prototipo e incluye la llave de desencriptado.

```
1 #include <SPI.h>
2 #include <LoRa.h>
3 #include <Wire.h>
4 #include "SSD1306.h"
5 #include "images.h"
6 #include <xtea-iot-crypt.h>
```

Listing 1. Librerías del nodo sensor

En el fragmento 2 del código, se definen ciertas constantes como algunos pines que se emplean en el del protocolo *Serial Peripheral Interface (SPI)*[7] el funcionamiento del botón de reset, igual se asigna un pin al IQR. Para el funcionamiento del chip SX1278 cuenta con el módulo del protocolo de comunicación LoRa. También se define en qué frecuencia se transmitirá la información por medio de LoRa, esto depende de la región donde se empleará el prototipo, en este caso es en México y la frecuencia libre de trasmisión es la 915 MHz.

```
1 #define SCK     5    // GPIO5  -- SX1278's SCK
2 #define MISO    19   // GPIO19 -- SX1278's MISnO
3 #define MOSI    27   // GPIO27 -- SX1278's MOSI
4 #define SS      18   // GPIO18 -- SX1278's CS
5 #define RST     14   // GPIO14 -- SX1278's RESET
6 #define DIO     26   // GPIO26 -- SX1278's IRQ(Interrupt Request)
7 #define BAND  868E6  // Transmission band
```

Listing 2. Constantes definidas para el chip SX1278 del nodo sensor

También es necesario definir los pines que se emplearán para la entrada y salida que tendrá el sensor, en este caso la salida es el disparador (trigger) el

[7]Es un protocolo de datos en serie síncrono utilizado por microcontroladores para comunicarse rápidamente con uno o más dispositivos periféricos en distancias cortas.

cual emitirá las ondas ultrasónicas y la entrada es el eco (echo) que se podría definir como el retorno de las ondas ultrasónicas, después de haber rebotado con algún objeto como la superficie del agua.

En el fragmento 3 se puede visualizar los pines de la tarjeta que se emplearán para la conexión de los pines trig y echo del sensor ultrasónico y en el fragmento 4 se declara si los pines funcionan como una entrada del sensor o una salida de éste.

```
1 const int trigPin = 32;
2 const int echoPin = 33;
```

Listing 3. Pines definidos de la tarjeta para el sensor

```
1  // Sensor ultrasonico
2   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
3   pinMode(echoPin, INPUT); // Sets the echoPin as an Input
```

Listing 4. Declaración entrada y salida del sensor.

Gracias a la librera xxtea-iot-crypt se pueden encriptar los mensajes que se envían hacia el nodo receptor, para desencriptar los mensajes se tiene que declarar una llave definida por el desarrollador (Proyecto ujat), presentada 5

```
1  // Key for Encrypt - ! Carefull no to more than 16 bytes !
2 uint8_t keybuf[] = "ProyectoUJAT";
```

Listing 5. Llave para desencriptar los mensajes.

Teniendo las constantes antes mencionadas, se puede inicializar la comunicación LoRa, en la programación proporcionada por el distribuidor, primero se inicializa la comunicación y después se realiza una prueba, para corroborar que se logró establecer la comunicación, ésta se declara en las líneas de código que se muestran en el fragmento 6 , esa parte del código se respetó ya que se consideró que era bueno evaluar si se lograba la comunicación LoRa.

```
1   Serial.begin(115200);
2   while (!Serial);
3   Serial.println();
4   Serial.println("LoRa Sender Test");
5
6   SPI.begin(SCK,MISO,MOSI,SS);
7   LoRa.setPins(SS,RST,DIO);
8   if (!LoRa.begin(915E6)) {
9     Serial.println("Fallo la inicializacion de LoRa!");
10    while (1);
11  };
```

Listing 6. Inicialización LoRa.

Una vez finalizados los procesos de la adquisición y procesamiento de los datos obtenidos del sensor ultrasónico, la encriptación del mensaje a enviar al nodo receptor e iniciación del protocolo LoRa, se transfiere el mensaje encriptado con los datos procesados hacia nodo receptor por medio de LoRa

3.2. Nodo receptor

Este nodo recibe los datos provenientes del nodo sensor, recibe la información por medio de un mensaje encriptado, mediante el protocolo de comunicación LoRa. Una vez que el nodo recibe el mensaje lo desencripta mediante una llave única, después comparte la información con la *Raspberry pi*.

3.2.1. Configuración

Este nodo no se le agrega un Hardware extra para su configuración, únicamente es necesario conectar la antena que trae la tarjeta TTGO LoRa32 antes de conectarlo a alguna fuente de alimentación. Al igual que el nodo sensor, se tienen que hacer las mismas configuraciones en la IDE de Arduino, como se puede observar en la figura 3.7.

3.2.2. Programación

En las primeras líneas de código (al igual que en el caso del nodo sensor) está ubicado el listado de librerías, la mayoría son las mismas implementadas en el código del nodo receptor con dos adiciones este listado, como se aprecia en el fragmento 7. Las librerías adicionales son *WiFi* y *PubSubClient*, las cuales ayudarán a realizar la conexión a una red de internet y la otra a implementar la comunicación MQTT.

```
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3
4  #include <SPI.h>
5  #include <LoRa.h>
6  #include <Wire.h>
7  #include "SSD1306.h"
8  #include "images.h"
9  #include <xxtea-iot-crypt.h>
```

Listing 7. Librerías nodo receptor.

En las líneas de código mostradas en el fragmento 8 tenemos contantes que son iguales a las del nodo sensor, por lo que se emplean en el funcionamiento del módulo SX1278 de la tarjeta y para la comunicación LoRa, de igual manera en este apartado se declara la banda de transmisión. Se puede ver que de la línea 1 a la 7.

```
1  #define SCK     5    // GPIO5  -- SX1278's SCK
2  #define MISO    19   // GPIO19 -- SX1278's MISO
3  #define MOSI    27   // GPIO27 -- SX1278's MOSI
4  #define SS      18   // GPIO18 -- SX1278's CS
5  #define RST     14   // GPIO14 -- SX1278's RESET
6  #define DIO     26   // GPIO26 -- SX1278's IRQ(Interrupt Request)
7  #define BAND    868E6  //Transmission band
```

Listing 8. Constantes definidas para el chip SX1278 del nodo receptor.

```
1 const char* ssid = "*********"; // Name of the internet
2 const char* password = "*********"; // Password of the internet
3
4 // Add your MQTT Broker IP address, example:
5 const char* mqtt_server = "*******";
```

Listing 9. Declaración de red y IP para la el protocolo MQTT.

En las líneas 1 y 2 se hace la comunicación con la red de internet, en la línea 1 se escribe el nombre de la red a la cual se desea hacer la conexión y en la línea 2 se escribe la contraseña de la red para tener acceso a ella (fragmento 9).

En la línea 5, fragmento 9, se escribe la dirección IP del Broker de MQTT. Esta dirección permitirá la transferencia de datos entre el nodo receptor y la Raspberry pi, ya que esta dirección IP es proporcionada por la Raspberry pi al inicializar el protocolo de comunicación MQTT igual conocido.

En el fragmento 9 se aprecia la llave de desencriptación del mensaje recibido proveniente del nodo sensor.

```
1 uint8_t keybuf[] = "ProyectoUJAT"; // Key for the decrytion
```

Listing 10. Llave para desencriptación.

En el fragmento 10 se visualiza la declaración del cliente WiFi en función de la librería de PubSubClient, que es necesario para la comunicación mediante el protocolo MQTT.

```
1 WiFiClient espClient;
2 PubSubClient client(espClient);
```

Listing 11. Declaración de cliente WiFi.

La recepción del mensaje y procesamiento del mensaje se pueden apreciar en la figura 11. En esta parte se recibe el mensaje proveniente del nodo sensor en forma de array, ya que de esa manera manda el mensaje encriptado el nodo

sensor para posteriormente ser desencriptado y poder ver los datos en texto plano.

```
1  // LoRa message processing
2  void cbk(int packetSize) {
3    char ch;
4    packet = "";
5    uint8_t plaintext[packetSize];
6
7    packSize = String(packetSize,DEC);
8    for (int i = 0; i < packetSize; i++) {
9      ch = (char) LoRa.read();
10     if(ch != '0'){
11       packet += ch;
12       buffer[i] =  ch;
13       plaintext[i] =  ch;
14     }
15   }
```

Listing 12. Recepción y procesamiento del mensaje por LoRa.

Las líneas para realizar la conexión a la red se pueden ver en el fragmento 12, empieza imprimiendo el nombre de la red, después en la línea 8 es donde se inicializa la conexión, se corrobora la red a la cual se desea conectar y valida la contraseña si es correcta, después se ejecuta un bucle while donde imprime puntos hasta que la conexión se haya realizado exitosamente y finalmente imprime la dirección IP.

```
void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Listing 13. Conexion a la red.

Una vez que el programa haya realizado los procesos de inicialización de las librerías y recepción mensaje encriptado. Se realiza el desencriptado del plaintext recibido mediante una sentencia if/else, de tal manera que el mensaje no sea desencriptado no continúe el código, una vez que el mensaje haya sido desencriptado se llama a la función procesa mensaje, donde se prepara la comunicación del protocolo MQTT (fragmento 13).

```
// Perform Decryption
  if(xxtea_decrypt(plaintext, packetSize) != XXTEA_STATUS_SUCCESS) {
    Serial.println(" Decryption Failed!");
    return;
  } else {
    Serial.print(" Desencriptando: ");
    Serial.println((char *)plaintext);
  }
  packet = (char *)plaintext;
  loraData();  //VMostrar en lcd
  procesaMensaje(packet); // enviar por wifi al mosquitto
}

void procesaMensaje(String readString){
  String sensor; //data String
  String nivel;
  String temperatura;

  int ind1; // , locations
  int ind2;
  int ind3;
```

Listing 14. Desencriptación del mensaje.

3.3. Access Point

El dispositivo utilizado como punto de acceso es un *smartphone*, específicamente un Huawei p10 lite. El motivo de la elección es por la comodidad al realizar las pruebas de comunicación, ya que por medio del *smartphone* es posible acceder a la red 4g de la compañía móvil gracias a los datos de navegación, lo cual ayuda a vincular el nodo receptor con Node Red mediante la red móvil, aunque se puede emplear cualquier dispositivo que dé acceso a Internet.

3.3.1. Configuración

Para compartir la red móvil con el nodo receptor y la Raspberry pi, es necesario activar la zona Wi-Fi del dispositivo, conocer el nombre de la red y la contraseña de la misma red. En este proyecto se emplea un *smartphone* se accede al apartado de ajustes y se selecciona la opción de conexiones inalámbricas y redes, donde se encuentra y se configura el nombre de la red 3.8 y la contraseña 3.9.

Figura 3.8. Nombre de la red Wi-Fi (Recuperado EMUI 8.0.0).

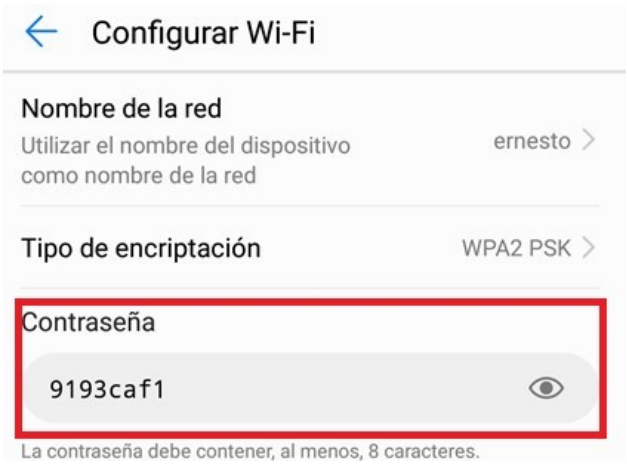

Figura 3.9. Contraseña de la red Wi-Fi (Recuperado EMUI 8.0.0).

Para realizar la conexión del nodo receptor, hay que escribir el nombre de la red y la contraseña en las líneas declaradas para ello, tomando de

referencia el fragmento de 8, en la líneas 9 y 10 se declara el nombre de la red y la contraseña respectivamente, en el caso de la Raspberry pi se realiza la conexión como si fuera una computadora común.

3.4. Raspberry pi

En esta sección se menciona cómo se obtuvo la API de Twitter, cómo fue configurado la Raspberry pi para mostrar, almacenar en una base de datos y publicar en Twitter los datos obtenidos por el nodo sensor.

3.4.1. API de Twitter

Es necesario obtener permiso de Twitter para emplear su API[8], lograr una comunicación entre la Raspberry pi y una cuenta de Twitter, se generó una cuenta específica para el proyecto.

Las personas interesadas en obtener la API de Twitter deberá solicitar una cuenta de desarrollador, una vez que Twitter aprueba la cuenta podrá emplear la API estándar o la API premium, por lo que para este proyecto se está empleando la API estándar.

Para solicitar una cuenta de desarrollador, vaya a http://developer.twitter.com e inicie sesión con su cuenta de Twitter. Después de haber seguido las instrucciones se deberá esperar para obtener la autorización de la API por parte de Twitter.

Después de haber sido autorizado, se puede crear la aplicación que nos proporcionará las claves API y los tokens de acceso que necesita usar en la aplicación (el flujo de twitter de Node-RED).

En la siguiente figura se puede apreciar la ventana donde se proporciona las claves 3.10, empleadas en el nodo de Twitter en Node-Red.

[8]Es un conjunto de funciones previamente implementadas que brindan al programador una interfaz a través de la cual comunicarse con un sistema determinado, añadiéndole nuevas funcionalidades.

Figura 3.10. Claves Api y tokens (Recuperado de la pagina de Twitter).

3.4.2. Node-Red

Node-Red es una herramienta que sirve para comunicar Hardware con servicios tecnológicos, su programación es visual y su estructura es mediante nodos que tiene una funcionalidad específica, estos nodos se conectan dependiendo de lo que el programador requiera, formando un flujo de programación en la figura 3.11 se puede visualizar el flujo de programación del proyecto.

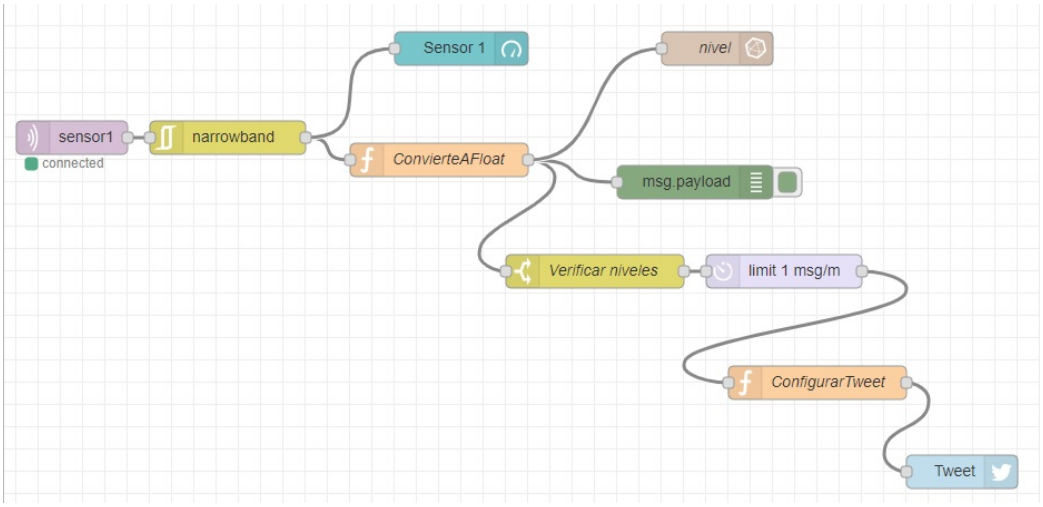

Figura 3.11. Flujo de programación del proyecto(Recuperado de Node-Red).

A continuación se presentan los nodos que conforman el flujo de programación.

Nodo mqtt *in*

Este nodo es utilizado para que se haga la transferencia del dato del nivel de agua desde el nodo receptor hacia la Raspberry pi, en este nodo se tiene que configurar el *broker mqtt* y el *topic*.

Al tratarse de un nodo de entrada, lo que hace es recibir los mensajes publicados en el *topic* que se declaró en el nodo receptor que este caso es sensor1, para suscribirse al *topic* hay que configurar el nodo, esto se hace dándole doble clic al nodo y así se abrirá el panel de configuración, una vez ahí hay que configurar el *mqtt broker* dándole clic en el lápiz 3.12.

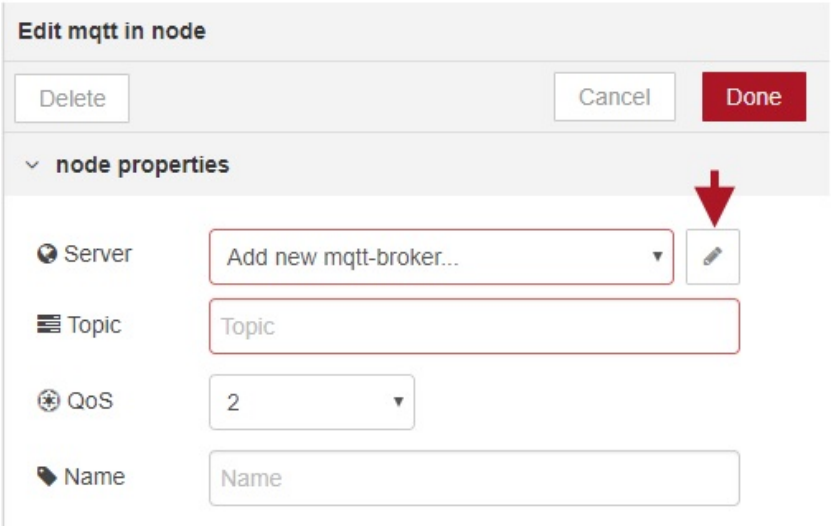

Figura 3.12. Panel de configuración del nodo mqtt in (Recuperado de Node-Red).

Esto abre un nuevo panel de configuración 3.13, donde nombrar el *mqtt server* y poner la dirección IP donde está instalado el *broker mqtt* y el puerto que emplea normalmente es el 1883, en el apartado de *server* en vez de poner la dirección IP establecemos *localhost*, ya que está instalado en el mismo servidor.

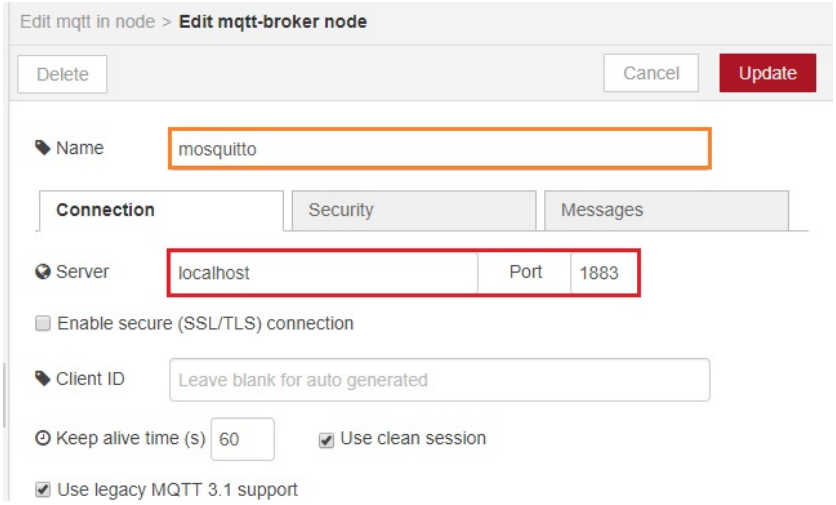

Figura 3.13. Panel de configuración del servidor mqtt (Recuperado de Node-Red).

Por último hay que rellenar le campo de *topic* 3.14, en este apartado se pone el que se declaró en el nodo receptor el cual es sensor1 y darle clic en done, eso sería lo esencial para la configuración del nodo *mqtt in*.

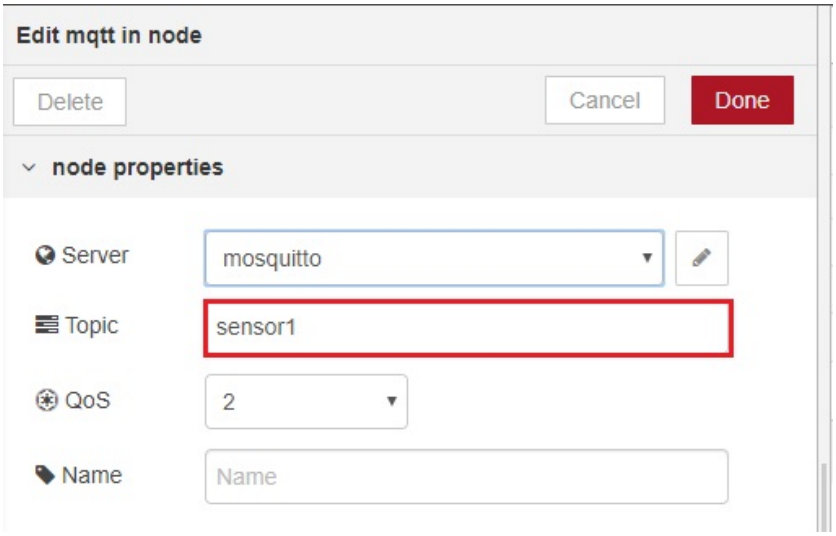

Figura 3.14. Llenado del campo de *topic* (Recuperado de Node-Red).

Nodo *narrowband*

Este nodo funciona como un filtro, ya que bloquea el dato recibido dependiendo de la configuración de éste, para su configuración se selecciona el modo que se requiere y darle ciertas condiciones, en este caso se seleccionó el modo *if value change is greater or equal to*, después se asigna un porcentaje de cambio aceptable y se selecciona el dato para comparar con el último dato que se ingresa en el nodo (figura 3.15).

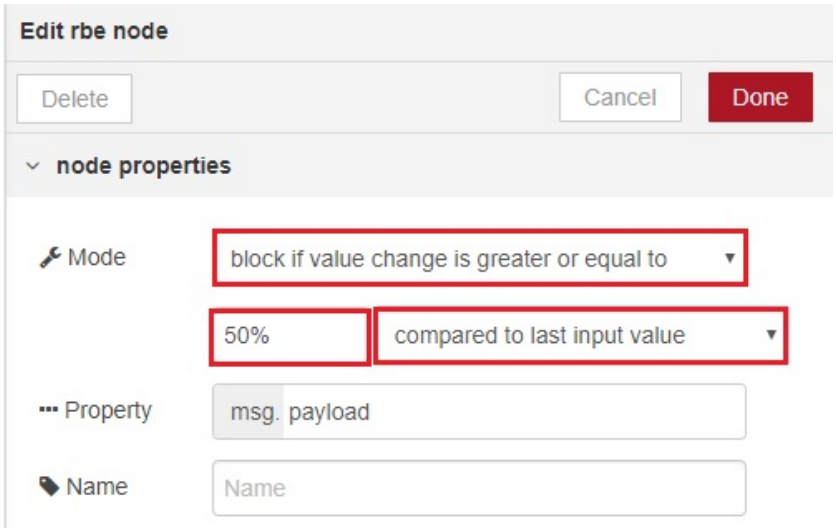

Figura 3.15. Configuración del Nodo *narrowband* (Recuperado de Node-Red).

Nodo *gauge*

Es el nodo que genera un *dashboard*, muestra los datos en tiempo real, requiere seleccionar el grupo al cual pertenezca, el tamaño que tendrá el display, seleccionar el tipo de display (para el proyecto se seleccionó un display de nivel), se le asigna un nombre al display, se selecciona la unidad de medida del display y por último se asigna el rango mínimo y máximo que tendrá el display (figura 3.16).

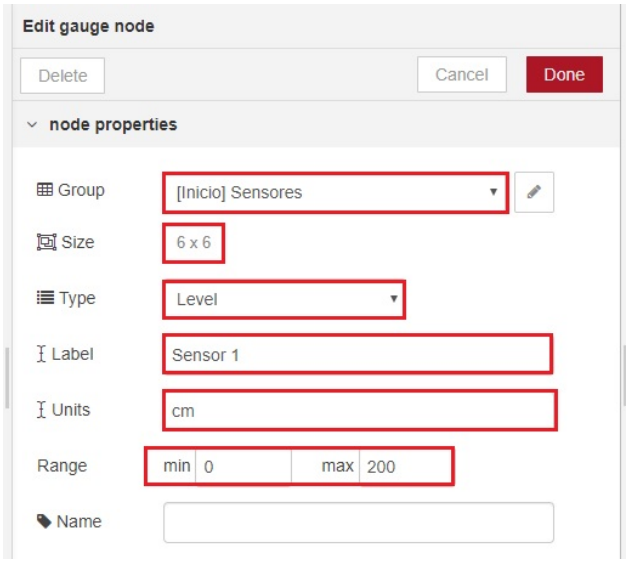

Figura 3.16. Panel de configuración del Nodo *gauge* (Recuperado de Node-Red).

Nodo *funtion*

Se usa para ejecutar código JavaScript con el objeto, este nodo se programó para volver la carga útil (msg.payload) tipo cadena en un número de punto flotante (figura 3.17).

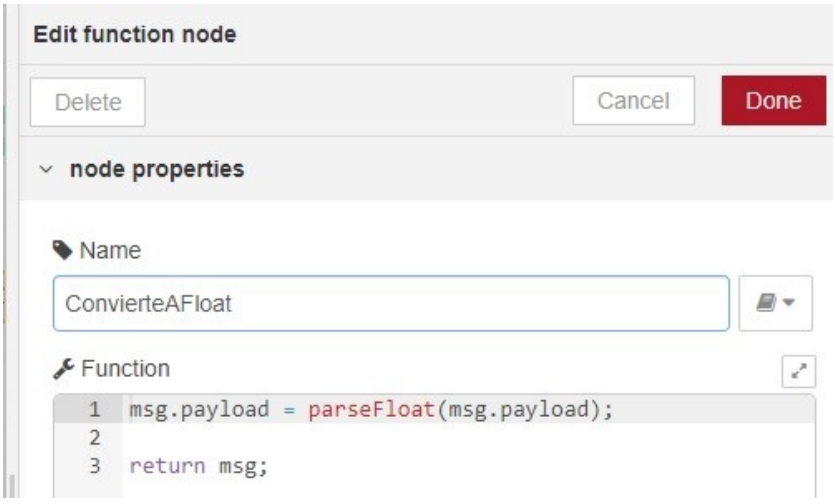

Figura 3.17. Programación del Nodo función (Recuperado de Node-Red).

Nodo *debug*

El nodo de depuración hace que se muestre cualquier mensaje en la barra lateral de depuración. Por defecto, solo muestra la carga útil del mensaje, este nodo viene configurado por defecto (figura 3.18), el cual en este caso muestra el dato del nivel de agua que envía el nodo receptor.

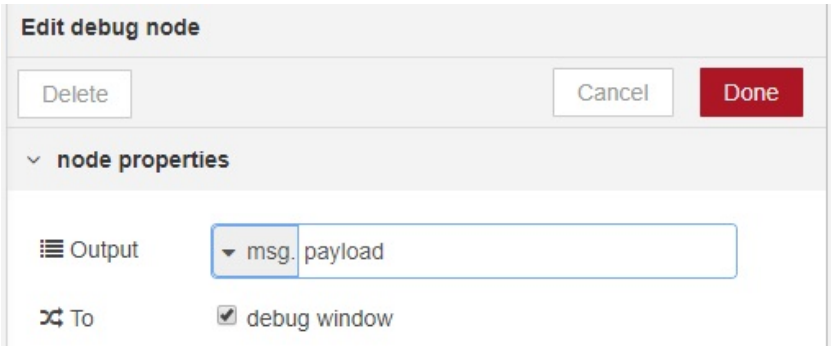

Figura 3.18. Panel del Nodo *debug* (Recuperado de Node-Red).

Nodo influxdb

Antes de poder ubicar este nodo en el flujo de programación hay que descargarlo, para ello hay que dar clic en la esquina superior derecha de Node-Red, hay que seleccionar la opción de *manage palette* (figura 3.19), posteriormente se abrirá una ventana donde hay que dar clic en la opción que dice *install* y en buscador se escribe el nombre del nodo: node-red-contrid-inflxdb y se da clic en *install* (figura 3.20), una vez finalizada la instalación se encuentra este nodo junto con los demás en el apartado de *storage*.

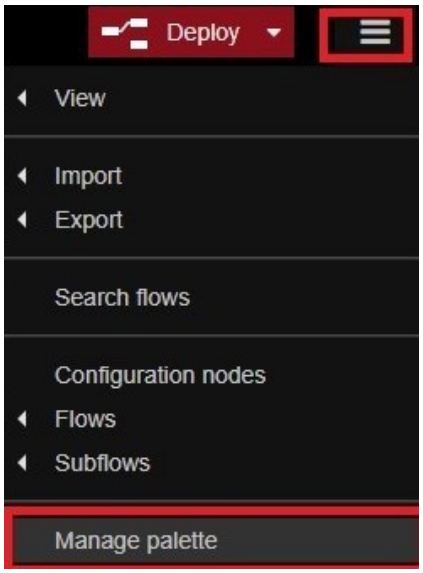

Figura 3.19. Paso 1 para instalar los nodos de InfluxDB (Recuperado de Node-Red).

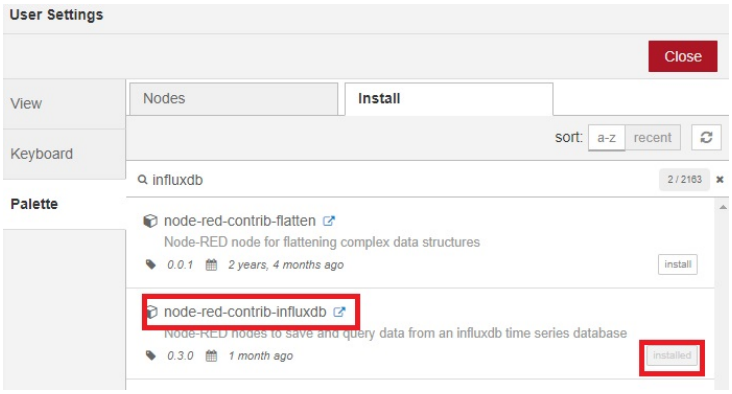

Figura 3.20. Paso 2 para instalar los nodos de influxdb (Recuperado de Node-Red).

Gracias a este nodo es posible utilizar el servicio de InfluxDB[9], para poder emplearlo se requiere instalar InfluxDB.chronograf.

Para la instalación de este servicio. hay que abrir la consola y escribimos los siguiente: curl -sL https://repos.influxdata.com/influx.key | sudo apt-key add - echo "deb https://repos.influxdata.com/debian stretch stable"| sudo

[9]Es una base de datos de serie de código abierto desarrollada por InfluxData.

tee/etc/apt/sources.list.d/influxdb.list (enter), una vez ejecutado lo anterior se actualizan los repositorios: sudo apt-get update, se instala influxb chronograf: sudo apt-get install influxb chronograf, se habilita el servicio de influxdb: sudo systemctl enable influxdb.service, se inicaliza influx: sudo systemctl start influxdb, se prosige a habilitar chronograf: sudo systemctl enable chronograf.service, se inicializa chronograf: sudo systemctl start chronograf.

Una vez instalado el servicio colocamos la IP proporcionada por el sistema mas la extensión:8888, la cual nos llevará a Chronograf donde habrá que hacer un usuario, una vez hecho los anterior se puede configurar la base de datos.

Para configurar la base de datos hay que ir a la opción de explore (figura 3.21), dar clic en *metaquery template*, en esa pestaña hay que seleccionar la opción de *create data base*, se le asigna un nombre a la base de datos y posteriormente dar clic en *summit* (figura 3.22).

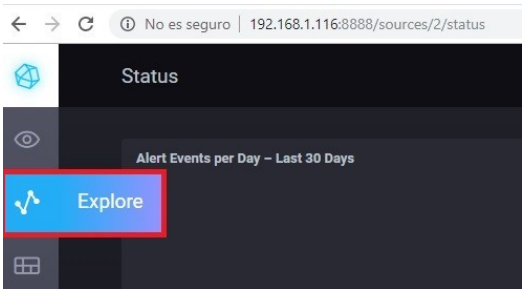

Figura 3.21. Paso 1 para configurar la base de datos (Recuperado de Influxb Chronograf).

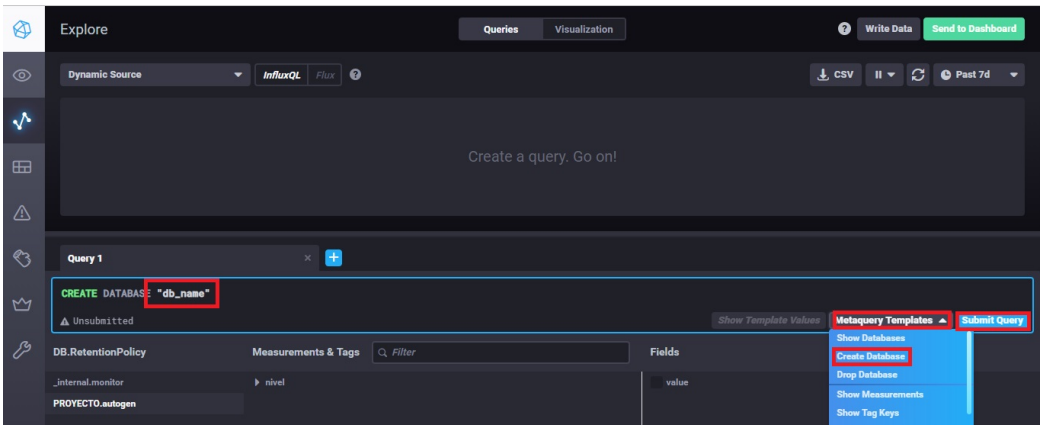

Figura 3.22. Paso 2 para configurar la base de datos (Recuperado de Influxb Chronograf).

Ahora hay que regresar a Node-Red damos clic en el nodo de influxdb, se desplegará el panel de configuración, primero habrá que configurar el servidor, para ello hay que dar clic en el ícono de lápiz que está a lado de campo de búsqueda. Por lo anterior, saldrá una serie de campo que hay que rellenar con los datos establecidos en la cuenta de usuario que se realizó en Chronograf los cuales son: la dirección ip (proporciona por el sistema) y el puerto, en *Database* es donde hay poner el nombre de la base de datos creada, en *username* se escribe el usuario de la cuenta, en el campo de *password* se escribe la contraseña de la cuenta y se le da clic en add (figura 3.23).

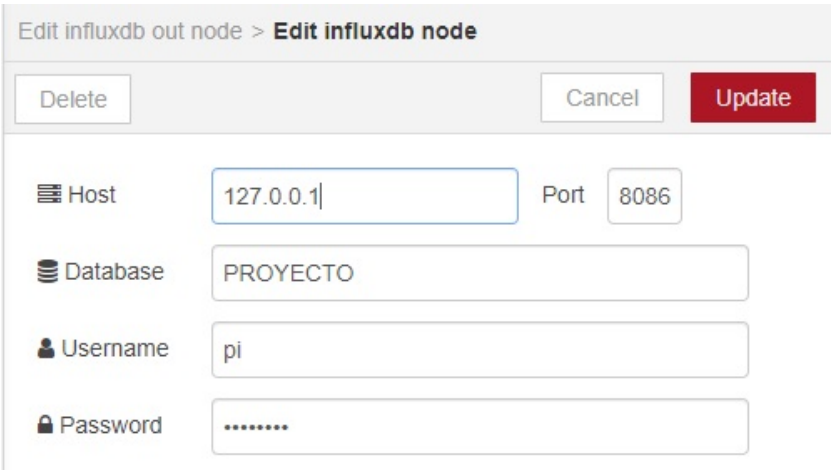

Figura 3.23. Datos de usuario de la base de datos (Recuperado de Node-Red).

Luego hay que regresar al panel anterior donde hay que nombrar el dato que se transmitirá a Chornograf y ponerle una etiqueta al nodo, como se ve en la figura 3.24.

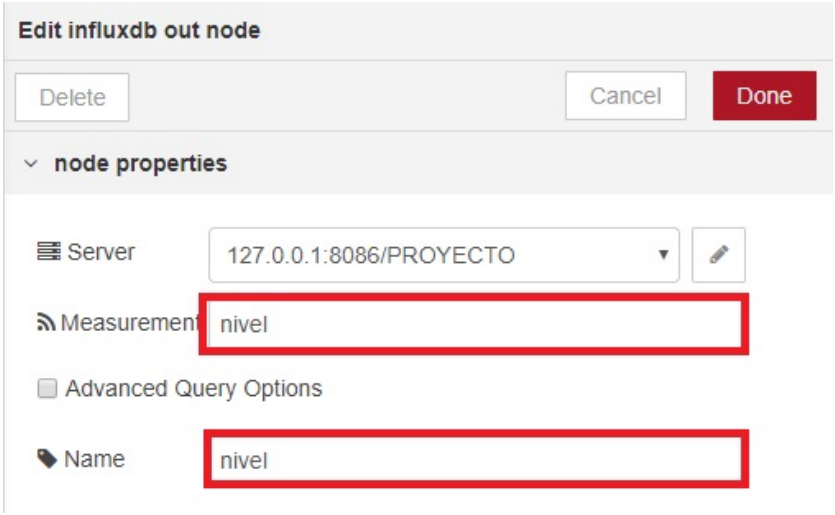

Figura 3.24. Panel de configuración del nodo influxdb (Recuperado de Node-Red).

Nodo *switch*

Este nodo se encarga de evaluar el dato recibido, dependiendo de las reglas declaradas el nodo determinará si el dato continuará el flujo hacia el siguiente nodo o no. En este caso la condición es que el valor recibido sea igual o mayor al establecido, como se muestra en la figura 3.25.

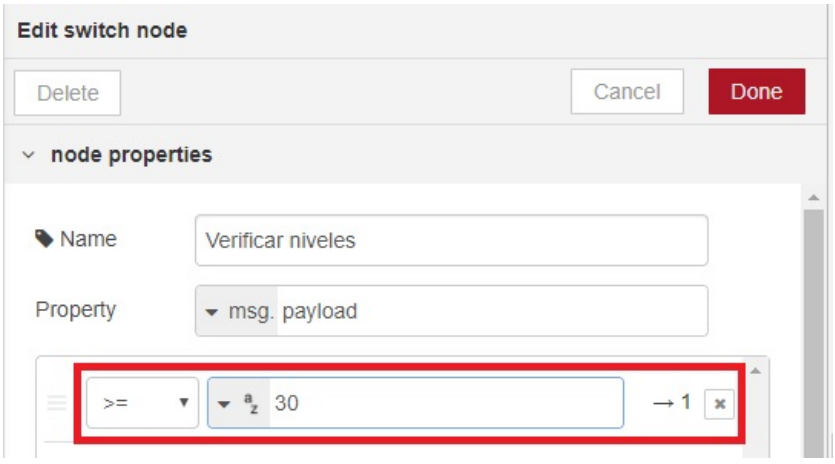

Figura 3.25. Condicionante (Recuperado de Node-Red).

Nodo *delay*

La función de este nodo es retrasar el dato, la configuración del nodo va a ser el tiempo que se va retrasar, también tiene la opción de solo tomar en cuenta el último dato recibido para seguir el flujo, se puede configurar para establecer cuantos mensajes van a pasar después de que termine el tiempo de espera, la configuración que se estableció para este nodo se puede ver en la figura 3.26.

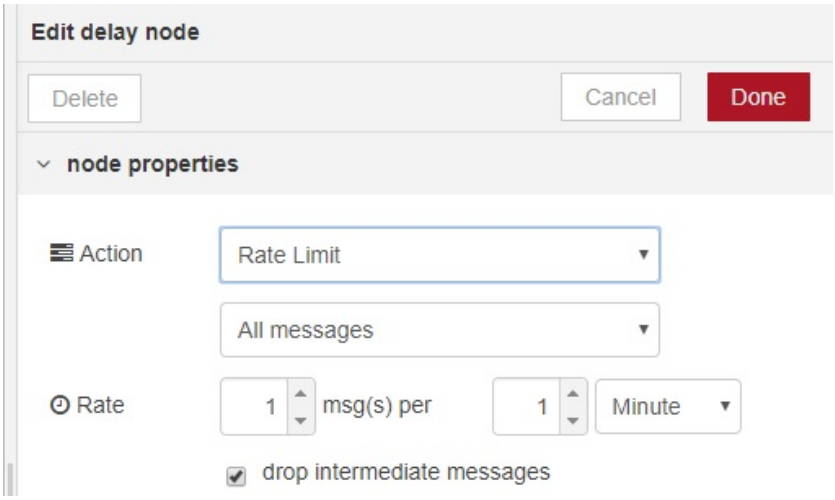

Figura 3.26. Configuración del nodo *delay* (Recuperado de Node-Red).

Nodo *function*

Se emplea otro nodo *function* para estructurar el mensaje que será publicado en Twitter, se invoca la carga útil (es el dato de nivel), se declaran dos variables texto1 y texto2, posteriormente se establece que la carga útil ahora será igual al texto1 + la carga útil (dato de nivel) + texto2 y de esa manera se configuró el tweet que publica el proyecto en la cuenta de Twitter, la programación se puede ver en la figura 3.27.

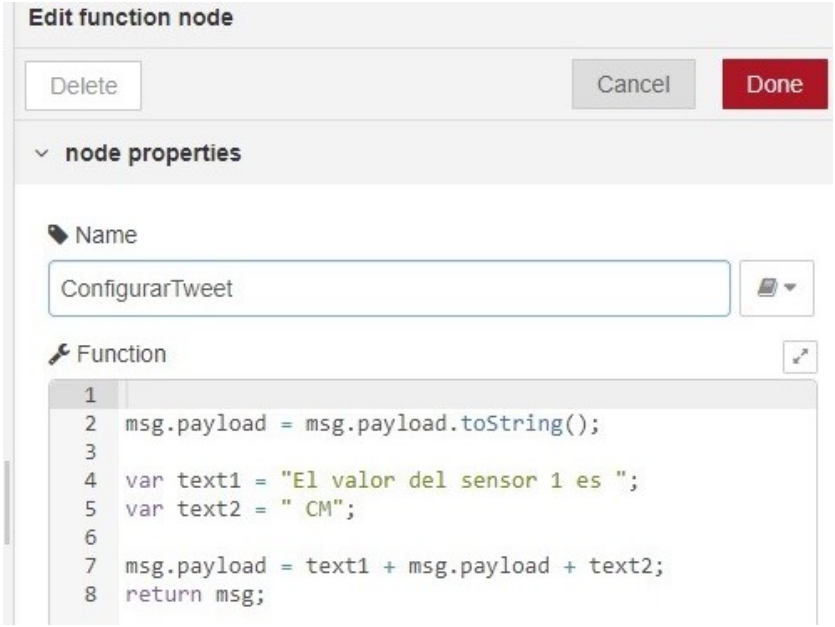

Figura 3.27. Configuracion de tweet (Recuperado de Node-Red).

Nodo twitter *out*

Gracias a este nodo y a la API, es posible realizar la publicación en Twitter sin la intervención humana, la forma de configurar este nodo es sencillo, en el panel de configuración se escribe el nombre de la cuenta de Twitter (figura 3.28), posteriormente damos clic al icono del lápiz, muestra otra ventana donde se escriben los códigos de la API previamente solicitada y los códigos de los *tokens*(figura 3.29) luego hay que hacer clic en *update*.

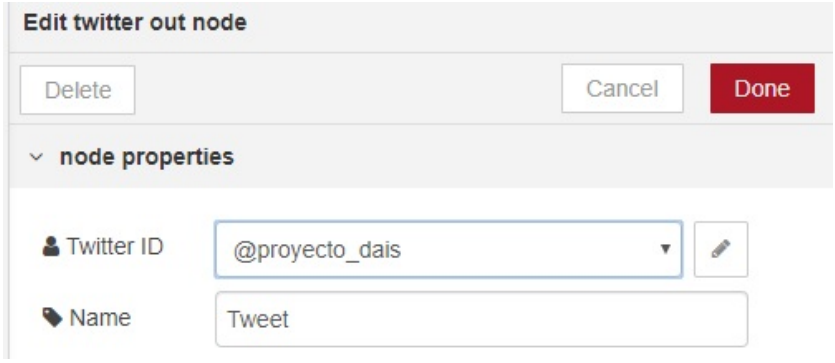

Figura 3.28. Panel de configuración del nodo twitter *out* (Recuperado de Node-Red).

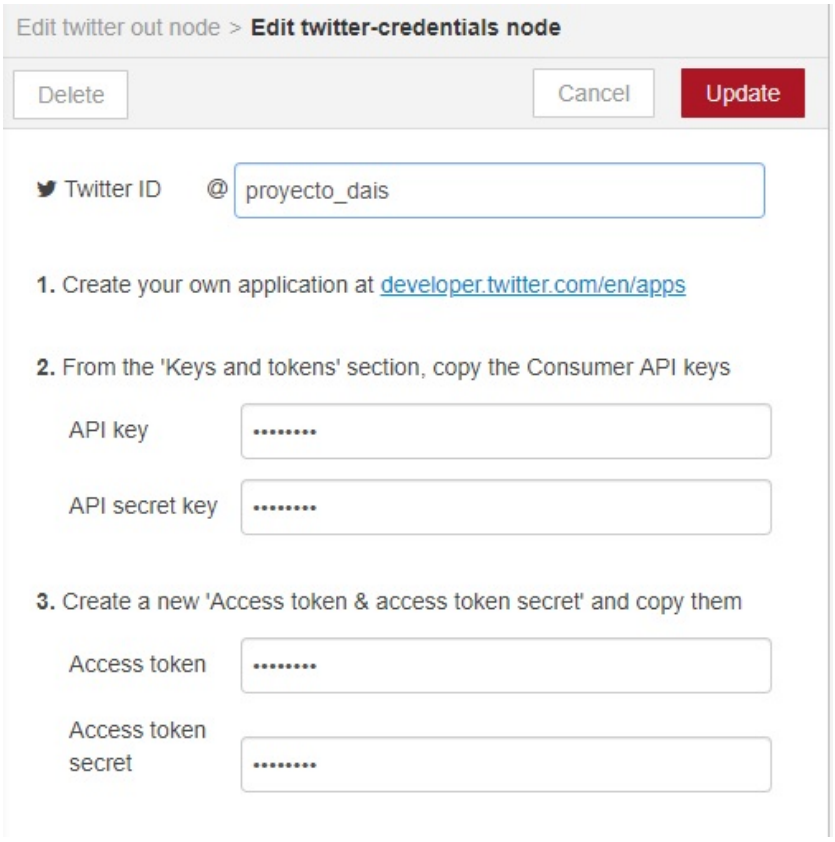

Figura 3.29. Configuración de las credenciales (Recuperado de Node-Red).

Capítulo 4

Experimentos y Resultados

En este capítulo se presentan las pruebas que se realizaron al prototipo y los resultados obtenidos.

4.1. Pruebas de distancia

Para poder corroborar que la medición del nodo sensor es exacta, se realizaron mediciones en tierra y en agua, los resultados de dichas pruebas fueron publicados en el artículo [23] y también se realizaron pruebas de entrega de tweets para determinar la eficacia que tiene el prototipo, al momento de comunicar/alertar por medio de twitter los datos que esté leyendo la red de sensores. Todas las pruebas se realizaron en un laboratorio, no se llevaron acabo pruebas en un entorno real, debido a que el hardware no es muy potente y tampoco se contaba con apoyo de protección civil para ir a realizar pruebas al borde de un río.

4.1.1. Pruebas en tierra

La prueba en tierra se realizó para determinar que el nodo sensor es capaz de medir distancias. La primera prueba se llevó acabo en tierra, con el fin de confirmar que el nodo sensor es capaz de obtener distancias de hasta 3 metros.

Las mediciones se realizaron poniendo un flexómetro en el piso y alineado en dirección a la pared, arriba del flexómetro se ubicaba el sensor ultrasónico para poder corroborar la distancia obtenida por el sensor, como se puede

visualizar en la figura 4.1, la figura se tomó en un plano normal.

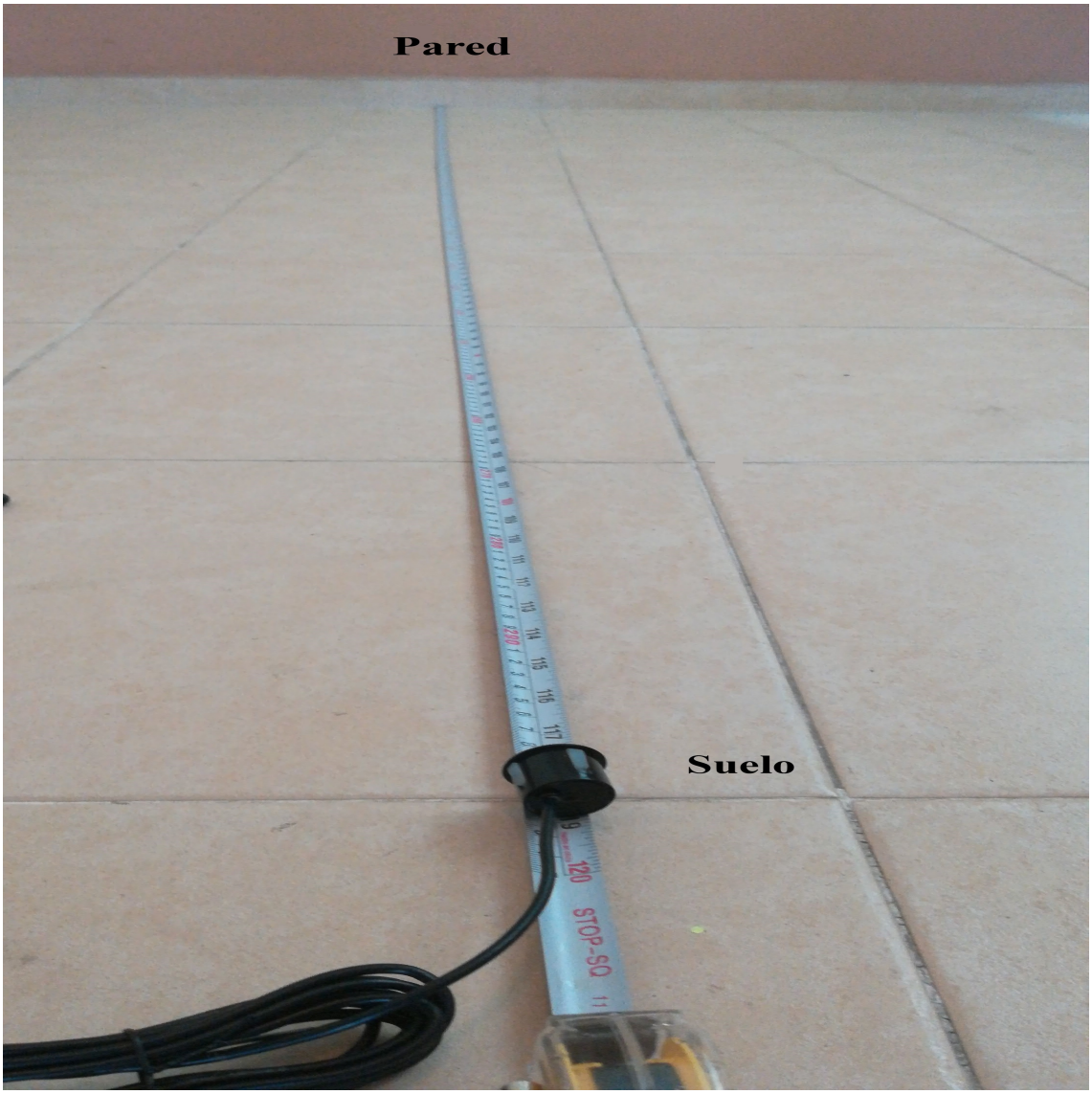

Figura 4.1. Medición de distancia en tierra (Foto del sensor Jsn-sr04t 2.0).

La medición que se obtenía mediante el sensor se visualiza en el nodo receptor (figura 4.2) de tal manera se prueba la exactitud de la medición y la comunicación LoRa entre nodo sensor y el nodo receptor.

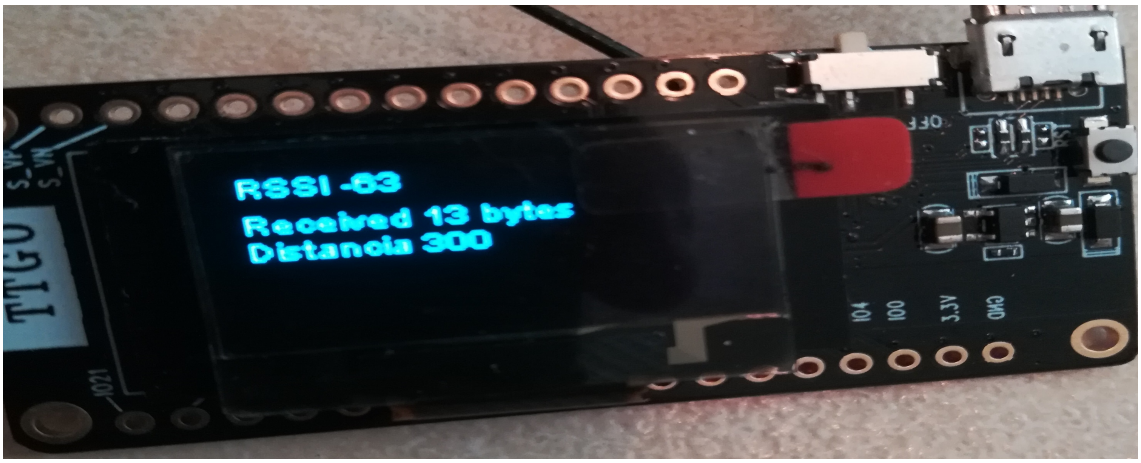

Figura 4.2. Nodo receptor recibiendo datos desde el nodo sensor(Foto de TTGO LoRa32).

Para realizar las pruebas se diseño un experimento de medir 6 tramos en intervalos de 50 cm, dichos tramos son 50 cm, 100 cm, 150 cm, 200 cm, 250 cm y 300 cm, de las cuales se tomaron 10 muestras por cada distancia, los resultados de esta prueba se muestran en la tabla4.1.

Tabla 4.1. Tabla de mediciones realizadas en tierra.

Prueba en tierra a 50 cm		Prueba en tierra a 100 cm		Prueba en tierra a 150 cm		Prueba en tierra a 200 cm		Prueba en tierra a 250 cm		Prueba en tierra a 300 cm	
Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm
50	50	100	100	150	150	200	200	250	251	300	300
50	50	100	100	150	150	200	200	250	250	300	300
50	50	100	100	150	151	200	201	250	251	300	301
50	50	100	100	150	150	200	200	250	250	300	299
50	50	100	100	150	150	200	200	250	250	300	300
50	50	100	100	150	150	200	200	250	250	300	300
50	50	100	100	150	150	200	200	250	251	300	300
50	50	100	101	150	151	200	201	250	250	300	300
50	50	100	101	150	150	200	200	250	250	300	301
50	51	100	100	150	150	200	200	250	250	300	300
Media	50.0991112	Media	100.199	Media	150.199	Media	200.1996	Media	250.300	Media	300.0995
SD	0.31622777	SD	0.4216	SD	0.4216	SD	0.422	SD	0.483	SD	0.568

En la tabla anterior se aprecian las distancias en las que se puso a prueba la medición del nodo sensor, empezando por 50 cm hasta los 300 cm, en intervalos de 50 cm, se tomaron 10 muestras por cada prueba, tomando en

cuenta los resultados de las muestras se obtuvo la media y la desviación estándar de estas mediciones.

4.1.2. Pruebas en agua

También se realizaron pruebas para determinar si el sensor podría medir el nivel de agua, se estableció la profundidad total del contenedor en el código de programación, a esa constante se le restaba la medición del sensor ubicado al borde del contenedor, el sensor medía desde el borde hasta la superficie del agua, esta prueba se realizó llenando de agua el contenedor, desde 20 cm hasta 70 cm, estas distancias se consideraron por que el experimento se realizó en un contenedor de 90 cm pero como el sensor empieza a medir a partir de 20 cm se le restaron a los 90 cm de altura del contenedor y la distancia máxima a medir fueron 70 cm, por lo cual se diseñó un experimento de llenar el contenedor a 4 alturas diferentes las cuales son a 20 cm, 40 cm, 60 cm y 70 cm, de cada medición se registraron 10 muestras, los resultados se muestran en la tabla4.2.

Tabla 4.2. Tabla de mediciones realizadas en agua.

Prueba en agua a 20 cm		Prueba en agua a 40 cm		Prueba en agua a 60 cm		Prueba en agua a 70 cm	
Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm	Flexómetro cm	Sensor cm
20	21	40	40	60	60	70	70
20	20	40	40	60	60	70	70
20	20	40	40	60	60	70	70
20	21	40	39	60	61	70	70
20	21	40	40	60	60	70	69
20	20	40	40	60	60	70	70
20	20	40	41	60	60	70	70
20	20	40	40	60	61	70	70
20	20	40	40	60	60	70	70
20	20	40	40	60	60	70	69
Media	20.295	Media	39.997	Media	60.199	Media	69.799
Desv. Estandar	0.483	Desv. Estandar	0.471	Desv. Estandar	0.422	Desv. Estandar	0.422

De igual manera se tomaron las 10 muestras a cada prueba, para sacar la media y la desviación estándar.

La medición se realizó poniendo el sensor en el borde del contenedor y el flexómetro a un lado del sensor, se medía desde el borde del contenedor hasta la superficie del agua como se muestra en la siguiente figura 4.3.

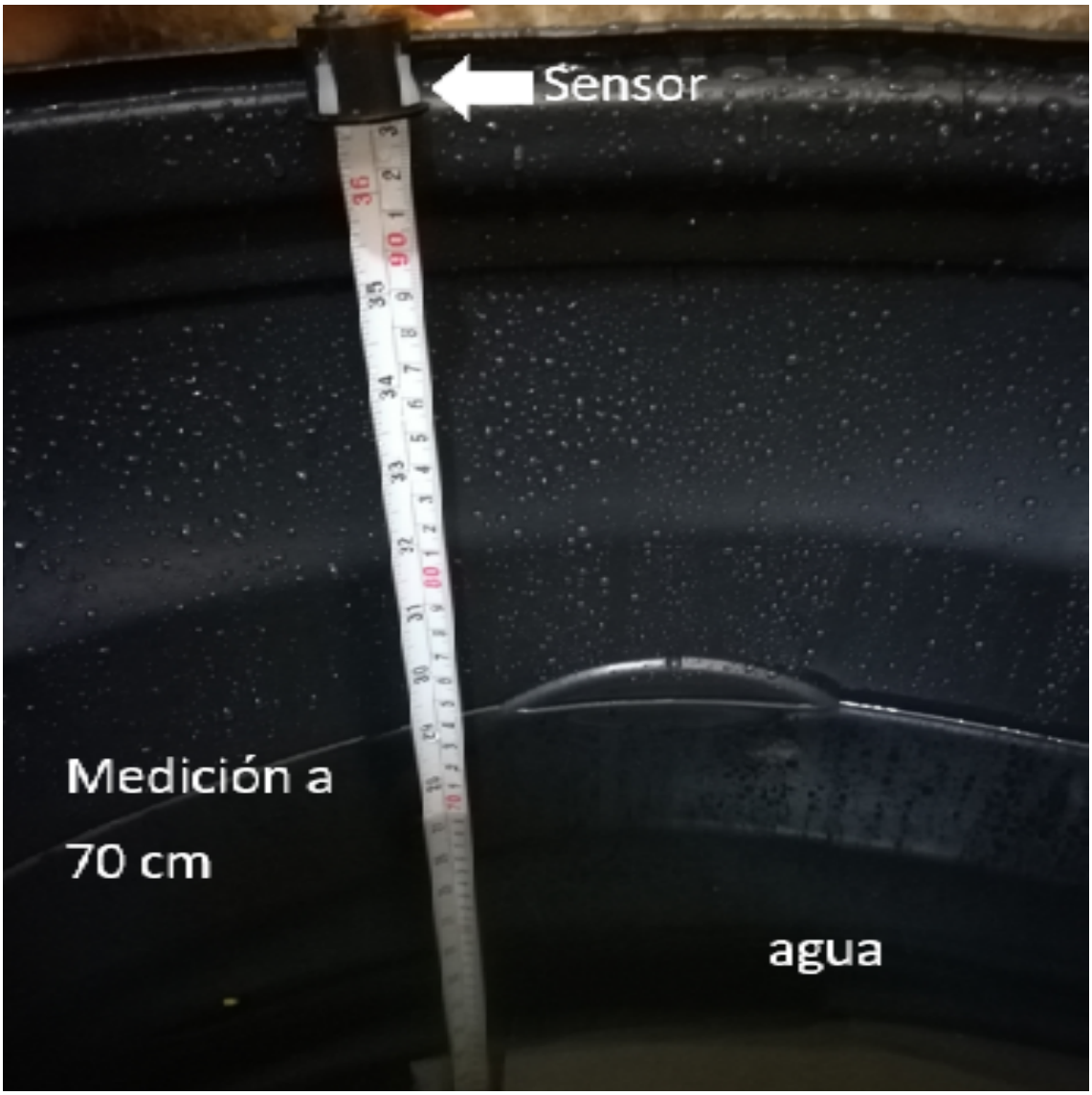

Figura 4.3. Medición de distancia en agua (Foto del sensor Jsn-sr04t 2.0)

La medición que se obtenía mediante el sensor se visualiza en el nodo receptor (figura 4.4) de tal manera se prueba la exactitud de la medición y la comunicación LoRa entre nodo sensor y el nodo receptor.

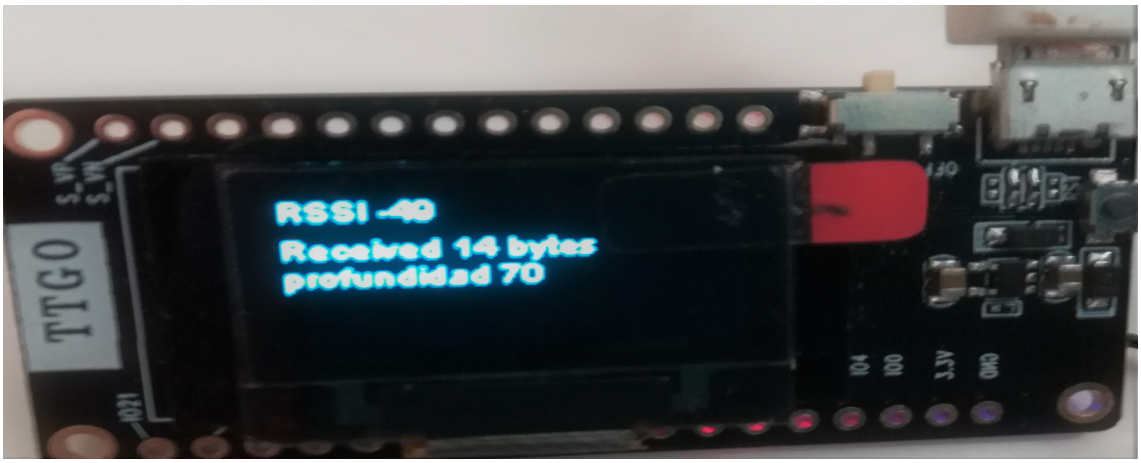

Figura 4.4. Nodo receptor recibiendo datos desde el nodo sensor(Foto de TTGO LoRa32).

En las pruebas realizadas en agua, el dato obtenido por el nodo sensor se visualiza en el nodo receptor, (como se aprecia en la figura 4.4), el contenedor tiene una profundidad total de 90 cm y el sensor estaba a una distancia de 25 cm de la superficie del agua y el nodo receptor indica una distancia de 70 cm, esto indica que si obtiene la profundidad, mediante la medición de la distancia correspondiente desde el sensor ultrasónico respecto de la superficie de agua, restando el dato obtenido por el sensor con la profundidad total del contenedor. La operación que se realiza para la obtención del nivel es una resta (4.1) la cual es: la distancia uno (D1), es el borde del cuerpo de agua, menos la distancia dos (D2), que es la superficie de la masa de agua y el resultado es la profundidad del agua (P).

$$P = D1 - D2 \tag{4.1}$$

En la figura 4.5 se visualiza como están determinadas las distancias, la D1 varía en cuestión de la superficie del agua del río y la D2 del fondo del río es una constante declarada en la placa TTGO, una vez tomada la distancia de la superficie del río con respecto al borde del río, está distancia se resta con la distancia D2 que está declara en la placa.

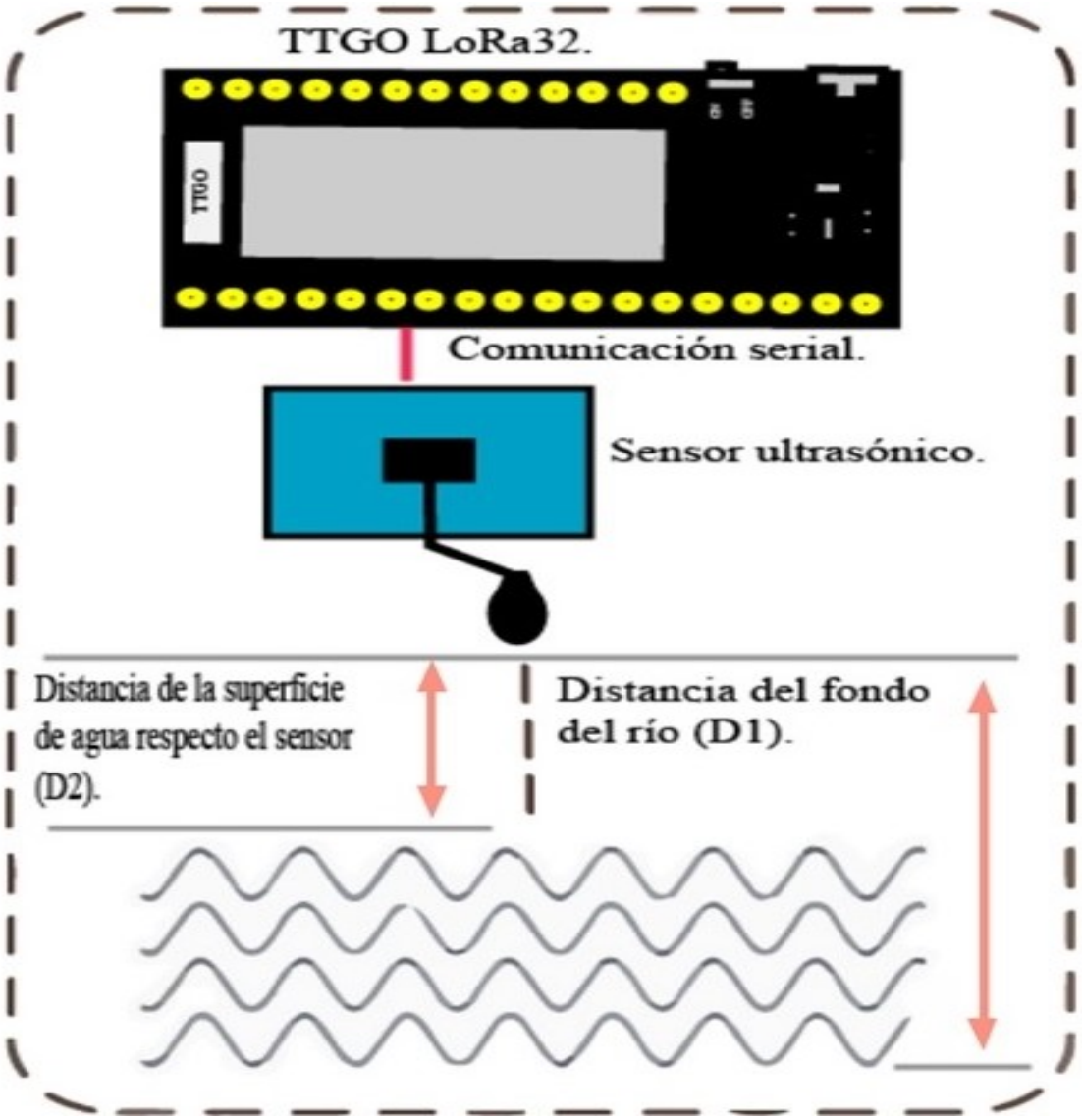

Figura 4.5. Premisa de obtención del nivel del agua.

4.2. Pruebas de publicación

Para conocer la eficacia que tiene el proyecto en mandar alertas por medio de Twitter, se realizaron pruebas en las cuales se notaron ciertas irregularidades en el funcionamiento del nodo receptor, la tarjeta cuando está en un

tiempo de mas o menos de 3 horas, en algunas pruebas se apagaba o simplemente dejaba de recibir los datos provenientes del nodo receptor, se desconoce a que se deba ese comportamiento.

Por el motivo expuesto en el párrafo anterior se propusieron pruebas de emitir 30 tweets en diferentes intervalos de minutos (1, 2, 3 y 4), establecidos en la programación de mediante el nodo delay en Node-Red, en las pruebas se tomaron en cuenta los tweets que estuvieron en el rango de tiempo de cada prueba; por ejemplo en las pruebas de intervalos de tweets, cada minutos se consideraron los tweets publicados en 30 minutos, en la de 2 minutos se tomaron en cuenta los tweets publicados en el rango de 60 minutos, en las pruebas de 3 min. los publicados en 90 minutos y en la prueba de 4 minutos los que se publicaron en un tiempo de 120 minutos, los primeros tweets de cada prueba tienen el valor de mayor o igual a 20 cm, debido a que el sensor empieza a captar distancias mayores o igual a 20 cm.

Para esta prueba se empleó un contenedor de 150 cm de altura, se utilizo el mismo principio empleado en las pruebas de medición de agua, poniendo el sensor al borde del contenedor, pero en este caso el valor se visualizaba en la cuenta de Twitter del proyecto, de donde igual se obtenía la hora de la publicación de cada tweet. Los resultados de las pruebas se muestran en las siguientes tablas.

Tabla 4.3. Tabla de tweets cada 1 minuto.

Prueba de tweets cada 1 min.			
Fecha			
21-sep-19			
Evento	Distancia	Hora - p.m.	Minutos de diferencia entre tweets
	cm	HH:MM	
1	23	04:25	0
2	26	04:26	1
3	29	04:27	1
4	31	04:28	1
5	36	04:32	4
6	39	04:33	1
7	45	04:34	1
8	48	04:35	1
9	50	04:36	1
10	53	04:37	1
11	56	04:38	1
12	59	04:40	2
13	62	04:41	1
14	64	04:42	1
15	66	04:43	1
16	68	04:44	1
17	71	04:45	1
18	72	04:46	1
19	74	04:47	1
20	76	04:48	1
21	78	04:49	1
22	82	04:50	1
23	86	04:51	1
24	88	04:52	1
25	92	04:53	1
26	94	04:54	1
27	99	04:55	1
28	102	04:56	1
29	108	04:57	1
30	110	04:58	1

Tabla 4.4. Tabla de tweets cada 2 minutos.

Prueba de tweets cada 2 min.			
Fecha			
20-sep-19			
Evento	Distancia	Hora - p.m.	Minutos de diferencia entre tweets
	cm	HH:MM	
1	21	04:01	0
2	24	04:03	2
3	27	04:05	2
4	31	04:07	2
5	35	04:09	2
6	41	04:11	2
7	43	04:13	2
8	46	04:15	2
9	50	04:17	2
10	56	04:20	3
11	58	04:22	2
12	61	04:24	2
13	66	04:26	2
14	68	04:28	2
15	72	04:30	2
16	76	04:33	3
17	81	04:35	2
18	84	04:37	2
19	88	04:39	2
20	93	04:41	2
21	85	04:45	4
22	95	04:47	2
23	98	04:49	2
24	104	04:51	2
25	109	04:53	2
26	113	04:55	2
27	117	04:57	2
28	120	05:01	4
29	125	05:06	5
30	129	05:08	2

Tabla 4.5. Tabla de tweets cada 3 minutos.

Prueba de tweets cada 3 min.			
Fecha			
19-sep-19			
Evento	Distancia	Hora - p.m.	Minutos de diferencia entre tweets
	cm	HH:MM	
1	22	02:35	0
2	27	02:38	3
3	28	02:42	4
4	31	02:45	3
5	35	02:48	3
6	38	02:51	3
7	41	02:54	3
8	44	02:57	3
9	47	03:00	3
10	49	03:03	3
11	51	03:06	3
12	53	03:09	3
13	56	03:12	3
14	58	03:15	3
15	60	03:18	3
16	61	03:21	3
17	64	03:24	3
18	67	03:27	3
19	69	03:30	3
20	71	03:33	3
21	75	03:36	3
22	77	03:39	3
23	82	03:42	3
24	84	03:46	4
25	88	03:49	3
26	93	03:52	3
27	95	03:55	3
28	98	03:59	4
29	65	04:03	4
30	105	04:07	4

Tabla 4.6. Tabla de tweets cada 4 minutos.

Fecha			
18-sep-19			
Evento	Distancia	Hora - p.m.	Minutos de diferencia entre tweets
	cm	HH:MM	
1	20	04:32	0
2	21	04:37	5
3	23	04:42	5
4	27	04:46	4
5	31	04:51	5
6	33	04:56	5
7	37	05:00	4
8	41	05:04	4
9	44	05:08	4
10	47	05:12	4
11	49	05:16	4
12	51	05:20	4
13	60	05:25	5
14	57	05:29	4
15	61	05:33	4
16	64	05:37	4
17	68	05:41	4
18	70	05:45	4
19	75	05:49	4
20	79	05:53	4
21	82	05:57	4
22	52	06:01	4
23	84	06:05	4
24	88	06:09	4
25	91	06:13	4
26	94	06:18	5
27	97	06:22	4
28	102	06:26	4
29	107	06:30	4
30	110	06:34	4

Los datos mostrados en las tablas anteriores se pueden cotejar en la cuenta específica del prototipo (@Proyecto_Dais), se emplearon para obtener la efectividad de alertar por medio de Twitter (ET[1]), utilizando los datos en una fórmula, conformada por los tweets que se publicaron en el tiempo correspondiente de cada prueba (tweet reales), los cuales se dividen entre la cantidad de tweets que se debieron publicar en el tiempo establecido para cada prueba (tweets esperados), el resultado de la división se multiplicará por 100 para así obtener la efectividad de los tweets.

$$ET = tweetsreales/tweetsesperados * 100 \tag{4.2}$$

La formula 4.2 se planteó modificando la fórmula de la efectividad 4.3.

$$Efectividad = produccionreal/capacidadproductiva * 100 \tag{4.3}$$

En la prueba de 1 tweet cada un minuto se esperaban 30 tweets en un tiempo de 30 minutos, de los cuales 27 fueron reales en los 30 minutos de la prueba.

Sustituyendo

$$ET = 27/30 * 100 = 90\,\% \tag{4.4}$$

En la prueba de 1 tweet cada dos minutos de igual manera se esperaban 30 tweets, pero en un tiempo de 60 minuto, de los cuales 28 fueron reales en ese lapso de tiempo.

Sustituyendo

$$ET = 28/30 * 100 = 93.33\,\% \tag{4.5}$$

En la prueba de 1 tweet cada 3 minutos se esperaban 30 tweets en un tiempo de 120 minutos(1 hora y media), en ese lapso de tiempo se publicaron 29 tweets.

Sustituyendo

$$ET = 29/30 * 100 = 96.66\,\% \tag{4.6}$$

En la prueba de 1 tweet cada 4 minutos se obtuvieron 29 tweets, en un tiempo de 150 minutos (2 horas y media) de los 30 esperados.

[1]significa efectividad de tweets.

Sustituyendo

$$ET = 29/30 * 100 = 96.66\,\% \tag{4.7}$$

4.3. Observaciones

En la ejecución de las pruebas, se observó que los tweets no se publicaban en un tiempo normalizado; por ejemplo, a veces se publicaban exactamente cada minuto y en otras ocasiones se tardaba unos minutos más en realizar la publicación del tweet, por ello se perdieron tweets efectivos en la prueba de publicación de tweet de cada 1 y 2 minutos, ya que en dado momento la suma de esos minutos llegaron a afectar en el desempeño de las publicaciones.

Igual afecta el tiempo de transmisión de los datos por medio de los nodos, puesto que se notó en las pruebas que se perdió en ocasiones por uno o tres minutos como máximo la transferencia de los datos, ello afectó la publicación de los tweets, lo cual sucedió en cada una de las pruebas.

4.4. Discusión

El prototipo de alerta temprana para desborde de ríos cumple con las especificaciones y arquitectura señaladas en el capítulo 3, sin embargo existen oportunidades para mejorarlo.

Una opción es invertir en un hardware mas potente que el empleado en este desarrollo, empezando por el sensor ultrasónico jsn-sr04t-2.0 el cual tiene un rango de 20 cm a 3 m, lo ideal sería sustituirlo por uno que tenga un rango de medición mas amplio.

Otro elemento a sustituir es la tarjeta programable TTGO LoRa32 para probar en un entorno real, ya que el módulo de radio que se utiliza para realizar la comunicación LoRa tiene una transmisión máxima de 700 m, por ello se debería sustituir por otra que cuente con un módulo de radio con mas alcance.

Además, es necesario contar con un módem con acceso a internet para comunicar los datos desde el nodo receptor a un sistema informático, también hay que considerar las restricciones que tiene Twitter ya que pueden variar.

Los resultados obtenidos por el prototipo son buenos y superan los resultados esperados, por lo cual se debe considerar hacer el cambio de hardware,

ver el comportamiento que éste tenga y emitir la alerta por otro medio, ya que las pruebas realizadas en el laboratorio son buenos por las pruebas realizadas de medición en tierra y agua, el sensor tiene un margen de error de 1 cm, y las pruebas de publicación muestran una efectividad muy buena de un 90 % de efectividad en la prueba de un minuto, en la prueba de 1 tweet cada 2 minutos se obtuvo una efectividad de 93 % y en las pruebas de un 1 cada 3 minutos y cada 4 minutos se obtuvo un 96 % de efectividad, superando la hipótesis planteada de un 80 % de efectividad.

Capítulo 5

Conclusiones, Contribuciones y Trabajos futuros

5.1. Conclusiones

El prototipo desarrollado es capaz de medir las variaciones de nivel de agua de un río mediante un nodo sensor, el cual transmite la información en tiempo real al nodo receptor, quien al mismo tiempo envía la información a una Raspberry pi, en consecuencia se encarga de registrar los datos en una base de datos y publicarlos en Twitter.

Esto puede informar a las personas prácticamente de manera inmediata para que de esa manera las personas puedan tomar sus precauciones. Lo que nos lleva a las siguientes conclusiones en relación con los objetivos planteados en este proyecto.

Por cuanto el objetivo general, se concluyó que el prototipo de alerta temprana para desbordes de ríos se realizó y fue probado a nivel laboratorio, de manera que emite alertas por medio de Twitter con el dato de nivel de agua. sin embargo no se probó en un entorno real (un río), debido a que no se contó con respaldo de las autoridades para llevar a cabo dichas pruebas.

Por lo que concierne a los objetivos específicos que se persiguieron alcanzar, como resultado del presente trabajo quedan logrados:

1. Se desarrolló una red de sensores, esta red de sensores consta de un nodo sensor y un nodo receptor, el nodo sensor que es la integración y configuración del sensor ultrasónico jsn-sr04t-2.0 con una tarjeta TT-GO LORA32, dicha tarjeta cuenta con un módulo de radio capaz de

emplear el protocolo de comunicación LoRa, este protocolo es utilizado para transmitir los datos al nodo receptor que es otra tarjeta TTGO LORA, de modo que se programó para recibir y enviar los datos obtenidos a un sistema informático, algunas complicaciones que se presentaron en este punto es que en el área en que se desarrolló este proyecto (Villahermosa, Tabasco), no hay distribuidores de hardware al menos no del que se empleó en este prototipo, de tal manera que se pidieron y el tiempo de espera de algunos elementos específicamente las tarjetas TTGO LoRa32 tardaron un lapso de 3 semanas en llegar, también la tarjeta TTGO LoRa32 no es compatible con el IDE de Arduino y se tuvo que realizar una serie de arreglos, lo que se explican en la sección de nodo sensor específicamente en la subsección de configuración, también los códigos de programación proporcionados por el distribuidor no venía muy bien documentado y se tuvo que investigar en páginas de internet y vídeos referentes a la tarjeta TTGO LoRa.

2. El segundo objetivo específico se consiguió, implementando el protocolo de comunicación LoRa (mencionado en el párrafo anterior) y el protocolo MQTT, que se utilizó para la transferencia del dato desde el nodo receptor hacia un sistema informático (Raspberry pi), en este punto no hubo mayor complicación, sin embargo debido a que se implementó una librería de encritación (xxtea-iot-crypt.h) hubo ciertos contratiempos ya que en el transcurso de estas pruebas fue actualizada y redujo los bytes que se podían encriptar, por lo cual el protocolo LoRa no hacía la transmisión de datos, estó se solucionó quitándo le al dato captado por el sensor las décimas y transmitiendo un número entero encriptado.

3. Se estableció una comunicación con Twitter, gracias a Node Red que facilitó la vinculación de una cuenta de Twitter, mediante un nodo de programación que permite la publicación de tweets de manera automática en intervalos de tiempo establecidos en la programación, en este lo único que podría verse como un contra tiempo es que hay que esperar respuesta de Twitter para que te concedan permiso para utilizar la API de Twitter, una vez que se autorice la utilización de la API es muy sencillo vincular la cuenta de Twitter mediante Node-Red.

De acuerdo con la hipótesis planteada, en cuanto a la efectividad que tiene el prototipo en alertar por medio de una red social, las pruebas dieron un 90 % en la prueba de alertar cada 1 minuto, en la prueba de alertar cada 2

minutos la prueba dio una efectividad de 93 % y las pruebas de alertar cada 3 minutos y cada 4 minutos lo resultados dieron un 96 % en cada una, por lo tanto quedan superadas las expectativas.

5.2. Contribuciones

Existen equipos comerciales los cuales miden el nivel del agua y comunican mediante el protocolo empleado en este prototipo, pero son costosos y de igual manera no tienen comunicación con una red social.

Este tipo de proyectos han sido poco explorados en la institución por lo que se espera que este proyecto sirva de motivación a la comunidad estudiantil.

Se utilizó exclusivamente Software libre y estándares abiertos que permiten que se pueda extender, configurar a la conveniencia y/o necesidad del usuario

La innovación que se está aplicando en el área de desarrollo de alertas tempranas es que se está empleando un protocolo de comunicación LoRa y no se necesita contar con acceso a Internet para poder transmitir datos obtenidos que en combinación con una computadora de placa única con acceso a internet también es capaz de emitir el estado del nivel de agua mediante una red social.

El desarrollo de este proyecto es un aporte a el internet de las cosas ya que se esta introduciendo un objeto en este caso un río este concepto, mediante el sensado de la red de sensores y la transmisión de los datos recabados hacia un sistema informático que guarde los datos en una base de datos, muestre en un panel y realice una alerta en una red social a través de internet.

5.3. Trabajos futuros

El desarrollo del prototipo se logró pero aun quedan trabajos por realizar como:

- Cambiar el hardware por uno mas potente.
- Utilizar energía solar para suministrar energía al prototipo.

- Establecer la ubicación del prototipo en Google Maps, para que los usuarios puedan ubicar el río de interés en el mapa y visualizar los datos del río.
- Probar el sistema de alerta temprana para desborde ríos en un escenario real.

Bibliografía

[1] Niels Aakvaag and Jan-Erik Frey. Redes de sensores inalámbricos. *Revista ABB*, 2:39–42, 2006.

[2] Rubén Adrián de la Cámara. Diseño de un sistema de monitorización remota de un depósito de agua mediante lora.

[3] Cristian Alberoni, Ernesto Leon, Miguel A Wister, and Jose A Hernandez-Nolasco. Wireless sensor network to collect training data from cycling. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 416–425. Springer, 2019.

[4] Aprendiendo Arduino. Ide arduino y configuración. https://aprendiendoarduino.wordpress.com/category/ide/, 6 2019.

[5] arduino.cl. ¿que es arduino? https://arduino.cl/que-es-arduino/, 2014.

[6] Pablo Esteban Avila Campos. Evaluación del rango de transmisión de lora para redes de sensores inalámbricas con lorawan en ambientes forestales. B.S. thesis, 2017.

[7] BricoGeek. Ttgo lora32 esp32 con oled. https://tienda.bricogeek.com/arduino-compatibles/1122-ttgo-lora32-esp32-con-oled-900-mhz.html.

[8] Gonzales Pinillos Art Charles Puente Isla Stephany Milagros Shica Julca Cristina Zenaida Calderón Flores Malú, Castillo Padilla Giancarlo. La informática y su relación con las otras ciencias. https://www.monografias.com/trabajos72/informatica-relacion-otras-ciencias/informatica-relacion-otras-ciencias2.shtml, 2009.

[9] Md Nasimuzzaman Chowdhury, Md Shiblee Nooman, and Srijon Sarker. Access control of door and home security by raspberry pi through internet. *Int. J. Sci. Eng. Res*, 4(1):550–558, 2013.

[10] Pedro Antonio Román Cisneros and Juan Carlos Parra Mena. Volumetric system, weighing and calculation of the shipping fee by standard method. In *2018 International Conference on Information Systems and Computer Science (INCISCOS)*, pages 115–125. IEEE, 2018.

[11] Melisa Andrea Acosta Coll. Sistemas de alerta temprana (sat) para la reducción del riesgo de inundaciones súbitas y fenómenos atmosféricos en el área metropolitana de barranquilla. *Scientia et technica*, 18(2):303–308, 2013.

[12] Conceptodefinicion.de. Definición de prototipo. https://conceptodefinicion.de/prototipo/, 2019.

[13] Blog Historia de la informatica. raspberry-pi. https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/, 12 2013.

[14] ISDR EWC III. Tercera conferencia internacional sobre alerta temprana. *Desarrollo de Sistemas de Alerta temprana*, 2006.

[15] Jose Facchin. Las redes sociales más utilizadas e importantes del mundo. https://josefacchin.com/lista-redes-sociales-mas-importantes-del-planeta/. Último acceso: 26 02 2018.

[16] Maray Garrido Monagas, Modesto R Gómez Crespo, and Alcides León Méndez. Sistema automatizado de alerta temprana ante el peligro de inundaciones. *Ingeniería Hidráulica y Ambiental*, 34(3):30–43, 2013.

[17] GeoEnciclopedia. Desastres naturales. http://www.geoenciclopedia.com/desastres-naturales/. Último acceso: 22 febrero 2018.

[18] J Gutiérrez and MA Porta-Gándara. Medidor ultrasónico de nivel de agua para estanques. *Ingeniería, investigación y tecnología*, 7(4):233–244, 2006.

[19] Mohannad Ibrahim, Abdelghafor Elgamri, Sharief Babiker, and Ahmed Mohamed. Internet of things based smart environmental monitoring using the raspberry-pi computer. In *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, pages 159–164. IEEE, 2015.

[20] influxdata. Chronograf. https://www.influxdata.com/time-series-platform/chronograf/.

[21] influxdata. Influxdb. https://www.influxdata.com/products/influxdb-overview/.

[22] Milica Lekić and Gordana Gardašević. Iot sensor integration to node-red platform. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE, 2018.

[23] Ernesto Leon, Cristian Alberoni, Miguel Wister, and Jose Hernández-Nolasco. Flood early warning system by twitter using lora. In *Multidisciplinary Digital Publishing Institute Proceedings*, volume 2, page 1213, 2018.

[24] Maryam Mohsin. Estadísticas de redes sociales que debes conocer en 2019. https://www.oberlo.com.mx/blog/estadisticas-redes-sociales, 2019.

[25] Multimedia. Lenguaje de programacion c++. http://lenguajedeprogramacion21.blogspot.com/, 2013.

[26] Christie Guadalupe Reyes Muñoz and Ezequiel Josué Flores González. Prototype early warning system using digital limnigraph. In *2017 IEEE Central America and Panama Student Conference (CONESCAPAN)*, pages 1–5. IEEE, 2017.

[27] naylamp mechatronics. Sensor ultrasonido jsn-sr04t. https://naylampmechatronics.com/sensores-proximidad/326-sensor-ultrasonido-jsn-sr04t.html.

[28] Dhira Negi, Ajit Kumar, Pradnya Kadam, and Bhairavi N Savant. Smart harvest analysis using raspberry pi based on internet of things. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–5. IEEE, 2018.

[29] Pierre Neumann, Julien Montavont, and Thomas Noël. Indoor deployment of low-power wide area networks (lpwan): A lorawan case study. In *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8. IEEE, 2016.

[30] Node-RED. Node-red. https://nodered.org/.

[31] Paul Pickering. Desarrollar con lora para aplicaciones iot de baja tasa y largo alcance. https://www.digikey.com.mx/es/articles/techzone/2017/jun/develop-lora-for-low-rate-long-range-iot-applications, julio 2017.

[32] María Estela Raffino. "hardware y software". https://concepto.de/hardware-y-software/, 2018.

[33] María Estela Raffino. Redes sociales. https://concepto.de/redes-sociales/, febrero 2019.

[34] Anoja Rajalakshmi and Hamid Shahnasser. Internet of things using node-red and alexa. In *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, pages 1–4. IEEE, 2017.

[35] Malik Abdul Rehman, Pratik Mallya, and Alistair Lobo. *Fire Detection by pattern matching using Camera and Raspberry PI*. PhD thesis, 2019.

[36] Mario Rodríguez Lallana et al. Diseño de un sensor de temperatura iot para la red lora. 2018.

[37] Edward Stiven Rodríguez Moreno, Víctor Felipe López Ordoñez, et al. Diseño e implementación de un sistema inteligente para un edificio mediante iot utilizando el protocolo de comunicación lorawan.

[38] Margaret Rouse. Twitter. https://whatis.techtarget.com/definition/Twitter, 12 2015.

[39] Syed Nazmus Sakib, Tanjea Ane, Nafisa Matin, and M Shamim Kaiser. An intelligent flood monitoring system for bangladesh using wireless sensor network. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 979–984. IEEE, 2016.

[40] Elena Sanz. ¿qué es el ïnternet de las cosas¿ https://www.muyinteresante.es/curiosidades/preguntas-respuestas/ique-es-el-qinternet-de-las-cosasq. Último acceso: 26 02 2018.

[41] Programo Ergo Sum. ¿qué es raspbian? https://www.programoergosum.com/cursos-online/raspberry-pi/232-curso-de-introduccion-a-raspberry-pi/instalar-raspbian.

[42] Telegrafia. Historia de los sistemas de alerta temprana y notificación de emergencias. http://www.sirenaselectronicas.com/blog/2017/02/08/historia-de-los-sistemas-de-alerta-previa-y-notificacion-de-emergencias/. Último acceso: 23 02 2018.

[43] Todo-Redes. Access point (punto de acceso). https://todo-redes.com/equipos-de-redes/access-point-punto-de-acceso, febrero 2019.

[44] TST. Mqtt. http://www.tst-sistemas.es/mqtt/.

[45] Damián Pérez Valdés. Qué son las bases de datos. http://www.maestrosdelweb.com/que-son-las-bases-de-datos/, 2007.

[46] Georgina Araceli Torres Vargas. El cómputo ubicuo y su importancia para la construcción del internet de las cosas y el big data/the ubiquitous computing and its importance for the construction of the internet of things and big data. *Revista general de información y documentación*, 24(2):217–232, 2014.

[47] Nico Varonas. Los mini ordenadores sbc y su futuro. https://www.neoteo.com/los-mini-ordenadores-sbc-y-su-futuro/comments-section, mayo 2012. Último acceso: 26 02 2018.

[48] Carlos Villagómez. Protocolo de comunicación. https://es.ccm.net/contents/275-protocolo-de-comunicacion, 2018.

[49] Miguel A Wister, José Adán Hernández-Nolasco, Pablo Pancardo, Francisco D Acosta, and Antonio Jara. Emergency population warning about floods by social media. In *2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pages 322–327. IEEE, 2016.

[50] Julián Pérez Porto y Ana Gardey. Sensor. https://definicion.de/sensor/, 2010.

[51] Julián Pérez Porto y Ana Gardey. Definición de api. https://definicion.de/api/, 2017.