



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA  
PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Implementación de una red de sensores inalámbricos  
enlazados con Xbee utilizando la plataforma thinger.io.**

AUTOR:

Cevallos Reinoso, Jorge Enrique

Componente práctico del examen complejo previo a la  
obtención del grado de **INGENIERO EN  
TELECOMUNICACIONES**

REVISOR:

Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

17 de Septiembre del 2018



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

## **CERTIFICACIÓN**

Certificamos que el presente **componente práctico del examen complejo**, fue realizado en su totalidad por **Cevallos Reinoso, Jorge Enrique** como requerimiento para la obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

REVISOR

---

Palacios Meléndez, Edwin Fernando

DIRECTOR DE CARRERA

---

Heras Sánchez, Miguel Armando

Guayaquil, a los 17 días del mes de Septiembre del año 2018



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Cevallos Reinoso, Jorge Enrique**

**DECLARÓ QUE:**

**El componente práctico del examen complejo, I Implementación de una red de sensores inalámbricos enlazados con Xbee utilizando la plataforma thinger.io.** previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 17 días del mes de Septiembre del año 2018

EL AUTOR

---

CEVALLOS REINOSO, JORGE ENRIQUE



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

### **AUTORIZACIÓN**

Yo, **Cevallos Reinoso, Jorge Enrique**

Autorizó a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del **componente práctico del examen complejo, Implementación de una red de sensores inalámbricos enlazados con Xbee utilizando la plataforma thinger.io.**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 17 días del mes de Septiembre del año 2018

EL AUTOR

---

CEVALLOS REINOSO, JORGE ENRIQUE

# REPORTE DE URKUND

The screenshot shows the URKUND interface with the following details:

- Documento:** [Cevallos Jorge EC FINAL 2018A.docx](#) (D41602984)
- Presentado:** 2018-09-19 18:38 (-05:00)
- Presentado por:** fernandopm23@hotmail.com
- Recibido:** edwin.palacios.ucsg@analysis.orkund.com
- Mensaje:** Revisión Examen Complexivo Jorge Cevallos [Mostrar el mensaje completo](#)  
1% de estas 15 páginas, se componen de texto presente en 2 fuentes.

The 'Lista de fuentes' (List of sources) panel on the right contains:

Categoría	Enlace/nombre de archivo
	<a href="#">TT-EDWIN-PINARGOTE-FINAL-20...</a>
	<a href="#">a Formato Complexivo FINAL 2...</a>
	<a href="#">Roberto Dender final.docx</a>
	<a href="https://doi.org/10.1049/cp.2010...">https://doi.org/10.1049/cp.2010...</a>

At the bottom right, there is a notification: **0 Advertencia**

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

Implementación

de una red de sensores inalámbricos enlazados con Xbee utilizando la plataforma thinger.io.

AUTOR: Cevallos Reinoso, Jorge Enrique

Componente práctico del examen complexivo

previo a la obtención del grado de INGENIERO EN TELECOMUNICACIONES

REVISOR: Palacios Meléndez, Edwin Fernando

## **DEDICATORIA**

Le dedico con todo mi cariño y amor a mis dos madres Anita Reinoso y Fabiola Tapia que me han ayudado a lo largo de mi formación personal y profesional, por su sacrificio y esfuerzo, por darme una carrera para mi futuro y por creer en mi capacidad. Gracias a ellas soy quien soy. Las amo.

EL AUTOR

CEVALLOS REINOSO, JORGE ENRIQUE

## **AGRADECIMIENTO**

Agradezco a Dios por siempre estar observándome y guiándome los pasos que doy en mi vida, por bendecirme siempre ayudándome a conquistar mis anhelos. Gracias al M. Sc. Edwin Fernando Palacios por todo el conocimiento y el apoyo brindado.

EL AUTOR

CEVALLOS REINOSO, JORGE ENRIQUE



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL**  
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO  
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

**TRIBUNAL DE SUSTENTACIÓN**

f. \_\_\_\_\_  
**MANUEL DE JESUS ROMERO PAZ**  
DECANO

f. \_\_\_\_\_  
**MIGUEL ARMANDO HERAS SÁNCHEZ**  
DIRECTOR DE CARRERA

f. \_\_\_\_\_  
**LUIS SILVIO CORDOVA RIVADENEIRA**  
OPONENTE

## ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	XI
ÍNDICE DE TABLAS.....	XIII
Resumen.....	XIV
CAPÍTULO 1: DESCRIPCIÓN DEL COMPONENTE PRÁCTICO.....	15
1.1. Introducción.....	15
1.2. Objetivo General.....	16
1.3. Objetivos Específicos.....	16
CAPÍTULO 2: Fundamentación teórica.....	17
2.1. Introducción del estándar IEEE 802.15.4.....	17
2.2. Aplicaciones de las redes de sensores inalámbricos.....	18
2.3. Topologías de las redes de sensores inalámbricos.....	18
2.3.1. Redes punto a punto (Peer-to-Peer, P2P).....	19
2.3.2. Redes estrella.....	19
2.3.3. Redes árbol.....	20
2.3.4. Redes enmalladas.....	20
2.4. XBee y sus estándares.....	21
2.4.1. Estándar IEEE 802.15.4.....	21
2.4.2. ZigBee.....	21
2.5.3. XBee.....	21
2.6. Motivo del uso del módulo XBee y Arduino UNO.....	22
Capítulo 3: Implementación de una red de sensores inalámbricos enlazados con Xbee utilizando la plataforma thinger.io.....	25
3.1. Elementos utilizados para el montaje de la red de sensores inalámbricos.....	25
3.1.1. Servidor Virtual Privado (VPS).....	25

3.1.2.	Arduino uno. ....	27
3.1.3.	Shield Ethernet w5100.....	28
3.1.4.	Módulo Xbee. ....	29
3.1.5.	Sensor DHT11.....	29
3.2.	Instalación y configuración del sistema de gestión en la VPS. ....	30
3.2.1.	Instalación de MongoDB.....	30
3.2.2.	Instalación del sistema thinger.io en VPS. ....	34
3.3.	Configuración de XBee para enlace de datos.....	36
3.4.	Implementación y configuración de sensores con Arduino y XBee. ....	39
3.4.1.	Configuración de sensores con arduino y módulos XBee. ....	39
3.4.2.	Configuración de servidor VPS para la presentación y recepción de datos de los sensores.....	43
	Conclusiones.....	47
	Recomendaciones.....	48
	Bibliografía.....	49

## ÍNDICE DE FIGURAS

### Capítulo 2:

Figura 2. 1: Esquemático de la red punto a punto (Peer-to-Peer, P2P). .....	19
Figura 2. 2: Esquemático de la red estrella. ....	20
Figura 2. 3: Esquemático de la red árbol. ....	20
Figura 2. 4: Esquemático de la red enmallada. ....	21

### Capítulo 3:

Figura 3. 1: VPS que proporciona la empresa DigitalOcean. ....	26
Figura 3. 2: Selección del sistema operativo. ....	26
Figura 3. 3: Selección del sitio donde se alojará la VPS. ....	27
Figura 3. 4: Placa electrónica Arduino UNO. ....	27
Figura 3. 5: Pinout Arduino uno. ....	28
Figura 3. 6: Shield Ethernet. ....	28
Figura 3. 7: Descarga de actualización del sistema operativo ROS. ....	29
Figura 3. 8: Sensor DHT11. ....	30
Figura 3. 9: Configuración de Putty. ....	31
Figura 3. 10: Putty. ....	31
Figura 3. 11: Putty. ....	32
Figura 3. 12: Descarga e instalación de MongoDB. ....	32
Figura 3. 13: Líneas de comandos para mongodb.service. ....	33
Figura 3. 14: Verificación de status de MongoDB. ....	33
Figura 3. 15: Instalación del aplicativo thinger.io. ....	34
Figura 3. 16: Verificación de estado de thinger.io. ....	34
Figura 3. 17: Interfaz de acceso a la plataforma web. ....	35
Figura 3. 18: Registro de nuevo usuario. ....	35
Figura 3. 19: Panel de control de thinger.io. ....	36
Figura 3. 20: X-CTU configuración de XBee. ....	36
Figura 3. 21: Configuración del módulo de nodo router. ....	37
Figura 3. 21: Configuración del módulo de nodo coordinador. ....	38
Figura 3. 23: Terminal de X-CTU para visualizar datos recibidos. ....	38
Figura 3. 24: Diagrama de proyecto propuesto. ....	39

Figura 3. 25: Diagrama de proyecto propuesto .....	39
Figura 3. 28: Declaración de variable y función setup. ....	40
Figura 3. 29: Diagrama utilizado para el transmisor. ....	40
Figura 3. 30: Encabezado del programa receptor y proceso de datos. ....	41
Figura 3. 31: Función Setup del programa. ....	41
Figura 3. 32: Función Setup del programa. ....	42
Figura 3. 33: Prototipo del transmisor. ....	42
Figura 3. 34: Prototipo del receptor. ....	43
Figura 3. 35: Crear dispositivo virtual en thinger. ....	43
Figura 3. 36: Crear dispositivo virtual en thinger. ....	44
Figura 3. 37: Crear dashboard en thinger. ....	45
Figura 3. 38: Dashboard en servidor VPS con thinger. ....	45

## ÍNDICE DE TABLAS

### Capítulo 2:

Tabla 2. 1: Ejemplos de los diferentes tipos de sensores utilizados en WSN. .....	17
Tabla 2. 2: Comparativa entre diferentes componentes utilizados como nodos en WSNs. ....	22

## Resumen

Una red de sensores inalámbricos (*Wireless Sensor Network, WSN*) es una combinación de hardware y software, en el cual ambos tienen un papel importante que jugar en la red. En el componente práctico del examen complejo se presenta el hardware utilizado, así como el software. Aunque individualmente tienen menos importancia, pero juntos pueden tener un gran impacto en el rendimiento de cualquier red. Se eligió trabajar con un número limitado de nodos. El hardware es la parte crucial para el componente práctico. Se eligieron componentes que sean confiables, funcionales y prácticos, pero más económicos. Después de explorar varias opciones, se escogieron los dispositivos electrónicos XBee y Arduino UNO que resultan una muy buena combinación para el presente trabajo. La tarjeta embebida Arduino UNO ha sido una parte integral del movimiento de código abierto, y XBee ha estado disponible durante algunos años como un componente de frecuencia de radio pequeño pero efectivo.

**Palabras claves: SENSORES, REDES, TOPOLOGÍAS, XBEE, ESTANDARES, WSN.**

## CAPÍTULO 1: DESCRIPCIÓN DEL COMPONENTE PRÁCTICO

### 1.1. Introducción.

De acuerdo con Sun, Lu, Wang, & Lin, (2010) la información es importante, especialmente cuando se trata de situaciones como el estado de salud de los pacientes. Mientras que Mohapatra, (2018) indica que otra aplicación es el seguimiento de los enemigos de los militares. Por ejemplo, el suministro de información sobre el nivel del agua para la prevención de inundaciones es también una aplicación de WSN. Estos son solo algunos de los principales ejemplos en los que se puede aprovechar las redes inalámbricas de sensores (WSN) para recibir información.

También hay muchas otras áreas donde WSN se usa para facilitar la vida. Se observa una mayor implementación de WSN del sector de la salud a la industria, desde la domótica hasta la seguridad nacional. Los investigadores ahora están invirtiendo sus recursos para su uso posterior en situaciones críticas, al proponer una plataforma estándar que sea más robusta, resistente e inteligente. Además, se desarrollan conceptos más nuevos para hacerlo más confiable, seguro y fácil de usar.

Una red de sensores inalámbricos multimedia, y los protocolos de enrutamiento juega un papel importante en el ahorro de los recursos limitados de los sensores. Permite a un nodo transmitir un contenido multimedia al nodo receptor, por ejemplo, una imagen. Además, WMSN se utiliza para el Internet de las Cosas (*Internet of Things, IoT*) para servir a diferentes propósitos a fin de hacer que nuestro mundo sea más inteligente. (Kbir & Bennani, 2016)

Uno de los desafíos en WSN es el suministro de energía. El suministro de energía de los nodos sensores es vital para la operación de WSN. En otras palabras, la energía es la propia vida de una WSN. Por ejemplo, en el procesamiento de imágenes, la tarea de un nodo sensor es capturar, procesar y encaminar las imágenes de la red hasta una estación de policía más cercana. (Shehu, Hulaj, & Bajrami, 2018)

## **1.2. Objetivo General.**

Realizar la Implementación de una red de sensores inalámbricos enlazados con Xbee utilizando la plataforma thinger.io. que permita la demostración del conocimiento práctico del examen complejo.

## **1.3. Objetivos Específicos.**

- a. Describir brevemente los fundamentos teóricos de las redes de sensores inalámbricos.
- b. Diseñar una WSN enlazados con módulos Xbee utilizando la plataforma Thinger.io.
- c. Obtener los resultados de los datos recibidos en los nodos sensores a través de la plataforma Thinger.io.

## CAPÍTULO 2: Fundamentación teórica.

### 2.1. Introducción del estándar IEEE 802.15.4.

El estándar IEEE 802.15.4 es el conjunto de especificaciones de facto recomendado para operar en redes de sensores inalámbricos (*Wireless Sensor Network, WSN*). La amplia gama de aplicaciones inspiradas en el advenimiento de WSN, junto con los diferentes requisitos de rendimiento de estas aplicaciones, han atraído a la comunidad investigadora a centrarse en mejorar el control de acceso al medio (*Medium Access Control, MAC*) IEEE 802.15.4 para mitigar sus deficiencias. El MAC de IEEE 802.15.4 ha estado bajo el interés de investigación desde hace más de una década, con diferentes objetivos que motivan a los diferentes grupos de investigación.

Estos objetivos incluyen principalmente la reducción del consumo de energía, la mejora de la utilización/rendimiento del canal, la mejora de la relación/fiabilidad de la entrega de paquetes, la reducción de la probabilidad de colisión y la reducción de los retardos de extremo a extremo. Para alcanzar estos objetivos, se han adoptado diferentes enfoques, para lo cual se revisa y discute estos enfoques en el presente capítulo.

Las WSNs pueden funcionar con cualquier cantidad de sensores según su tipo de detección. Se pueden clasificar por su capacidad de detección diferente. Como ejemplo, la tabla 2.1 muestra algunos sensores.

Tabla 2. 1: Ejemplos de los diferentes tipos de sensores utilizados en WSN.

Tipos de sensores	
Temperatura	Caudal
Presión	Posición
Óptico	Electromagnético
Acústico	Químico
Movimiento, vibración	Humedad
Radiación	

Elaborado por: Autor.

## **2.2. Aplicaciones de las redes de sensores inalámbricos.**

Las WSNs se pueden usar en muchos campos y aplicaciones. Estas aplicaciones son bastante diversas y no se limitan solo a la implementación industrial. Se han utilizado ampliamente en el sector de la salud, la garantía de la calidad del medio ambiente y la prevención de la intrusión militar. En esta sección se describen algunas de las principales contribuciones de WSN en diversas aplicaciones.

Las aplicaciones ambientales incluyen monitoreo de la calidad del aire, detección de incendios forestales, seguimiento y monitoreo de la vegetación. Las aplicaciones militares brindan muchos servicios diferentes, como la supervisión de fuerzas, equipos y municiones, y la detección de ataques nucleares, biológicos y químicos. La supervisión de la seguridad es fundamentalmente la misma que la del entorno, pero no recopila ningún dato. Busca anomalías y si encuentra algo, le da una señal al usuario para que tome las medidas necesarias.

En el cuidado de la salud, las WSNs se utilizan ampliamente para monitorear los datos fisiológicos de un paciente, controlar la administración de medicamentos y enviar datos en tiempo real a los interesados. Para las personas mayores, esto se puede usar para monitorear la detección de caídas, la detección de la inconsciencia y el monitoreo del ejercicio. Las WSNs en aplicaciones domésticas incluyen monitoreo de temperatura, control de intrusión, activación de actividad basada en eventos, conmutación de luz automática y más.

## **2.3. Topologías de las redes de sensores inalámbricos.**

Junto con el avance en las tecnologías de redes inalámbricas de sensores, también se produjeron cambios importantes en la distribución de la red. Se han introducido muchas formas de redes para llevar a cabo diferentes actos y tienen diferentes ventajas y limitaciones. Aquí se resumen algunas topologías de renombre que se utilizan en las redes de sensores inalámbricos.

### 2.3.1. Redes punto a punto (Peer-to-Peer, P2P).

Esta topología es de naturaleza ad-hoc y admite la comunicación de nodo a nodo. Un nodo puede comunicarse directamente con cualquier otro nodo siempre que estén dentro del alcance. Si no están dentro del alcance, un nodo también se puede comunicar a través de otros nodos. El aspecto importante de esta red P2P es su autoorganización y autorrecuperación (la capacidad de recuperación de fallas).

Esta topología puede implementarse en la supervisión de la industria o en el seguimiento de inventario, ya que los datos pasan directamente de un nodo a otro en el menor tiempo posible. Los nodos actúan no solo como un receptor o un emisor, sino también como un enrutador en este tipo de topología de red. La figura 2.1 muestra una red punto a punto.

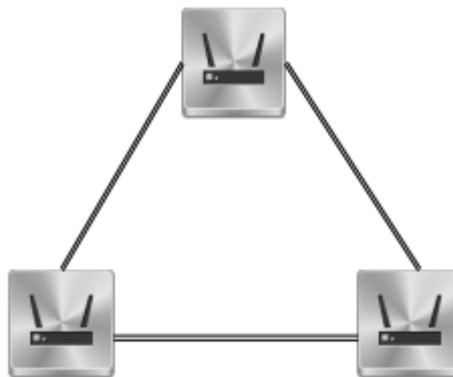


Figura 2. 1: Esquemático de la red punto a punto (Peer-to-Peer, P2P).  
Fuente: (Raj P. M., Mohan, & Srinivasa, 2018)

### 2.3.2. Redes estrella.

A diferencia de una red punto a punto, en una red estrella, los nodos no se pueden comunicar entre sí. Esta red tiene un nodo coordinador que funciona como un nodo principal y todos los demás nodos se conectan con él. Entonces, todos los datos deberían ir al nodo coordinador primero. En una red más grande, los nodos de los sensores usualmente funcionan con una batería, pero el nodo coordinador recibe energía directamente de la electricidad para mantener la demanda de comunicación con otros nodos de sensores para obtener datos ininterrumpidos. Aunque los nodos de sensor se conectan al nodo coordinador, son operadores independientes.

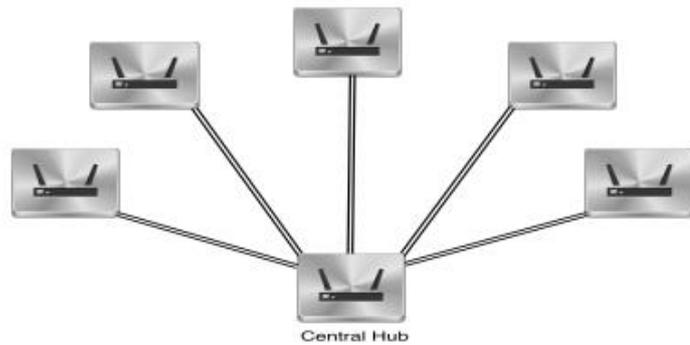


Figura 2. 2: Esquemático de la red estrella.  
Fuente: (Maier, 2004; Narmada & Rao, 2012)

### 2.3.3. Redes árbol.

La topología de árbol a menudo se denomina estrella híbrida y topología de red punto a punto. Un concentrador central se conecta con otros nodos, así como con un nodo raíz y, por lo tanto, hace que la situación sea similar a la topología de árbol. Los nodos se conectan con un concentrador central y luego desde el concentrador central, una señal va al nodo raíz. Entonces, cuando un nuevo nodo quiere unirse a esta red, primero tiene que conectarse con el concentrador central. La figura 2.3 muestra un ejemplo de una topología de red en árbol para facilitar la comprensión del concepto.

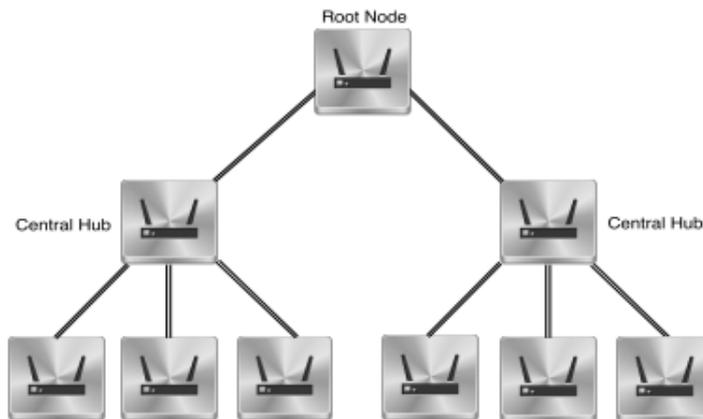


Figura 2. 3: Esquemático de la red árbol.  
Fuente: (Eiselt & Sandblom, 2000)

### 2.3.4. Redes enmalladas.

Una red enmallada permite que los nodos se comuniquen de un extremo a otro saltando a través de otros nodos, lo que hace que se autorrecupere. Cada nodo está conectado con otros nodos disponibles en la misma área. Una topología de malla puede ser muy costosa debido a su estructura inclusiva.

Es la estructura más compleja entre las topologías de red. La figura 2.4 muestra la estructura de una red enmallada.

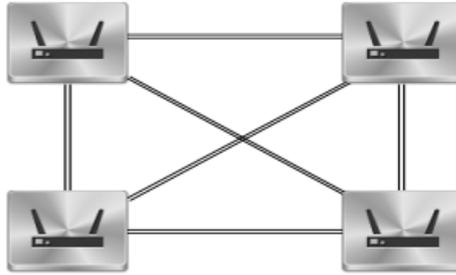


Figura 2. 4: Esquemático de la red enmallada.  
Fuente: (Ahmed, Butt, Khan, & Mehmood, 2016)

## 2.4. XBee y sus estándares.

XBee es el nombre del módem de radiofrecuencia (*Radio Frequency, RF*) producido por Digi International que funciona como un transceptor inalámbrico. Antes de hablar de XBee en detalle en el capítulo 3, primero se presenta el estándar y el protocolo en el que se basa.

### 2.4.1. Estándar IEEE 802.15.4.

El Instituto de Ingenieros Eléctricos y Electrónicos (*Institute of Electrical and Electronic Engineers, IEEE*) introdujo el estándar 802.15.4 para cumplir con el requisito de un cierto segmento de comunicación que necesita una velocidad de datos baja, baja potencia y conectividad simple. El estándar IEEE 802.15.4 especifica que la comunicación debe ocurrir en el rango de 5 MHz y en la banda de 2.4 GHz.

### 2.4.2. ZigBee.

ZigBee es un protocolo que fue presentado por ZigBee Alliance, una organización que se ocupa del desarrollo de un protocolo para aplicaciones de baja velocidad de datos. ZigBee fue presentado como una mejora del estándar 802.15.4. La topología de malla es el uso más popular en WSN para ZigBee.

### 2.5.3. XBee.

Basado en estos dos estándares y protocolos, Digi International presentó dos series diferentes de XBee. XBee 802.15.4 (Serie 1) está desarrollado para

cumplir con el requisito IEEE 802.15.4 para admitir comunicaciones punto a punto de bajo costo. Mientras que, los módulos XBee ZNet2.5 (Serie 2) proporcionan malla compatible con ZigBee. El presente trabajo se utiliza XBee 802.15.4 para comunicación por radio.

## 2.6. Motivo del uso del módulo XBee y Arduino UNO.

Antes de decidir utilizar XBee y Arduino UNO para el presente trabajo, se compararon los módulos con diferentes parámetros que tienen mayor prioridad en el presente componente práctico. Como ejemplo, cuando se consideró un módulo, el tamaño y el peso tenían menor prioridad. Se puso mayor énfasis en módulos económicos, en especial el de código abierto, flexible y que cumpla con el propósito del trabajo. Después de analizar estos parámetros, que se muestran en la tabla 2.2, es evidente que el módulo XBee y Arduino UNO son las opciones más óptimas para el propósito de la investigación.

Tabla 2. 2: Comparativa entre diferentes componentes utilizados como nodos en WSNs.

<b>Factores</b>	<b>TelosB TPR2400CA</b>	<b>Micaz</b>	<b>IRIS</b>	<b>Waspnote</b>	<b>Xbee- Arduino</b>
<b>Precio (USD)</b>	110	400	110	215	50
<b>Tasa de datos (Tx)</b>	250 kbps	250 kbps	250 kbps	250 kbps	250 kbps
<b>Alcance en interiores (Indoor)</b>	20m a 30m	20m a 30m	> 50m	30m	30m
<b>Alcance en exteriores (Outdoor)</b>	75m a 100m	75m a 100m	> 300m	250m	90m
<b>RAM</b>	10 KBytes	4 KBytes	8 KBytes	8 KBytes	2 Kbytes
<b>EEPROM</b>	16 KBytes	4 KBytes	4 KBytes	4 KBytes	1 KByte
<b>Programa de memoria flash</b>	48 KBytes	128 KBytes	128 KBytes	128 KBytes	32 KBytes
<b>Tamaño (mm)</b>	65x31x6	58x32x7	58x32x7	73.5x51x13	74x53x22
<b>Peso (g)</b>	23	18	18	20	5+28=33

Elaborado por: Autor.

El módulo XBee y Arduino de código abierto, son económicos y no se han utilizado ampliamente en redes inalámbricas de sensores con fines de

investigación. Se buscaba algo con memoria o velocidad más pequeña, para poder utilizar el código más eficiente, en lugar de algo con memoria más grande o velocidad más alta. La mayor ventaja del módulo XBee y Arduino es que son personalizables. De esta manera, diferentes partes del módulo pueden ser reemplazadas, agregadas y cambiadas fácilmente, a diferencia de los módulos especificados por el fabricante tales como TelosB, IRIS, Mica o MicaZ.

Como ejemplo, para los otros módulos, se debe usar un paquete completo (con procesador) incluso con una PC, aunque una PC tenga un procesador potente, y el pequeño procesador en ese módulo es un desperdicio de un recurso. En XBee y Arduino UNO, XBee se puede desmontar de Arduino UNO (el procesador) y solo XBee (que es el transceptor) debe utilizarse con una PC. De esta manera, XBee utiliza la mayor potencia de procesador de la PC. Lo mismo se aplica a los sensores, como en el módulo XBee y Arduino UNO. Se pueden usar varios sensores simplemente conectando los sensores con Arduino UNO. Para otros módulos específicos del fabricante, un usuario debe comprar un dispositivo separado con un sensor por separado que de nuevo es un desperdicio de un procesador.

La tabla 2.2 muestra diferentes módulos que se comparan entre sí según algunos parámetros. Si se dispone cuenta el precio, existe una clara diferencia entre las motas fabricadas por Crossbow Technology y las demás, aunque la velocidad de transmisión (TX) es exactamente la misma para todas. La velocidad de datos es un aspecto importante para un dispositivo porque esta es la velocidad a la que los datos se transfieren de un extremo a otro. La mayoría de los dispositivos para interiores y exteriores son casi iguales. Si se considera a IRIS por su mejor alcance, se debe recordar que su fabricante mejoró su gama para exteriores.

La tabla 2.2 muestra diferentes módulos que se comparan entre sí según algunos parámetros. Si se tiene en cuenta el precio, existe una clara diferencia entre los dispositivos fabricadas por Crossbow Technology y las demás,

aunque la velocidad de transmisión (Tx) es exactamente la misma para todas. La velocidad de datos es un aspecto importante para un dispositivo porque esta es la velocidad a la que los datos se transfieren de un extremo a otro. La mayoría de los dispositivos para interiores y exteriores son casi iguales. Si se considera a IRIS por su mejor alcance, se debe recordar que su fabricante mejoró su gama para exteriores.

## **Capítulo 3: Implementación de una red de sensores inalámbricos enlazados con Xbee utilizando la plataforma thinger.io.**

En la actualidad para desarrollar un sistema de sensores inalámbricos utilizando las tecnologías actuales es muy fácil, pero se tienen varios obstáculos como son las interferencias, limitación del espectro radioeléctrico, solapamiento, es por ello por lo que es necesario implementarlo eficientemente utilizando dispositivos accesibles y de mejor rendimiento en el mercado. En este capítulo se describirá el proceso de implementación de una red de sensores inalámbricos utilizando el módulo de comunicación XBee que permita enviar los datos a un servidor virtual privado para analizar y almacenar los datos en la nube.

### **3.1. Elementos utilizados para el montaje de la red de sensores inalámbricos.**

En esta sección del capítulo se detalla los elementos a utilizar para el proyecto que se presenta.

#### **3.1.1. Servidor Virtual Privado (VPS)**

Un servidor virtual privado no es más que una sección de un servidor físico que se puede designar los recursos necesarios para ello, teniendo varios servidores en uno solo. Cada uno puede tener su propio sistema operativo independiente proporcionando la administración y acceso al usuario. En el mercado existen varias empresas que proporcionan este tipo de servicios a precios competitivos, entre ellas se encuentra la empresa DigitalOcean.

En la figura 3.1 se puede observar la adquisición de una VPS a un costo de 5 dólares mensuales con 1 GB de memoria RAM, 25 GB de almacenamiento en un disco duro de estado sólido, 1vCPU, y transferencia de datos de 1 TB.

## Choose a size

### Standard Droplets

Balanced virtual machines with a healthy amount of memory tuned to host and scale applications like blogs, web applications, testing / staging environments, in-memory caching and databases.

MEMORY	vCPUs	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	\$5/mo \$0.007/hr
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr
3 GB	1 vCPU	60 GB	3 TB	\$15/mo \$0.022/hr
2 GB	2 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
1 GB	3 vCPUs	60 GB	3 TB	\$15/mo \$0.022/hr
4 GB	2 vCPUs	80 GB	4 TB	\$20/mo \$0.030/hr

Figura 3. 1: VPS que proporciona la empresa DigitalOcean.  
Elaborado por: Autor.

Una vez seleccionado el VPS necesario para el proyecto, se necesita definir el sistema operativo y de que parte del mundo se desea que este alojado, como se muestra en la figura 3.2 y 3.3, esta selección no afecta al costo ni al rendimiento del servidor.

### Choose an image ?

Distributions Container distributions One-click apps Snapshots

 Ubuntu 16.04.5 x64	 FreeBSD Select version	 Fedora Select version	 Debian Select version	 CentOS Select version
--	--	---	---	---

Figura 3. 2: Selección del sistema operativo.  
Elaborado por: Autor.

Choose a datacenter region

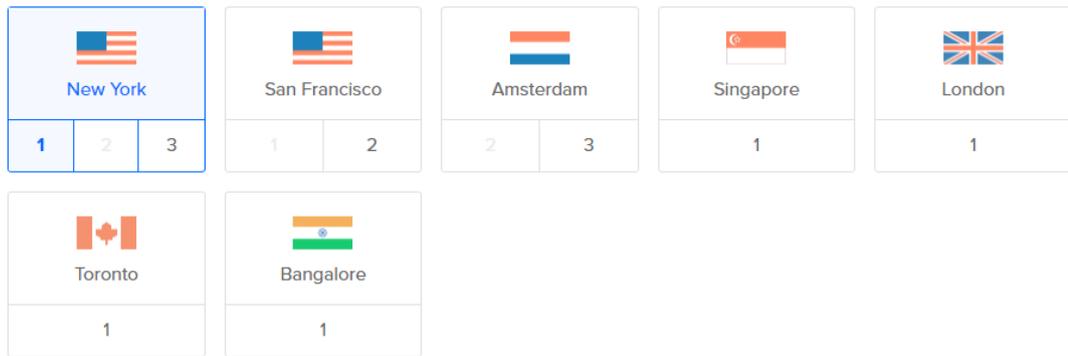


Figura 3. 3: Selección del sitio donde se alojará la VPS.  
Elaborado por: Autor.

Cuando se finaliza la creación del servidor llegara al correo electrónico las credenciales que permitirán el acceso por cualquier terminal utilizando el protocolo SSH puerto 22 al terminal donde se administrara los recursos y aplicaciones que se deseen o necesiten.

### 3.1.2. Arduino uno.

Las placas embebidas están últimamente siendo muy utilizadas en los proyectos de IoT y otras aplicaciones, es por eso que se decidió utilizar este tipo de dispositivos ya que ayudan mucho en la simplificación de hardware y software por su gran aporte que tiene por parte de la comunidad. La placa Arduino contiene un microcontrolador Atmega 328p, siendo este muy superior a otros conocidos en varios aspectos, entre ellos son las interrupciones, las salidas de PWM, entradas analógicas e información. En la figura 3.4 se puede apreciar su presentación.

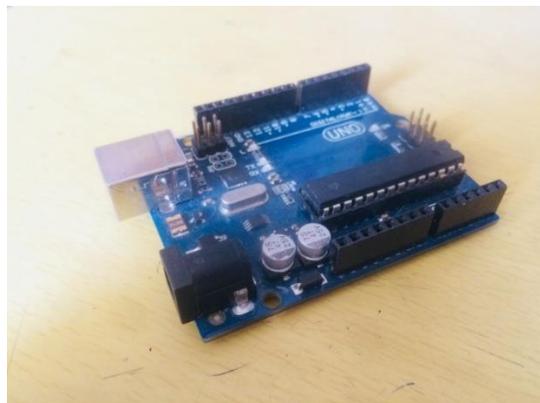


Figura 3. 4: Placa electrónica Arduino UNO.  
Elaborado por: Autor.

Una de las ventajas de esta placa embebida aparte de su robusto microcontrolador, es la cantidad de Shield y módulos que existen en el mercado para implementar cualquier tipo de prototipos. En la figura 3.5 se puede apreciar el pinout de la placa, con esta información es posible tener una idea de las aplicaciones que se pueden realizar utilizando arduino.

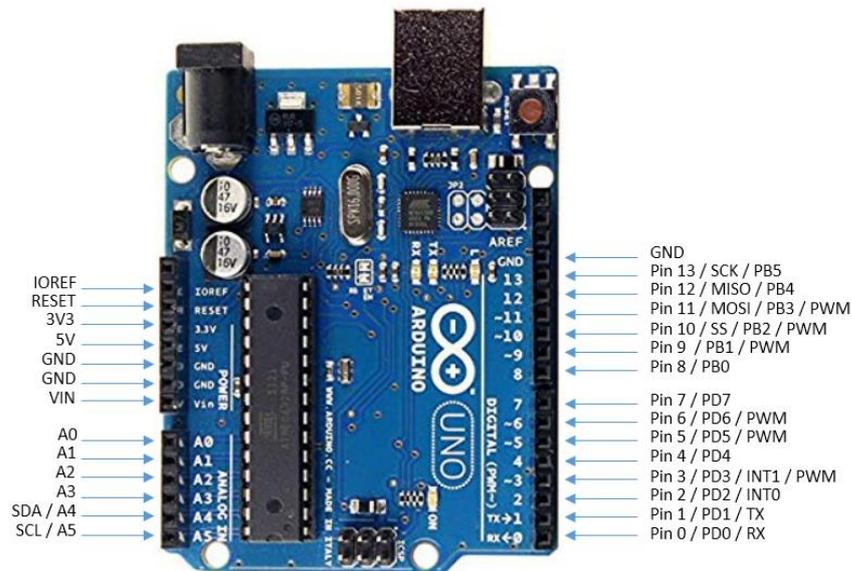


Figura 3. 5: Pinout Arduino uno.  
Elaborado por: Autor.

### 3.1.3. Shield Ethernet w5100

El módulo de ethernet w5100 es una placa diseñada para trabajar directamente con arduino uno proporcionando la capacidad de conectar a una red ethernet. El Shield (véase la figura 3.6) posee un conector RJ45 estándar y un socket para tarjeta SD, además tiene conectores que se acoplan directamente con el diseño de la placa embebida Arduino UNO, y para su comunicación con el microcontrolador utiliza los pines 10, 11, 12, 13, SPI.



Figura 3. 6: Shield Ethernet.  
Elaborado por: Autor.

Este dispositivo posee el chip ethernet W5100, de la empresa WIZnet facilitando la conectividad a internet sin necesidad de un sistema operativo, entre sus aplicaciones se puede resumir en la aplicación de conectividad de dispositivos pequeños, comunicación serial – ethernet, usb – ethernet, sistemas de seguridad, monitoreo de equipos.

#### **3.1.4. Módulo Xbee.**

Los Xbee (véase figura 3.7) son módulos de comunicaciones inalámbricas utilizando el protocolo IEEE 802.15.4, estos pueden crear redes punto a punto y punto a multipunto, estos fueron diseñados para un alto tráfico de datos y baja latencia, y tienen un gran alcance.

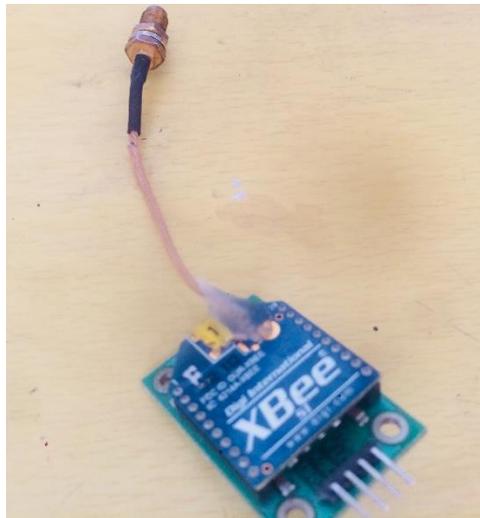


Figura 3. 7: Descarga de actualización del sistema operativo ROS.  
Elaborado por: Autor.

Las distancias de transmisión dependen mucho de la serie con la que se trabaje, tanto la serie 1 y 2 están disponibles con dos diferentes potencias, siendo la PRO la más avanzada llegando hasta 24 kilómetros según las condiciones.

#### **3.1.5. Sensor DHT11.**

Este proyecto del componente práctico está orientado a la utilización de sensores inalámbricos, pero al ser así el costo de implementación se elevaría, es por ello que se decidió utilizar el DHT11 que no es más que un sensor de temperatura y humedad de bajo costo que gracias a un microcontrolador y el

módulo XBee detallado anteriormente se puede transferir estos datos inalámbricamente.

El sensor DHT11 es digital por lo tanto no preocupa mucho el tema de pérdida de datos por ruido, este se comunica por un pin digital y el microcontrolador interpreta los resultados para transmitirlos inalámbricamente al XBee. Entre las características de este sensor (véase la figura 3.8) están el bajo consumo de energía siendo apenas de 2.5 mA con una alimentación que puede variar entre 3.5 V y 5V. El rango de temperatura va desde 0°C a 50°C, con una precisión de  $25^{\circ}\text{C} \pm 2^{\circ}\text{C}$ , y su humedad el rango va de 20% RH a 90% RH con una precisión entre  $0^{\circ}\text{C}$  y  $50^{\circ}\text{C} \pm 5\% \text{ RH}$ .

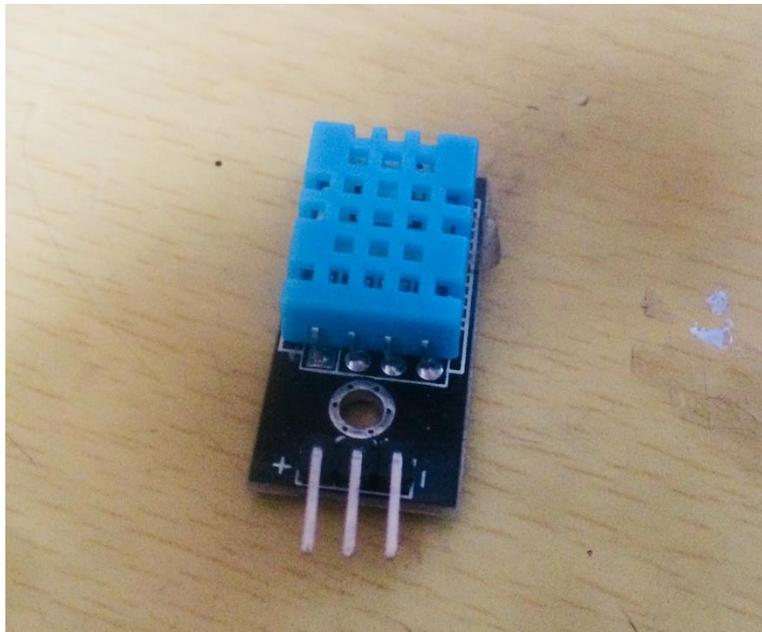


Figura 3. 8: Sensor DHT11.  
Elaborado por: Autor.

### **3.2. Instalación y configuración del sistema de gestión en la VPS.**

En esta sección del capítulo se describirá la instalación y configuración del servicio de thinger.io en una VPS con sistema operativo Ubuntu server.

#### **3.2.1. Instalación de MongoDB.**

El proceso de preparación de la VPS a utilizar en el proyecto se realizó remotamente utilizando el software Putty (véase figura 3.8) con el que se tiene acceso via SSH con las credenciales correspondientes.

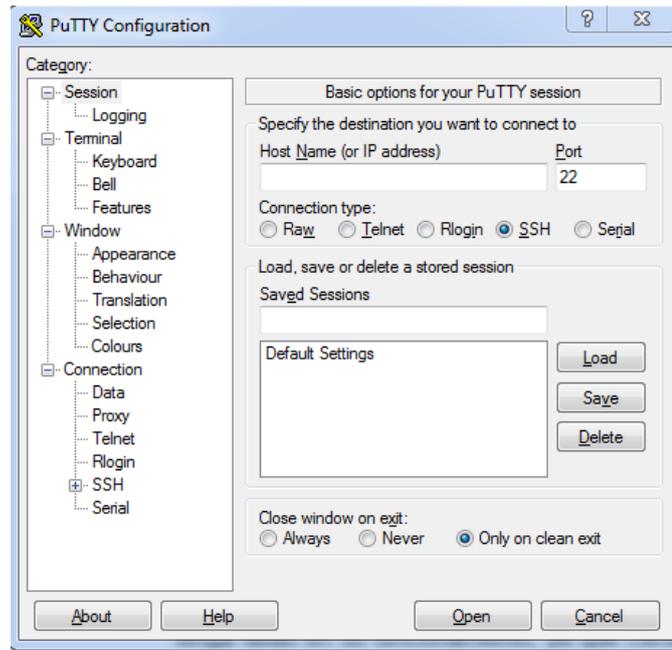


Figura 3. 9: Configuración de Putty.  
Elaborado por: Autor.

Lo primero que se realizó una vez ingresado a la terminal fue actualizar el sistema operativo para tener al día los repositorios ya que este sistema operativo está en constante actualización gracias al aporte de la comunidad de software libre, tal como se muestra en la figura 3.10.

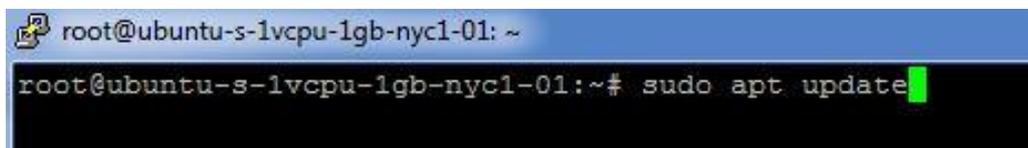
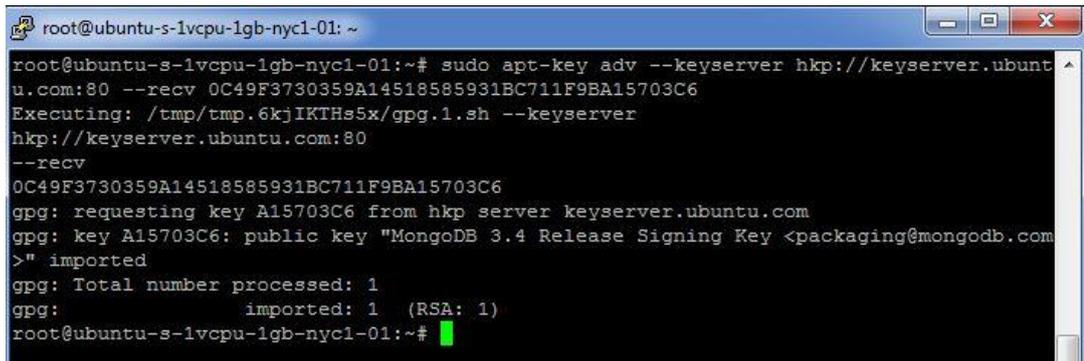


Figura 3. 10: Putty.  
Elaborado por: Autor.

A continuación, se debe proporcionar a la lista de repositorios las llaves que permitirán la instalación de MongoDB, sin estas credenciales es imposible realizar dicho proceso, en la figura 3.11 se presenta el resultado de este proceso.

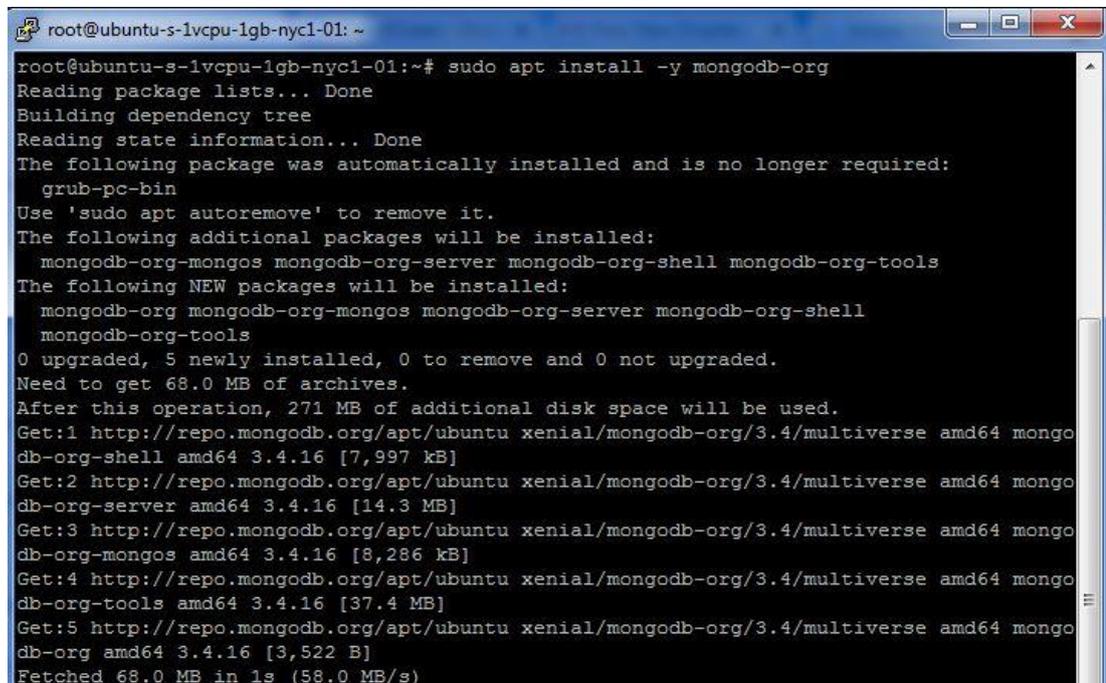
Una vez actualizado el sistema operativo y obtenido las llaves para la aplicación se procede a realizar la instalación del sistema MongoDB que será la encargada de almacenar los datos recibidos por los sensores en el servidor, con este recurso es posible tener resultados anteriores a la fecha y hora

proporcionando un análisis de los datos enviados por el sensor, tal como se muestra en la figura 3.12.



```
root@ubuntu-s-1vcpu-1gb-nyc1-01: ~  
root@ubuntu-s-1vcpu-1gb-nyc1-01:~# sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC711F9BA15703C6  
Executing: /tmp/tmp.6kjIKTHs5x/gpg.1.sh --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC711F9BA15703C6  
gpg: requesting key A15703C6 from hkp server keyserver.ubuntu.com  
gpg: key A15703C6: public key "MongoDB 3.4 Release Signing Key <packaging@mongodb.com>" imported  
gpg: Total number processed: 1  
gpg: imported: 1 (RSA: 1)  
root@ubuntu-s-1vcpu-1gb-nyc1-01:~#
```

Figura 3. 11: Putty.  
Elaborado por: Autor.

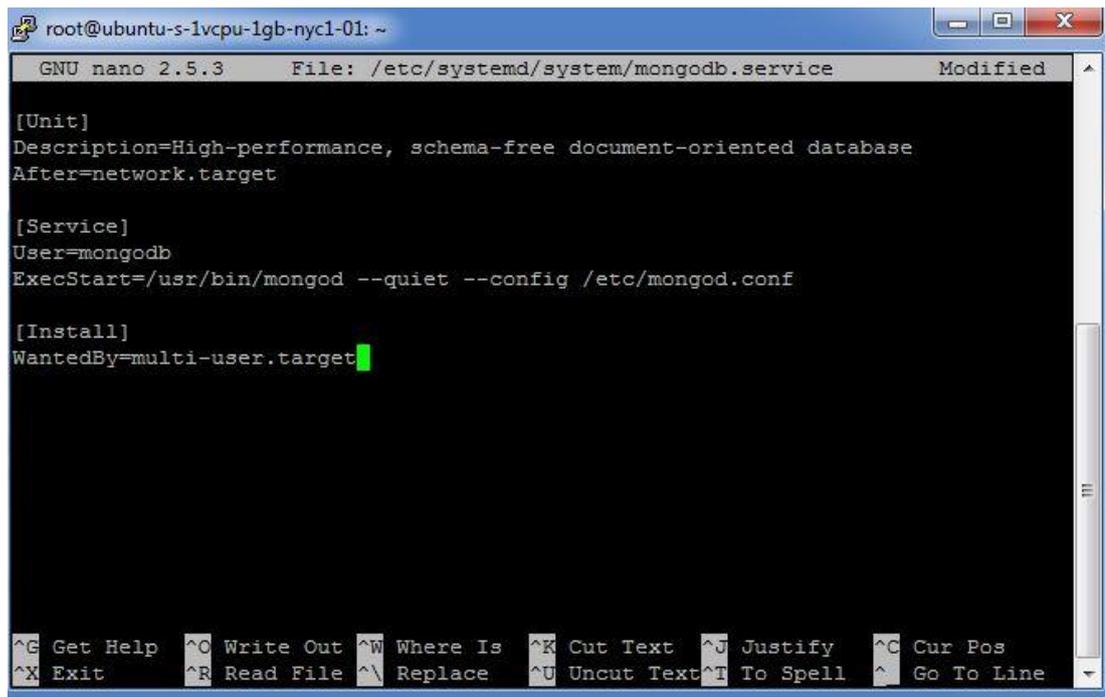


```
root@ubuntu-s-1vcpu-1gb-nyc1-01: ~  
root@ubuntu-s-1vcpu-1gb-nyc1-01:~# sudo apt install -y mongodb-org  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  grub-pc-bin  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  mongodb-org-mongos mongodb-org-server mongodb-org-shell mongodb-org-tools  
The following NEW packages will be installed:  
  mongodb-org mongodb-org-mongos mongodb-org-server mongodb-org-shell  
  mongodb-org-tools  
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.  
Need to get 68.0 MB of archives.  
After this operation, 271 MB of additional disk space will be used.  
Get:1 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4/multiverse amd64 mongo  
db-org-shell amd64 3.4.16 [7,997 kB]  
Get:2 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4/multiverse amd64 mongo  
db-org-server amd64 3.4.16 [14.3 MB]  
Get:3 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4/multiverse amd64 mongo  
db-org-mongos amd64 3.4.16 [8,286 kB]  
Get:4 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4/multiverse amd64 mongo  
db-org-tools amd64 3.4.16 [37.4 MB]  
Get:5 http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4/multiverse amd64 mongo  
db-org amd64 3.4.16 [3,522 B]  
Fetched 68.0 MB in 1s (58.0 MB/s)
```

Figura 3. 12: Descarga e instalación de MongoDB.  
Elaborado por: Autor.

Mongodb no es más que una base de datos orientada a documentos, es decir, que en vez de guardar los datos en registros como lo hace normalmente SQL simplemente los guarda en documentos siendo almacenados en BSON. Mongodb es muy diferente a las bases de datos relacionales, no es necesario seguir un esquema ya que los documentos de una colección pueden tener esquemas diferentes para almacenar clasificatoriamente los datos obtenidos por cualquier fuente.

Una de las características de este software es que está escrito en C++, pero las consultas se hacen en JSON como parámetro, dados que los propios documentos se almacenan, como ya se mencionó anteriormente en BSON. A continuación, se procede a crear un archivo de configuración de MongoDB para poder activar o desactivar el servicio cuando sea llamado. La figura 3.13 se presentan los comandos que se alojará en el directorio `/etc/systemd/system/mongodb.service`.



```
root@ubuntu-s-1vcpu-1gb-nyc1-01: ~
GNU nano 2.5.3 File: /etc/systemd/system/mongodb.service Modified
[Unit]
Description=High-performance, schema-free document-oriented database
After=network.target

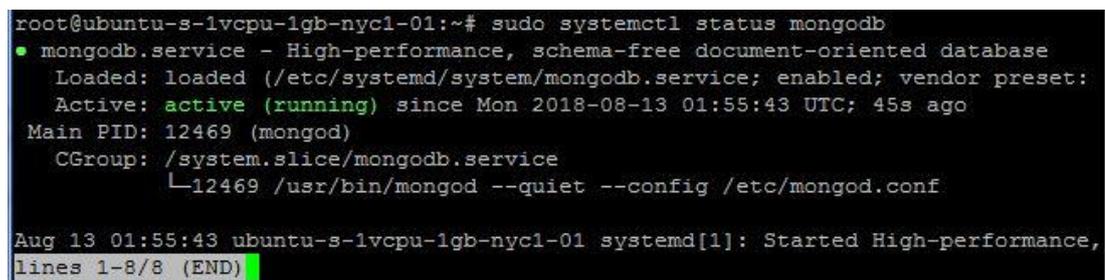
[Service]
User=mongodb
ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

[Install]
WantedBy=multi-user.target

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Figura 3. 13: Líneas de comandos para `mongodb.service`.  
Elaborado por: Autor.

Una vez instalado y configurado con el archivo creado recientemente se procede a activar el servicio con la línea de comando `sudo systemctl start mongodb.service`, al ser dado de alta en el servidor se podrá verificar el status de ello, tal como se muestra en la figura 3.14.



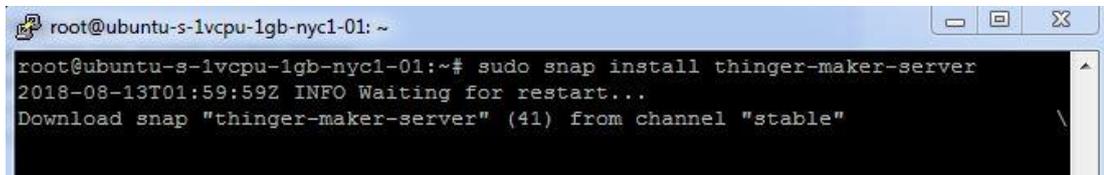
```
root@ubuntu-s-1vcpu-1gb-nyc1-01:~# sudo systemctl status mongodb
● mongodb.service - High-performance, schema-free document-oriented database
   Loaded: loaded (/etc/systemd/system/mongodb.service; enabled; vendor preset:
   Active: active (running) since Mon 2018-08-13 01:55:43 UTC; 45s ago
   Main PID: 12469 (mongod)
   CGroup: /system.slice/mongodb.service
           └─12469 /usr/bin/mongod --quiet --config /etc/mongod.conf

Aug 13 01:55:43 ubuntu-s-1vcpu-1gb-nyc1-01 systemd[1]: Started High-performance,
lines 1-8/8 (END)
```

Figura 3. 14: Verificación de status de MongoDB.  
Elaborado por: Autor.

### 3.2.2. Instalación del sistema thinger.io en VPS.

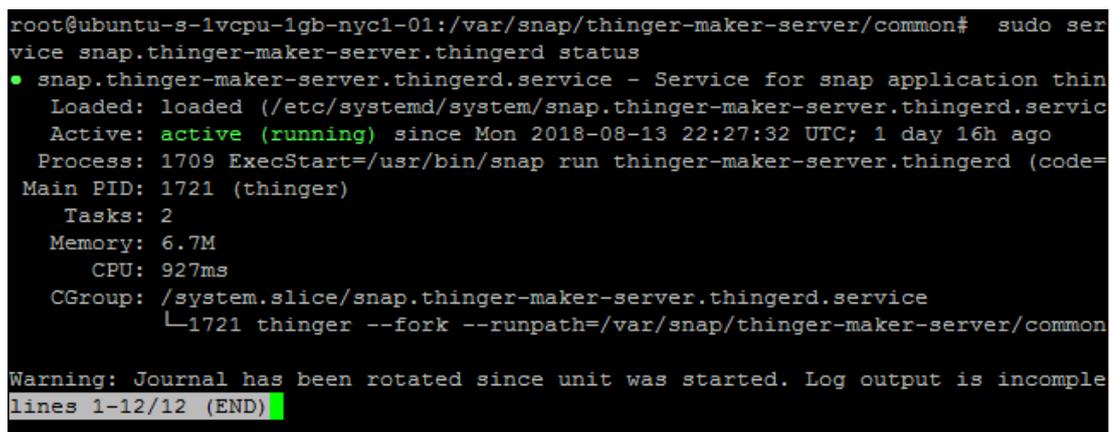
La instalación del paquete thinger.io es sencillo, pero es necesario realizarlo posteriormente a todo el proceso descrito en esta documentación, además es necesario configurar su interfaz personalizando los accesos para seguridad de los datos que se pueden almacenar en la base de datos del sistema. El sistema de gestión thinger.io es totalmente gratuita por lo que es un gran candidato para la utilización de este tipo de proyectos abaratando costos de implementación con resultados profesionales y eficientes como se pudo obtener en la investigación realizada. En la figura 3.15 se presenta la línea de comando que se requiere para la instalación del aplicativo de gestión thinger.io en la terminal del sistema operativo del servidor VPS.



```
root@ubuntu-s-1vcpu-1gb-nyc1-01: ~
root@ubuntu-s-1vcpu-1gb-nyc1-01:~# sudo snap install thinger-maker-server
2018-08-13T01:59:59Z INFO Waiting for restart...
Download snap "thinger-maker-server" (41) from channel "stable"
```

Figura 3. 15: Instalación del aplicativo thinger.io.  
Elaborado por: Autor.

Una vez concluida la instalación es posible determinar el status del servicio y tener información si está funcionando correctamente o si existe algún problema, como en el VPS se instaló los requerimientos necesarios, aunque previamente no deberá existir ningún tipo de inconsistencia, tal como se observa en la figura 3.16.



```
root@ubuntu-s-1vcpu-1gb-nyc1-01:/var/snap/thinger-maker-server/common# sudo ser
vice snap.thinger-maker-server.thingerd status
• snap.thinger-maker-server.thingerd.service - Service for snap application thin
  Loaded: loaded (/etc/systemd/system/snap.thinger-maker-server.thingerd.servic
  Active: active (running) since Mon 2018-08-13 22:27:32 UTC; 1 day 16h ago
  Process: 1709 ExecStart=/usr/bin/snap run thinger-maker-server.thingerd (code=
  Main PID: 1721 (thinger)
  Tasks: 2
  Memory: 6.7M
  CPU: 927ms
  CGroup: /system.slice/snap.thinger-maker-server.thingerd.service
          └─1721 thinger --fork --runpath=/var/snap/thinger-maker-server/common

Warning: Journal has been rotated since unit was started. Log output is incomple
lines 1-12/12 (END)
```

Figura 3. 16: Verificación de estado de thinger.io.  
Elaborado por: Autor.

Como se indicó anteriormente, el proceso es totalmente sencillo y casi automático, una vez realizado el proceso de instalación es necesario verificar

el servidor ingresando a un navegador web de cualquier lugar apuntando a la dirección ip del servidor VPS, si está todo bien pronto se presentará la ventana de acceso solicitando usuario y contraseña (ver figura 3.17).

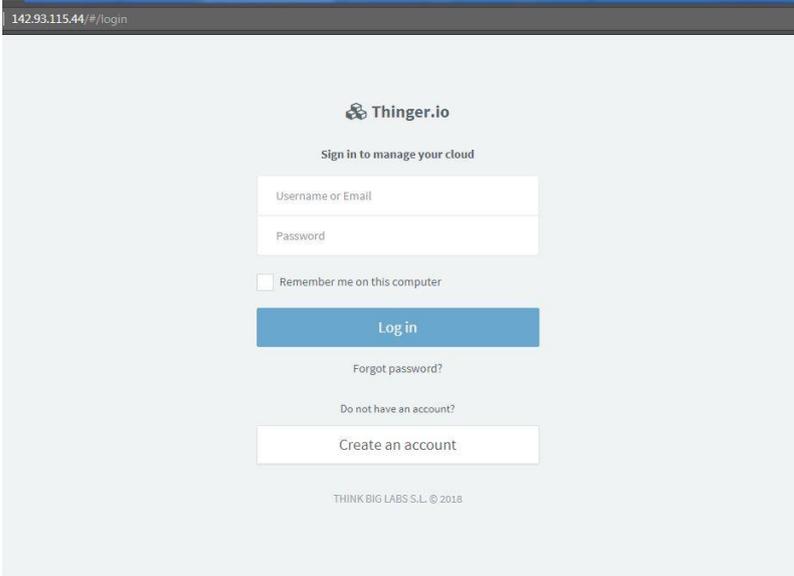
The image shows a web browser window with the address bar displaying '142.93.115.44/#/login'. The main content area features the Thinger.io logo at the top, followed by the text 'Sign in to manage your cloud'. Below this, there are two input fields: 'Username or Email' and 'Password'. A checkbox labeled 'Remember me on this computer' is positioned below the password field. A prominent blue 'Log in' button is centered below the checkbox. Underneath the button, there are two links: 'Forgot password?' and 'Do not have an account?'. At the bottom of the form area is a white 'Create an account' button. The footer of the page reads 'THINK BIG LABS S.L. © 2018'.

Figura 3. 17: Interfaz de acceso a la plataforma web.  
Elaborado por: Autor.

A continuación, es necesario la creación de una cuenta para la administración del sistema, para ello se accede a Create an account, y nos indica que se requiere llenar un pequeño registro (véase la figura 3.18).

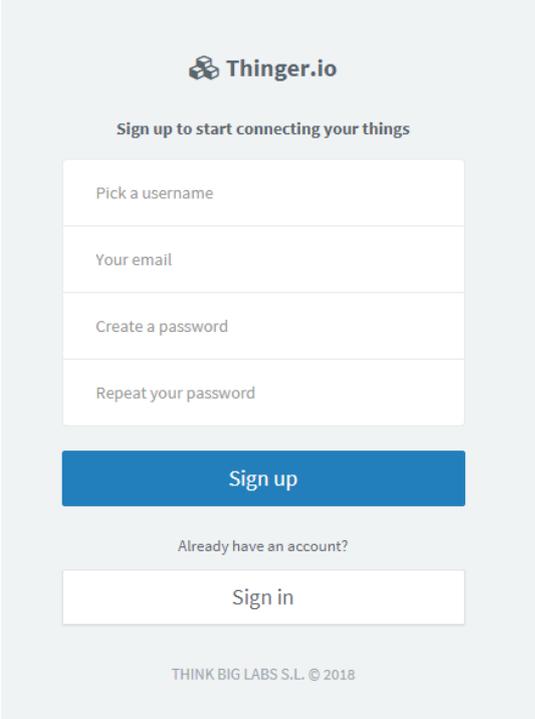
The image shows the registration page of Thinger.io. It features the Thinger.io logo at the top, followed by the text 'Sign up to start connecting your things'. The registration form consists of four input fields: 'Pick a username', 'Your email', 'Create a password', and 'Repeat your password'. A large blue 'Sign up' button is centered below these fields. Below the button, there is a link that says 'Already have an account?'. At the bottom of the form area is a white 'Sign in' button. The footer of the page reads 'THINK BIG LABS S.L. © 2018'.

Figura 3. 18: Registro de nuevo usuario.  
Elaborado por: Autor.

Al realizar el registro ya es posible acceder al panel de control de los dispositivos, recursos, bases de datos, entre otros, tal como se muestra en figura 3.19.

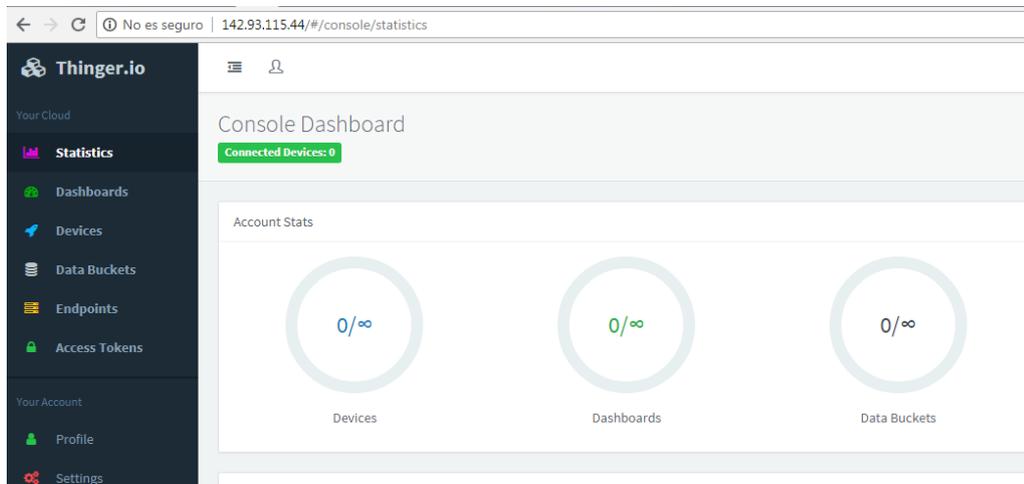


Figura 3. 19: Panel de control de thinger.io.  
Elaborado por: Autor.

### 3.3. Configuración de XBee para enlace de datos.

A continuación, se procede a describir el proceso realizado para la configuración de los módulos Xbee que permitan la comunicación entre ellos. Se conecta el XBee por medio de un convertidor serial USB a la computadora y con el software X-CTU se lee el puerto en el que se reconoció el dispositivo (véase la figura 3.20) que en este caso será el puerto COM6.

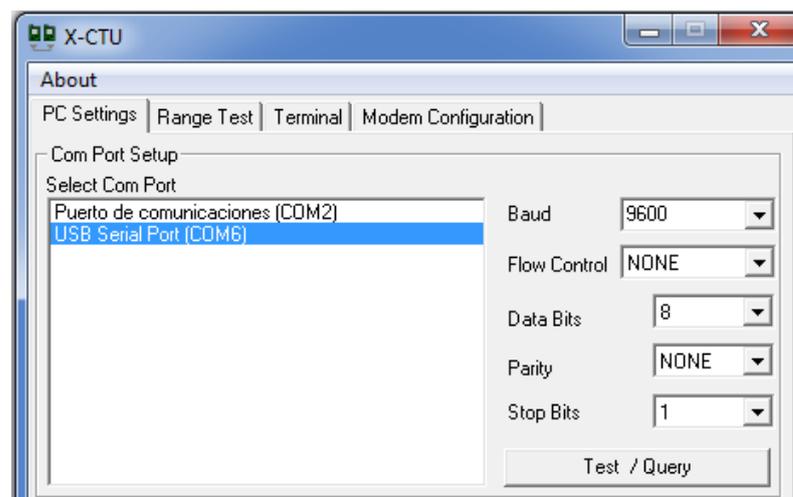


Figura 3. 20: X-CTU configuración de XBee.  
Elaborado por: Autor.

Cabe recalcar que los módulos XBee deben ser configurados correctamente como nodo coordinador y router para ello se accede al

apartado de modem configuration y una vez ahí se procede a leer la configuración actual del módulo donde se permite cambiar parámetros definiendo si es nodo coordinador o router. En la figura 3.21 se configura el módulo del nodo router quien será que envíe la información de los sensores al coordinador.

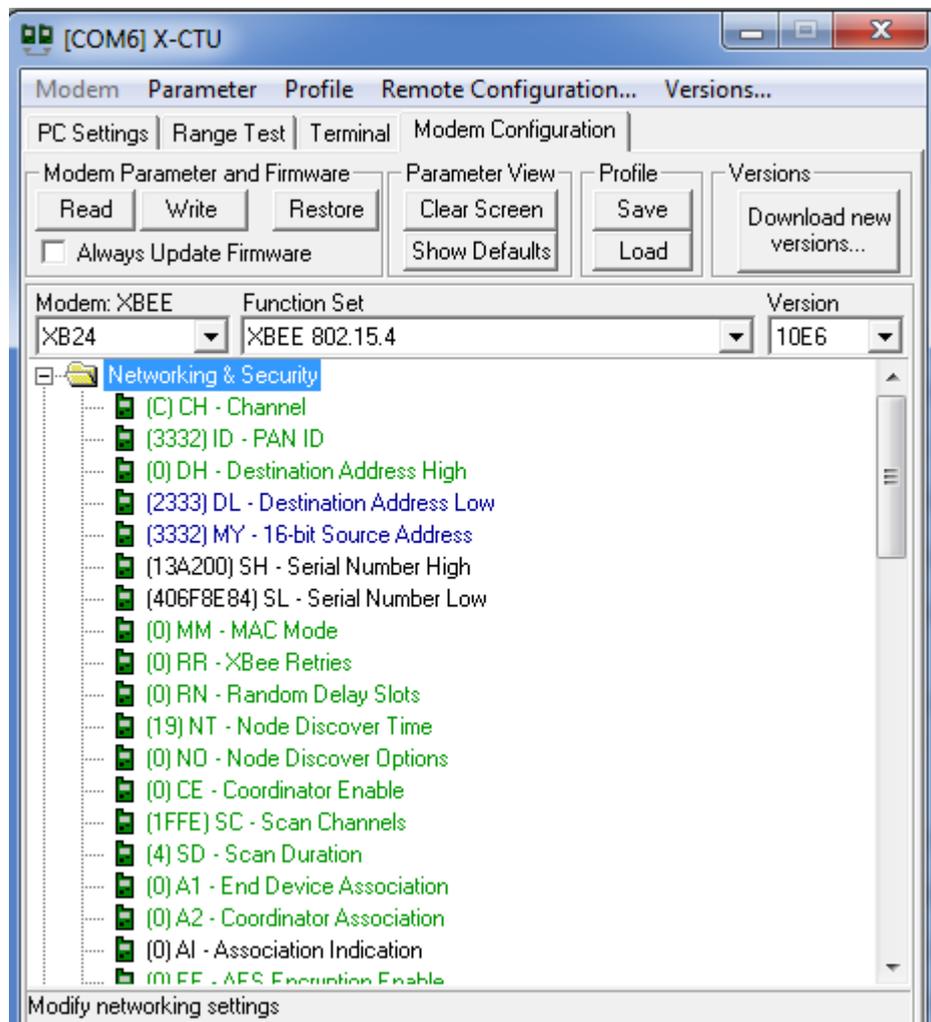


Figura 3. 21: Configuración del módulo de nodo router.  
Elaborado por: Autor.

Donde para ello se configura DL, MY, CE, con la numeración que se presenta en la figura 3.21, una vez concluido el proceso se realiza lo mismo con el siguiente modulo (véase la figura 3.22), que será el nodo coordinador de la red IEEE 802.15.4 y que recibirá la información de los sensores que a su vez después de un proceso de procesamiento enviará los datos por internet al servidor VPS.

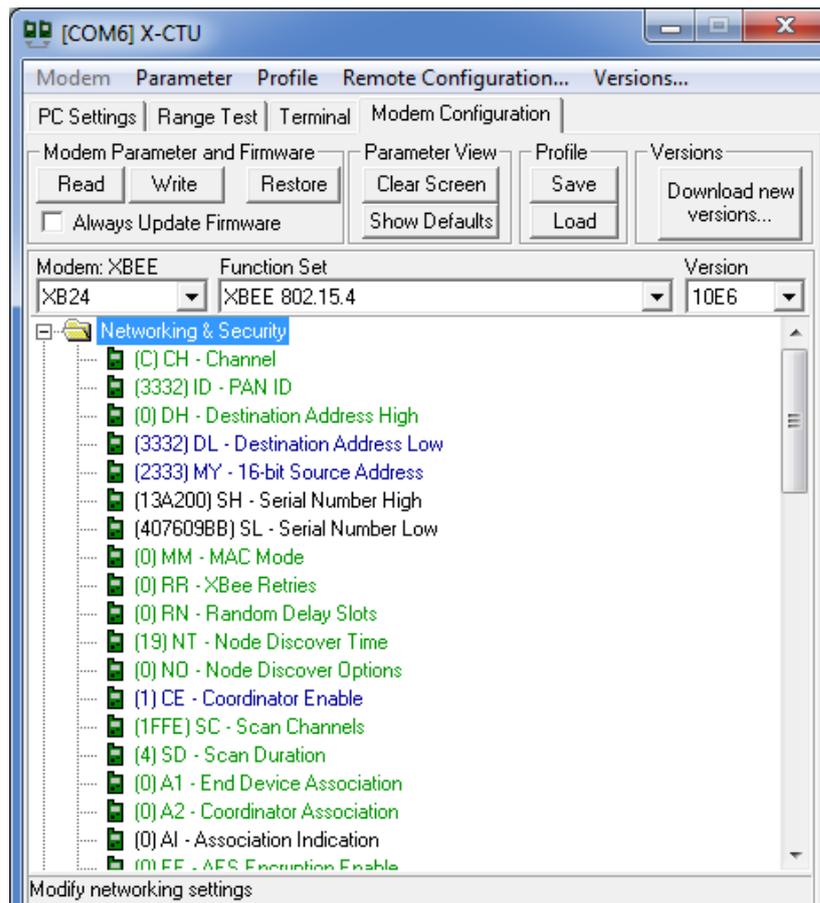


Figura 3. 22: Configuración del módulo de nodo coordinador.  
Elaborado por: Autor.

Al concluir el proceso de configuración de los módulos de transmisión de datos, XBee puede realizar una prueba de comunicación en él, transmitiendo ciertos caracteres desde un microcontrolador al nodo coordinador conectado al PC, en el apartado Terminal se puede observar los datos recibidos (ver figura 3.23).

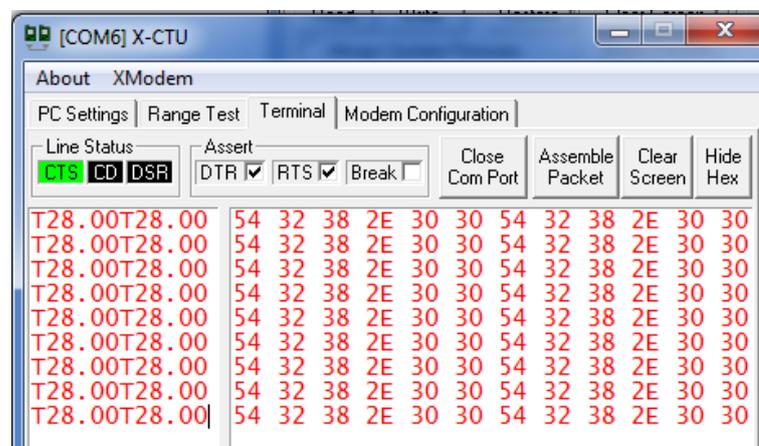


Figura 3. 23: Terminal de X-CTU para visualizar datos recibidos.  
Elaborado por: Autor.

### 3.4. Implementación y configuración de sensores con Arduino y XBee.

A continuación, se describe la configuración de los sensores con las placas embebidas Arduino UNO y su conectividad inalámbrica para enviar datos al servidor VPS previamente descrito. En la figura 3.24 se muestra un diagrama de bloque que describe la propuesta de implementación.

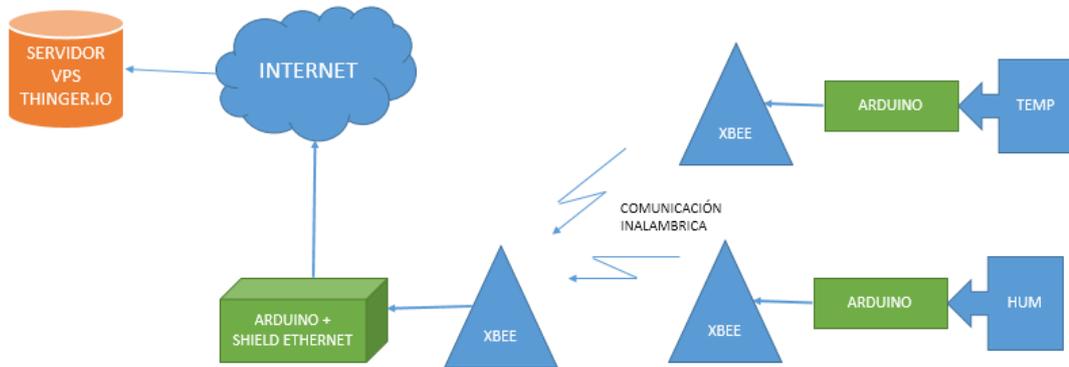


Figura 3. 24: Diagrama de proyecto propuesto  
Elaborado por: Autor.

#### 3.4.1. Configuración de sensores con arduino y módulos XBee.

En este proyecto se utiliza el sensor DHT11 ya descrito previamente, pero para que este funcione es necesario una placa arduino para procesar datos de los resultados obtenidos por este y así poder transmitir vía serial al XBee quien se encarga de realizar la comunicación a otra placa arduino que tenga un Shield ethernet. La figura 3.25 muestra el prototipo utilizando el programa Fritzing.

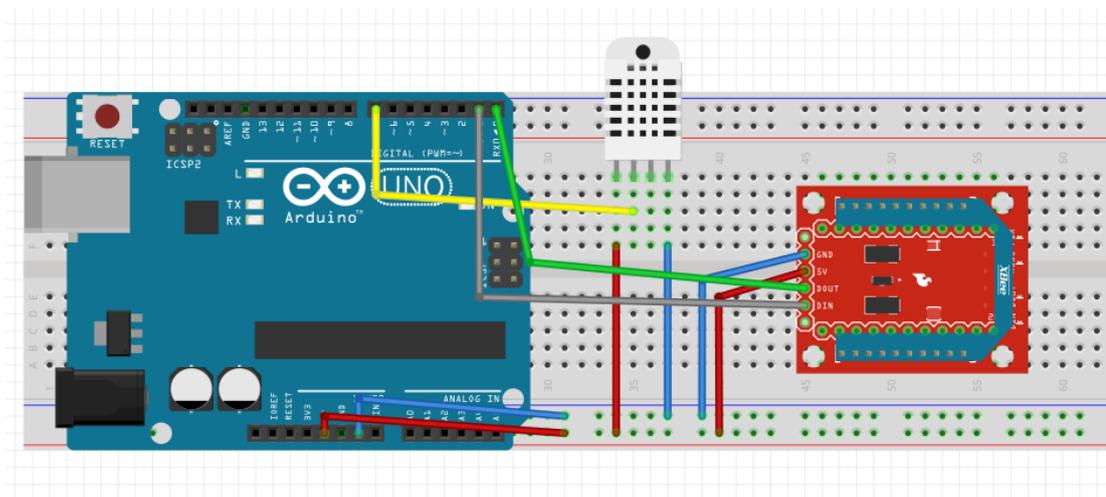


Figura 3. 25: Diagrama de proyecto propuesto  
Elaborado por: Autor.

Con el prototipo ensamblado en un protoboard se procede a cargar la programación realizada en la placa Arduino UNO. A continuación, se describen las partes del código realizado para este caso en particular. Para iniciar el transmisor de los datos del sensor DHT11 es necesario llamar a la librería DHT.h (véase la figura 3.25), que se encarga de realizar la interpretación de las tramas recibidas por el módulo en Arduino UNO, después de ello se define el pin donde se conecta para transferencia de datos.

Se definen valores del sensor cada determinado tiempo, luego lo aloja en la variable anteriormente declarada y se arma la trama con una letra al inicio para interpretarla en el otro punto, con este identificador se puede enviar datos de varios sensores y así mismo interpretarlos en el punto que enviara al servidor los datos. En la figura 3.28 se puede observar dicho proceso.

```
void loop(void)
{
    sensorValue = dht.readTemperature();
    Serial.print("T");
    Serial.print(sensorValue);
    delay(1000);
}
```

Figura 3. 26: Declaración de variable y función setup.  
Elaborado por: Autor.

Continuando con el desarrollo del proyecto, es necesario también configurar el receptor que será la encargada de recibir los datos del sensor inalámbricamente y a su vez enviará a la plataforma thinger.io para ser almacenada, analizada y presentada. La figura 3.29 muestra el esquema prototipo del diagrama que se utilizó utilizando el programa Fritzing.

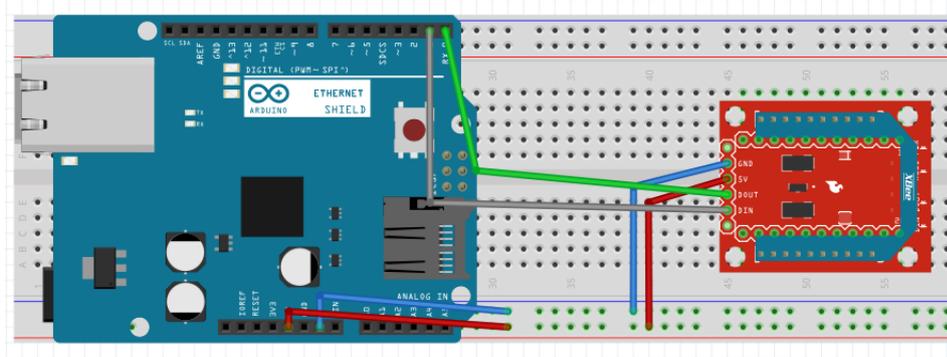


Figura 3. 27: Diagrama utilizado para el transmisor.  
Elaborado por: Autor.

El código de Arduino UNO que se utilizó para recibir los datos desde el XBee se presenta a continuación en las siguientes imágenes detallando por segmentos las funciones de cada parte.

En primer lugar, antes de declarar las librerías y variables a utilizar es necesario definir la dirección ip del servidor VPS (véase figura 3.30), donde está instalado el sistema de gestión thinger, ya que al llamar a la librería de thinger direcciona el llamado del servidor inmediatamente, posteriormente se llama a las librerías SPI.h, Ethernet.h y thingerEthernet.h, siendo las dos primeras necesarias para funcionar correctamente el Shield ethernet que se mostró anteriormente.

```
#define THINGER_SERVER "142.93.115.44"

#include <SPI.h>
#include <Ethernet.h>
#include <ThingierEthernet.h>

#define USERNAME "telcotelemetry"
#define DEVICE_ID "telem001"
#define DEVICE_CREDENTIAL "TtwMYGX2LFm!"

ThingierEthernet thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
int valorTemp;
```

Figura 3. 28: Encabezado del programa receptor y proceso de datos.  
Elaborado por: Autor.

Después se declara la variable entera llamada valorTemp que es la encargada de almacenar el dato recibido por el puerto serie para posteriormente transmitirla al servidor. A continuación, se define la función void setup que simplemente inicializa el puerto serie del microcontrolador y se configura el pin 2 como salida para la Shield ethernet, tal como se ilustra en la figura 3.31.

```
void setup() {
    pinMode(2, OUTPUT);
    Serial.begin(9600);
}
```

Figura 3. 29: Función Setup del programa.  
Elaborado por: Autor.

En la figura 3.32 se describe la función principal del programa diseñado para este propósito, en él se puede apreciar cómo se inicializa declarando un arreglo de una variable tipo char que servirá como almacén de los datos recibidos por el puerto serie, luego se presenta un while para que comience a leer los datos apenas se reciba el identificador T anteriormente definido en el programa del transmisor.

```
void loop() {  
  char n[4]; // Tamaño de 4 asumiendo que máximo solo habrán tres dígitos y valores positivos  
  while (Serial.read() != 'T'); // Buscar el inicio de la trama  
  
  n[Serial.readBytesUntil(' ', n, sizeof(n) - 1)] = 0; // Leo la primera cifra  
  valorTemp = atoi(n); // Conversión de texto a valor  
  
  thing["temperature"] >> [](pson& out){  
    out = valorTemp;  
  };  
  delay(1000);  
  thing.handle();  
}
```

Figura 3. 30: Función Setup del programa.  
Elaborado por: Autor.

Se lee la primera cifra y se hace conversión de texto a un valor entero en la variable global valorTemp, a continuación, la línea thing (“temperatura”) envía el dato con esa etiqueta al servidor para que el sistema lo pueda reconocer y esto es posible ya que al inicio del programa se definió las credenciales que permiten al dispositivo enviar la información por internet. Al final de todo se puede apreciar como ambos comienzan a trabajar en conjunto enviando constantemente la información desde el transmisor al receptor inalámbricamente. La figura 3.33 muestra el transmisor adquiriendo la información del sensor para posteriormente ser enviada al receptor, tal como se muestra gen la figura 3.34.

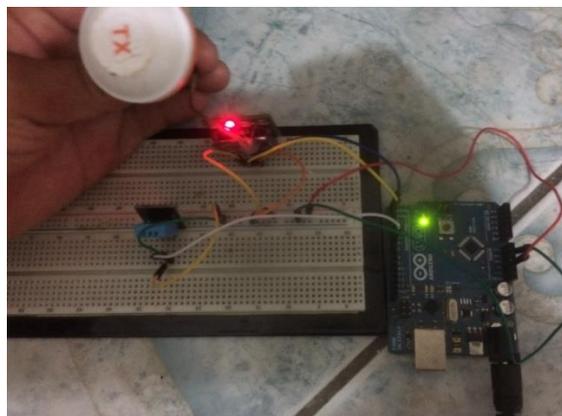


Figura 3. 31: Prototipo del transmisor.  
Elaborado por: Autor.

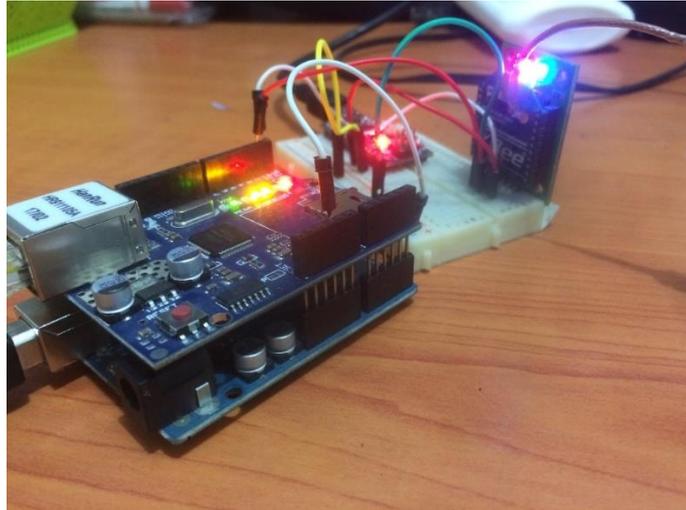


Figura 3. 32: Prototipo del receptor.  
Elaborado por: Autor.

### 3.4.2. Configuración de servidor VPS para la presentación y recepción de datos de los sensores.

En este apartado se presenta la configuración del sistema thinger.io en el servidor privado virtual (VPS) que se adquirió y configuro anteriormente para el proyecto. Para empezar, se debe ingresar con el usuario y contraseña anteriormente creados en la interfaz web, al ingresar se puede apreciar el panel de control de todo el sistema donde se pueden introducir diferentes dispositivos, recursos, dashboard y bases de datos.

Device details

Device Id ⓘ  
TELE001

Device description ⓘ  
microcentral

Device credentials ⓘ  
jHndaMqk@bg\$  
Generate Random Credential

✓ Add Device

Figura 3. 33: Crear dispositivo virtual en thinger.  
Elaborado por: Autor.

La ventaja de crear por cuenta propia el servidor thinger, es que no se tiene límite de dispositivos agregados al contrario que sucede cuando se quiere agregar desde la página oficial del sistema con la cuenta gratuita. Se inicia agregando un device que no es más que un dispositivo que se comunicará con el servidor, en este caso será el Arduino UNO anteriormente descrito.

En la figura 3.35 se aprecia los términos que requieren para realizar el proceso. De la figura 3.35 es necesario proporcionar ciertos detalles como es el identificador del dispositivo, una breve descripción y la credencial o token de este que se puede generar aleatoriamente. Una vez que el dispositivo sea creado correctamente, inmediatamente se puede verificar si este está en línea (ver figura 3.36) recibiendo datos cada determinado tiempo según el código de programación del receptor.

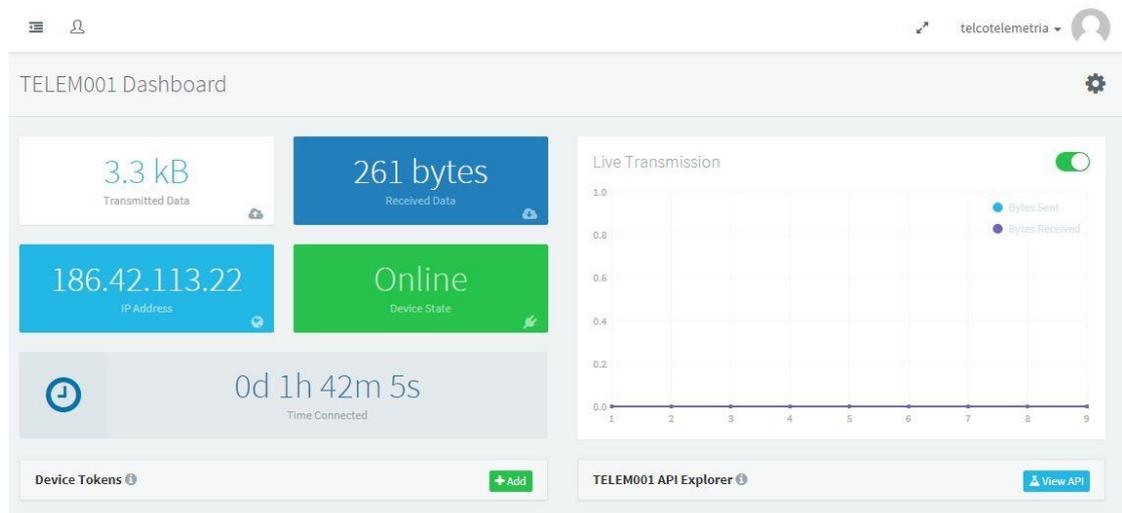


Figura 3. 34: Crear dispositivo virtual en thinger.  
Elaborado por: Autor.

En los datos se puede observar ciertos detalles como por ejemplo si está o no online, la IP desde donde se está conectando al internet, el tiempo sin caídas, y un pequeño cuadro char que visualiza el tamaño de datos que transmite. Al constatar que todo está funcionando correctamente es necesario crear un dashboard en la misma plataforma que permita la presentación en tiempo real de los datos recibidos por los sensores. En la figura 3.37 se presenta dicho proceso, en este se debe proporcionar ciertos datos como son el identificador, nombre y una breve descripción, concluido este paso se

procede a acceder en él para agregar dispositivos y recursos que sean necesarios.

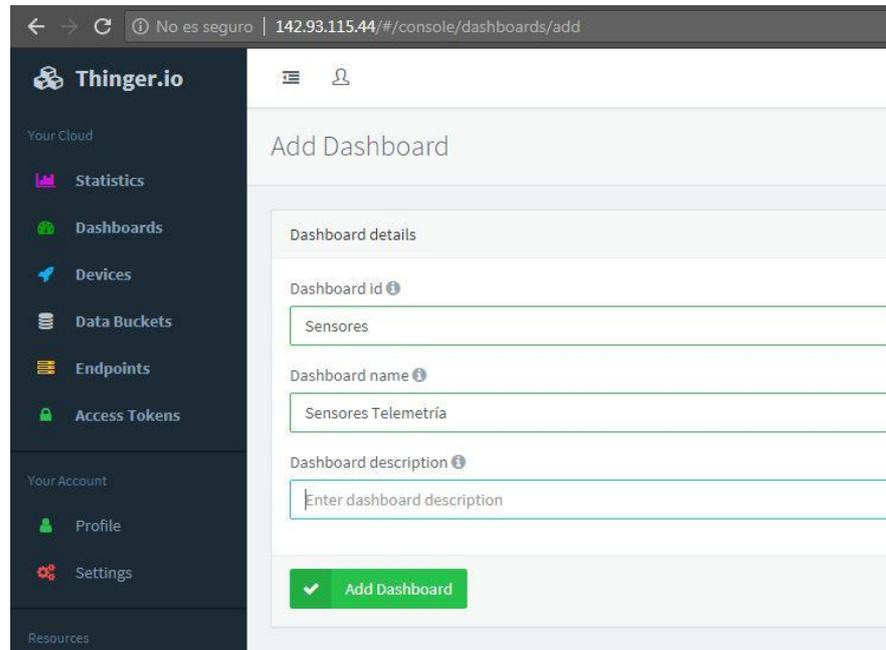


Figura 3. 35: Crear dashboard en thinger.  
Elaborado por: Autor.

En la figura 3.38 se puede apreciar el dashboard terminado, en este caso se presenta el resultado de un solo sensor por limitaciones del hardware adquirido, este presenta la temperatura en tiempo real, pero si se desea agregar otro sensor se deberá anclar a la red de XBee con protocolo IEEE 802.15.4 y proporcionar un identificador a la trama para que el receptor pueda interpretarlos y almacenarlo en otra variable, así mismo enviar resultados del segundo punto.

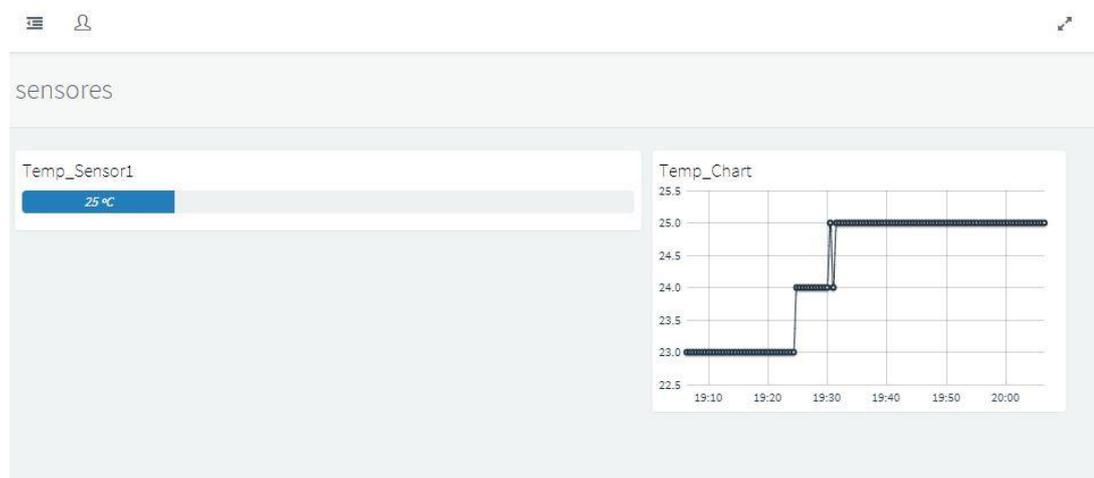


Figura 3. 36: Dashboard en servidor VPS con thinger.  
Elaborado por: Autor.

Estos datos se actualizarán como sea definido al agregar el dispositivo en el dashboard, una de las ventajas de este sistema es que es muy intuitivo y depende mucho de la forma que se aplique puede ser muy eficiente en el campo donde se necesita tomar medidas para diferentes tipos de industrias.

## **Conclusiones.**

- El componente práctico del examen complejo demuestra que es posible realizar una implementación a bajo costo ya que se utilizan elementos accesibles en el mercado nacional.
- Con este sistema implementado de una WSN se demostró la eficiencia de la comunicación de los módulos XBee que se comportaban como nodos sensores en lugares pequeños sin pérdida alguna de datos tanto en la transmisión como recepción.

## **Recomendaciones.**

- Realizar trabajos relacionados al uso de redes de sensores inalámbricos utilizando técnicas de enrutamiento para aplicaciones en el procesamiento de imágenes.
- Realizar comparativas de los módulos de comunicación Xbee y ZigBee en redes inalámbricas de sensores.

## Bibliografía.

- Ahmed, F., Butt, Z., Khan, A., & Mehmood, J. (2016). Wireless Mesh Network IEEE 802.11s. *International Journal of Computer Science and Information Security*, 14(12), 803–809.
- Eiselt, H. A., & Sandblom, C.-L. (2000). Tree Networks. En H. A. Eiselt & C.-L. Sandblom, *Integer Programming and Network Models* (pp. 261–281). Berlin, Heidelberg: Springer Berlin Heidelberg. Recuperado de [http://link.springer.com/10.1007/978-3-662-04197-0\\_12](http://link.springer.com/10.1007/978-3-662-04197-0_12)
- Kbir, M. A., & Bennani, M. (2016). Round time adaptation for image transferring over wireless sensor networks leach protocol. En *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)* (pp. 896–900). Tangier, Morocco: IEEE. <https://doi.org/10.1109/CIST.2016.7805014>
- Maier, M. (2004). Star Networks. En M. Maier, *Metropolitan Area WDM Networks* (pp. 53–76). Boston, MA: Springer US. Recuperado de [http://link.springer.com/10.1007/978-1-4615-0511-2\\_3](http://link.springer.com/10.1007/978-1-4615-0511-2_3)
- Mohapatra, H. (2018). EB-LEACH Algorithm in WSN. <https://doi.org/10.13140/rg.2.2.21496.96009>
- Narmada, A., & Rao, P. S. (2012). Zigbee Based WSN with IP Connectivity. En *2012 Fourth International Conference on Computational Intelligence, Modelling and Simulation* (pp. 178–181). Kuantan, Malaysia: IEEE. <https://doi.org/10.1109/CIMSim.2012.39>
- Raj P. M., K., Mohan, A., & Srinivasa, K. G. (2018). Peer-To-Peer Networks. En K. Raj P.M., A. Mohan, & K. G. Srinivasa, *Practical Social Network Analysis with Python* (pp. 101–108). Cham: Springer International Publishing. Recuperado de [http://link.springer.com/10.1007/978-3-319-96746-2\\_6](http://link.springer.com/10.1007/978-3-319-96746-2_6)

- Shehu, A., Hulaj, A., & Bajrami, X. (2018). An Algorithm for Edge Detection of the Image for Application in WSN. En K. Ntalianis & A. Croitoru (Eds.), *Applied Physics, System Science and Computers* (Vol. 428, pp. 207–213). Cham: Springer International Publishing. Recuperado de [http://link.springer.com/10.1007/978-3-319-53934-8\\_25](http://link.springer.com/10.1007/978-3-319-53934-8_25)
- Sun, D., Lu, J., Wang, W., & Lin, Z. (2010). Design of WSN nodes and network performance analysis in a tea plantation. En *IET International Conference on Wireless Sensor Network 2010 (IET-WSN 2010)* (pp. 144–147). Beijing, China: IET. <https://doi.org/10.1049/cp.2010.1043>



## <DECLARACIÓN Y AUTORIZACIÓN

Yo, **CEVALLOS REINOSO, JORGE ENRIQUE** con C.C: # 092290165-7 autor del Trabajo de Titulación: **IMPLEMENTACIÓN DE UNA RED DE SENSORES INALÁMBRICOS ENLAZADOS CON XBEE UTILIZANDO LA PLATAFORMA THINGER.IO.** previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 17 de Septiembre de 2018

f. \_\_\_\_\_

Nombre: CEVALLOS REINOSO, JORGE ENRIQUE

C.C: 092290165-7

## **REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA**

### **FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN**

<b>TÍTULO Y SUBTÍTULO:</b>	IMPLEMENTACIÓN DE UNA RED DE SENSORES INALÁMBRICOS ENLAZADOS CON XBEE UTILIZANDO LA PLATAFORMA THINGER.IO.		
<b>AUTOR(ES)</b>	CEVALLOS REINOSO, JORGE ENRIQUE		
<b>REVISOR(ES)/TUTOR(ES)</b>	M. Sc. EDWIN F. PALACIOS MELÉNDEZ		
<b>INSTITUCIÓN:</b>	Universidad Católica de Santiago de Guayaquil		
<b>FACULTAD:</b>	Facultad de Educación Técnica para el Desarrollo		
<b>CARRERA:</b>	Ingeniería en Telecomunicaciones		
<b>TÍTULO OBTENIDO:</b>	Ingeniero en Telecomunicaciones		
<b>FECHA DE PUBLICACIÓN:</b>	17 de Septiembre de 2018	<b>No. DE PÁGINAS:</b>	50
<b>ÁREAS TEMÁTICAS:</b>	Análisis de Señales y Sistemas, Procesamiento Digital de Señales e Instrumentación Virtual,		
<b>PALABRAS CLAVES/ KEYWORDS:</b>	Sensores, Redes, Topologías, Xbee, Estándares, WSN.		
<b>RESUMEN/ABSTRACT (150-250 palabras):</b>			
<p>Una red de sensores inalámbricos (Wireless Sensor Network, WSN) es una combinación de hardware y software, en el cual ambos tienen un papel importante que jugar en la red. En el componente práctico del examen complejo se presenta el hardware utilizado, así como el software. Aunque individualmente tienen menos importancia, pero juntos pueden tener un gran impacto en el rendimiento de cualquier red. Se eligió trabajar con un número limitado de nodos. El hardware es la parte crucial para el componente práctico. Se eligieron componentes que sean confiables, funcionales y prácticos, pero más económicos. Después de explorar varias opciones, se escogieron los dispositivos electrónicos XBee y Arduino UNO que resultan una muy buena combinación para el presente trabajo. La tarjeta embebida Arduino UNO ha sido una parte integral del movimiento de código abierto, y XBee ha estado disponible durante algunos años como un componente de frecuencia de radio pequeño pero efectivo.</p>			
<b>ADJUNTO PDF:</b>	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
<b>CONTACTO CON AUTOR/ES:</b>	<b>Teléfono:</b> +593-9-93825498	<b>E-mail:</b> <a href="mailto:jorge-ceba_1992@hotmail.com">jorge-ceba_1992@hotmail.com</a>	
<b>CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE</b>	<b>Nombre:</b> Palacios Meléndez Edwin Fernando		
	<b>Teléfono:</b> +593-9-68366762		
	<b>E-mail:</b> <a href="mailto:edwin.palacios@cu.ucsg.edu.ec">edwin.palacios@cu.ucsg.edu.ec</a>		
<b>SECCIÓN PARA USO DE BIBLIOTECA</b>			
<b>Nº. DE REGISTRO (en base a datos):</b>			
<b>Nº. DE CLASIFICACIÓN:</b>			
<b>DIRECCIÓN URL (tesis en la web):</b>			