



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

TEMA:

**Implementación de una red de sensores inalámbricos mediante sistema
embebido Raspberry Pi**

AUTOR:

Navarrete Armijos, José Ignacio

Trabajo de Titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES

TUTOR:

M. Sc. Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

9 de marzo del 2021



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

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr. **Navarrete Armijos, José Ignacio** como requerimiento para la obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

TUTOR

M. Sc. Palacios Meléndez, Edwin Fernando

DIRECTOR DE CARRERA

M. Sc. Heras Sánchez, Miguel Armando

Guayaquil, a los 9 días del mes de marzo del año 2021



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE TELECOMUNICACIONES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Navarrete Armijos, José Ignacio**

DECLARÓ QUE:

El trabajo de titulación **“Implementación de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi”** 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 9 días del mes de marzo del año 2021

EL AUTOR

NAVARRETE ARMIJOS, JOSÉ IGNACIO



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

AUTORIZACIÓN

Yo, **Navarrete Armijos, José Ignacio**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **“Implementación de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 9 días del mes de marzo del año 2021

EL AUTOR

NAVARRETE ARMIJOS, JOSÉ IGNACIO

REPORTE DE URKUND

URKUND Fernando Palacios Meléndez (edwin_palacios)

Documento	tesis pepe navarrete final 3.0 FINAL.docx (D96938174)
Presentado	2021-03-01 21:27 (-05:00)
Presentado por	fernandopm23@hotmail.com
Recibido	edwin.palacios.ucsg@analysis.orkund.com
Mensaje	Revisión TT José Navarrete Mostrar el mensaje completo 1% de estas 21 páginas, se componen de texto presente en 3 fuentes.

Lista de fuentes

Categoría	Enlace/nombre de archivo
	http://repositorio.ucsg.edu.ec/bitstream/3317/10...
	2020-09-03 Tesis Gonzalo Torres.docx
	TESIS KEVIN CHOEZ 25-08-17.docx
	Roberto_Dender_final.docx
	TT Milton Alfonzo Guzman.docx
	TESIS WILLIAM MERA (1).docx
	https://robotzero.one/hetec-wifi-kit-32/

0 Advertencias. Reiniciar Exportar Compartir

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

mediante sistema embebido Raspberry Pi

AUTOR: Navarrete Armijos, José Ignacio

Trabajo de Titulación previo a la obtención del título de INGENIERO EN TELECOMUNICACIONES

TUTOR: M. Sc. Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

25 de Febrero del 2021

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL

TUTOR



M. Sc. Palacios Meléndez, Edwin Fernando

DEDICATORIA

El trabajo de titulación va dedicado primero a Dios por la oportunidad de realizarme como profesional y sus bendiciones, también a mis padres por el apoyo día a día brindado en todos estos cinco años de carrera.

Gracias a ellos aprendí que con esfuerzo y dedicación en la vida el deseo de superación es posible, estuvieron de inicio a fin apoyándome en cada logro y también en los resbalones que tuve, por eso y muchas cosas este trabajo de titulación va para mis padres y sobre todo a Dios que siempre camino conmigo.

EL AUTOR

NAVARRETE ARMIJOS, JOSÉ IGNACIO

AGRADECIMIENTO

Agradezco a la Universidad Católica Santiago de Guayaquil por haberme abierto las puertas desde el primer día que entre a estudiar en gran distinguida institución, la cual a cargo de sus profesores y demás autoridades me supo guiar como verdadero profesional.

Agradezco a mi tutor M. Sc. Edwin Fernando Palacios Meléndez por haberme apoyado y haber estado conmigo durante estos últimos meses. Y de esta forma haber logrado un gran proyecto de titulación, que cabe mencionar no fue una tarea sencilla, pero que con su tiempo y paciencia conseguimos alcanzar las expectativas antes planteadas.

A mis compañeros que también su apoyo incondicional aportó mucho en mi crecimiento profesional, el trabajo en equipo y las demás actividades realizadas por parte de la universidad, siempre lo voy a llevar dentro de mí.

Finalmente agradecer a Dios por su inalcanzable amor y por haber confiado en mí potencial dado en cada año de mi carrera, también no puede faltar el de mis padres y hermanos que con gran esfuerzo estuvieron prestos para ayudarme y aconsejarme en los que necesitaba durante este tiempo.

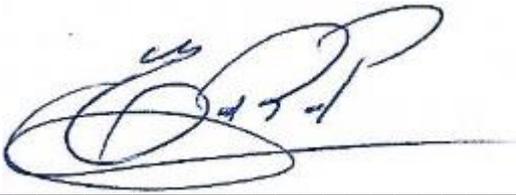
EL AUTOR

NAVARRETE ARMIJOS, JOSÉ IGNACIO



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

TRIBUNAL DE SUSTENTACIÓN

f. 

M. Sc. ROMERO PAZ, MANUEL DE JESUS
DECANO

f. 

M. Sc. HERAS SÁNCHEZ, MIGUEL ARMANDO
DIRECTOR DE CARRERA

f. 

M. Sc. PHILCO ASQUI, LUIS ORLANDO
OPONENTE

Índice General

Índice de Figuras	XII
Índice de Tablas	XIII
Resumen	XIV
Capítulo 1: Descripción General del Trabajo de Titulación.....	2
1.1. Introducción.....	2
1.2. Antecedentes.	3
1.3. Definición del Problema.....	3
1.4. Justificación del Problema.....	4
1.5. Objetivos del Problema de Investigación.....	4
1.5.1. Objetivo General.....	4
1.5.2. Objetivos Específicos.....	5
Capítulo 2: Fundamentación Teórica	6
2.1. Introducción de las comunicaciones inalámbricas,	6
2.2. Tecnologías Inalámbricas.....	6
2.2.1. Redes Inalámbricas de área personal (WPAN).....	7
2.2.2. Bluetooth	7
2.2.3. IRDA.....	8
2.2.4. ZigBee	8
2.2.5. XBEE	9
2.3. Redes inalámbricas de área local (WLAN)	9
2.3.1. IEEE 802.11	10
2.3.2. Capas IEEE 802.11	10
2.3.3. Capa de enlace (MAC)	11
2.3.4. Capa física PHY	11
2.3.5. Estándar IEEE 802.11b	11
2.3.6. Estándar IEEE 802.11 a	12
2.4. Redes inalámbricas de área amplia (WWAN).....	12
2.4.1. Red de telefonía móvil	12
2.4.2. Satélite	12
2.5. Redes de sensores inalámbricos (WSN).....	13
2.5.1. Motes.....	13
2.5.1.1. Microcontrolador	14

2.5.1.2. Memoria Externa	14
2.5.1.3. Batería	14
2.5.2. Nodo Coordinador	14
2.5.3. Nodo Padre	15
2.6. Espectro electromagnético	16
2.6.1. Infrarrojos (IF)	16
2.6.2. Microondas (MW)	16
2.6.3. Radiofrecuencias (RF)	16
2.7. Topologías	16
2.7.1. Arquitectura general de un nodo sensor	17
2.7.2. Aplicaciones de la WSN.....	18
2.7.3. Desafíos y limitaciones de las WSN.....	18
2.7.4. Consumo de energía	19
2.7.5. Distribución de nodos	19
2.8. Domótica	19
2.8.1. Beneficios y Aplicaciones de la domótica	20
2.8.1.1. Confort y ahorro de energía	20
2.8.1.2. Seguridad	21
2.8.2. Red domótica	21
2.9. WirelessHART.....	22
2.9.1. Descripción.....	22
2.9.2. Uso en sistemas de control industrial.....	23
Capítulo 3: Diseño, Implementación y resultados	24
3.1. Aplicación práctica 1: Redes de sensores con #ESP32 #LORA.....	24
3.1.2. Core de Lora en Arduino	24
3.1.3. Esp32.....	25
3.1.4. Esp32 Tarjeta TIGO Lora32- OLED V1	25
3.1.5. Administrador de librerías.....	25
3.1.6. Lectura de sensor STH30.....	26
3.1.7. Esquema de conexión del sensor.....	27
3.1.8. Compilación del sensor Esp32	28
3.2. Aplicación práctica 2: Sensores ESP32 LORA	28
3.2.2. Definición de dispositivo emisor y receptor por códigos	29
3.2.3. Dispositivo en función de emisor	30

3.2.4. Dispositivo en función de receptor.....	30
3.3 Aplicación práctica 3: Sensores LORA, Serial, a RaspberryPi4 y Thingspeak.....	31
3.3.1. Recepción de paquetes.....	31
3.3.2. Configuración multipunto.....	32
3.3.3. Funcionamiento de SD Card y Raspberry Pi	32
3.3.4. MobaXterm Python.....	33
3.3.5. ThingSpeak.....	33
3.3.6. Compilación de códigos por medio de comandos	34
Recomendaciones.....	36
Anexos.	37
Bibliografía.....	46

Índice de Figuras

Capítulo 2

Figura 2. 1: Redes Inalámbricas	6
Figura 2. 2: Clasificación de redes inalámbricas	7
Figura 2. 3: Red dispersa Bluetooth formada de dos picoredes. El maestro de la picored A es un esclavo en la picored B.	8
Figura 2. 4: Comunicación IrDA entre una PDA y una impresora (punto a punto)	8
Figura 2. 5: Diagrama de una red de estructura ZigBee	9
Figura 2. 6: Conexiones mínimas para conectar XBee	9
Figura 2. 7: Esquema de una WLAN en el hogar	10
Figura 2. 8: Capas del IEEE 802.11	11
Figura 2. 9: Estándares IEEE 802.11 en la capa física PHY	11
Figura 2. 10: Redes de telefonía móvil y de satélite	13
Figura 2. 11: Estructura de una red de sensores	13
Figura 2. 12: Mote MicaZ: tamaño (mm): 58x32x7, peso (gramos):18 (Sin baterías).	14
Figura 2. 13: Aporte de energía de batería a los diferentes módulos	14
Figura 2. 14: Componentes del nodo configurado como configurador	15
Figura 2. 15: Diagrama electromagnético	16
Figura 2. 16: Topologías del estándar IEEE 802.15.4	17
Figura 2. 17: Arquitectura de un nodo sensor.	18
Figura 2. 18: Ejemplo de un sistema domótico	20
Figura 2. 19: Pasarela residencial.....	22
Figura 2. 20: Infraestructura con dispositivos HART y WirelessHART	23

Capítulo 3

Figura 3. 1: Sensores Esp32 Lora	24
Figura 3. 2: Settings Network Arduino.....	24
Figura 3. 3: Instalador Esp32.....	25
Figura 3. 4: Tarjeta actual	25
Figura 3. 5: Librerías (SPI-I2C).....	25

Índice de Tablas

Capítulo 2

Tabla 3. 1: Esquema de conexión del sensor.....	27
---	----

Resumen

El proyecto de titulación básicamente se trata de una implementación de sensores inalámbricos mediante un sistema embebido Raspberry Pi que es una serie de ordenadores de placa reducida para proyectos de informática y creaciones digitales, y pues que se el trabajo conjunto con cuatro nodos. El primer nodo tiene conectado el sensor de temperatura y humedad con conexión I2C que utiliza dos pines y conecta múltiples sensores y actuadores. Con el módulo Esp32 Lora y por medio de I2C realiza la lectura de temperatura y humedad enviándolo por radiofrecuencia a nuestro nodo posterior que es el 2. El nodo dos recibe los datos de nuestro primer nodo y además lee los resultados de temperatura y humedad de su propio sensor, y esas ds informaciones van hacia el nodo tres. El nodo 3 presenta humedad y temperatura de los nodos anteriores, y claro también presento los mismos datos de su sensor. El nodo tres envía los datos anteriores al nodo principal que es el coordinador y este envía por puerto serial a la Raspberry Pi la información de los nodos. La Raspberry Pi tiene un código en Python que se ejecuta y por medio del puerto serial mencionado envía esos datos a una aplicación en la nube llamada Ubidots. La Raspberry Pi tiene instalado el sistema operativo raspbian que está en una microSD. Para concluir para acceder a la Raspberry Pi lo hacemos por MobaXterm y por SSH inalámbrica con una red local conectada a internet, y transmitir a Ubidots.

Palabras claves: NODOS, SENSORES, REDES, PROTOCOLOS, RADIOFRECUENCIA, SISTEMA.

Capítulo 1: Descripción General del Trabajo de Titulación

1.1. Introducción.

La red de sensores está compuesta por un grupo de sensores con funciones sensibles y funciones inalámbricas, que pueden formar una red autoorganizada sin una infraestructura física especial o una gestión centralizada. Estas redes de sensores inalámbricos se denominan Redes de sensores y actuadores (Redes de sensores y actuadores inalámbricos, WSAN) con siglas en inglés (Redes de sensores inalámbricos, WSN). Estos sensores autónomos se utilizan para monitorear varios tipos de condiciones físicas o ambientales. Como temperatura, sonido, presión, etc. Las redes más modernas son bidireccionales, y dan un control en la actividad del sensor.

Actuaciones recientes en tecnología de sistemas microelectromecánicos, comunicaciones inalámbricas y electrónica digital han permitido el desarrollo de dispositivos sensores inalámbricos multifuncionales, de bajo costo y bajo consumo, de tamaño pequeño y capaces de comunicarse en distancias cortas. Estos dispositivos, que consisten en componentes de detección, procesamiento de datos y comunicación, aprovechan la idea de redes de sensores basadas en el esfuerzo colaborativo de una gran cantidad de nodos. Las redes de sensores inalámbricas se pueden definir como una colección de dichos nodos de sensores.

Para hablar de sensores inalámbricos móviles, en inglés conocidos (*Mobile Wireless Sensor Network, MWSN*) se dice que son conocidos como sensores de red sin cables de interconexión (WSN) en donde los nodos de estos sensores son móviles. MWSN es más pequeño y más versátil que las redes de sensores estáticos porque se pueden implementar en cualquier escenario y se enfrentan a rápidos cambios tecnológicos. Caracterizadas por su facilidad de despliegue y por ser autoconfigurables, y que con el tiempo ha convertido el transmisor y el receptor también proporcionan servicios de enrutamiento entre nodos y registran datos relacionados con los sensores locales de cada nodo.

1.2. Antecedentes.

La influencia es cada vez mayor en sensores electrónicos ya que su industria está ocasionada la necesidad de proponer métodos que ayuden a reducir costos de cableado como de canalización, por lo tanto, las soluciones inalámbricas suelen ser muy provechosas, sin embargo, no hay un método fijo que permita su implementación independiente de la tecnología que se use. Por ello, hay metodologías indicadas para el diseño e implementación de una red inalámbrica de sensores (*Metodología para el diseño de una red de sensores inalámbricos por Pérez, Juan Urdaneta, Elizabeth Custodio, Ángel*).

Las redes de sensores inalámbricos han alcanzado el nivel de madurez con un nivel aceptable de rendimiento para entornos del mundo real. Sin embargo, existen varios desafíos sin resolver, como protocolos, datos en tiempo real, administración de energía, abstracciones de programación, seguridad y privacidad. Por lo general, las redes de sensores se forman con una gran cantidad de nodos de sensores.

1.3. Definición del Problema.

El propósito del proyecto es realizar un estudio profundo de la red de sensores inalámbricos (WSN) la que permita su aplicación en cualquiera área de la ingeniería y agroindustria, la domótica implica el desarrollo de protocolos para manejar y organizar la comunicación de diferentes elementos de la red. Actualmente, la investigación se ha completado de toda la WSN, especialmente de términos en consumo de energía, debido a que estos sistemas son enfocados para sitios con difícil acceso y no utiliza una fuente de energía fija (Khan, Khan y Ali, 2016).

Otros estudios se han basado a la capacidad del sistema de auto-configurable dejando de lado una infraestructura fija y dándole movilidad a sus elementos (Loo, Mauri, & Ortiz, 2012). Los estudios han dado resultados en Con el desarrollo de protocolos más eficientes, de esta manera, hemos encontrado algunos protocolos comerciales para redes de sensores inalámbricos, como ZigBee y 6LoWPAN.

El costo de implementación de varios de sus protocolos y sistemas existentes es muy alto, ya que para mejorar la calidad, utilizan los siguientes componentes que para ciertas aplicaciones no son de extrema necesidad, por ejemplo, la implementación del Sistema de Posicionamiento Global (GPS) se utiliza para geolocalizar la posición de los nodos sensores en el sistema (Guo, Low y Nguyen, 2011). Con todo, en este trabajo, presentamos el desarrollo de un sistema de intercambio básico. La misma particularidad que los ya dados en la actualidad, pero a menor coste, esto ha llevado a la necesidad de realizar mayores esfuerzos para adaptar los protocolos de red modificados desarrollados para adaptarse a estos nuevos elementos.

1.4. Justificación del Problema.

Los avances en tecnología y publicidad en diferentes campos de las redes de sensores inalámbricos hacen que esta investigación sea imprescindible sobre estos sistemas que tienen enormes ventajas y han sido probados (Polo, Hornero, Duijneveld, García y Casas, 2015). La ventaja de esto es que podemos monitorear mejor las variables que intervienen en el proceso, acceder y controlar los datos de forma remota y automatizar la toma de decisiones. Debido a grandes costos en sus dispositivos y protocolo los propietarios han privatizado a muchos sectores y a sus clientes con el fin de que se puedan beneficiar con estos avances tecnológicos.

Por otro lado, la investigación y desarrollo de WSN tiene un gran valor académico porque no es una materia universal en el país, ya que ayuda a conocer la estructura interna de estas, componentes y protocolos necesarios para que estos se pongan en trabajo. Como también la utilización de metodologías nuevas y sugerencias de acuerdo a demandas de cada situación planteada.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Implementar una red de sensores inalámbricos mediante sistema embebido Raspberry Pi utilizando software libre Python y Arduino.

1.5.2. Objetivos Específicos.

- Describir los fundamentos teóricos de los sistemas de comunicaciones inalámbricas y de sensores inalámbricos.
- Elaborar los códigos de programación en Python y Arduino para las aplicaciones experimentales de laboratorio de redes de sensores inalámbricos.
- Implementar prácticas para el laboratorio de redes de sensores inalámbricos para la malla rediseñada de Telecomunicaciones.

Capítulo 2: Fundamentación Teórica

2.1. Introducción de las comunicaciones inalámbricas,

La comunicación inalámbrica, o también conocido como sin cables es la comunicación (emisor/receptor) que no se encuentra unida por el medio de propagación físico, más bien se utiliza modulación de ondas electromagnéticas por medio del espacio. De esta forma, los dispositivos físicos solo están presentes en los emisores y receptores de la señal, y en lo que encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.

Los dispositivos que utilizan redes inalámbricas incluyen computadoras portátiles, computadoras de escritorio, notebooks, asistentes digitales personales (PDA), teléfonos móviles, tabletas y dispositivos de localización. El modo de trabajo de una red inalámbrica es básicamente el mismo que el de una red cableada, pero la red inalámbrica tiene la capacidad de hacer que las señales de información sean más adecuadas para la transmisión por aire.



Figura 2. 1: Redes Inalámbricas
Fuente: (Bhavsar & Sharma, 2014)

2.2. Tecnologías Inalámbricas

Podemos clasificarlas en cuatro grupos específicos, según el campo de aplicación y el alcance de la red de área personal inalámbrica (WPAN), las redes inalámbricas de área local (*Wireless Local Area Networks, WLAN*), también red de área metropolitana inalámbrica (WMAN) y red de área amplia inalámbrica (WWAN). En la red inalámbrica tiene dos partes: corta distancia y larga distancia. La distancia corta es una red confinada a un área limitada, adecuada para redes de área local (LAN), ya sean edificios de empresas, campus de escuelas y universidades, fábricas o casas con frecuencia de 2,4

GHz y 5 GHz, de largo alcance si nos referimos a redes que abarquen grandes áreas, como un área metropolitana (WMAN), un estado, un país, que su objetivo es una buena cobertura inalámbrica global.

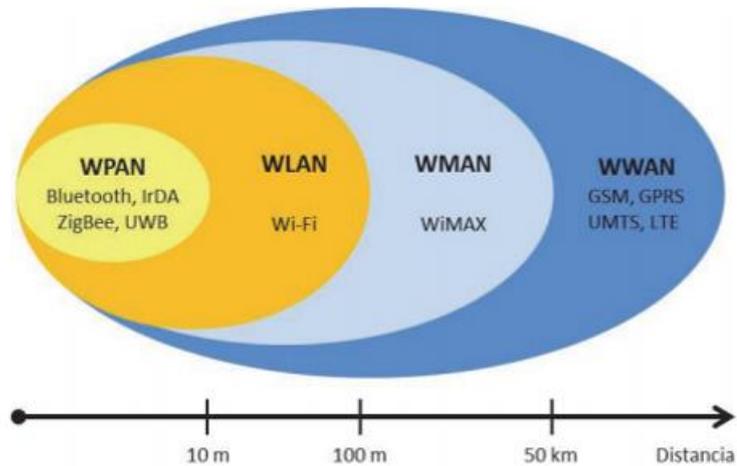


Figura 2. 2: Clasificación de redes inalámbricas
Fuente: (Salazar, 2019)

2.2.1. Redes Inalámbricas de área personal (WPAN)

A diferencia de otras redes inalámbricas una conexión que se hace por medio de una WPAN hace que generalmente se tenga poca o muchas veces ninguna infraestructura o conectividad directa fuera del enlace establecido. Este tipo de redes se caracterizan por su bajo consumo energético y baja velocidad de transmisión. Están basados en tecnologías como Bluetooth, IrDA, ZigBee o UWB.

2.2.2. Bluetooth

Es parte del estándar IEEE 802.15.1. Inicialmente, Bluetooth fue creado para lograr comunicación omnidireccional, bajo consumo de energía, corta distancia y dispositivos económicos, reemplazó el uso de cables y conectó varios dispositivos en conexiones temporales de radio. La red Bluetooth denominada piconet (piconet) consta de 8 dispositivos activos relacionados con maestro-esclavo. El primer dispositivo con función Bluetooth es el dispositivo maestro y los otros dispositivos son dispositivos esclavos. Tienen la comunicación directa con el maestro.

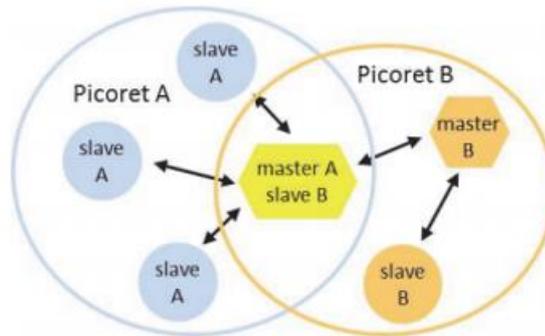


Figura 2. 3: La red descentralizada de Bluetooth consta de dos redes pico. El anfitrión de Piconet A es el esclavo de Piconet B.

Fuente: (Salazar, 2019)

2.2.3. IRDA

Son la unión de datos por infrarrojos, que especifica un conjunto completo de estándares para tener comunicación por medio de infrarrojos. IrDA son las normas que se utilizan para mandar una conectividad inalámbrica a los dispositivos que generalmente son utilizados para cables de conectividad.



Figura 2. 4: Comunicación IrDA entre PDA e impresora (punto a punto)

Fuente: (Salazar, 2019)

2.2.4. ZigBee

Tomado en cuenta del El estándar IEEE 802.15.4, creado como un estándar global abierto, puede satisfacer las necesidades de las redes de dispositivos inalámbricos para una fácil aplicación, alta confiabilidad, bajo costo, bajo consumo y baja velocidad de transmisión de datos. Es apto para bandas de frecuencia sin licencia de 2,4 GHz, 900 MHz y 868 MHz, y su velocidad máxima de transmisión es de 250 Kbps, fundamental para el uso de sensores y automatización en redes inalámbricas. La red ZigBee participa en diferentes tipos de dispositivos: dispositivos de función completa (FFD) y dispositivos de función reducida (RFD).

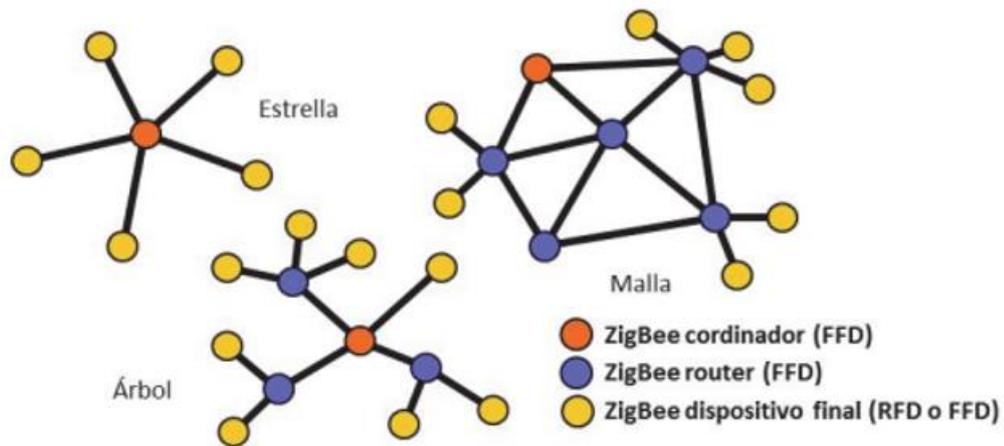


Figura 2. 5: Diagrama de una red de estructura ZigBee
 Fuente: (Crespo, 2016)

2.2.5. XBEE

Creados por la empresa Digi, proporcionan conectividad inalámbrica con un dispositivo de la misma serie por un precio barato respecto a otras empresas fabricantes. Ofrecen 2 series, la Serie 1 tiene conjuntos de chips Freescale para redes punto a punto o multipunto, y la Serie 2 tiene conjuntos de chips Ember para aplicaciones que requieren repetidores o redes de malla.

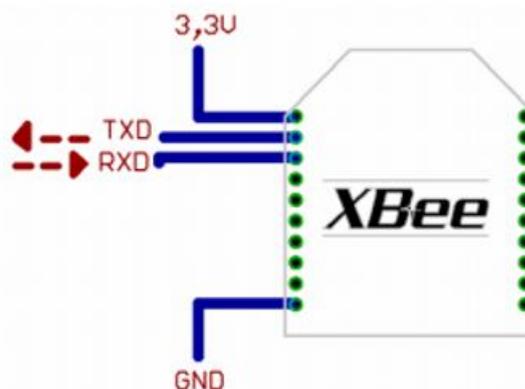


Figura 2. 6: Conexiones mínimas para conectar XBee
 Fuente: (MACSANGAR90, 2014)

2.3. Redes inalámbricas de área local (WLAN)

Diseñadas para proporcionar acceso inalámbrico para lugares que proporcionan alcance universal de hasta 100 metros, se puede utilizar en el hogar, la escuela, la sala de ordenadores y el entorno de oficina. Ofrecen al usuario la capacidad de moverse alrededor de un área que está cubierto localmente y vinculado a la red estándar 802.11 IEEE y se vende bajo la

marca Wi-Fi. Utilizan una misma banda de frecuencia 2,4 GHz que usa su estándar. Cuenta con una velocidad de transferencia para el enlace inalámbrico IEEE 802.11g con un 54 Mbps.



Figura 2. 7: Esquema de una WLAN en el hogar
Fuente: (Salazar, 2019)

2.3.1. IEEE 802.11

Se dice que este sistema de comunicaciones tiene la finalidad de transmitir información entre dos puntos divididos por una distancia, y unidos físicamente por una estructura física ya sea (natural o artificial) dispuesto a ser usada para ello. Por lo tanto, se define esta transmisión como un proceso donde se puede enviar o transportar información de un punto como fuente hasta el punto de destino por medio de un canal o un puente para realizar la transmisión.

2.3.2. Capas IEEE 802.11

Cubre la capa física o de enlace MAC, esta capa física tiene tres capas físicas diferentes: espectro ensanchado por secuencia directa (DSSS), infrarrojos y el espectro ensanchado por salto en frecuencia (FHSS). La mencionada capa de enlace o MAC es habitual entre las tres capas físicas, dando una interfaz única en los protocolos de capas superiores. MAC tiene funciones como por ejemplo la de Fragmentación, Retransmisión y Aceptación de paquetes.

Data Link Layer	802.2		
	802.11 MAC		
PHY Layer	FH	DS	IR

Figura 2. 8: Capas del IEEE 802.11
Fuente: (Crespo, 2016)

2.3.3. Capa de enlace (MAC)

Teniendo un protocolo de acceso al medio en redes inalámbricas es más difícil que realizarlo en redes cableadas, hay que tener en cuenta las dos topologías, la de red inalámbrica ya sea en modo independiente o por infraestructura). También hay que saber que se pueden tener factores que pueden causar problemas y son:

- Interferencias
- Variaciones en potencia de la señal.
- Conexiones y desconexiones en la red
- Roaming o itinerancia, en nodos móviles

2.3.4. Capa física PHY

Este estándar IEEE 802.11 proporciona una serie de esquemas para la codificación y transmisión de las comunicaciones inalámbricas, los esquemas dados más conocidos son el Espectro Ensanchado por Salto de Frecuencia.

	802.2 Logical Link Control (LLC)					
MAC	CSMA/CA					
PHY	802.11 2.4 GHz FHSS	802.11b 2.4 GHz DSSS	802.11a 5 GHz OFDM	802.11g 2.4 GHz OFDM	802.11n 2.4/5 GHz OFDM	802.11ac 5 GHz OFDM

Figura 2. 9: Estándares IEEE 802.11 en la capa física PHY
Fuente: (Salazar, 2019)

2.3.5. Estándar IEEE 802.11b

Este es el estándar en la capa física y admite velocidades de transmisión muy altas. El estándar permite dos velocidades adicionales en la banda de 2,4 GHz, 5,5 Mbps y 11 Mbps.

2.3.6. Estándar IEEE 802.11 a

Trabaja a velocidades de máximo 54 Mbps y tiene también una banda de frecuencia de 5 GHz. Reemplazando a DSSS, este estándar usa OFDM, el cual hace Los datos se transmiten en paralelo por subportadoras, lo que proporciona una mejor capacidad anti interferente y una mayor velocidad de transmisión.

2.4. Redes inalámbricas de área amplia (WWAN)

Se pueden ampliar a más de 50 kilómetros, se suelen utilizar estas frecuencias autorizadas. Estos tipos de redes se mantienen en grandes áreas (como ciudades o países) a través de múltiples sistemas o ubicaciones de satélites que posean antena, siempre con la atención de un proveedor de servicios de Internet.

2.4.1. Red de telefonía móvil

Su cobertura se da en la celda. Un transmisor celular o una estación base en un área pequeña está diseñado para brindar servicio a una sola celda. Estos dispositivos móviles están vinculados a la estación base y otros dispositivos móviles están vinculados a la central telefónica conmutada que conecta el teléfono móvil a la red telefónica con cable.

1G, era analógica dada únicamente para llamadas de voz, 2G es basada en tecnología digital y cuenta con infraestructura de red (GSM), la cual permite mensajes de texto, la velocidad de datos es de hasta 64 Kbps. En la era 3G creada en 2000, la velocidad de datos era de hasta 2 Mbps, la cuarta generación, 4G ofrece velocidades de hasta 1 Gbps. Mientras que la nueva generación 5G se espera para el año 2020.

2.4.2. Satélite

Debido a su gran altura, la transmisión por satélite cubre una gran área sobre la tierra. Esto es muy útil para usuarios en áreas remotas o islas sin cables submarinos para cuando hay estos acontecimientos necesitan teléfonos satelitales. Cada satélite está compuesto con una variedad de transpondedores que constan de transceptores y antenas. La señal de

entrada se amplifica y luego se retransmite por medio de una frecuencia diferente.



Figura 2. 10: Redes de telefonía móvil y de satélite
Fuente: (Salazar, 2019)

2.5. Redes de sensores inalámbricos (WSN)

Constituidas por sensores autónomos que se distribuyen espacialmente con la finalidad de ser capaces de tener comunicación con poco consumo de energía y entregando sus datos. Estos se componen por cientos o incluso miles de estos sensores electrónicos que trabajan con baterías, que tiene el nombre de nodos sensores conocidos como “motes ” distribuidos por un lugar de interés particular.

Cada nodo sensor es ubicado por la red inalámbrica con la capacidad de recoger datos, y también para enrutamiento al nodo recolector (sink node) a través de una arquitectura ad hoc de múltiples saltos. El nodo recolector se comunica con el nodo administrador que es el gestor de tareas vía Internet, vía satélite, y de manera directa.

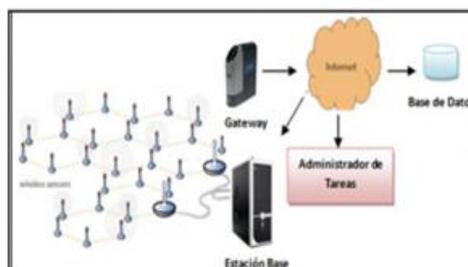


Figura 2. 11: Estructura de una red de sensores
Fuente: (Perez, Urdaneta, & Custodio, 2014)

2.5.1. Motes

Son dispositivos muy pequeños que se conectan inalámbricamente uno al otro, pasando información de tipo de temperaturas, humedades y

almacenajes de sus baterías. Tiene como función registrar datos del medio de donde se lo utiliza.



Figura 2. 12: Mote MicaZ: tamaño (mm): 58x32x7, peso (gramos):18 (Sin baterías).
Fuente: (Perez, Urdaneta, & Custodio, 2014)

2.5.1.1. Microcontrolador

Es un circuito integrado el cual su componente principal es una aplicación embebida. Es una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida.

2.5.1.2. Memoria Externa

Almacena datos del nodo y la memoria necesaria es dependiendo de cada aplicación que se use.

2.5.1.3. Batería

Aporta la energía al nodo para su funcionamiento así sea prendido o apagado.

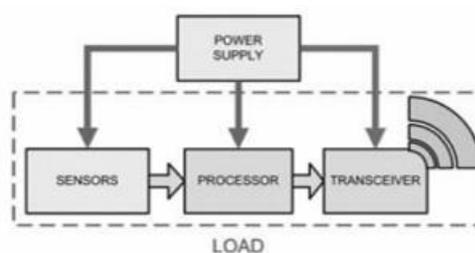


Figura 2. 13: Aporte de energía de batería a los diferentes módulos
Fuente: (Crespo, 2016)

2.5.2. Nodo Coordinador

Controla datos recibidos en forma inalámbrica de otros nodos que conforman la red a través de un módulo XBEE S2C, se dice que contiene un sensor FC-28 y un reloj externo RTC DS3231 configurado para realizar mediciones cada hora durante 15 días. Tiene como función también la

comunicación hacia la base de datos MySQL la que almacena valores necesarios en sensores que se usen en nuestra red. Hay que mencionar que un nodo ESP8266 ya viene incorporado un módulo Wifi que se enlaza a nuestra red LAN y después a internet con el propósito de mostrar datos en tiempo real por medio del interfaz dado por una aplicación móvil.



(a) Vista interior: Componentes



(b) Vista Lateral: Protector de plástico

Figura 2. 14: Componentes del nodo configurado como configurador
Fuente: (Crespo, 2016)

2.5.3. Nodo Padre

Es una subred del sistema, administrada por un número de sensores/actuadores inteligentes que se comunican y comparten los recursos del sistema con otros nodos principales. El nodo padre tiene las siguientes funciones:

- Proporcione direcciones para nuevos nodos secundarios en la red.
- Descubra las rutas disponibles a nodos secundarios específicos
- Escriba el actuador para que esté disponible en un nodo secundario determinado.
- Recolecta datos para hacer una autoevaluación en la subrd NP-NH.
- Comparta datos en los nodos secundarios con otros nodos principales.

2.6. Espectro electromagnético

El espectro electromagnético tiene un rango en sus frecuencias para las ondas electromagnéticas las que se pueden propagar a través de un espacio libre, y que se ordenan por su longitud de onda y su frecuencia antes mencionada.

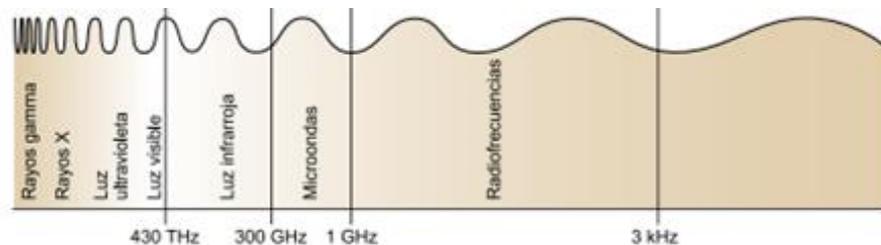


Figura 2. 15: Diagrama electromagnético
Fuente: (MACSANGAR90, 2014)

2.6.1. Infrarrojos (IF)

Utilizan en comunicaciones punto a punto de corto alcance, direccionables para que no puedan atravesar obstáculos. Utilizados generalmente para el mando de distancia de tv y hasta hace tiempo era un sistema de comunicación utilizado para conectar dispositivos situados el uno al lado del otro.

2.6.2. Microondas (MW)

Rango de frecuencias acomodado para transmisiones en largo recorrido como las comunicaciones por satélite, comunicaciones terrestres punto a punto que es alternativa al cable coaxial o la fibra óptica. Recaltar también que la mayoría de las tecnologías inalámbricas habituales que existen actualmente como UMTS, Bluetooth o WLAN).

2.6.3. Radiofrecuencias (RF)

Las transmisiones de radio (FM, AM) y televisión digital terrestre (TDT). Estas radiofrecuencias son omnidireccionales y pueden atravesar obstáculos sin ningún alguno.

2.7. Topologías

Este estándar IEEE 802.15.4 que es el de más uso en redes de sensores inalámbricos, aquí podremos ver todos los beneficios y lo que este estándar

soporta. Permite tener topologías en estrella y peer-to-peer (Yang, 2014). Estrella tenemos la posibilidad de formar redes estrella y árbol, mientras que topología peer-to-peer son más para desarrollar redes de árbol y malla clúster.

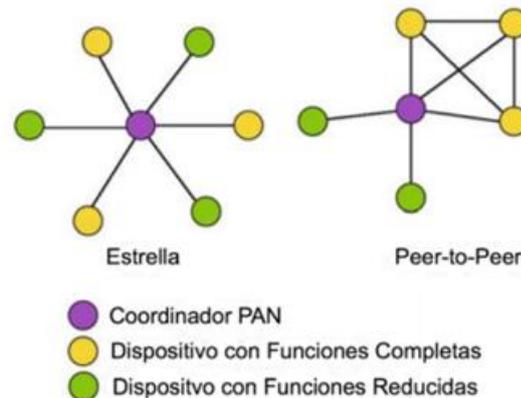


Figura 2. 16: Topologías del estándar IEEE 802.15.4
Fuente: (Salazar, 2019)

2.7.1. Arquitectura general de un nodo sensor

Por el gran número de nodos sensores que se ponen en marcha en una red de sensores inalámbricos, pues estos suelen ser pequeños, baratos, eficaces en relación con energía con la capacidad de acumulación, informática y comunicación. Por eso que los nodos sensores no usan baterías de larga duración y de mucha capacidad en la alimentación principal que es la fuente de energía debido a su límite de tamaño. Debido a que el modelo de este nodo sensor sus características son reducidas por capacidades necesarias en informática y comunicación. Estos nodos sensores poseen cuatro subsistemas principales:

- Un subsistema informático compuesto por un microcontrolador y una memoria que se utiliza para almacenar y procesar los datos recopilados por el subsistema que los ubica.
- El subsistema de localización se comprende de uno o más sensores que monitorean el medio físico.
- El subsistema de informática que tiene su sistema de radio con corto alcance para comunicar sin cables.
- El subsistema que provee energía, normalmente necesario en el uso de baterías, este alimenta a todo el nodo sensor. Tiene un generador

de energía que se lo incluye al sistema, empleando tecnologías de recopilación energética.

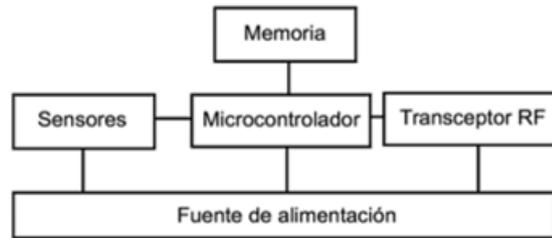


Figura 2. 17: Arquitectura de un nodo sensor.
Fuente: (Quiñones, González, Torres , & Jumbo , 2017)

2.7.2. Aplicaciones de la WSN

Estas redes de sensores son utilizadas para aplicaciones de alto nivel, y con sistemas que detectan amenazas y radiaciones nucleares, aplicaciones biomédicas, detección de hábitats y seguimiento de terremotos. En las aplicaciones que existen y que tienen potencial hay muchos tipos de redes de sensores, que incluyen detección militar, seguridad física, control del tráfico aéreo, vigilancia del tráfico, video vigilancia, automatización industrial y de fabricación, control de procesos, gestión de inventario, robots distribuidos, detección meteorológica, monitoreo ambiental, fronteras nacionales y construcción. construcción y vigilancia, mostradas a continuación:

- Monitoreo de fuerzas adversas
- Monitoreo con fuerzas aliadas
- Microclimas
- Detección de inundaciones
- Agricultura
- Control y monitoreo de datos fisiológicos
- Automatización en el hogar
- Detección de vehículos

2.7.3. Desafíos y limitaciones de las WSN

Es importante saber que las redes de sensores inalámbricos contienen cierta similitud con sistemas de distribución, sujetas a desafíos y limitaciones especiales. Las restricciones que se tiene afectan todo diseño de esta red de

sensores, esto hace que se usen protocolos y algoritmos que reducen en sus contrapartes a los sistemas distribuidos.

2.7.4. Consumo de energía

Los nodos sensores que contienen WSN suelen tener fuentes de alimentación limitadas. Suelen funcionar con pilas, sustitutas o recargables al momento de que se acabe la batería. La batería se recarga y no afecta tanto a la estrategia en este consumo de energía. Gracias a la limitación, los protocolos WSN tratan del ahorro de energía (Fischione, 2014).

2.7.5. Distribución de nodos

La posición de los nodos no es necesario ser predeterminada. Esto hace que la distribución nodal sea aleatoria en lugares no accesibles en operaciones de socorro. Se puede decir que este repartimiento aleatorio requiere el proceso de desarrollo en protocolos de auto-organización para el grupo de protocolos para comunicación.

Es importante que estos nodos tengan la capacidad de configurar, operar y colaborar con otros nodos de esta forma para adaptarse a errores y cambios en el entorno. La mayoría de las WSN funcionan sin supervisión, por lo que puede personalizarlas, mantenerlas y repararlas personalmente. En redes de sensores con energía limitada, estas características deben tener un diseño e implementación de modo que no ocurran en gastos excesivos de la energía.

2.8. Domótica

La domótica se refiere al control y automatización de electrodomésticos y equipos eléctricos, ya sean bombillas, indicadores de temperatura o cerraduras de puertas (Andersson et al., 2015). El Internet de las cosas (IoT- Internet of Things) está vinculado a la automatización del hogar. Los dispositivos de Internet de las cosas, como los termómetros inteligentes, se pueden controlar mediante teléfonos inteligentes, por lo que con la ayuda de Internet, se pueden realizar capacidades globales.

La automatización del hogar con la ayuda de IoT se puede optimizar en términos de confort y consumo de energía, porque puede presentar automáticamente el modo de ahorro de energía del dispositivo al usuario en tiempo real bajo ciertas circunstancias, logrando así un modo de ahorro de energía.

La configuración de los sistemas domóticos usualmente se maneja por un portal con accesibilidad de un computador de escritorio y dispositivos móviles. Aquí les mostramos un sistema domótico con una variedad de equipos eléctricos inteligentes.

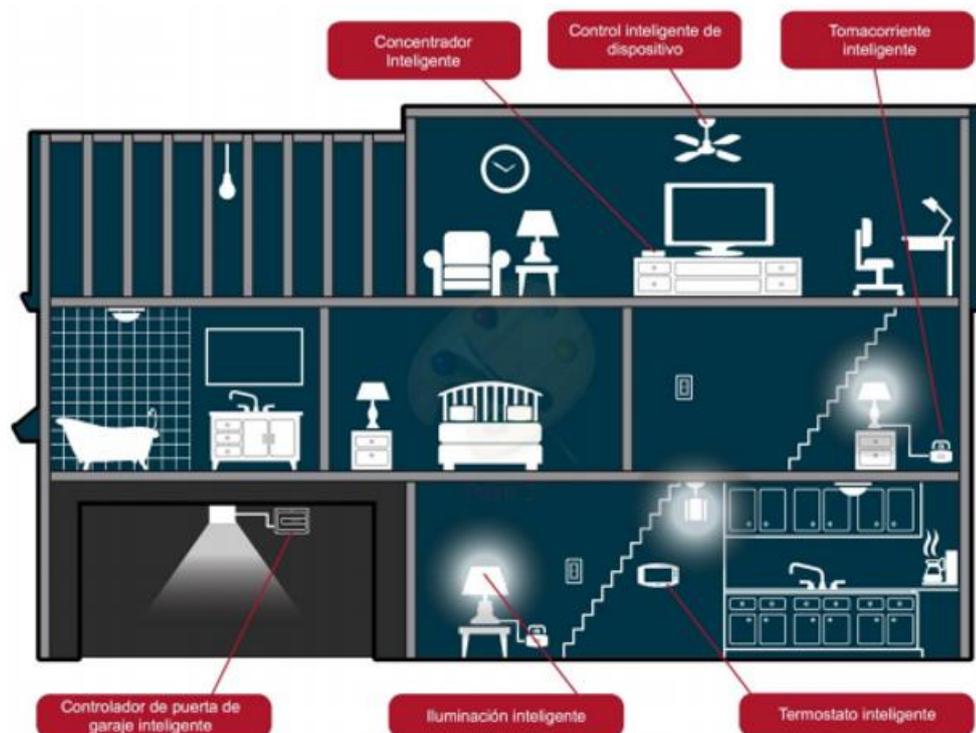


Figura 2. 18: Ejemplo de un sistema domótico

Fuente: (Salazar, 2019)

2.8.1. Beneficios y Aplicaciones de la domótica

En cuanto a las funciones básicas de los sistemas domóticos, incluimos algunas de las aplicaciones domóticas más habituales (Miller, 2013):

2.8.1.1. Confort y ahorro de energía

- Iluminación exterior e interior
 - Las luces se encienden y apagan automáticamente para simular que alguien en la casa protege contra el robo.

- Sincronización de luces con sensoriales para encendido automático cuando alguien desee entrar a un cuarto.
- Calefacción con aire acondicionado
 - En climas cálidos, encienda el aire acondicionado desde fuera de la casa.
 - Programe la temperatura de la casa para ajustar automáticamente la temperatura de acuerdo con las personas en la casa para lograr el mejor consumo de energía.

2.8.1.2. Seguridad

- Sistemas de seguridad
 - Telnet en el sistema de seguridad para habilitarlo o deshabilitarlo.
 - Enviar una alerta por correo electrónico o mensaje, si no, ningún sensor detecta movimiento al lado de la caja fuerte, armario de armas, etc.
 - Control de cámaras de seguridad.
- Candados
 - Si está más allá de la distancia predefinida, y si la puerta no se puede cerrar, recibiremos una alerta de seguridad directamente en su dispositivo móvil, lo que le permitirá cerrar la puerta de forma remota.
 - Desbloquea puertas desde un teléfono inteligente para que una visita entre a la casa.

2.8.2. Red domótica

Hay tres tipos de redes necesarios en la actualidad para ámbitos de comunicación, control y automatización, y entretenimiento, construidas por el mismo soporte físico o con soporte físicos diferentes:

- **Red de datos:** Es la usamos en una misma red para distribuir ficheros por ordenadores, compartir dispositivos y aplicaciones.
- **Red de entretenimiento:** Es la red que se orienta a distribuir audio y video en el hogar.

- **Red domótica:** Se puede integrar dispositivos y sensores para así automatizar el buen control del hogar.

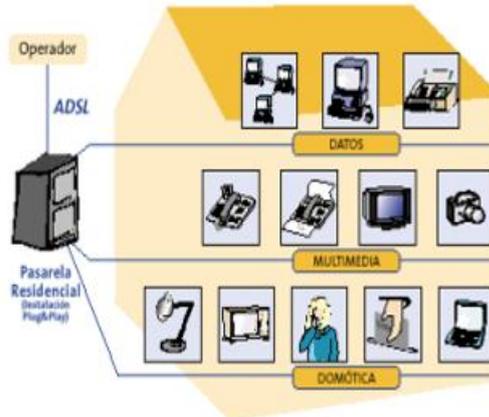


Figura 2. 19: Pasarela residencial
Fuente: (Torres, Gaspoz, & Nicola, 2018)

2.9. WirelessHART

WirelessHART está basado de la misma forma como en ZigBee, ya que este estándar IEEE 802.15.4 en las conexiones inalámbrica se usan los dispositivos de campo HART en industria.

2.9.1. Descripción

Tiene como función principal proveer comunicaciones inalámbricas a dispositivos con protocolo HART. En 2010 se le dio luz verde para ser estándar IEC 62591. Existe una serie de características:

- Transmisión y manipulación confidencial.
- Enrutamiento Full-Mesh con alta fiabilidad.
- Consumo de energía bajo debido a la comunicación sincronizada.
- 2,4 GHz.
- Cuenta con 15 canales de emisión.

En estas condiciones tecnológicas podemos ver que aparecen ciertas ventajas en medios cableados:

- Sin autorizaciones y retrasos.
- Uso de diferentes tipos de instrumentos y diagnóstico para dispositivos HART con tradición cableada.

- No necesita una extensa planificación o estudio en emisiones de radio.

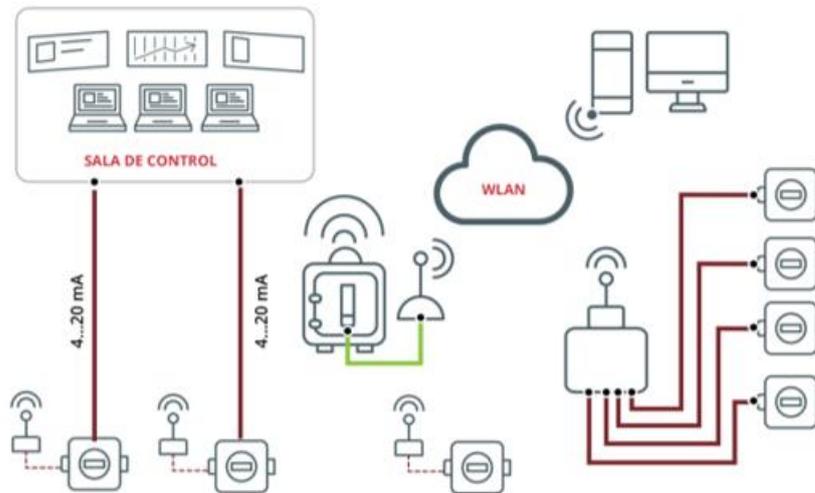


Figura 2. 20: Infraestructura con dispositivos HART y WirelessHART
Fuente: (Torres, Gaspoz, & Nicola, 2018)

2.9.2. Uso en sistemas de control industrial

WirelessHART en los sistemas de control industrial es exclusivo para el envío de señales definidas en el protocolo HART. En estos sistemas utilizados en esta tecnología podemos mencionar:

- Monitoreo de equipos médicos.
- Monitoreo al medio ambiente, gestionando su energía.
- Equipamiento rotatorio

Capítulo 3: Diseño, Implementación y resultados

3.1. Aplicación práctica 1: Redes de sensores con #ESP32 #LORA

El siguiente procedimiento es para adquirir datos del sensor SHT30 I2C, hacia el ESP32 Lora y mostrarlo en su pantalla OLED.



Figura 3. 1: Sensores Esp32 Lora
Elaborado por: Autor

3.1.2. Core de Lora en Arduino

Agregar el core de lora en Arduino, para ello dirigirse al Arduino Ide, en File-Preferencia-Tarjetas Adicionales agregar el siguiente enlace:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

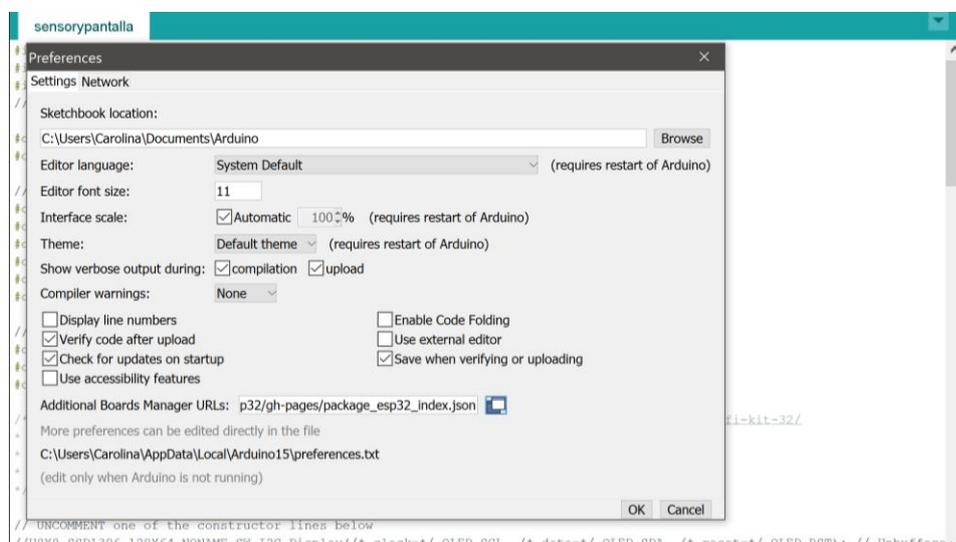


Figura 3. 2: Settings Network Arduino
Elaborado por: Autor

3.1.3. Esp32

Cuando finalice ese proceso dirigirse a Tool-board-boardmanager, e instalar el esp32.



Figura 3. 3: Instalador Esp32

Elaborado por: Autor

3.1.4. Esp32 Tarjeta TIGO Lora32- OLED V1

Se elige la tarjeta TIGO LoRa32- OLED V1 en tools - board "tarjeta actual"

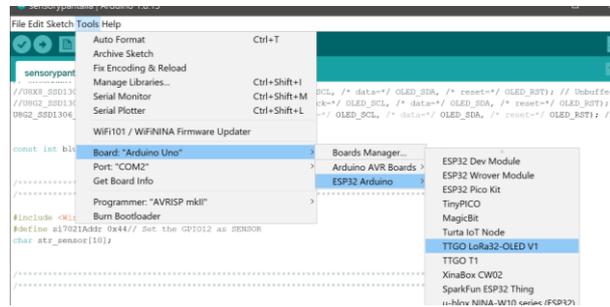


Figura 3. 4: Tarjeta actual

Elaborado por: Autor

3.1.5. Administrador de librerías

Hay que también dirigirse al administrador de librerías y nos colocamos específicamente en agregar las librerías (SPI-I2C) Adafruit BusIO, U8g2 y LoRa.



Figura 3. 5: Librerías (SPI-I2C)

Elaborado por: Autor

3.1.6. Lectura de sensor STH30

Para una buena lectura de nuestro sensor STH30, se realiza por comunicación I2C por ende necesitamos usar la librería I2C y para mostrar la pantalla oled se lo hace por SPI. De esta forma procedemos a construir nuestro programa para sensar y mostrar por pantalla la temperatura y la humedad.

Luego definimos simplemente las constantes para leds, comunicación I2C y para sensor, el sensor y el dispositivo Lora Radio se lo inicia desde el setup, y se realiza las funciones de lectura del sensor.

```
#include <Wire.h>
#include <SPI.h>
#include <LoRa.h>
#include <U8g2lib.h>
```

```
// For LED
#define OFF 0
#define ON 1

// SPI LoRa Radio
#define LORA_SCK 5 // GPIO5 - SX1276 SCK
#define LORA_MISO 19 // GPIO19 - SX1276 MISO
#define LORA_MOSI 27 // GPIO27 - SX1276 MOSI
#define LORA_CS 18 // GPIO18 - SX1276 CS
#define LORA_RST 14 // GPIO14 - SX1276 RST
#define LORA_IRQ 26 // GPIO26 - SX1276 IRQ (interrupt request)

// I2C OLED Display works with SSD1306 driver
#define OLED_SDA 4
#define OLED_SCL 15
#define OLED_RST 16

U8G2_SSD1306_128X64_NONAME_F_SW_I2C Display(U8G2_R0, /* clock=*/ OLED_SCL, /* data=*/ OLED_SDA,
/* reset=*/ OLED_RST); // Full framebuffer, SW I2C

const int blueLED = LED_BUILTIN;

//For Sensor
#define si7021Addr 0x44 // Set the GPIO12 as SENSOR
char str_sensor[10];
```

```
void setup() {
  Serial.begin(115200);
  while (!Serial);
  Wire.begin();
  Wire.beginTransmission(si7021Addr);
  Wire.endTransmission();
  delay(300);
  Display.begin();
  Display.enableUTF8Print(); // enable UTF8 support for the Arduino print() function
  Display.setFont(u8g2_font_ncenB08_te); //u8g2_font_ncenB10_tr , u8g2_font_ncenB08_te
  , u8g2_font_ncenB08_tn , u8g2_font_ncenB08_tr

  // Very important for SPI pin configuration!
  SPI.begin(LORA_SCK, LORA_MISO, LORA_MOSI, LORA_CS);
  // Very important for LoRa Radio pin configuration!
  LoRa.setPins(LORA_CS, LORA_RST, LORA_IRQ);
  pinMode(blueLED, OUTPUT); // For LED feedback
}
```

```

void loop(){
  unsigned int data[6];
  // Start I2C Transmission
  Wire.beginTransmission( si7021Addr);
  // Send measurement command
  Wire.write(0x2C);
  Wire.write(0x06);
  // Stop I2C transmission
  Wire.endTransmission();
  delay(500);
  // Request 6 bytes of data
  Wire.requestFrom( si7021Addr, 6);
  // Read 6 bytes of data
  // cTemp msb, cTemp lsb, cTemp crc, humidity msb, humidity lsb, humidity crc
  if (Wire.available() == 6)
  {
    data[0] = Wire.read();
    data[1] = Wire.read();
    data[2] = Wire.read();
    data[3] = Wire.read();
    data[4] = Wire.read();
    data[5] = Wire.read();
  }
  // Convert the data
  float cTemp = (((data[0] * 256.0) + data[1]) * 175) / 65535.0) - 45;
  float fTemp = (cTemp * 1.8) + 32;
  float humidity = (((data[3] * 256.0) + data[4]) * 100) / 65535.0);
  Display.clearBuffer();
  Display.setCursor(0,10); Display.print("Temp: "+(String)cTemp);
  Display.setCursor(0,30); Display.print("Humedad: "+(String)humidity);
  Display.sendBuffer();
}

```

Figura 3. 6: Sensor de temperatura y humedad
Elaborado por: Autor

3.1.7. Esquema de conexión del sensor

Para esto necesitamos conectar nuestro sensor a los pines SDA y SCL respectivamente acorde a la siguiente imagen y tabla.

Tabla 3. 1: Esquema de conexión del sensor

ESP32 LORA	SENSOR SHT30 I2C
VCC	RED
GND	BLACK
SDA	GREEN
SCL	YELLOW

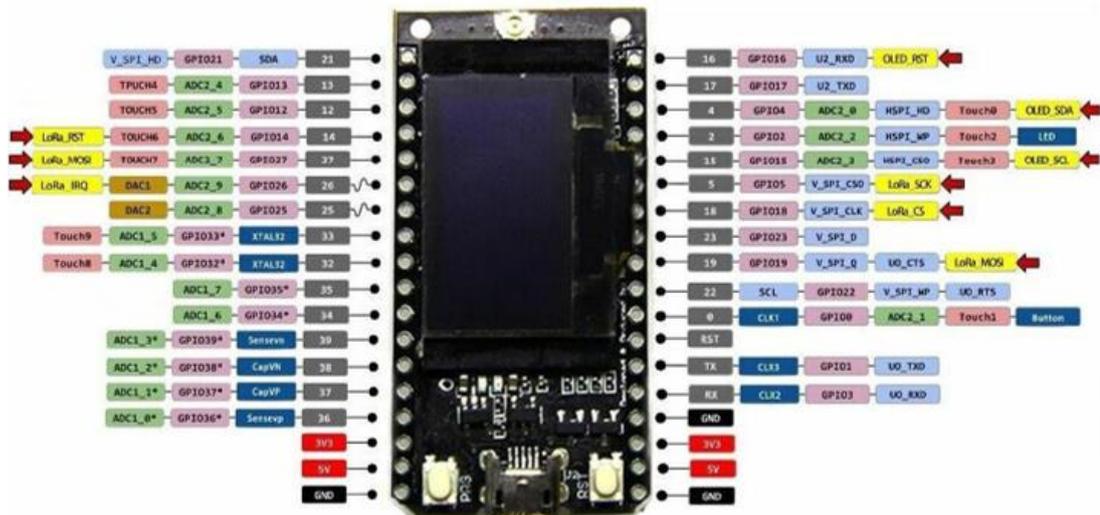


Figura 3. 7: Conectividad del sensor
Elaborado por: Autor

3.1.8. Compilación del sensor Esp32

Configurar la velocidad de la tarjeta en 115200 y verificar que la frecuencia se encuentre en 80MHz, por siguiente proceder a compilar el programa.

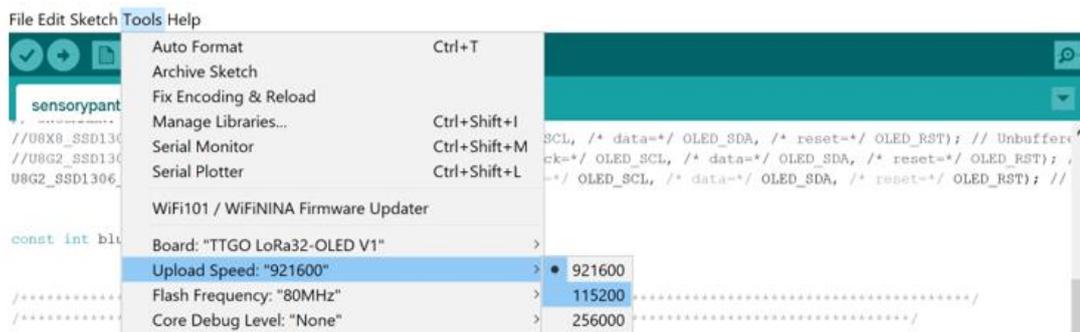


Figura 3. 8: Compilación en sensor Esp32
Elaborado por: Autor

3.2. Aplicación práctica 2: Sensores ESP32 LORA

Implementamos nuestro sensor ESP32 Lora, mostrando los valores y los enviamos a su respectivo receptor, basado en la siguiente figura.



Figura 3. 9: Implementación sensor Esp32 Lora
Elaborado por: Autor

3.2.1. Tarjeta TIGO LORA 32

Elegimos la tarjeta TIGO LORA 32- OLED V1 en tools - board “tarjeta actual”

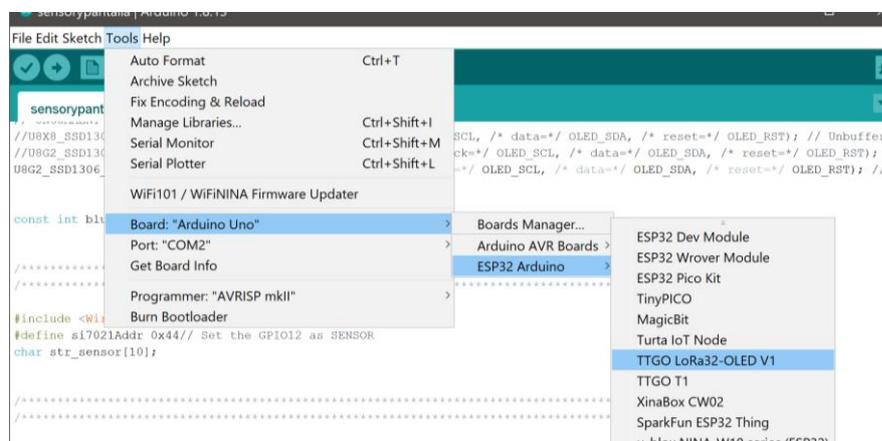


Figura 3. 10: Tarjeta Tigo Lora 32
Elaborado por: Autor

3.2.2. Definición de dispositivo emisor y receptor por códigos

Los dos códigos son para el primer Lora el cual lee la señal de temperatura y humedad, la muestra por pantalla OLED y lo envía por radiofrecuencia. El segundo código recibe encapsulado el dato, lee los datos de temperatura y humedad y muestra los dos valores en la pantalla OLED.

A su vez lo transmite los cuatro valores a otro LORA para crear una red de sensores. Uno Lora se encuentra más distanciado a otro en una central como se muestra en la siguiente imagen.

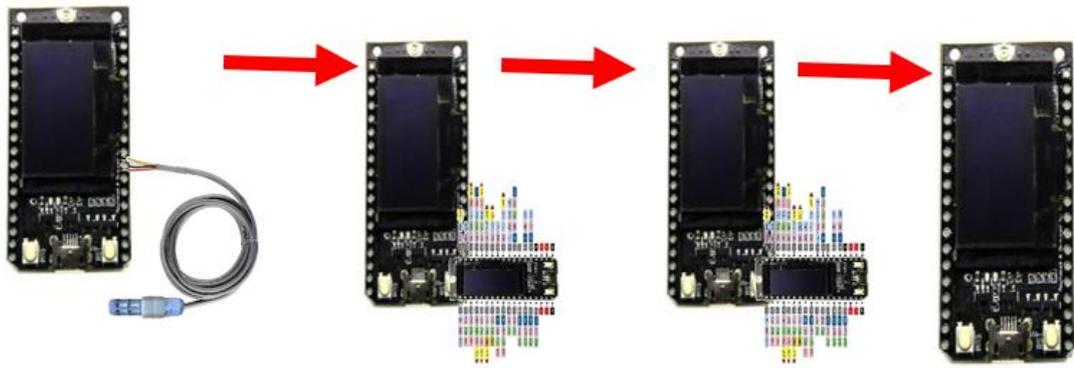


Figura 3. 11: Valores de temperatura y humedad en redes de sensores Lora
Elaborado por: Autor

3.2.3. Dispositivo en función de emisor

Se comienzan definiendo las librerías, constantes y pines necesarios para el emisor, agregando las variables y constante como la banda en que se transmite que depende de la región que se encuentra. En el bloque SetUP agregamos la configuración del sensor y de la tarjeta ESP32 para verificar la banda definiendo así el factor de frecuencia en baud para comunicarse por Lora con un factor de 10.

Después el comportamiento en el lazo loop del dispositivo emisor ESP32 Lora Radio, se agrega sección para enviar datos con una función que recibe el dato encapsulado. Definimos una función que permita enviar datos a otro dispositivo de Lora, Esp32 Lora Radio, para ello se necesita el dato encapsulado, la dirección de destino, dirección local y un mensaje contador como clave de seguridad para enviar datos a un dispositivo Lora en específico.

3.2.4. Dispositivo en función de receptor

Se inicia definiendo librerías, constantes y pines necesarios para el emisor, además reconfigurar y agregar las variables y constantes necesarias. Usamos el mismo bloque set up del emisor reconfigurable en el siguiente comportamiento del lazo loop en bloque de comunicación del dispositivo receptor ESP32 Lora Radio.

Luego se reconfigura el siguiente comportamiento en el lazo loop en bloque COMUNICACIÓN del dispositivo receptor ESP32 Lora Radio. Definir una función que permita recibir datos a otro dispositivo de Lora, Esp32 Lora Radio y agregar la función de enviar datos usando el emisor, y para la función de recibir los datos se usa el tamaño del paquete para verificar si se ha recibido por completo.

Al final se verifica el valor de contador para asegurarse que se recibe datos de un dispositivo Lora en específico, además usar la misma función para enviar datos a otro dispositivo Lora del Emisor y enviar por serial para luego poder conectar con una raspberry con paquete enviado.

3.3 Aplicación práctica 3: Sensores LORA, Serial, a RaspberryPi4 y Thingspeak

Adquirimos datos del sensor SHT30 I2C, e implementar el ESP32 Lora, mostrando así los valores, y enviamos los valores a su respectivo receptor, haciendo comunicación con una raspberry para enviar a la nube.



Figura 3. 12: Aplicación con Lora Serial a RaspberryPi
Elaborado por: Autor

3.3.1. Recepción de paquetes

Definimos el punto final en el cual se recibirá los paquetes como se muestra en la siguiente imagen.

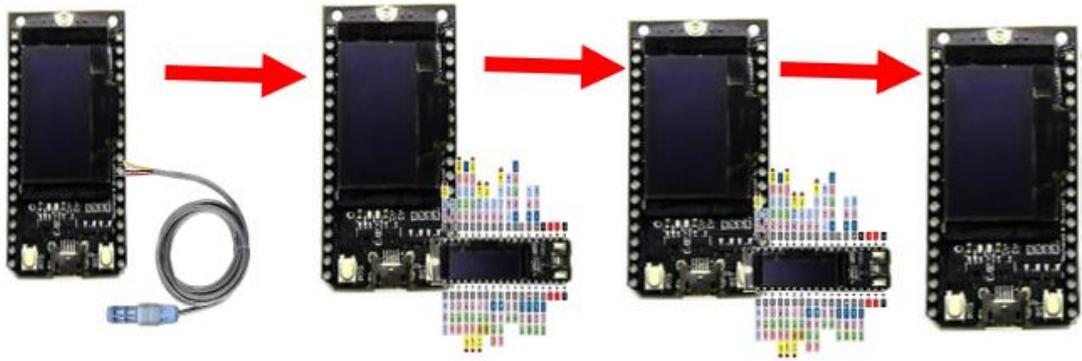


Figura 3. 13: Recepción de paquetes
Elaborado por: Autor

3.3.2. Configuración multipunto

Usamos la configuración de multipunto1 y multipunto 2, de esa forma definimos un multipunto3 con el programa cambiando la variable de referencia local, y recibimos un multipunto2 enviando los datos a un último punto mediante el código dado. Después definimos el último punto que recibe datos del multipunto3, procediendo a enviar los datos a la Raspberry Pi.

3.3.3. Funcionamiento de SD Card y Raspberry Pi

Ingresamos una SD Card a la Raspberry Pi para el encendido de la misma, se necesita descargar la aplicación fing – escaner de red en el celular para encontrar la dirección ip de la raspberry Pi, registrarse en la aplicación, seleccionar Buscar dispositivo y encontrar la ip de la Raspberry Pi.

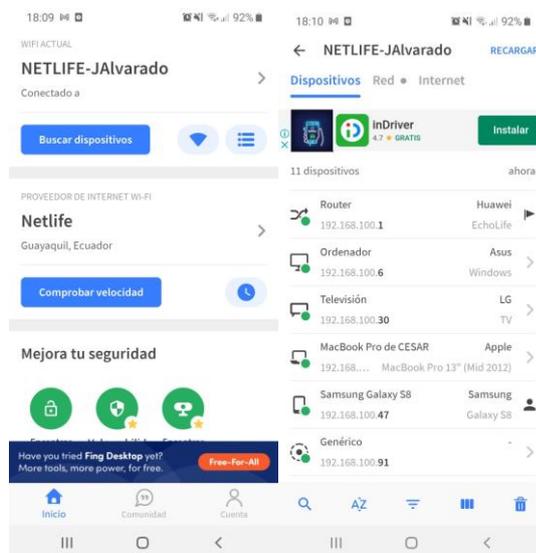
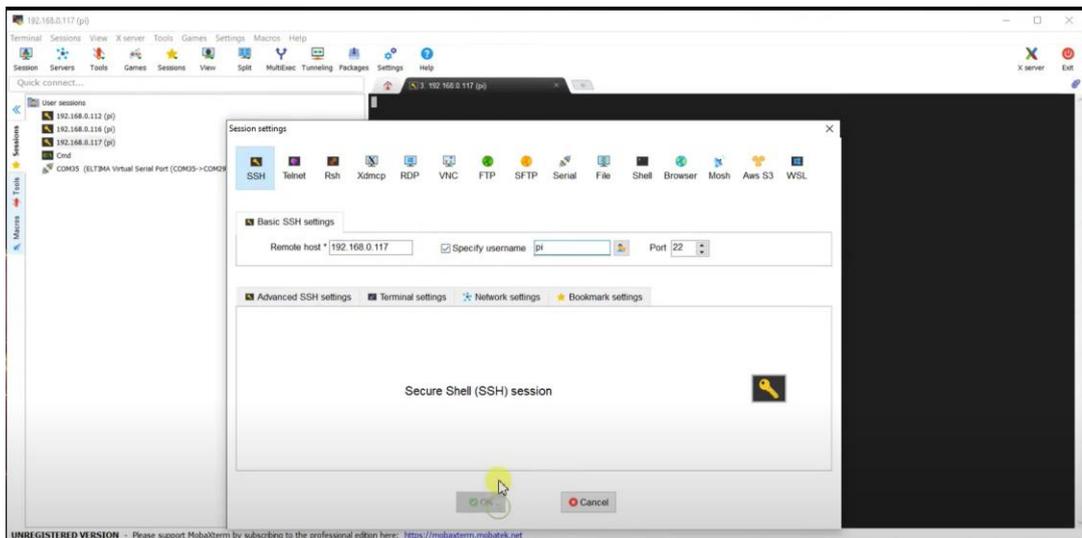


Figura 3. 14: Dirección Ip
Elaborado por: Autor

3.3.4. MobaXterm Python

Colocamos la ip en Remote Host, escribimos usuario Pi y se da click en ok, y luego realizamos la configuración en la raspberry para la comunicación serial. Se crea una carpeta para los archivos del programa de lora y crear un archivo Python a conectarse en thingspeak, Finalmente Verificamos los puertos seriales de la raspberry, colocando en puerto serial. El puerto encontrado que esté usando el dispositivo Esp32 Lora con el código de lectura de temperatura y humedad enviando por serie estos datos.



```
sudo apt-get upgrade  
sudo apt-get install python-serial
```

```
sudo mkdir esp32lora  
cd esp32lora  
sudo nano lorathings.py
```

```
python -m serial.tools.list_ports
```

Figura 3. 15: MobaXterm Python
Elaborado por: Autor

3.3.5. ThingSpeak

Se crea un nuevo canal con nombre Lora Pi configurando field1 y field2 para temperatura y humedad, luego vamos a la sección de API keys y usamos el api key de escritura de canal.

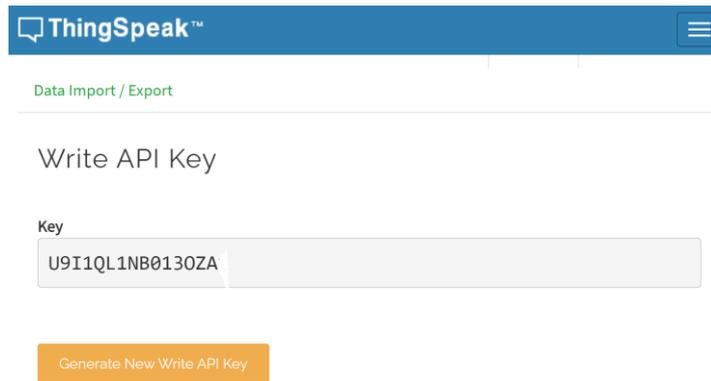


Figura 3. 16: Write API key
Elaborado por: Autor

3.3.6. Compilación de códigos por medio de comandos

Escribimos `sudo python3 lorathings.py`, y finalmente en ThingSpeak se observan los datos de temperatura y humedad recibidos en las gráficas.

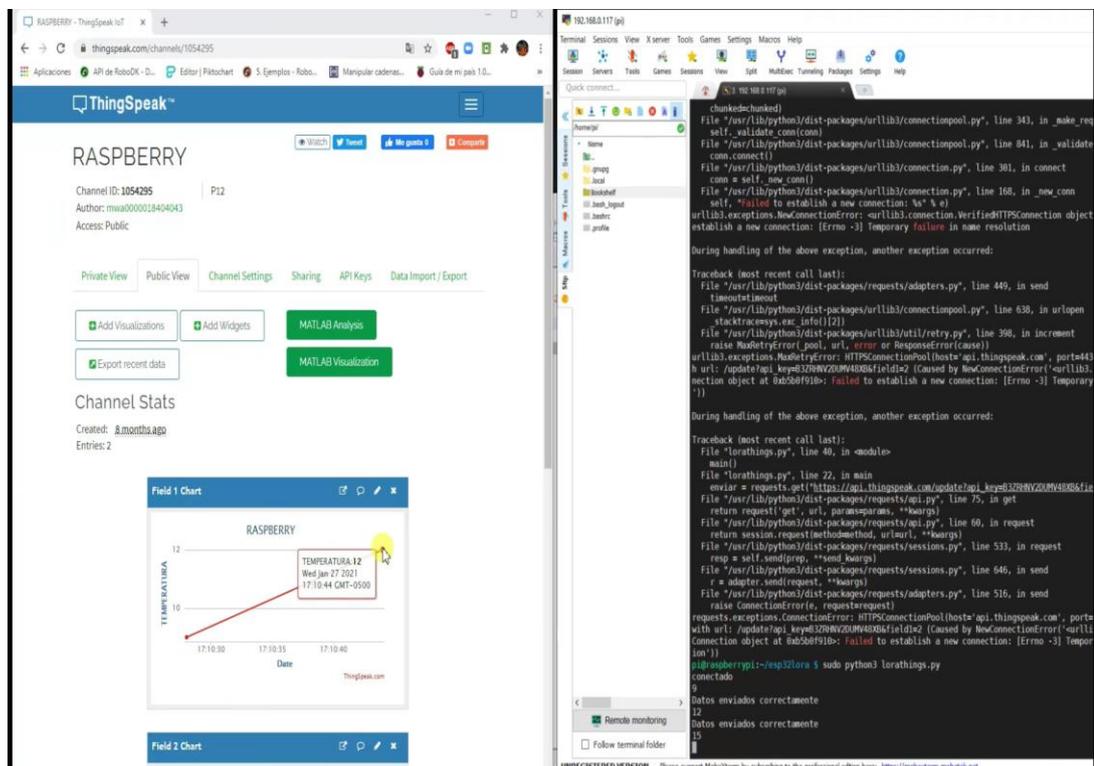


Figura 3. 17: Compilación final del sistema
Elaborado por: Autor

Conclusiones.

- Pudimos realizar la lectura de temperatura y humedad de cada sensor y así pasar sus datos por medio de radiofrecuencia hacia el otro modulo siguiente o también llamado nodo.
- El nodo coordinador presenta la información de los 3 sensores de temperatura y humedad sin necesidad de actuar como sensor, y por su consecuente envía estos datos a la Raspberry Pi.
- La Raspberry Pi posee un código en Python que pude ejecutarlo y también con su recepción de datos por puerto serial los envía a una aplicación en la nube llamada Ubidots.
- Para poder conectarnos a nuestra Raspberry necesitamos conectarnos a nuestro programa MobaXterm y accedemos por SSH de manera inalámbrica usando la red local

Recomendaciones.

- Hacer una investigación más a profundidad de las comunicaciones inalámbricas de sus ventajas y tecnologías para implementar en medios físicos y ambientales como temperatura, presión, sonido, humedad ya antes mencionados.
- Que nuestra aplicación en sensores inalámbricos nos permita en cualquier área de la ingeniería, agroindustria, y domótica un desarrollo de protocolos para tratar y organizar una comunicación de los diferentes elementos de la red.
- Actualizarnos siempre en nuevas innovaciones de programas en Python y Arduino para de esa forma tener un mejor resultado en nuestra compilación de datos e informaciones, en nuestro caso con la Raspberry.
- Es importante mencionar que la red local debe tener acceso a internet para enviar los datos a la nube por medio del programa Ubidots.

Anexos.

```
#include <SPI.h>
#include <LoRa.h>
#include <U8g2lib.h> // https://github.com/olikraus/U8g2_Arduino
// #include <U8x8lib.h>

#define OFF 0 // For LED
#define ON 1

// SPI LoRa Radio
#define LORA_SCK 5 // GPIO5 - SX1276 SCK
#define LORA_MISO 19 // GPIO19 - SX1276 MISO
#define LORA_MOSI 27 // GPIO27 - SX1276 MOSI
#define LORA_CS 18 // GPIO18 - SX1276 CS
#define LORA_RST 14 // GPIO14 - SX1276 RST
#define LORA_IRQ 26 // GPIO26 - SX1276 IRQ (interrupt request)

// I2C OLED Display works with SSD1306 driver
#define OLED_SDA 4
#define OLED_SCL 15
#define OLED_RST 16

/* Pick One. Hardware I2C does NOT work! This article helped:
   https://robotzero.one/heltec-wifi-kit-32/
* TTGo boards similar to Heltec boards, LED_BUILTIN = 2 instead of pin 25
* Some OLED displays don't handle ACK correctly so SW I2C works better. Thank
  you Olikraus!
* TTGo OLED has pin 16 reset unlike other OLED displays
*/

// UNCOMMENT one of the constructor lines below
//U8X8_SSD1306_128X64_NONAME_SW_I2C Display(/* clock=*/ OLED_SCL, /*
  data=*/ OLED_SDA, /* reset=*/ OLED_RST); // Unbuffered, basic graphics,
  software I2C
```

```

//U8G2_SSD1306_128X64_NONAME_1_SW_I2C Display(U8G2_R0, /* clock=*/
  OLED_SCL, /* data=*/ OLED_SDA, /* reset=*/ OLED_RST); // Page buffer,
  SW I2C
U8G2_SSD1306_128X64_NONAME_F_SW_I2C Display(U8G2_R0, /* clock=*/
  OLED_SCL, /* data=*/ OLED_SDA, /* reset=*/ OLED_RST); // Full
  framebuffer, SW I2C

String sensorBlink = "ON"; // inicializa on/off: 1/0

//Banda LoRa - ISM en Región 915Mhz
#define BAND 915E6 // 433E6,868E6,915E6

// Mensaje a enviar por direcciones
byte dir_local = 0xD1; // Dispositivo 1
byte dir_destino = 0xC1; // Concentrador 1
// identificador de mensaje
byte msjContador = 1;
// tiempo entre lecturas
long t_anterior = 0;
int t_intervalo = 5000;

// Mensajes por Puerto Serial
volatile boolean serial_msj = true;

const int blueLED = LED_BUILTIN;

/*****|lectura de
  sensor*****/
/*****
  *****/

#include <Wire.h>
#define si7021Addr 0x44// Set the GPIO12 as SENSOR

```

```
char str_sensor[10];
```

```
/*  
*****/  
/*  
*****/
```

```
void setup() {
```

```
  Serial.begin(115200);
```

```
  while (!Serial);
```

```
  /******sensor*****/
```

```
  Wire.begin();
```

```
  Wire.beginTransmission(si7021Addr);
```

```
  Wire.endTransmission();
```

```
  delay(300);
```

```
  /******sensor*****/
```

```
  Display.begin();
```

```
  Display.enableUTF8Print();    // enable UTF8 support for the Arduino print()  
  function
```

```
  Display.setFont(u8g2_font_ncenB08_te);           //u8g2_font_ncenB10_tr
```

```
    u8g2_font_ncenB10_tf u8g2_font_ncenB08_te      u8g2_font_ncenB08_tn
```

```
    u8g2_font_ncenB08_tr
```

```
  // Very important for SPI pin configuration!
```

```
  SPI.begin(LORA_SCK, LORA_MISO, LORA_MOSI, LORA_CS);
```

```
  // Very important for LoRa Radio pin configuration!
```

```
  LoRa.setPins(LORA_CS, LORA_RST, LORA_IRQ);
```

```
  pinMode(blueLED, OUTPUT); // For LED feedback
```

```

if (!LoRa.begin(BAND)) {
  //Serial.println("Starting LoRa failed!");
  while (1);
}
// The larger the spreading factor the greater the range but slower data rate
// Send and receive radios need to be set the same
LoRa.setSpreadingFactor(10); // ranges from 6-12, default 7 see API docs

// Change the transmit power of the radio
// Default is LoRa.setTxPower(17, PA_OUTPUT_PA_BOOST_PIN);
// Most modules have the PA output pin connected to PA_BOOST, gain 2-17
// TTGO and some modules are connected to RFO_HF, gain 0-14
// If your receiver RSSI is very weak and little affected by a better antenna, change
  this!
//LoRa.setTxPower(14, PA_OUTPUT_RFO_PIN);

}

void loop(){
  unsigned int data[6];

  digitalWrite(blueLED, HIGH); //EMPIEZA A LEER SENSOR

  // Start I2C Transmission
  Wire.beginTransmission( si7021Addr);
  // Send measurement command
  Wire.write(0x2C);
  Wire.write(0x06);
  // Stop I2C transmission
  Wire.endTransmission();
  //delay(5000);

```

```

// Request 6 bytes of data
Wire.requestFrom( si7021Addr, 6);

digitalWrite(blueLED, LOW); //TERMINA DE LEER SENSOR

// Read 6 bytes of data
// cTemp msb, cTemp lsb, cTemp crc, humidity msb, humidity lsb, humidity crc
if (Wire.available() == 6)
{
  data[0] = Wire.read();
  data[1] = Wire.read();
  data[2] = Wire.read();
  data[3] = Wire.read();
  data[4] = Wire.read();
  data[5] = Wire.read();
}

// Convert the data
float cTemp = (((data[0] * 256.0) + data[1]) * 175) / 65535.0 - 45;
float fTemp = (cTemp * 1.8) + 32;
float humidity = (((data[3] * 256.0) + data[4]) * 100) / 65535.0;

// Enviar mensajes entre intervalos
long t_ahora = millis();
long t_transcurrido = t_ahora - t_anterior;

if (t_transcurrido >= t_intervalo){

  String paqueteEnv = (String)cTemp+", "+(String)humidity;
  enviarlora(dir_destino, dir_local,

```

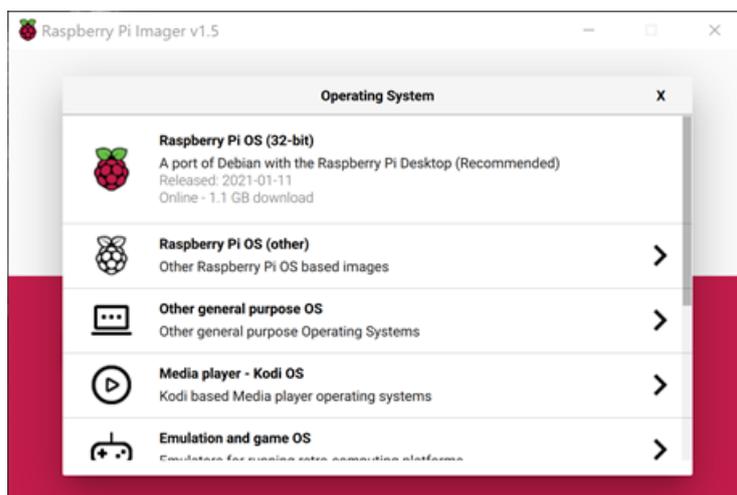
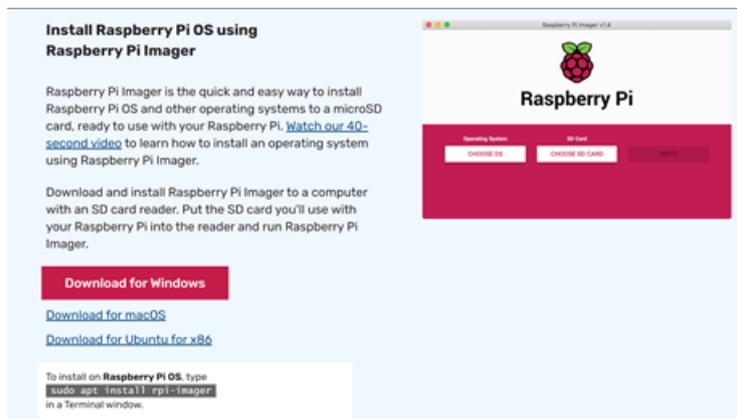
```

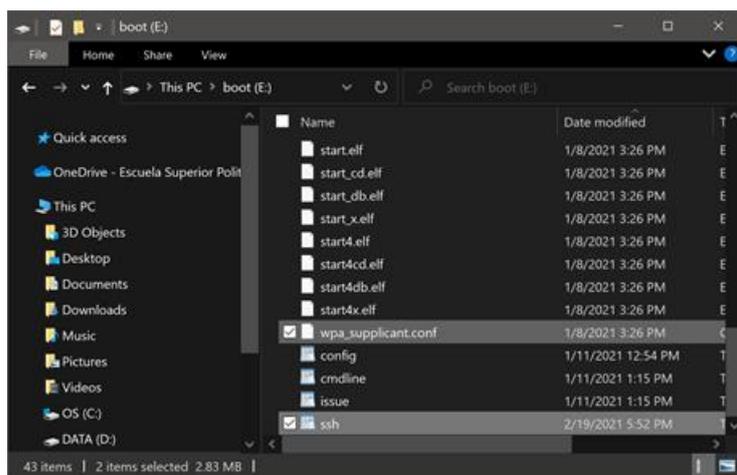
msjContador, paqueteEnv);

yield();

// mensaje a serial
if (serial_msj==true){
  Serial.print(String(dir_destino,HEX));
  Serial.print(",");
  Serial.print(String(dir_local,HEX));
  Serial.print(",");
  Serial.print(msjContador);Serial.print(",");
  Serial.println(paqueteEnv);
}

```





```
# /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GRUOUP=netdev
update_config=1
country=EC
network{
  ssid="nombre de tu red SSID"
  psk = "contraseña"
  key_mgmt=WPA-PSK
```



Lora Pi TyH

Channel ID: 1309337
Author: macaagui
Access: Private

- Private View
- Public View
- Channel Settings
- Sharing
- API Keys
- Data Import / Export

- Add Visualizations
- Add Widgets
- Export recent data

Channel Stats

Created: about a minute ago
Entries: 0

```

import time
import serial
import requests

def main():
    try:
        ser = serial.Serial('PUERTO SERIAL ',115200,timeout = 3.0)
        print("conectado")

        while True:

            ser.reset_input_buffer()
            time.sleep(5)
            if (ser.in_waiting > 0):
                var1 = ser.readline().strip()
                #lista = line.decode().split(',')
                print(var1.decode())
                #Escritura de datos a Thingsboard
                #if len(lista) == 2:
                enviar =
requests.get("https://api.thingspeak.com/update?api_key=XXXXXXXXXXXXXXXXXX&fi
eld1="+str(lista[0])+"&field2="+str(lista[1])) #cuando se quiere enviar dos o mas
datos
                #enviar =
requests.get("https://api.thingspeak.com/update?api_key=XXXXXXXXXXXXXXXXXX&fie
ld1="+str(var1.decode()))
                if enviar.status_code == requests.codes.ok:
                    if enviar.text != '0':
                        print("Datos enviados correctamente")
                    else:
                        print("Tiempo de espera insuficiente (>15seg)")
                else:
                    print("Error en el request: ",enviar.status_code)
                #else:
                #print("La cadena recibida no contiene 2 elementos, sino:",len(lista),"elementos")

            time.sleep(10)

    except KeyboardInterrupt: #Cierra el serial cuando el usuario cierra forzosamente el
proceso
        print ()
        ser.close()

if __name__ == '__main__':
    main()

```

Bibliografía.

- Crespo, E. (2016). *Aprendiendo Arduino*. Obtenido de <https://aprendiendoarduino.wordpress.com/acerca-de/>
- MACSANGAR90. (2014). *noescomolocuantan*. Obtenido de <https://noescomolocuantan.wordpress.com/2014/11/21/primeros-pasos-con-xbee-y-nueva-version-xctu/>
- Perez, J., Urdaneta, E., & Custodio, Á. (2014). *Metodología para el diseño de una red de sensores inalámbricos*. Obtenido de http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-48212014000100002
- Quiñones, M., González, V., Torres, R., & Jumbo, M. (2017). *Redalyc*. Obtenido de https://www.researchgate.net/figure/Sensor-Node-Architecture_fig1_334093763
- Salazar, J. (2019). *Redes Inalámbricas*.
- Torres, J., Gaspoz, P., & Nicola, R. (2018). *DOMOTICA E INMOTICA*.



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Navarrete Armijos, José Ignacio** con C.C: # 070430246-2 autor del Trabajo de Titulación: **Implementación de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi** 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, 9 de Marzo del 2021

f. 

Nombre: Navarrete, José Ignacio

C.C: 070430246-2



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Implementación de una red de sensores inalámbricos mediante sistema embebido Raspberry Pi		
AUTOR(ES)	Navarrete Armijos, José Ignacio		
REVISOR(ES)/TUTOR(ES)	MSc. Edwin Fernando, Palacios Meléndez		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	De Telecomunicaciones		
TITULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	9 de marzo del 2021	No. DE PÁGINAS:	46
ÁREAS TEMÁTICAS:	Telecomunicaciones, Programación, Electrónica		
PALABRAS CLAVES/ KEYWORDS:	Nodos, Sensores, Redes, Protocolos, Radiofrecuencia, Sistema.		
RESUMEN/ABSTRACT (150-250 palabras):			
<p>El proyecto de titulación básicamente se trata de una implementación de sensores inalámbricos mediante un sistema embebido Raspberry Pi que es una serie de ordenadores de placa reducida para proyectos de informática y creaciones digitales, y pues que se el trabajo conjunto con cuatro nodos. El primer nodo tiene conectado el sensor de temperatura y humedad con conexión I2C que utiliza dos pines y conecta múltiples sensores y actuadores. Con el módulo Esp32 Lora y por medio de I2C realiza la lectura de temperatura y humedad enviándolo por radiofrecuencia a nuestro nodo posterior que es el 2. El nodo dos recibe los datos de nuestro primer nodo y además lee los resultados de temperatura y humedad de su propio sensor, y esas ds informaciones van hacia el nodo tres. El nodo 3 presenta humedad y temperatura de los nodos anteriores, y claro también presento los mismos datos de su sensor. El nodo tres envía los datos anteriores al nodo principal que es el coordinador y este envía por puerto serial a la Raspberry Pi la información de los nodos. La Raspberry Pi tiene un código en Python que se ejecuta y por medio del puerto serial mencionado envía esos datos a una aplicación en la nube llamada Ubidots. La Raspberry Pi tiene instalado el sistema operativo raspbian que está en una microSD. Para concluir para acceder a la Raspberry Pi lo hacemos por MobaXterm y por SSH inalámbrica con una red local conectada a internet, y transmitir a Ubidots</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593981447512	E-mail:	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Palacios Meléndez, Edwin Fernando		
	Teléfono: +593-9-67608298		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			