

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Despliegue de redes malladas ZigBee para aplicaciones de edificios
inteligentes

PROYECTO INTEGRADOR

Previo la obtención del Título de:

INGENIERO EN TELEMÁTICA

Presentado por:

Carvajal Cabay Héctor Byron

Mora Pérez Claudia Melina

GUAYAQUIL - ECUADOR

Año: 2018

DEDICATORIA

Primeramente, a Dios por la vida y todas sus bendiciones, a nuestros padres, amigos, profesores, ayudantes y demás familiares.

AGRADECIMIENTOS

Al Centro de información e innovación, CTI-Espol por habernos abierto las puertas en sus prestigiosos laboratorios, a nuestra querida profesora PhD. Rebeca Estrada, por toda la paciencia y apoyo, a nuestro querido tutor PhD. Federico Domínguez por guiarnos con amor y sabiduría en cada fase del proyecto.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Héctor Byron Carvajal Cabay, Claudia Melina Mora Pérez* y damos nuestro consentimiento para que la ESPOLE realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Héctor B. Carvajal
Cabay

Claudia M. Mora Pérez

EVALUADORES

PhD. REBECA ESTRADA PICO

PROFESOR DE LA MATERIA

PhD. FEDERICO DOMINGUEZ

PROFESOR TUTOR

RESUMEN

El Centro de Tecnologías de Información, CTI-Espol, es un espacio de investigación considerada un área de acceso controlado, por medio de tarjetas las cuales permiten el ingreso y egreso de los investigadores y personal, sin embargo, este sistema representa una limitación para los visitantes del CTI.

El presente proyecto planteo diseñar un sistema de timbre inteligente que implemente redes malladas de ZigBee, para el análisis de alta seguridad y accesibilidad en áreas restringidas, para lo cual se requirió de la utilización de módulos XBee, Arduino UNO, Raspberry pi 3, se desarrolló una App y se usó de un servidor MQTT bróker, este a su vez notifica en el móvil y con advertencias visuales (leds) muestra al usuario el correcto funcionamiento.

Así pues, al accionar el botón del timbre el sistema activa el led amarillo, que indica el envío de un mensaje al nodo más cercano de la red Zigbee, posteriormente el receptor procesa el mensaje para el aplicativo móvil, donde se presentan las notificaciones del timbre a los usuarios suscritos a un tópico (código QR - puerta), para esperar un mensaje de aceptación que confirma al usuario demandante que su pedido será atendido, encendiendo el led verde que interrumpe el parpadeo del led amarillo mencionado anteriormente, caso contrario el sistema mostrará al usuario demandante por medio de un led rojo.

Finalmente, se aclara que el sistema permite notificar la presencia de un visitante en tiempo real a los usuarios suscritos a determinada puerta, es escalable y de potencial crecimiento ya que permite aumentar considerablemente la cantidad de nodos y además tiene la capacidad de reconfigurar la red automáticamente si falla un nodo, todo esto fue sustentado mediante el plan de pruebas que se ejecutó para realizar la comparativa de nuestro sistema con el que implementa Wifi, donde se logró obtener los resultados esperados, que se presentan con detalle en el capítulo 4.

Palabras Clave: Timbre Inteligente, Internet de las cosas, Protocolo ZigBee, Módulos XBee.

ABSTRACT

The Information Technology Center, CTI-Espol, is a research space considered a controlled access area, by means of cards that allow the entry and exit of researchers and staff, however, this system represents a limitation for visitors of the CTI.

The present project proposed to design an intelligent doorbell system that implements ZigBee mesh networks, for the analysis of high security and accessibility in restricted areas, where XBee, Arduino UNO, Raspberry pi 3 modules were used, an App was developed and an MQTT broker server was used, this system through notifications on the mobile and visual warnings (LEDs) show the user the correct operation of the same.

Thus, when the bell button is pressed, the system activates the yellow LED, which indicates that it sent a message to the nearest node of the Zigbee network, then the receiver processes the message for the mobile application, where the bell notifications are presented. to users subscribed to a topic (QR code - door), to wait for an acceptance message that confirms to the requesting user that their request will be attended, turning on the green led that interrupts the yellow LED mentioned above, otherwise the system will show to the demanding user by means of a red LED.

Finally, it is clarified that the system allows to notify the presence of a visitor in real time to users subscribed to a certain door, it is scalable and of potential growth since it allows to increase considerably the number of nodes and also has the ability to reconfigure the network automatically if a node fails, all this was supported by the test plan that was executed to make the comparison of our system with the one that implements Wifi, where the expected results were obtained, which are presented in detail in chapter 4.

Keywords: Smart DoorBell, IoT, ZigBee protocol, XBee modules

ÍNDICE GENERAL

EVALUADORES	5
RESUMEN	6
<i>ABSTRACT</i>	7
ÍNDICE GENERAL.....	8
ABREVIATURAS.....	11
SIMBOLOGÍA.....	12
ÍNDICE DE FIGURAS	13
ÍNDICE DE TABLAS	14
CAPÍTULO 1	15
1. Introducción	15
1.1 Descripción del problema.....	15
1.2 Delimitación del problema	16
1.3 Posibles soluciones.....	16
1.4 Solución Propuesta	17
Realizar una conexión en malla que se conecte a un servidor, el cual notificará mediante el envío de un mensaje al aplicativo móvil cuando alguien presione el timbre y a su vez mediante leds indicadores muestre a la persona que accione el timbre el estado de su petición.....	17
1.5 Justificación.....	17
1.6 Objetivos	19
1.6.1 Objetivo General.....	19
1.6.2 Objetivos Específicos	19
1.7 Metodología	19

1.7.1	Actividad 1	20
1.7.2	Actividad 2	20
1.7.3	Actividad 3	21
1.8	Resultados Esperados	21
CAPÍTULO 2		22
2.	MARCO TEÓRICO	22
2.1	Módulo XBee S2	22
2.2	Adaptador USB XBee	23
2.3	Arduino UNO.....	24
2.4	Arduino Wireless Proto Shield.....	25
2.5	Raspberry Pi 3	26
2.6	Tarjeta Micro SD	27
2.7	IDE DE ARDUINO.....	28
2.8	SD CARD FORMATTER.....	29
2.9	WIN32 DISK IMAGER.....	29
2.10	XCTU	30
2.11	RASPBIAN.....	31
2.12	PYTHON.....	32
2.13	Elementos utilizados	33
2.14	Costos del proyecto	34
CAPÍTULO 3		35
3.	Descripción de la solución.....	35
3.1	Descripción general	35
3.1.1	Recursos para la implementación del timbre inteligente	35

3.2	Método de la solución	36
3.2.1	Configuración de Módulos XBee S2	36
3.2.2	Configuración de Arduino UNO	38
3.3	Configuración de la placa Raspberry	41
3.4	APLICACIÓN MÓVIL PARA TIMBRE INTELIGENTE.....	42
3.5	CONFIGURACIÓN MQTT BRÓKER	43
CAPÍTULO 4		44
4.	DeTalles de las pruebas	44
4.1	Resultados de la primera prueba	44
4.2	Resultados de la segunda prueba.....	45
4.3	Resultados de la tercera prueba	45
4.4	Resultados de las pruebas de rendimiento	45
4.5	Estabilidad de comunicación del dispositivo	49
4.6	Estudio Energético del Prototipo	50
CONCLUSIONES Y RECOMENDACIONES		52
ANEXOS		54

ABREVIATURAS

ESPOL Escuela Superior Politécnica del Litoral

CTI Centro de Tecnología e Investigación

IEEE Institute of Electrical and Electronics Engineers

XCTU XBee Configuration and Test Utility

SIMBOLOGÍA

mV	Milivoltio
V	Voltio
m	Metro
s	Segundo
A	Amperio

ÍNDICE DE FIGURAS

Figura 1.4 Describe la solución de manera general. [1]	17
Figura 2.1 Módulo XBee S2. [1]	23
Figura 2.2 Adaptador USB XBee. [1].....	24
Figura 2.3 Arduino UNO. [1].....	25
Figura 2.4 Arduino Wireless Proto Shield. [1].....	26
Figura 2.5 Raspberry Pi 3. [1]	27
Figura 2.6 Tarjeta Micro SD. [1]	28
Figura 2.7 Entorno grafico del IDE de Arduino. [1].....	28
Figura 2.8 Vista general del entorno del Software SD Card Formatter. [1]	29
Figura 2.9 Vista general de Entorno de la herramienta Win32 Disk Imager. [1].....	30
Figura 3.2.1 Topología Mesh basado en protocolo Zigbee que se visualiza en la aplicación X-CTU.[1]	38
Figura 3.2.2 Grafico ilustrativo del montaje del módulo XBee (End Device) con Arduino UNO, en la fase de implementación en protoboard. [1]	40
Figura 3.2.3 Esquemático del circuito para el timbre inteligente.[1]	40
Figura 3.2.4 PCB del circuito para el timbre inteligente.[1].....	41
Figura 4.5.1 Grafico comparativo de tiempo promedio de respuesta vs. Distancia entre nodos [1]	50
Figura 4.6.1 Grafico de nivel de voltaje. [1].....	51

ÍNDICE DE TABLAS

Tabla 1.3.1 Descripción de ideas para posible solución del problema 1	16
Tabla 1.5.2 Análisis comparativo de ventajas y desventajas acerca de las posibles soluciones del problema con sus respectivos costos referenciales. [1].....	18
Tabla 2.14.1 Tabla de componentes utilizados con sus respectivos costos y costo total del proyecto. [1].....	34
Tabla 3.2.1 Parámetros configurados en los módulos XBee S2 empleados en la implementación de la topología de red Mesh. [1].....	37
Tabla 3.2.2 Conexión de pines entre Arduino UNO y Proto Wireless Shield. [1]	39
Tabla 4.4.1 Tiempo promedio de encendido de ambas tecnologías. [1]	46
Tabla 4.4.2 Tiempo promedio de respuesta de ambas tecnologías. [1].....	47
Tabla 4.5.1 Tiempo promedio de respuesta entre nodos a diferentes distancias.[1]	49

CAPÍTULO 1

1. INTRODUCCIÓN

El edificio inteligente del Centro de Tecnologías de Información, CTI-Espol, es un espacio de investigación donde se desarrollan proyectos reales en los cuales convergen tecnologías emergentes y participación multidisciplinaria, por esta razón se la considera un área de acceso controlado, por medio de tarjetas que permiten el ingreso y egreso de los investigadores y personal, sin embargo este sistema representa una limitación para los visitantes, ante esta necesidad cabe la búsqueda de una solución que posibilite de manera efectiva el control y la atención adecuada a los invitados del CTI.

1.1 Descripción del problema

Actualmente, el acceso al CTI se realiza a través de un número limitado de tarjetas, que son facilitadas por el guardia de seguridad, estas posibilitan el acceso al edificio y a las áreas restringidas que debido a la seguridad lo requieren. Cabe indicar, que cuando se acaban las tarjetas los visitantes deben esperar que quienes ingresaron previamente egresen y devuelvan la tarjeta, esta espera fluctúa entre 5-30 minutos lo que representa descortesía e incomodidad para quienes acuden al CTI. A lo antes expuesto se debe agregar el inconveniente que tiene el visitante quien una vez que ha ingresado, no podrá movilizarse entre las diversas áreas.

Para resolver la necesidad planteada se propone la implementación de un timbre inteligente que mejore la atención a los visitantes y facilite la movilidad de estos dentro del CTI.

1.2 Delimitación del problema

Se realizará un sistema de timbre inteligente basado en despliegue de redes malladas Zigbee para aplicaciones de edificios inteligentes que se instalará en el edificio CTI-Espol, en la puerta principal del área DOMO que da acceso al Laboratorio de prototipado, el cual envía notificaciones a la aplicación móvil que alertará a los usuarios suscritos la presencia de visitantes externos e indicará a dichos visitantes mediante leds indicadores el estado del mensaje de petición del timbre.

1.3 Posibles soluciones

Con la premisa de la necesidad de un timbre inteligente en el edificio CTI-Espol, se desarrollaron diferentes alternativas, las cuales registran en la siguiente tabla:

Tabla 1.3.1 Descripción de ideas para posible solución del problema [1]

Idea	Descripción
1	Brindar un código QR a los visitantes, el cual indicará la ruta a tomar para llegar al área de destino.
2	Revisar diferentes tipos de protocolos de comunicación, tales como SigFox, bluetooth y WIFI para comparar entre dispositivos.
3	Implementar una red Mesh conectada a un servidor MQTT broker, la cual se activará cada vez que se presione el timbre que notifica a los usuarios conectados a través de una App.
4	Establecer por medio de una cámara un sistema que permita la comunicación con personas asociadas a una Red.

1.4 Solución Propuesta

Realizar una conexión en malla que se conecte a un servidor, el cual notificará mediante el envío de un mensaje al aplicativo móvil cuando alguien presione el timbre y a su vez mediante leds indicadores muestre a la persona que accione el timbre el estado de su petición.

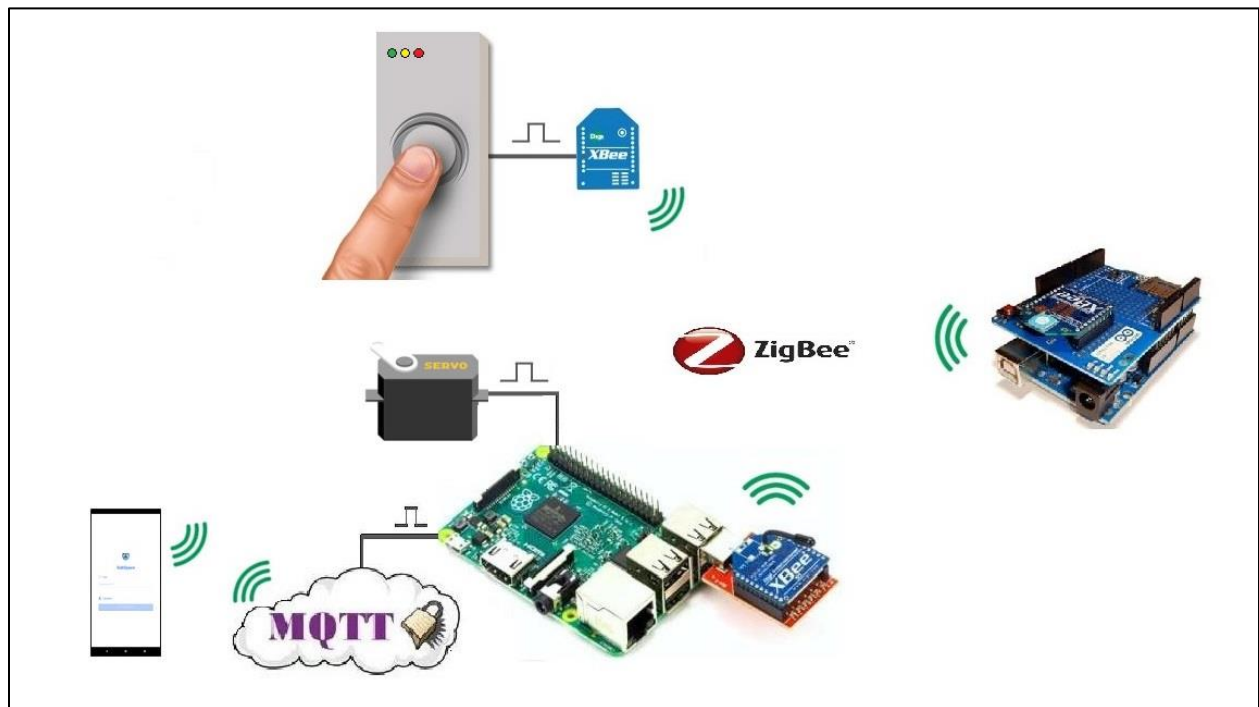


Figura 1.4 Describe la solución de manera general. [1]

1.5 Justificación

La propuesta de solución a la necesidad planteada es conveniente debido a que en la actualidad existe un sistema de ingreso al CTI, que no satisface adecuadamente a visitantes ni a algunos investigadores que laboran en el centro, pues ante la presencia de visitantes, estos no tienen posibilidades de un acceso adecuado a las áreas restringidas, lo cual en muchas ocasiones interfiere con el desarrollo de las actividades laborales.

La trascendencia de la instalación de un nuevo sistema de ingreso contribuirá con la calidad del servicio, ya que los visitantes podrán contactarse con los

investigadores con un toque del timbre que se reflejará en la aplicación móvil instalada en cada celular y además recibir retroalimentación visual del estado de solicitud del timbre mediante leds indicadores. Este acceso incorpora la posibilidad de que el visitante se movilice con mayor libertad entre las distintas áreas.

En la siguiente tabla, se dan a conocer de forma general las ideas a las posibles soluciones del problema del timbre inteligente.

Tabla 1.5.2 Análisis comparativo de ventajas y desventajas acerca de las posibles soluciones del problema con sus respectivos costos referenciales. [1]

Idea	Ventajas	Desventajas	Costos
1	<ul style="list-style-type: none"> - Fácil de usar - Económico - Fácil de implementar - Alto rendimiento 	Es una red muy dependiente del estado del internet.	\$100
2	<ul style="list-style-type: none"> - Mejor diversificación - Uno de los más Económicos - Mejor enfoque del producto - Alto rendimiento 	Requiere de mucho tiempo de investigación para las diferentes comparaciones.	\$250
3	<ul style="list-style-type: none"> - Escalabilidad - Soporte para múltiples topologías de red - Son más baratos y de construcción más sencilla - Alto rendimiento 	Requiere de implementar el protocolo Zigbee.	\$200
4	<ul style="list-style-type: none"> - Fácil uso - Fácil de instalar - Rápido manejo - Alta seguridad 	Requiere de mucho estudio sobre la conexión y transferencia de datos.	\$300

Con referencia a la tabla 1.5.2 y a la matriz de impacto dificultad (Anexo A), después de realizar un análisis comparativo de costos referenciales, ventajas y desventajas se determinó que la idea 3 es la que más se adaptó a los

requerimientos especificados a resolver del problema de timbre inteligente en el edificio del CTI, debido a que en sus laboratorios tienen los materiales ideales para implementar el diseño de la idea propuesta y además esta solución sugiere un alto impacto y baja dificultad.

1.6 Objetivos

1.6.1 Objetivo General

Diseñar un timbre inteligente con un sistema que implemente redes malladas de Zigbee, para el análisis de alta seguridad y accesibilidad en áreas restringidas que requieran este tipo de servicio.

1.6.2 Objetivos Específicos

1. Identificar las tecnologías de IoT para la construcción de un prototipo comercial, de fácil manejo y bajo consumo energético.
2. Diseñar una topología basada en redes malladas para el desarrollo de un timbre inteligente.
3. Contrastar, analizando ventajas y desventajas existentes, entre el diseño del timbre inteligente con tecnología WIFI/Ipv4 y uno que implemente redes malladas de Zigbee estableciendo la eficiencia de cada uno de estos.

1.7 Metodología

En relación a los objetivos específicos propuestos se detalla a continuación los aspectos metodológicos puntuales a desarrollar para la elaboración del proyecto.

1.7.1 Revisión de la literatura e infraestructura existente

Se investigaron los diferentes tipos de Brókers existentes para MQTT, paralelamente se consultaron en documentos académicos sobre proyectos que han implementado IoT y utilizaron el protocolo ZigBee.

La consulta en las fuentes permitió establecer las configuraciones básicas del módulo XBee S2 programándolo desde el X-CTU. Así mismo se encontraron los códigos referenciales que contribuyeron a la construcción de la conexión entre el Bróker y la malla ZigBee.

1.7.2 Configuración de módulos e instalación de software.

Para asignar los parámetros necesarios a fin de establecer la comunicación pertinente, se configuraron los módulos XBee que conformaron la malla ZigBee mediante el programa XCTU. Este diseño requirió instalación del programa antes mencionado, en esta fase se configuraron en el Arduino UNO los códigos de transmisión, recepción y retroalimentación visual mediante leds, para la comunicación bidireccional entre la aplicación móvil y la malla, a través del MQTT instalado en la Raspberry.

Previamente a la actividad antes descrita se instalaron: el broker con su respectivo Topic, las librerías pyserial, paho y digi-xbee a fin de que funcionen en python3.

Simultáneamente al proceso antes mencionado, se desarrolló una aplicación híbrida denominada SubSpace que permitirá a los colaboradores del CTI la recepción de notificaciones del timbre inteligente.

1.7.3 Validación del proyecto y comparación con la versión implementada con wifi.

Para realizar las pruebas necesarias que permiten contrastar y analizar nuestro sistema frente una versión anterior que implementa Wifi, se procedió a elaborar un plan de pruebas detallado, donde se dio a conocer los lineamientos y directrices generales, para llevar a cabo cada una de ellas, así como también se especificaron de las condiciones en la que fueron tomados los datos en cada sistema.

Posteriormente, para el análisis de resultados a cada prueba, se aplicaron conocimientos de estadística general y a través de gráficos se ilustraron las muestras tomadas de ambos sistemas para su comparación respectiva, detallando de manera sencilla la eficiencia del presente proyecto.

1.8 Resultados Esperados

Un sistema compuesto por módulos de Xbee S2 que formen una red mallada y a su vez permita establecer conexión entre sus dispositivos terminales con un servidor MQTT; este servidor enviará las notificaciones al aplicativo móvil y será de capaz de:

1. Mejorar el tiempo de espera de los visitantes.
2. Evitar el uso de las tarjetas de ingreso que se agotaban fácilmente.
3. Controlar el acceso a áreas restringidas por los trabajadores de dichas áreas específicas.
4. Enviar las notificaciones al aplicativo móvil de manera rápida.

CAPÍTULO 2

2. MARCO TEÓRICO

Este capítulo describe los aspectos teóricos acerca de las características y operatividad de los diferentes componentes empleados para resolver la propuesta planteada a la solución y utilizados para la elaboración del diseño y el desarrollo del proyecto.

2.1 Módulo XBee S2

Los módulos inalámbricos XBee S2 son descritos por Digi como módulos XBee con antena Whip, que permite la comunicación e interconexión de manera inalámbrica. Estos utilizan un protocolo llamado IEEE 802.15.4, que les permite crear redes; por lo tanto, en el presente proyecto se debe crear un despliegue mallado de Zigbee que estará conformada por tres de estos módulos configurados en cada una de sus diferentes maneras a) Router; b) Coordinador y c) Dispositivo final.

Las principales características de los módulos XBee S2 son las siguientes:

- 3.3V a 40mA
- Velocidad de datos máxima de 250 kbps
- Interfaz UART predeterminada: 9600, 8-N-1
- Salida de 2mW (+ 3dBm)
- Alcance de 400 pies (120 m)
- Antena incorporada
- Completamente certificado por la FCC

- 6 pines de entrada ADC de 10 bits
- 8 x pines IO digitales
- Cifrado de 128 bits
- Configuración local o aérea
- Conjunto de comandos AT o API. (DIGI, 2018, pp. 11-12)



Figura 2.1 Módulo XBee S2. [1]

2.2 Adaptador USB XBee

El dispositivo adaptador USB para Xbee es el encargado de parrear los módulos Xbee S2 con la aplicación de escritorio X-CTU donde se pueden observar y configurar los parámetros de estos módulos y así asegurar la comunicación entre todos los dispositivos que conforman la red.

A continuación, se describen las especificaciones clave de este adaptador:

- Requisitos de alimentación: 3.3 V paso a través del módulo XBee
- Comunicación: Paso de serie al módulo XBee
- Temperatura de funcionamiento: -40 a + 158 ° F (-40 a + 70 ° C)
- Dimensiones de PCB: 1.5 x 1.0 pulg. (3.81 mm x 2.54 cm). (Parallax, 2010, p. 1)



Figura 2.2 Adaptador USB XBee. [1]

2.3 Arduino UNO

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. De acuerdo a las consultas realizadas se conceptualiza a las placas Arduino como:

Aquellas que pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirla en una salida: activar un motor, encender un LED y publicar algo en línea. Puede decirle a su tarjeta qué debe hacer enviando un conjunto de instrucciones al microcontrolador en la tarjeta. Para hacerlo, utiliza el lenguaje de programación Arduino (basado en Wiring) y el software Arduino (IDE), basado en el procesamiento. (ARDUINO, 2018, p. 1)

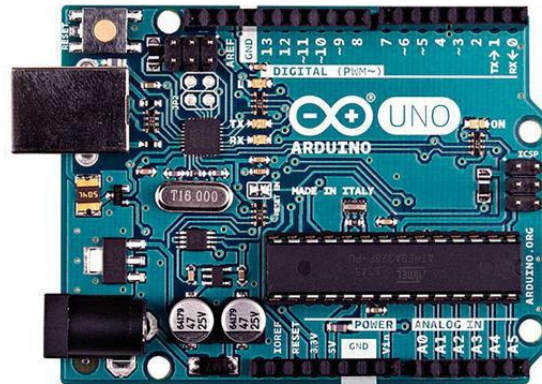


Figura 2.3 Arduino UNO. [1]

Para la realización de este proyecto utilizaremos las placas de Arduino UNO.

2.4 Arduino Wireless Proto Shield

Permite enlazar una placa de Arduino UNO para que se comunice con módulos inalámbricos de XBee, pero puede usar cualquier módulo con la misma huella, se utiliza para configurar una variedad de opciones en los módulos a través de códigos de Arduino y la librería especializada para Xbee que esta placa posee, además tiene un interruptor integrado que permite que el módulo inalámbrico se comunice con el convertidor de USB a serie o con el microcontrolador.

(ARDUINO Store, 2018, pp. 1-2)



Figura 2.4 Arduino Wireless Proto Shield. [1]

2.5 Raspberry Pi 3

La placa Raspberry PI 3 funciona como una computadora de bajo costo que cuenta con un procesador de cuatro núcleos de 64 bits que funciona a 1.4 GHz, doble banda de 2.4 GHz y LAN inalámbrica de 5 GHz, Bluetooth 4.2 / BLE, Ethernet más rápida y capacidad PoE a través de un PoE HAT separado, tiene una LAN inalámbrica de banda dual que viene con certificación de cumplimiento modular, permitiendo que en la placa se diseñen productos finales con una reducción significativa pruebas de conformidad de LAN inalámbrica, mejorando tanto el costo como el tiempo de comercialización. (RASPBERRY, n.d., pp. 1-2)

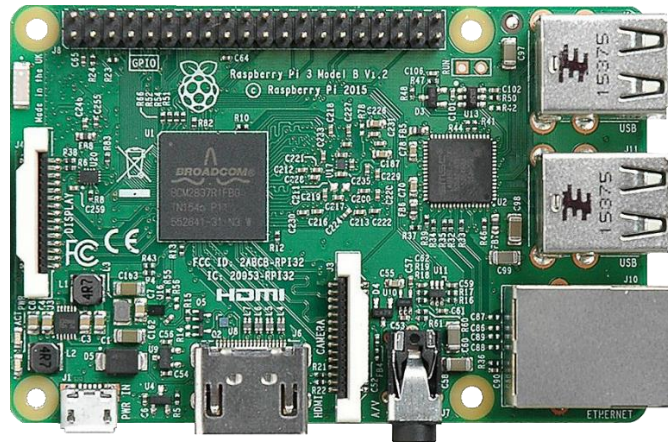


Figura 2.5 Raspberry Pi 3. [1]

2.6 Tarjeta Micro SD

Una tarjeta SD, o cualquiera de sus derivados, es una pequeña tarjeta recubierta de plástico donde se pueden almacenar fotos, música, vídeos y cualquier tipo de archivo o datos. Al igual que un disco duro o una memoria USB, podemos grabar y borrar los datos que en ella se encuentren tantas veces como queramos. A día de hoy las podemos encontrar en tres formatos distintos: SD, miniSD y microSD. La diferencia entre cada uno de estos formatos es el tamaño que ocupan y los dispositivos en los que se pueden utilizar. (Barroso, 2012, p. 2)

En nuestro proyecto utilizaremos una tarjeta Micro SD, la cual se encuentra interactuando directamente con la placa Raspberry pi3 descrita anteriormente.



Figura 2.6 Tarjeta Micro SD. [1]

2.7 IDE DE ARDUINO

El entorno de desarrollo integrado de Arduino, o el software Arduino (IDE), contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino y Genuino para cargar programas y comunicarse con ellos. (SM, 2017, p. 1)

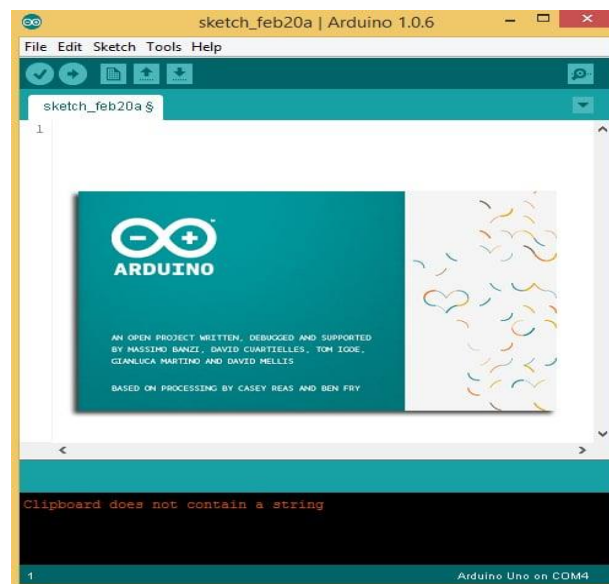


Figura 2.7 Entorno grafico del IDE de Arduino. [1]

2.8 SD CARD FORMATTER

De acuerdo a investigaciones realizadas se define al “SD Card Formatter es un software que proporciona un acceso rápido y sencillo a todos los formatos de tarjetas de memoria SD, SDHC y SDXC diseñado para que puedas deshacerte de una vez de todo el contenido almacenado en tu memoria SD” (SD Association, 2013, p. 2).

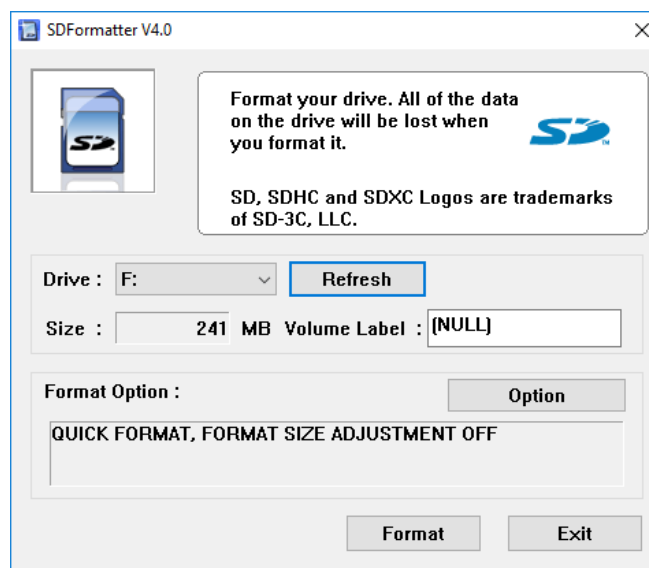


Figura 2.8 Vista general del entorno del Software SD Card Formatter. [1]

2.9 WIN32 DISK IMAGER

Estudios realizados acerca de este programa definen al Win32 Disk Imager “Es una sencilla aplicación de código abierto que graba imágenes de CD o DVD en tu memoria USB o en tu tarjeta SD, creando un lector de discos virtual” (Tobin, 2013, p. 1).

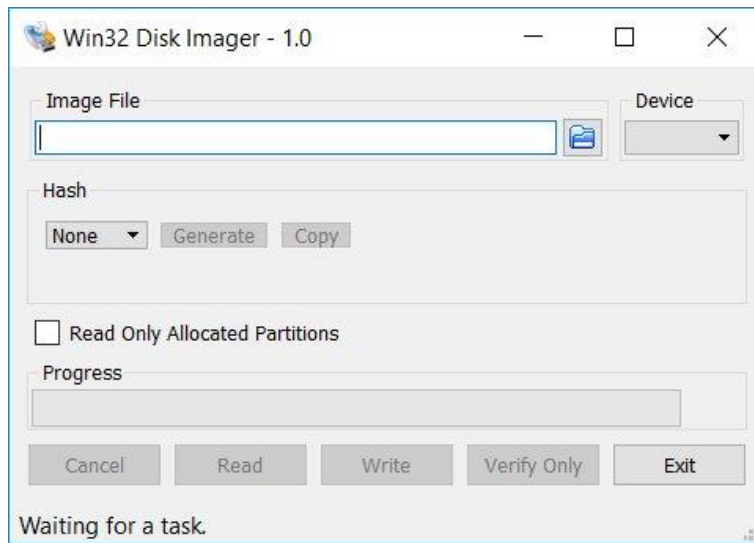


Figura 2.9 Vista general de Entorno de la herramienta Win32 Disk Imager. [1]

2.10 XCTU

XCTU es una aplicación multiplataforma gratuita diseñada para permitir a los desarrolladores interactuar con los módulos de RF de Digi a través de una interfaz gráfica fácil de usar. Incluye nuevas herramientas que facilitan la configuración, configuración y prueba de los módulos de RF XBee.

XCTU incluye todas las herramientas que un desarrollador necesita para comenzar a utilizar XBee rápidamente. Las características únicas como la vista de red gráfica, que representa gráficamente la red XBee junto con la potencia de la señal de cada conexión, y el generador de marcos de API XBee, que intuitivamente ayuda a crear e interpretar marcos API para XBees que se utilizan en el modo API, se combinan para hacer el desarrollo de XBee más fácil. (Digi Products, 2018, pp. 1-2)

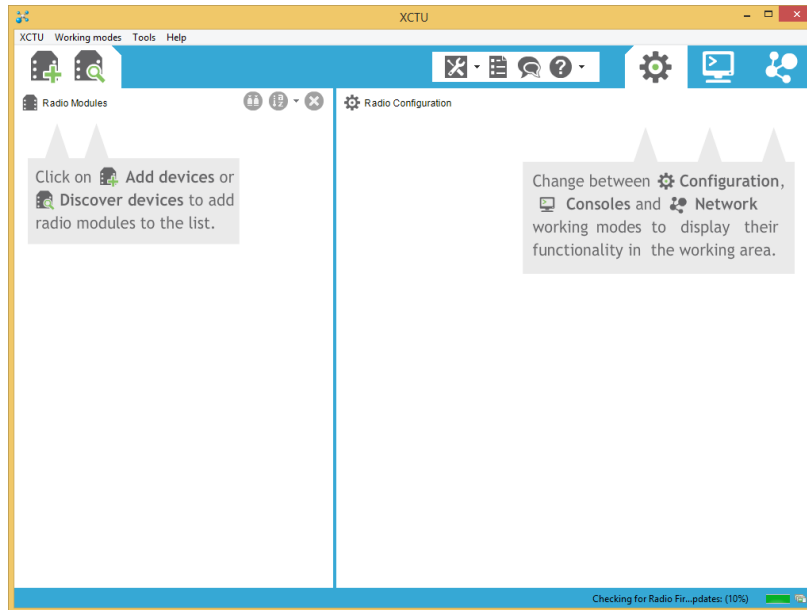


Figura 2.10 Entorno de la aplicación multiplataforma XCTU. [1]

2.11 RASPBIAN

Raspbian es el sistema operativo recomendado para Raspberry Pi, al estar optimizado para su hardware y se basa en una distribución de GNU/Linux llamada Debian.

Para instalar Raspbian en nuestra Raspberry Pi disponemos de dos versiones; una más completa con entorno gráfico y otra más reducida sin entorno gráfico:

Raspbian Pixel: Versión completa con entorno gráfico de Raspbian, es decir, la versión de escritorio con menús, ventanas, iconos, fondos de pantalla, etc. utilizado por la mayoría de los usuarios como ordenador de sobremesa.

Raspbian Lite: Versión reducida sin entorno gráfico, es decir, la versión en modo consola sin gráficos. Esta opción generalmente es para usuarios avanzados con

conocimientos de Linux que utilizan la Raspberry Pi como servidor. (Abellán, n.d., pp. 1-4)

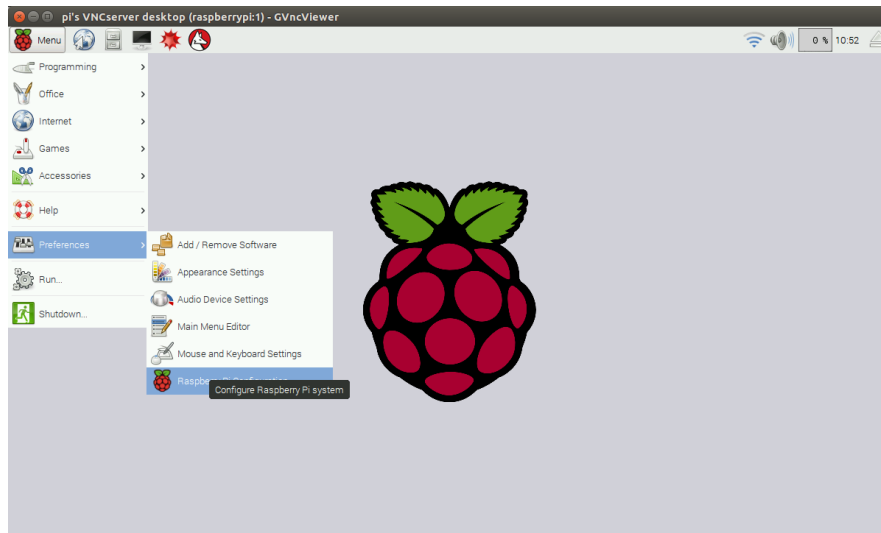


Figura 2.11 Entorno grafico de Raspbian que se muestra en Versión completa.[1]

2.12 PYTHON

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, <http://www.python.org/>, y puede distribuirse libremente. El mismo sitio

contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional.

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables. (van Rossum, 2009, p. 7)

2.13 Elementos utilizados

Se ha descrito en su mayoría las características y funcionalidad de los componentes, tanto de hardware como de software, a utilizar en el desarrollo e implementación del proyecto. Además, existen otros componentes electrónicos y accesorios de igual importancia, que se enlistan a continuación:

- a) Resistencias (1 K Ω)
- b) Botonera Push-Pull
- c) Cables Jumpers
- d) Cable USB Tipo B para Arduino
- e) Fuente cargador para Raspberry pi
- f) Monitor
- g) Teclado
- h) Mouse
- i) Protoboard
- j) LEDs (rojo, verde y amarillo)
- k) Baquelita
- l) Carcasa
- m) Soldador e hilo de estaño.

2.14 Costos del proyecto

Para la implementación del timbre inteligente que se utilizará para la puerta DOMO y de acceso restringido en el edificio CTI, se usaron los dispositivos electrónicos que proponen mejoras significativas al sistema actual de ingreso a estas áreas. A continuación, se detalla una tabla de costos referenciales de cada uno de los componentes utilizados para la elaboración de este diseño, así como el costo total del proyecto.

Tabla 2.14.1 Tabla de componentes utilizados con sus respectivos costos y costo total del proyecto. [1]

Descripción	Cantidad	P. Unitario	Valor
Módulo XBee S2	3	\$36,00	\$108,00
Arduino UNO	2	\$15,00	\$30,00
Tarjeta microSD (32G)	1	\$16,00	\$16,00
Resistencias	2	\$0,10	\$0,20
LED	3	\$0,30	\$0,30
Cables Jumpers	1	\$3,00	\$3,00
Raspberry pi 3	1	\$57,00	\$57,00
Carcasa	1	\$6,00	\$6,00
Adaptador USB XBee	1	\$10,00	\$10,00
Arduino Proto Wireless Shield	2	\$9,00	\$18,00
Botonera Push-Pull	2	\$0,15	\$0,30
Total			\$247,80

CAPÍTULO 3

3. DESCRIPCIÓN DE LA SOLUCIÓN

3.1 Descripción general

Este proyecto permite, mediante el envío de notificaciones al celular y usando una aplicación móvil, mejorar la atención a visitantes externos e internos a las áreas restringidas del edificio inteligente CTI-Espol.

La propuesta se desarrolla en 2 etapas: En la primera el dispositivo del timbre es accionado por un usuario (visitante externo o interno del CTI), el cual podrá visualizar por medio de leds el estado del requerimiento, mientras, en la segunda etapa llega la notificación al celular mediante el aplicativo móvil, esto ocurre a través de la difusión de mensajes tipo broadcast, empleando el protocolo de comunicación MQTT (bróker). Cabe señalar, que el mensaje generado llega a todos usuarios suscritos, para que uno de ellos dé el acceso, simultáneamente desaparece el mensaje del aplicativo. Además, existe un súper usuario con permiso para abrir la puerta desde la app.

3.1.1 Recursos para la implementación del timbre inteligente

Para la implementación del hardware se utilizaron placas de Arduino UNO, Raspberry pi, Wireless Proto Shield, PCB diseñada y módulos XBee S2, a las cuales se asignaron diferentes pines establecidos previamente de acuerdo con la función específica que requería cumplir cada uno de estos.

En la parte del software se detallaron todas las configuraciones requeridas, así como los comandos específicos empleados para dar la funcionalidad necesaria, por último, cabe recalcar que el aplicativo móvil ha sido desarrollado para Android.

3.2 Método de la solución

Para solucionar el problema se trabajó con el método Design Thinking, que parte del análisis general del problema, lluvia de ideas, entrevistas a los actores, entre otras actividades.

El método permitió identificar diferentes opciones de solución que se evaluaron con el fin de mejorar el sistema actual del timbre, estas constan en la Tabla 1.3.1.

Una vez valoradas las alternativas planteadas se decidió aplicar la tercera opción que consiste en implementar una red Mesh conectada a un servidor MQTT broker, la idea seleccionada proporciona:

- a) Respuesta rápida del sistema
- b) Dispositivo amigable
- c) Reconfiguración automática

3.2.1 Configuración de Módulos XBee S2

Los módulos Xbee S2 utilizados para crear la malla Zigbee, que se encarga de la comunicación bidireccional, la cual permite la recepción y transmisión de la señal entre el módulo Xbee configurado como dispositivo final (End device) y el módulo Xbee configurado como coordinador (Coordinator), se

realiza a través de la aplicación de escritorio X-CTU, además la malla cuenta con un módulo Xbee que funciona como router.

En base a la referencia del manual de usuario del módulo XBee S2 publicado por Digi en su página oficial, se configuró el módulo asignado como End device en modo AT, y los dispositivos dispuestos como Coordinator y Router se configuraron en modo API respectivamente.

A continuación, se detallan en la tabla 3.2.1 los parámetros establecidos, modo de operación, función, etc. con los que se configuraron los módulos, así como la versión de firmware que se estableció en cada uno de ellos y otros parámetros de fábrica que se encuentran establecidos por defecto como la MAC Address.

Tabla 3.2.1 Parámetros configurados en los módulos XBee S2 empleados en la implementación de la topología de red Mesh. [1]

Módulo	MAC Address	Net Addr	PAN ID	Función	Modo	Versión firmware
XBee1	0013A20040E91644	0000	C001BEE	Coordinator	API	21A7
XBee2	0013A200405CE758	AD07	C001BEE	Router	API	23A7
XBee3	0013A200403CC13B	93F5	C001BEE	End device	AT	28A7

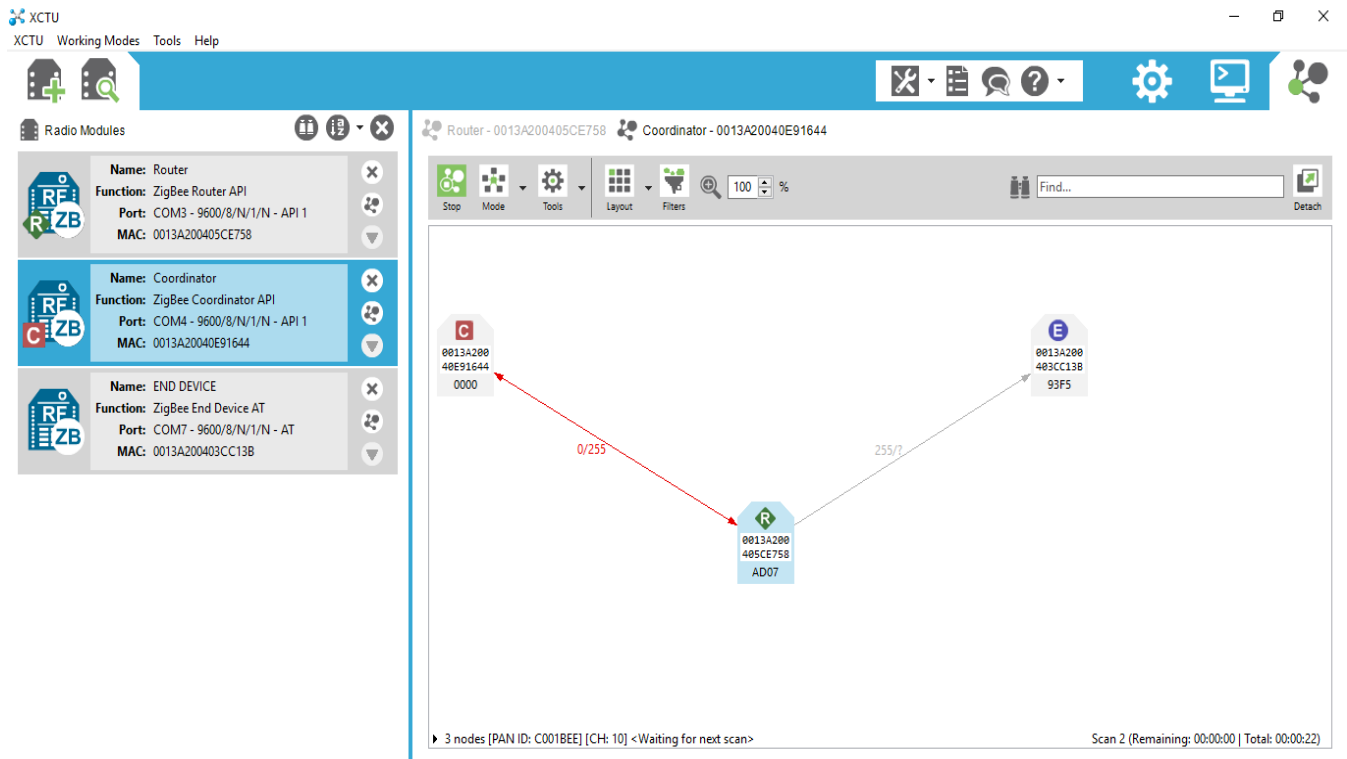


Figura 3.2.1 Topología Mesh basado en protocolo Zigbee que se visualiza en la aplicación X-CTU.[1]

3.2.2 Configuración de Arduino UNO

El Arduino UNO es el encargado de transmitir un carácter, esto se lo verificó a través de consola usando el puerto serial para extraer el carácter 'T', por lo tanto en cada ocasión que se presione el botón del timbre inteligente se realizará el envío del carácter, el mismo que activa los ciclos informativos que utilizan 3 leds: amarillo notifica el envío del mensaje, verde interrumpe la acción del amarillo y se enciende cuando el mensaje ha sido aceptado, y finalmente el rojo que alerta que el mensaje no fue aceptado o fue ignorado por algún suscriptor siendo necesario presionar nuevamente para empezar el ciclo que dura aproximadamente 1 min.

Así mismo debido a que el módulo XBee está montado sobre el Arduino, el envío se efectuará de manera inalámbrica; para esto la placa de Arduino UNO se configuró previamente mediante el código respectivo, que permitió la acción mencionada (Anexo B). Por otra parte, la conexión de la placa con el módulo XBee S2 dispuesto como End device, se realizó a través de una placa adaptadora Proto Wireless Shield específica para este montaje.

Por otro lado, se realizó un pequeño circuito constituido por una botonera, una resistencia y un switch. El primer elemento accionará el timbre cada vez que sea presionado; paralelamente la resistencia de 10k[Ohms] con 5% de tolerancia, permite el paso de corriente sistema. El Switch, que activa o desactiva un Reset configurado mediante la conexión de los pines RST(reset) con GND(tierra), que permite que el Arduino UNO, ya configurado, otorgue la capacidad de ingresar mediante XCTU al módulo XBee S2.

La tabla 3.2.2 detalla los pines utilizados para la conexión entre la placa de Arduino UNO, el adaptador Wireless Proto Shield, la botonera, la resistencia y los leds.

Tabla 3.2.2 Conexión de pines entre Arduino UNO y Proto Wireless Shield. [1]

Arduino UNO Pines	Descripción
2	Pin analógico utilizado para leer el estado de la botonera
+5V	Alimentación de Arduino UNO
GND	Referencia a tierra
RST	Pin de reset

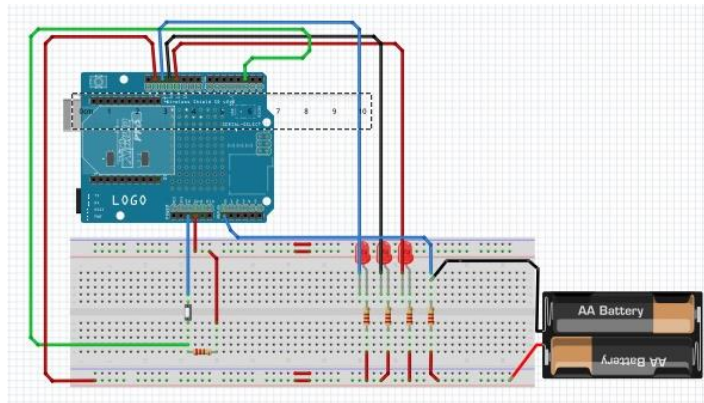


Figura 3.2.2 Grafico ilustrativo del montaje del módulo XBee (End Device) con Arduino UNO, en la fase de implementación en protoboard. [1]

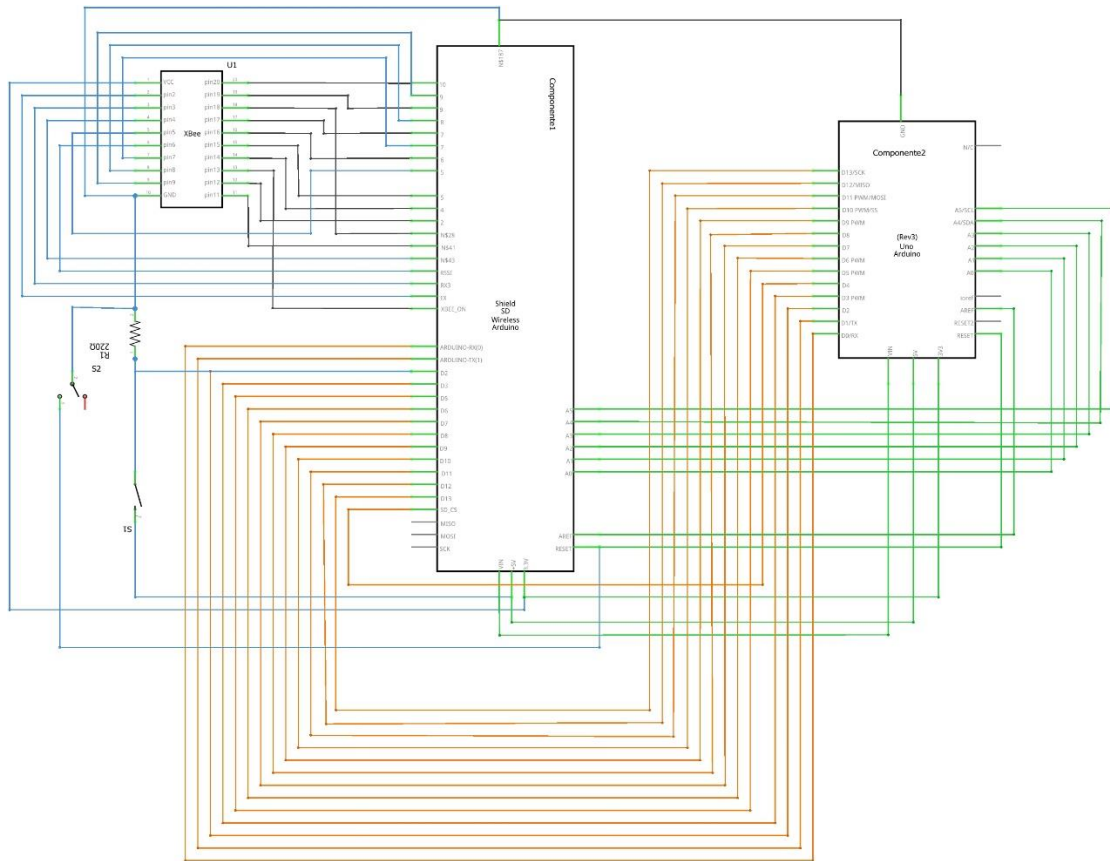


Figura 3.2.3 Esquemático del circuito para el timbre inteligente.[1]

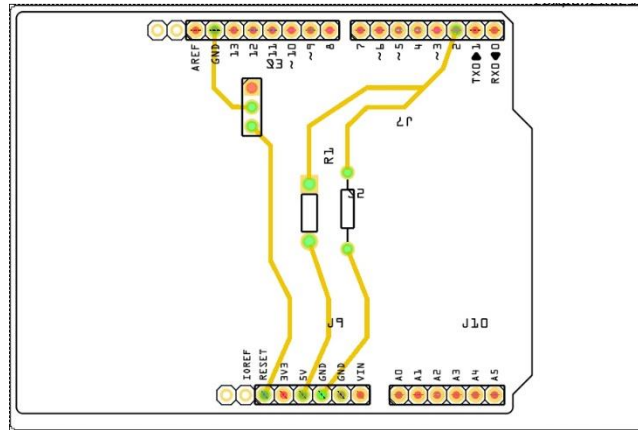


Figura 3.2.4 PCB del circuito para el timbre inteligente.[1]

3.3 Configuración de la placa Raspberry

Se instaló en la placa una distribución del sistema operativo GNU/Linux llamada Raspbian que permite alojar un servidor MQTT, el cual brinda la capacidad de comunicarse entre el dispositivo y los usuarios que se encuentren enlazados al servidor mediante la aplicación móvil.

Esta placa contiene a su vez los códigos en Python que permiten establecer la comunicación mediante el puerto serial que comunica la placa de Raspberry con el módulo XBee S2 configurado como coordinador (Anexo C), para realizar estas configuraciones se utilizaron los comandos que se describen a continuación:

- **Instalación de librería MQTT para usar con Python 3.5**

```
pip3 install paho-mqtt
```

- **Instalación de Serial**

```
python -m pip install pyserial
```

Esta librería solo funciona en el modo API

- **Instalación de librería XBee para usar con Python 3.5**

pip3 install digi-xtbee

La librería solo funciona con el modo API.

El tres especifica que siempre se va a correr en Python 3.

- **Línea de comando para enlistar los dispositivos que estén conectados en los puertos seriales**

dmesg | grep tty

3.4 APLICACIÓN MÓVIL PARA TIMBRE INTELIGENTE

Para la implementación de la aplicación se usó el Framework IONIC, debido a que es un SDK completo, de código abierto el cual se utiliza para el desarrollo de aplicaciones android.

3.4.1. Instalación de IONIC

El funcionamiento adecuado de IONIC depende de una correcta instalación de: AngularJs, Apache Cordova, NodeJS y CLI.

Estas dependencias tienen el siguiente orden de instalación:

- **NodeJS:** Se instaló de acuerdo a la documentación del sitio oficial, a partir de este momento se usaron los manejadores de paquetes llamados npm los cuales facilitan la instalación de dependencias.
- **CLI:** Sirvió para instalar Apache Cordova lo cual debemos hacerlo a través del manejador de paquetes npm.

Una vez instalado todo lo anterior se procedió a instalar IONIC 4.5.2 por medio de npm, llegando a este punto, para poder usar Ionic se necesitó instalar tanto java como Android Studio, este último para el uso del android sdk.

3.5 CONFIGURACIÓN MQTT BRÓKER

Para la configuración del MQTT bróker se tuvo que habilitar la comunicación del web sockets, con el puerto 9001, ya que este nos permite enviar los mensajes a cualquier tópico para que este sea usado en la aplicación móvil y, además, el puerto 1884 para los servicios de MQTT, Anexo D.

Para realizar las configuraciones antes mencionadas, se utilizaron los comandos que se describen a continuación:

- **Editar la configuración del documento mosquitto.conf**
sudo nano /etc/mosquitto/mosquitto.conf
- **Agregamos en mosquitto.conf**
listener 1884
listener 9001
protocol websockets

Cabe recalcar que, sí no se usa web Sockets la aplicación no podrá usar MQTT.

CAPÍTULO 4

4. PRUEBAS Y RESULTADOS

En el presente capítulo se comprobó mediante pruebas de rendimiento, Anexo H, la conectividad de los dispositivos, servicios y la aplicación; comenzamos probando el sistema planteado tres veces, con el fin de solucionar pequeños defectos que sean encontrados en el mismo, los cuales se detallan en los ítems 4.1, 4.2 y 4.3 con la solución a dichos fallos respectivamente.

Luego se realizó la prueba de tiempo de respuesta del sistema, necesaria para comparar nuestro sistema con el sistema que implementa WIFI, los resultados de dicha prueba están descritos en la sección 4.4 de este capítulo. A continuación, en el ítem 4.5, se detalla la prueba referente a la estabilidad de comunicación del dispositivo entre sus nodos XBee a diferentes distancias, las cuales fueron tomadas desde 1 metro de distancia hasta los 15 metros, realizando para ello un muestreo considerable y calculando el valor promedio de dichos tiempos de respuesta del sistema para cada escenario descrito anteriormente.

Por último, se realizó una prueba referente al estudio energético de nuestro sistema detallado en el apartado 4.5, donde se exponen las condiciones iniciales del sistema y los resultados obtenidos en esta última prueba.

4.1 Prueba de configuración en Arduino UNO y el Aplicativo móvil

Después de haber realizado las pruebas con el Arduino UNO, se tomó la decisión de mejorar la precisión del tiempo de ejecución del código en el dispositivo mencionado, para esto se utilizó la función `millis()`, dado que la acción de envío y recepción de mensajes después de tocar la botonera era muy imprecisa.

Otro problema que se encontró fue el de configurar un tiempo de espera, para que una vez accionado el timbre este no pueda volverse a accionar hasta que se

termine dicho tiempo o en su debido caso se conteste con un mensaje de confirmación por parte de alguno de los usuarios suscritos que recibieron la notificación del sistema.

Por otra parte, en la aplicación se consideró notificar a los demás usuarios suscritos a la puerta cuando se aceptó la notificación para ir a abrir la puerta, permitiendo de esta manera a los demás usuarios verificar que no sea un usuario externo a el laboratorio el que este interfiriendo, caso contrario el mensaje desaparece y la aplicación vuelve a esperar otra notificación de que existe otro usuario en la puerta.

4.2 Prueba de comunicación.

Después de efectuarse esta prueba se logró presenciar un error en el canal de comunicación por parte de la Raspberry pi3 ya que este dispositivo solo recibía mensajes por medio de un canal, mas no obstante al tratar de difundir la confirmación por parte de los usuarios suscritos al tópico, el puerto se bloqueaba debido a que se programó mal la inicialización del dispositivo.

Por esto se elimina de las funciones en donde se inicializaba en cada momento el dispositivo XBee por una inicialización global al comienzo del programa, gracias a esto se solucionó el problema del bloqueo del puerto permitiendo usar un único recurso para el envío y recepción de la información.

4.3 Prueba transmisión de datos.

Cuando se realizaron numerosas pruebas de manera periódica, se requirió mejorar la tasa de envío, por ende, se consultó en el manual de XBee como hacerlo, para ello se modificó la tasa de transmisión de datos de 9600 bits/s a 115200 bits/s, dando como resultado mayor eficiencia en la comunicación.

4.4 Pruebas de rendimiento

Para las pruebas de rendimiento, un factor clave es la confiabilidad de envío de mensajes del sistema, esta prueba relaciona el tiempo de encendido del XBee(end

device), ya que este tiene un tiempo de demora corto en el encendido cuando este ha estado hibernando, para posteriormente recibir una confirmación por parte de los usuarios suscritos al tópico en cuestión. Para solucionar este problema, se decidió realizar varios envíos en un determinado tiempo junto a la instrucción de encender el led, esto se consigue gracias a la implementación de millis() mencionada anteriormente en el ítem 4.1.

Consecuentemente se muestra las tablas con los datos obtenidas mediante las pruebas realizadas a ambas tecnologías, necesarias para la realización de comparativas:

Tabla 4.4.1 Tiempo promedio de encendido de ambas tecnologías. [1]

Prueba	Wifi [s]	XBee [s]
1	9,53	0,85
2	7,81	1
3	7,99	0,53
4	8,93	1,44
5	7,92	0,59
6	18,87	1,31
7	10,06	0,79
8	14,5	0,66
9	11,76	1,11
10	9,81	0,85
11	8,03	1,32
12	12,69	1,15
13	14,98	0,85
14	15,24	1,7
15	12,62	1,1
16	9,09	1,32
17	11,04	2,1
18	13,48	1,11
19	8,03	1
20	9,69	1,25

21	7,95	1,17
22	13,95	1,43
23	14,52	0,85
24	9,88	1,12
25	13,78	1,32
26	12,8	1,15
27	10,7	1,05
28	9,29	1,27
29	11,78	1,31
30	9,97	1,26
Promedio	11,223	1,132

Para realizar el análisis comparativo de estos datos, se debe de tomar en consideración que el tiempo de inicialización del Foton Particule es mucho mayor que el de los módulos XBee, además de que el Foton Particule presenta ciertos problemas al reconectarse a una red Wifi. De la tabla 4.4.1. podemos concluir que el sistema que implementa tecnología ZigBee es mucho más eficiente que el que implementa WIFI, llevando una ventaja de 10.09 [s] en tiempo promedio de respuesta del sistema.

A continuación, en el siguiente análisis comparativo de acuerdo a la tabla 4.4.2, los datos fueron tomados bajo las condiciones que se describen seguidamente: el Fotón ya se encuentra despierto y conectado a la red WIFI no congestionada, garantizando así su funcionamiento óptimo.

Tabla 4.4.2 Tiempo promedio de respuesta de ambas tecnologías. [1]

Prueba	XBee [s]	Wifi[s]
1	0,85	0,92
2	1	1,56
3	0,53	0,92

4	1,44	1,11
5	0,59	0,72
6	1,31	0,66
7	0,79	0,79
8	0,66	0,46
9	1,11	0,79
10	0,85	0,66
11	1,32	0,99
12	1,15	0,92
13	0,85	1,11
14	1,7	0,79
15	1,1	1,24
16	1,32	1,09
17	2,1	0,66
18	1,11	0,65
19	1	1,05
20	1,25	0,91
21	1,17	0,59
22	1,43	1,05
23	0,85	0,98
24	1,12	1,24
25	1,32	0,85
26	1,15	0,85
27	1,05	1,83
28	1,27	1,05
29	1,31	1,05
30	1,26	0,59
Promedio	1,132	0,936

De la tabla 4.4.2 podemos concluir que el tiempo de respuesta promedio del sistema implementado con tecnología WIFI aparenta una pequeña ventaja en el tiempo de respuesta vs. Zigbee, no obstante debemos recordar que estas últimas mediciones fueron tomadas en condiciones ideales para el sistema implementado con WIFI lo que evito que este sistema tuviera que reconectarse a la red como es su funcionamiento habitual por lo que se mantiene firme el criterio de que el sistema que implementa Zigbee es más estable que este.

4.5 Estabilidad de comunicación del dispositivo

Para establecer el rango de confianza de la comunicación entre nodos se realizaron pruebas del timbre a distancias diferentes, en donde los valores demostraron un comportamiento lineal entre el tiempo de respuesta del sistema y su distancia entre nodos. En las primeras pruebas efectuadas a el sistema, se comprobó que si el XBee muestra algún problema este presenta conflictos al enviar la señal de activación, por ende, se procedió a realizar pruebas a diferentes distancias con la finalidad de verificar el estado del XBee (end device), cuando se ha establecido la comunicación en todo el sistema.

A continuación, se muestra la tabla de los tiempos de respuesta promedio tomados a cada distancia las cuales van de 1 a 15 m respectivamente:

Tabla 4.5.1 Tiempo promedio de respuesta entre nodos a diferentes distancias.[1]

Distancia [m]	Tiempo de Respuesta [s]
1	0,90
2	0,98
3	1,15
4	1,10
5	1,05
6	1,17
7	1,14
8	1,15
9	1,33
10	1,29
11	1,38
12	1,36
13	1,36
14	1,56
15	1,50

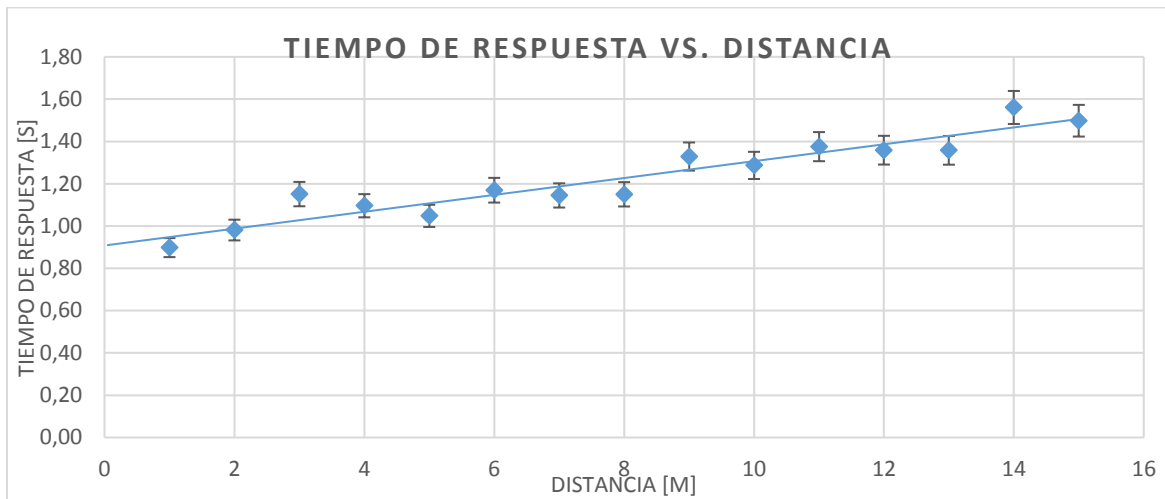


Figura 4.5.1 Gráfico comparativo de tiempo promedio de respuesta vs. Distancia entre nodos [1]

Se realizó el gráfico comparativo de tiempo promedio de respuesta vs. Distancia entre nodos XBee de la figura 4.5.1 de acuerdo a la tabla 4.5.1, en esta tabla podemos apreciar claramente el comportamiento del sistema a diferentes distancias, el error estándar calculado en este gráfico para validar la veracidad del promedio de tiempo de respuesta fue calculado en un intervalo de confianza al 95%, lo cual nos da la certeza de que este valor de tiempo en base a las mediciones de prueba y error tiene alto grado de confiabilidad, además podemos notar que mientras más alejados se encuentran los nodos el sistema posee un tiempo de respuesta más alto. Para terminar, debemos resaltar que estas mediciones fueron tomadas en una distancia a línea recta.

4.6 Estudio Energético del Prototipo

El estudio energético se realizó por medio de una pila de 9 [V] (nominal) y 200m [A], para el cual se realizaron mediciones antes y después de las pruebas. Después de cada prueba se logró ver que el dispositivo presenta un gasto energético del 3% mínimo y 6% máximo, dado que las pruebas fueron hechas durante una hora sin pausa. Según los valores obtenidos, el consumo del Arduino UNO es de unos 46m [A] en reposo, por otra parte, el XBee S2 es de bajo consumo, el cual es menor a 50m [A], cuando está en funcionamiento y menor a 10u [A] cuando está en modo Sleep.

Para el presente caso, se sabe que el Arduino + XBee consumen en promedio unos 96m [A], en estado de funcionamiento y dado que el sistema se encuentra funcionando por medio de una pila con apenas 200m [A], lo significa que el sistema tendría 2.083 horas de uso autónomo aproximadamente tal como se muestra en el cálculo.

$$\frac{200\text{mA}}{96\text{mA}} = 2.083$$

Cabe recalcar que el sistema puede ser mejorado con sistemas de ahorro energético tales como el Arduino Micro, el cual consta con librerías dedicadas a esto tales como LowPower.h.

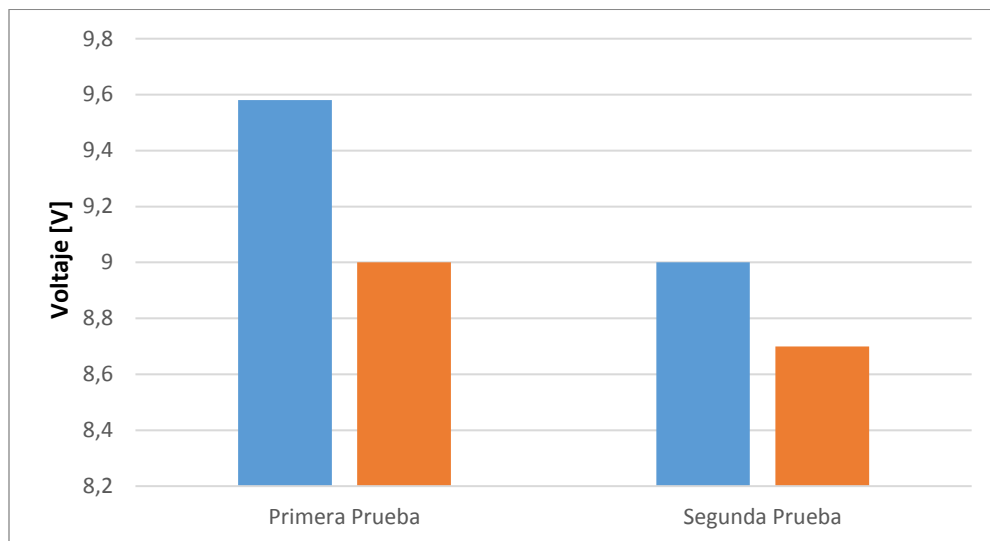


Figura 4.6.1 Grafico de nivel de voltaje. [1]

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

La implementación de tecnología IoT ayuda a promover nuevas soluciones y mejorar antiguas que favorezcan a todos aquellos que se encaminan en este mundo.

Cuando las tecnologías IoT son complementadas correctamente con herramientas de desarrollo web para móviles o computadoras presentan un valor muy grande de uso debido a su amplio campo de contenido y facilidades, en nuestro caso, las herramientas usadas para prototipado están presentes en la comunicación e interacción con un usuario en tiempo real por medio de un dispositivo móvil, de esta manera se logra demostrar que no es necesario un alto costo de materiales para la investigación y desarrollo de proyectos que puedan tener un alcance del tipo comercial.

Se logró concluir que las herramientas usadas en este sistema presenten un buen alcance en la distancia de comunicación en comparación con el modelo desarrollado anteriormente, además de ser más estable que las del sistema anterior.

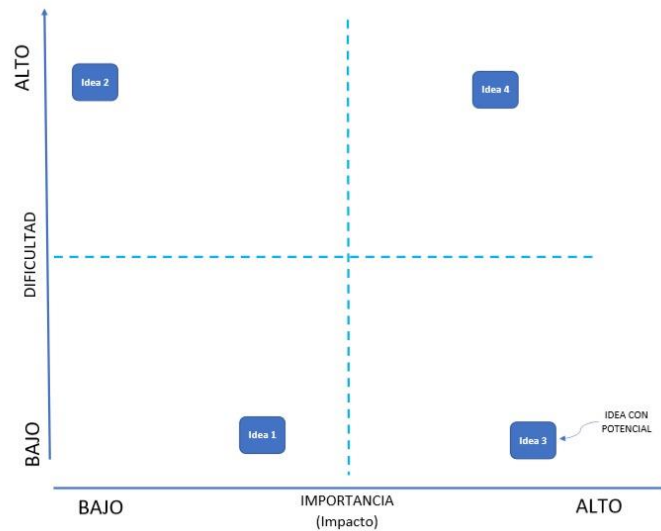
RECOMENDACIONES

Para poder recrear e implementar nuevos proyectos de prototipado en dispositivos IoT es recomendable leer los manuales que las compañías brindan, por cuestiones de limitaciones que muchas veces tienen presentes estos dispositivos evitando de esta manera posibles complicaciones en el desarrollo de estos sistemas, además de ayudar a comprobar que el dispositivo está funcionando correctamente según lo indicado.

Para posibles mejoras del proyecto se sugiere realizar un estudio para el ahorro del consumo energético y manejo de datos de una cámara de baja resolución (VGA) para mejorar la parte de seguridad que brinda el dispositivo de IoT.

ANEXOS

Anexo A. Matriz impacto dificultad.



Anexo B. Código del emisor

```
//-----  
//Declara puertos de entradas y salidas y variables  
//-----  
int boton = 2; //Pin donde se encuentra el pulsador, entrada  
  
//-----  
//Funcion principal  
//-----  
void setup() // Se ejecuta cada vez que el Arduino se inicia  
{  
  Serial.begin(9600); //Inicia comunicación serial  
  pinMode(boton,INPUT); //Configura el pulsador como una entrada  
}  
|  
//-----  
//Funcion ciclicla  
//-----  
void loop() // Esta funcion se mantiene ejecutando  
{ // cuando este energizado el Arduino  
  
  //Pregunta si el pulsador esta oprimido  
  if(digitalRead(boton)==HIGH){  
    Serial.print('D'); //Imprime vía serial la letra 'D'  
    delay(10); //Retardo para no congestionar la escritura serial  
  }  
  
}
```

Anexo C. Código de la Raspberry

```
from digi.xbee.devices import XBeeDevice #Libreria xbee
import paho.mqtt.client as paho #Libreria MQTT

#Direccion ip broker
broker="192.168.1.17"

#Puerto
port=1884

#Puerto serial de XBEE
PORT = "/dev/ttyUSB0"

BAUD_RATE = 9600

def main():
    print(" +-----+")
    print(" | XBee Python Library Receive Data Sample |")
    print(" +-----+\n")

    #Instanciacion de dispositivo xbee
    device = XBeeDevice(PORT, BAUD_RATE)
    try:
        device.open()

        #Metodo que publica a un topic de mqtt
        def data_receive_callback(xbee_message):
            msg = xbee_message.data.decode()
            if "D" in msg:
                client1 = paho.Client("control1")

                #Conexion de broker
                client1.connect(broker,port)

                #Publicacion del mensaje
                ret = client1.publish("test","Hola Mundo")

                device.add_data_received_callback(data_receive_callback)

                print("Waiting for data...\n")
                input()

    finally:
        if device is not None and device.is_open():
            device.close()

if __name__ == '__main__':
    main()
```

Anexo D. Configuración MQTT bróker

```
pi@raspberrypi:~ $ sudo nano /etc/mosquitto/mosquitto.conf

# Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

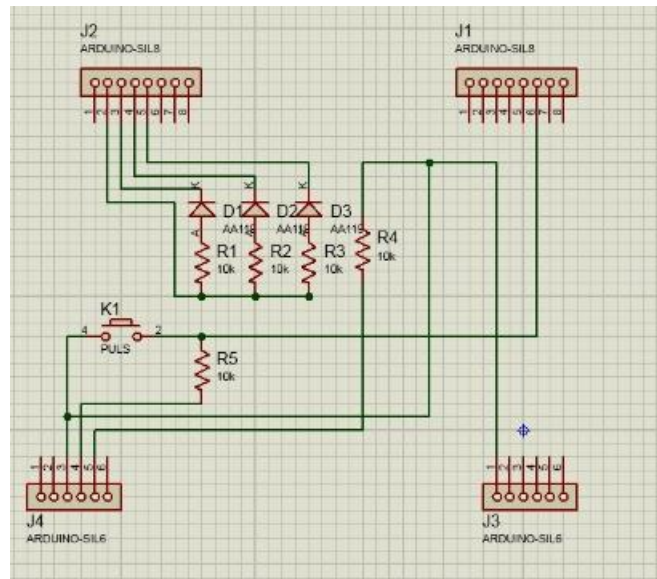
#pid_file /var/run/mosquitto.pid

#persistence true
#persistence_location /var/lib/mosquitto/

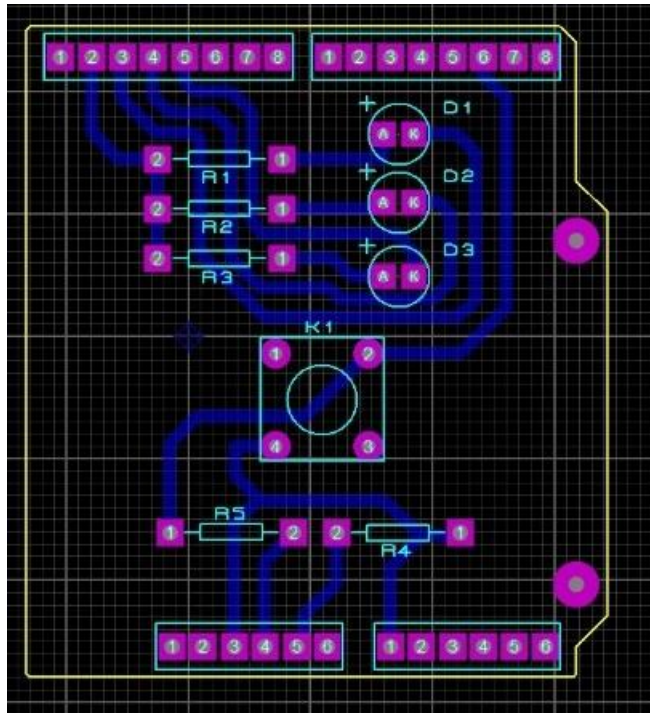
#log_dest file /var/log/mosquitto/mosquitto.log
listener 1884
#protocol mqtt

listener 9001
protocol websockets
```

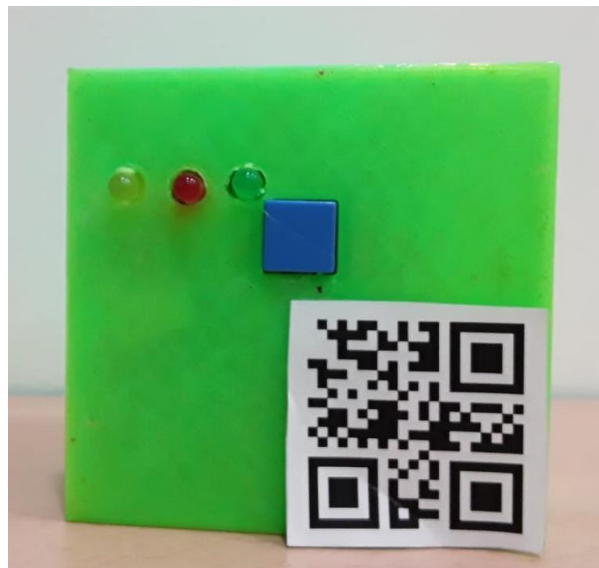
Anexo E. Esquemático del circuito implementado



Anexo F. Esquemático del PCB



Anexo G. Impresión 3D





FIEEC

Facultad de Ingeniería en Electricidad y Computación

Nombre: *Fernando David Eras Delgado*

Fecha: *14/02/2019*

Marca la columna que mejor refleje tu opinión acerca del proyecto que analizas entre el 4 (muy evidente) y el 1 (poco evidente). El valor (se utiliza cuando el criterio no se puede aplicar o no es evidente. Puedes añadir comentarios al final de cada bloque.

I. Preparación del proyecto

	Muy evidente	Bastante evidente	Evidente	Poco evidente	No evidente
1. Los objetivos de aprendizaje fueron claramente definidos	X				
2. Los objetivos y contenido del proyecto encajan con lo deseado.		X			
3. El proyecto es el resultado de la integración de objetivos, contenidos y criterios de evaluación de diferentes materias o áreas de conocimiento		X			
4. Se observa una clara relación entre las actividades a desarrollar en el proyecto y el desarrollo de las competencias básicas de los estudiantes		X			
5. Se han establecidos indicadores de éxito del proyecto	X				

Comentarios:

II. Análisis del proyecto

6. Se Cumplió con los objetivos que se persiguen con el desarrollo del proyecto	X				
7. Se siguió todos los pasos a seguir y secuencia establecida para el desarrollo del proyecto		X			
8. Los conocimientos demostrados por los estudiantes fueron los deseados		X			
9. La revisión completa de los conocimientos fundamentales en el desarrollo del proyecto dieron una respuesta acertada		X			
10. Considera que se puede mejorar el proyecto			X		

Comentarios:

Fernando Eras
Firma

leaded94@hotmail.com
Correo



FIEC

Facultad de Ingeniería en Electricidad y Computación

Nombre: Leonardo David Eras Delgado

Fecha: 14/02/2019

1. ¿Considera que el proyecto cumplió con lo planteado?

Si X No _____

2. ¿Considera que se debe mejorar algo del proyecto?

Si X No _____

En caso de marcar el sí, Explique ¿Por qué?:

Necesita revisión en consumo de energía.

3. ¿Considera que el diseño del proyecto es el adecuado?

Si X No _____

¿Por qué?

Usan diversos equipos y protocolos IoT para efectuar la tarea

4. ¿Considera que el proyecto es de gran utilidad para el desarrollo de la tecnología IoT?

Si X No _____

5. ¿Si el proyecto estuviera disponible para uso del hogar lo recomendará?

Si X No _____

6. ¿Cree que tiene un gran impacto tecnológico y de innovación?

Si X No _____

¿Por qué?

Permite comparar y contrastar con otras tecnologías IoT

Leonardo Eras
Firma

leoded94@hotmail.com
Correo

PLAN DE PRUEBAS

VERSIÓN [1.0]

HOJA DE RESUMEN DE MODIFICACIONES

Version	Fecha	Punto	Cambios respecto a la version anterior	Preparado por	Aprobado por
[1.0]	18/01/2019		Versión Inicial	Carvajal Cabay Mora Perez	

INTRODUCCIÓN

Proyecto	Tipo de Proyecto
Despliegue de redes malladas ZigBee para aplicaciones de edificios inteligentes	Proyecto tecnológico IoT
Documentos Evaluacion relacionados	

Documento de titulacion	
Equipo de Proyecto	Carvajal Cabay, Mora Perez
Tutor	Federico Domínguez, PhD.

Objetivo

El presente documento tiene como finalidad entregar los pasos a seguir para la correcta aplicación de pruebas necesarias en el sistema del timbre inteligente, con el fin de verificar las funciones y los procesos en el módulo de software como en el de hardware, para de esta manera encontrar los posibles errores o fallos que se presenten durante el periodo de pruebas. Además de validar si el sistema cumple con los requerimientos que se contemplan en las especificaciones del funcionamiento total del mismo.

A continuación, se detallan las fases de pruebas que se realizaran:

- 1. Planificación de pruebas:** Identificar los recursos necesarios para realizar las pruebas. Desarrollar la estrategia de pruebas. Generar plan de pruebas
- 2. Diseño de pruebas:** Desarrollo de las pruebas. Identificar y describir los casos de prueba.
- 3. Implementación de pruebas:** Establecer el entorno de prueba. Desarrollar las clases de prueba, los componentes y datos de prueba.
- 4. Ejecución de pruebas:** Ejecutar los casos de prueba. Evaluar la ejecución del proceso de prueba. Verificar los resultados. Investigar los resultados no esperados.
- 5. Evaluación de las pruebas:** Evaluar la cobertura de los casos de prueba. Analizar defectos. Determinar si se han alcanzado los criterios de las pruebas. Crear informes de evaluación de pruebas.

ALCANCE DE LAS PRUEBRAS

En esta sección se presentan los módulos del sistema, cada uno con sus requerimientos definidos para ser llevados a cabo con éxito.

Componentes a ser probados:	Sistema: <ul style="list-style-type: none">✓ Conectividad✓ Estabilidad✓ Escalabilidad Software: <ul style="list-style-type: none">✓ Tiempo de respuesta (notificaciones)✓ Soporte plataforma (Android)
Objetivos de las pruebas:	Se realizaran pruebas para validar: <ul style="list-style-type: none">✓ Verificar conectividad de los dispositivos , servicios y la aplicación.✓ Facil usabilidad para el usuario✓ Estimar rendimiento del sistema✓ Facilitar informacion detallada (manual de usuario)
Detalle del orden de ejecucion de los componentes:	Se deben ejecutar en orden y de manera independiente:

	<ul style="list-style-type: none"> ✓ Configurar modulos XBee ✓ Crear la red Mesh ✓ Establecer Servicio MQTT ✓ Ejecutar codigos python en la Raspberry ✓ Establecer conexión ✓ Probar conexión
Responsabilidad de la Prueba:	Las pruebas son responsabilidad del equipo de proyecto en conjunto con el tutor encargado asignado por el CTI, centro de tecnologías e investigacion.

ENTORNO Y CONFIGURACION DE LAS PRUEBAS

Para el proceso de pruebas del proyecto se requiere la disponibilidad de los siguientes entornos:

- a. **Equipos Cliente:** Equipo de prueba como módulos XBee, placa Arduino UNO, Raspberry pi, módulos adaptadores.
- b. **Equipos laboratorio:** Equipos de medición como multímetro, cronometro.
- c. **Dispositivos móviles usuarios:** Celulares con Android.

REGISTRO DE LOS RESULTADOS DE LAS PRUEBAS

Objetivos de la prueba	Prueba #1. Comprobar la conectividad de los dispositivos, servicios y la aplicación.						
Técnicas	Configurar todos los dispositivos del sistema.						
Casos de prueba	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center; background-color: #e1eef6;">Formato de casos de pruebas</td> </tr> <tr> <td> Tipo de Prueba: Revisión del prototipo. Objetivo: Validar la funcionalidad del sistema. </td> </tr> <tr> <td style="text-align: center; background-color: #e1eef6;">Caso No. 1</td> </tr> <tr> <td> Descripción: Configuración correcta de cada dispositivo </td> </tr> <tr> <td> Entradas: Parámetros XBee, configuración raspberry, instalación de la aplicación en dispositivo móvil. </td> </tr> <tr> <td> Salidas esperadas: Notificación en el móvil cada vez que se acciona el timbre. </td> </tr> </table>	Formato de casos de pruebas	Tipo de Prueba: Revisión del prototipo. Objetivo: Validar la funcionalidad del sistema.	Caso No. 1	Descripción: Configuración correcta de cada dispositivo	Entradas: Parámetros XBee, configuración raspberry, instalación de la aplicación en dispositivo móvil.	Salidas esperadas: Notificación en el móvil cada vez que se acciona el timbre.
Formato de casos de pruebas							
Tipo de Prueba: Revisión del prototipo. Objetivo: Validar la funcionalidad del sistema.							
Caso No. 1							
Descripción: Configuración correcta de cada dispositivo							
Entradas: Parámetros XBee, configuración raspberry, instalación de la aplicación en dispositivo móvil.							
Salidas esperadas: Notificación en el móvil cada vez que se acciona el timbre.							
Resultados	<p>Todos los casos de prueba se han ejecutado.</p> <p style="text-align: center;">Todos los defectos identificados se han considerado.</p> <p style="text-align: center;">Se logró mejorar la presión de la notificación cuando se pulsa el timbre.</p>						
Observaciones	<p>Se observó impresión de tiempo en la ejecución de códigos, que afectaban el tiempo de envío de la notificación una vez presionada la botonera, se resolvió el mismo usando la función millis().</p>						

Objetivos de la prueba	Prueba #2. Comprobar la conectividad de los dispositivos, servicios y la aplicación.
Técnicas	Configurar todos los dispositivos del sistema.
Casos de prueba	Formato de casos de pruebas
	Tipo de Prueba: Revisión del prototipo.
	Objetivo: Validar la funcionalidad del sistema.
	Caso No. 1
	Descripción: Configuración correcta de cada dispositivo
	Entradas: Parámetros XBee, configuración raspberry, instalación de la aplicación en dispositivo móvil.
Salidas esperadas: Notificación en el móvil cada vez que se acciona el timbre.	
Resultados	<p>Todos los casos de prueba se han ejecutado.</p> <p>Todos los defectos identificados se han considerado.</p> <p>Se logró evitar el bloqueo del puerto serial.</p>
Observaciones	El puerto serial se bloqueaba por un error en el canal de comunicación al tratar de difundir el mensaje de confirmación, se solucionó usando un solo recurso para efectuar ambos mensajes.

Objetivos de las prueba	Prueba #3. Comprobar la conectividad de los dispositivos, servicios y la aplicación.							
Tecnicas	Configurar todos los dispositivos del sistema.							
Casos de prueba	<table border="1"> <tr> <td data-bbox="737 552 1409 625" style="text-align: center;"> Formato de casos de pruebas </td> </tr> <tr> <td data-bbox="737 625 1409 699"> Tipo de Prueba: Revision del prototipo. </td> </tr> <tr> <td data-bbox="737 699 1409 821"> Objetivo: Validar la funcionalidad del sistema. </td> </tr> <tr> <td data-bbox="737 821 1409 894" style="text-align: center;"> Caso No. 1 </td> </tr> <tr> <td data-bbox="737 894 1409 1016"> Descripcion: Configuracion correcta de cada dispositivo </td> </tr> <tr> <td data-bbox="737 1016 1409 1203"> Entradas: Parametros XBee, configuracion raspberry, instalacion de la aplicación en dispositivo movil. </td> </tr> <tr> <td data-bbox="737 1203 1409 1333"> Salidas esperadas: Notificacion en el movil cada vez que se acciona el timbre. </td> </tr> </table>	Formato de casos de pruebas	Tipo de Prueba: Revision del prototipo.	Objetivo: Validar la funcionalidad del sistema.	Caso No. 1	Descripcion: Configuracion correcta de cada dispositivo	Entradas: Parametros XBee, configuracion raspberry, instalacion de la aplicación en dispositivo movil.	Salidas esperadas: Notificacion en el movil cada vez que se acciona el timbre.
Formato de casos de pruebas								
Tipo de Prueba: Revision del prototipo.								
Objetivo: Validar la funcionalidad del sistema.								
Caso No. 1								
Descripcion: Configuracion correcta de cada dispositivo								
Entradas: Parametros XBee, configuracion raspberry, instalacion de la aplicación en dispositivo movil.								
Salidas esperadas: Notificacion en el movil cada vez que se acciona el timbre.								
Resultados	<p>Todos los casos de prueba se han ejecutado.</p> <p style="text-align: center;">Todos los defectos identificados se han considerado.</p> <p style="text-align: center;">Se logro establecer mejor eficiencia en la comunicación.</p>							
Observaciones	Esta última prueba permitió ver la necesidad de mejorar la tasa de envío de datos.							

Objetivos de la prueba	Comprobar tiempo promedio de respuesta del sistema.
Técnicas	Configurar todos los dispositivos del sistema, tanto software como hardware.
Casos de prueba	Formato de casos de pruebas
	Tipo de Prueba: Revisión de rendimiento del sistema.
	Objetivo: Validar la funcionalidad del sistema en un entorno de peticiones del timbre.
	Caso No. 1
	Descripción: Configuración correcta de cada dispositivo, sistema funcionando.
	Entradas: Notificaciones del timbre enviadas a usuarios desde su móvil a través de la aplicación.
Salidas esperadas: Tiempo de respuesta favorable.	
Resultados	<p>Todos los casos de prueba se han ejecutado.</p> <p>Todos los defectos identificados se han considerado.</p>
Observaciones	

Objetivos de la prueba	Establecer estabilidad de comunicación entre nodos del sistema a diferentes distancias							
Técnicas	Configurar todos los dispositivos del sistema.							
Casos de prueba	<table border="1"> <tr> <th data-bbox="737 552 1409 625">Formato de casos de pruebas</th> </tr> <tr> <td data-bbox="737 625 1409 699">Tipo de Prueba: Revisión del prototipo.</td> </tr> <tr> <td data-bbox="737 699 1409 877">Objetivo: Validar la estabilidad de la comunicación del sistema, entre sus nodos a diferentes distancias.</td> </tr> <tr> <th data-bbox="737 877 1409 951">Caso No. 1</th> </tr> <tr> <td data-bbox="737 951 1409 1077">Descripción: Configuración correcta de cada dispositivo nuevo anexo a la red</td> </tr> <tr> <td data-bbox="737 1077 1409 1255">Entradas: Nodos XBee separados a diferentes distancias que van desde 1 hasta 15 metros.</td> </tr> <tr> <td data-bbox="737 1255 1409 1386">Salidas esperadas: Comunicación estable y funcional.</td> </tr> </table>	Formato de casos de pruebas	Tipo de Prueba: Revisión del prototipo.	Objetivo: Validar la estabilidad de la comunicación del sistema, entre sus nodos a diferentes distancias.	Caso No. 1	Descripción: Configuración correcta de cada dispositivo nuevo anexo a la red	Entradas: Nodos XBee separados a diferentes distancias que van desde 1 hasta 15 metros.	Salidas esperadas: Comunicación estable y funcional.
Formato de casos de pruebas								
Tipo de Prueba: Revisión del prototipo.								
Objetivo: Validar la estabilidad de la comunicación del sistema, entre sus nodos a diferentes distancias.								
Caso No. 1								
Descripción: Configuración correcta de cada dispositivo nuevo anexo a la red								
Entradas: Nodos XBee separados a diferentes distancias que van desde 1 hasta 15 metros.								
Salidas esperadas: Comunicación estable y funcional.								
Resultados	<p>Todos los casos de prueba se han ejecutado.</p> <p>Todos los defectos identificados se han considerado.</p>							
Observaciones								

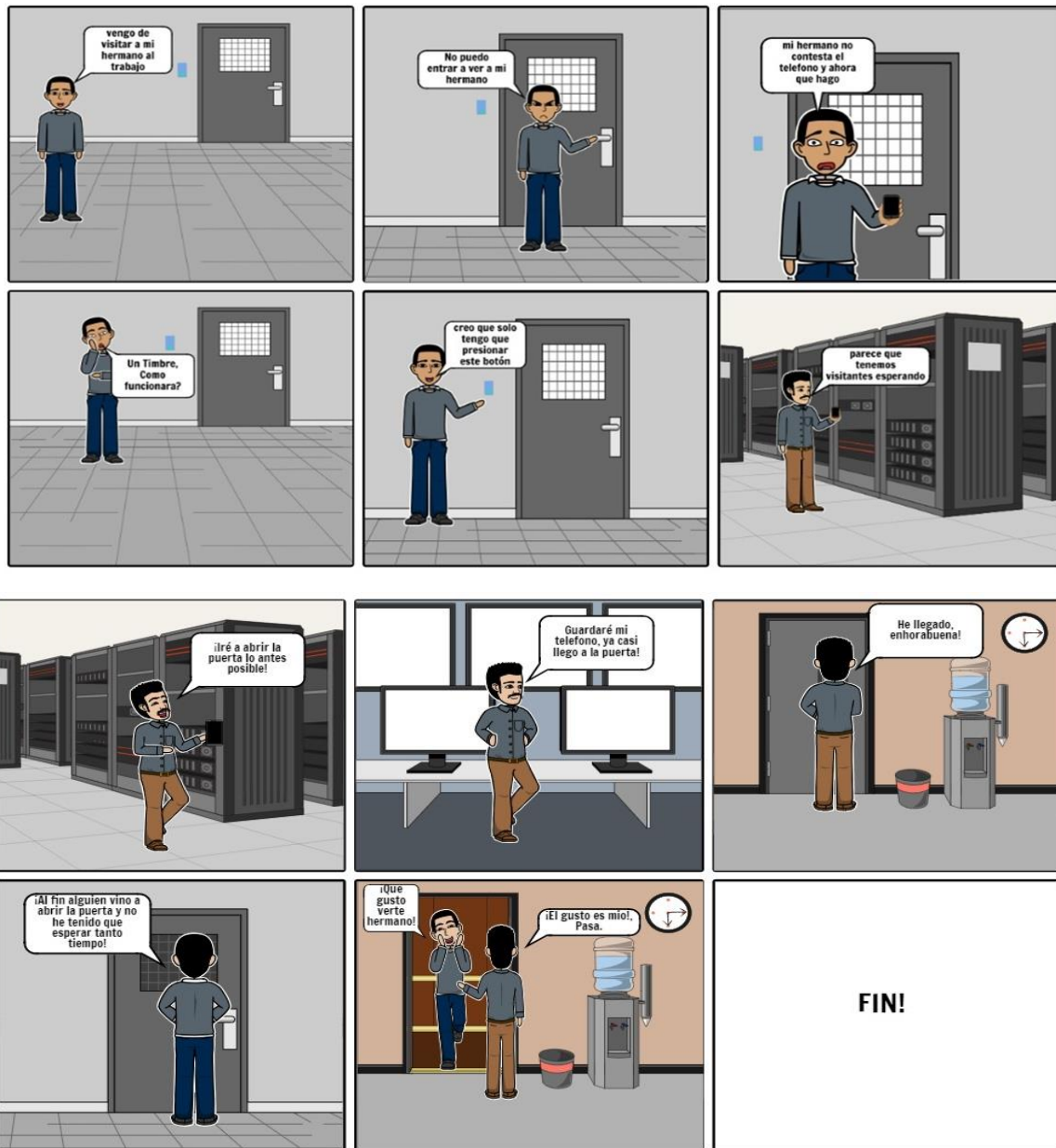
Objetivos de la prueba	Realizar un estudio energético del sistema de timbre inteligente							
Técnicas	Configurar todos los dispositivos del sistema.							
Casos de prueba	<table border="1"> <tr> <th data-bbox="737 552 1409 625">Formato de casos de pruebas</th> </tr> <tr> <td data-bbox="737 625 1409 699">Tipo de Prueba: Revisión del prototipo.</td> </tr> <tr> <td data-bbox="737 699 1409 825">Objetivo: Analizar el rendimiento energético del sistema planteado.</td> </tr> <tr> <th data-bbox="737 825 1409 898">Caso No. 1</th> </tr> <tr> <td data-bbox="737 898 1409 1024">Descripción: Configuración correcta de cada dispositivo nuevo anexo a la red</td> </tr> <tr> <td data-bbox="737 1024 1409 1150">Entradas: Medición de la batería antes de iniciar las pruebas.</td> </tr> <tr> <td data-bbox="737 1150 1409 1276">Salidas esperadas: Medición del nivel de la batería al finalizar las pruebas</td> </tr> </table>	Formato de casos de pruebas	Tipo de Prueba: Revisión del prototipo.	Objetivo: Analizar el rendimiento energético del sistema planteado.	Caso No. 1	Descripción: Configuración correcta de cada dispositivo nuevo anexo a la red	Entradas: Medición de la batería antes de iniciar las pruebas.	Salidas esperadas: Medición del nivel de la batería al finalizar las pruebas
Formato de casos de pruebas								
Tipo de Prueba: Revisión del prototipo.								
Objetivo: Analizar el rendimiento energético del sistema planteado.								
Caso No. 1								
Descripción: Configuración correcta de cada dispositivo nuevo anexo a la red								
Entradas: Medición de la batería antes de iniciar las pruebas.								
Salidas esperadas: Medición del nivel de la batería al finalizar las pruebas								
Resultados	<p>Todos los casos de prueba se han ejecutado.</p> <p>Todos los defectos identificados se han considerado.</p>							
Observaciones								

ANEXO I. Propuesta de prototipo de bajo nivel

Proyecto: Despliegue de redes malladas ZigBee para aplicaciones de edificios inteligentes

Propuesta del prototipo de bajo nivel

Storyboard:



Propuesta para el prototipo de Bajo nivel:

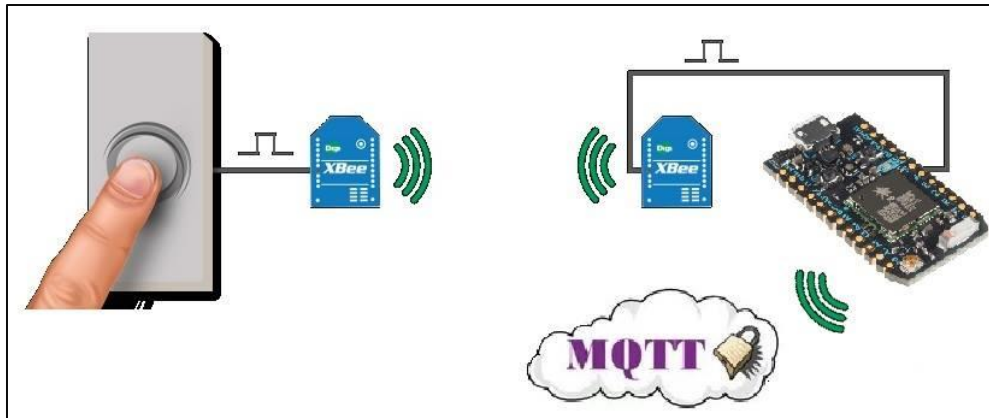
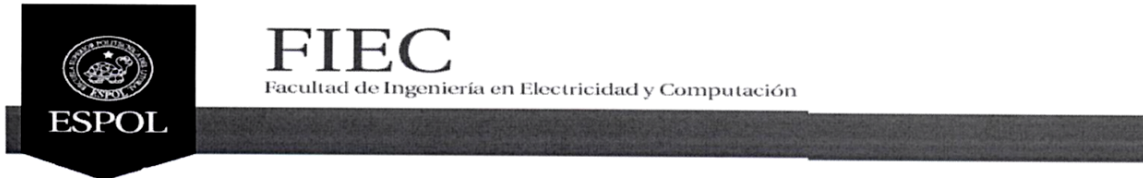


Gráfico 1.4 Esquema general y descriptivo de la solución del problema.

Descripción de la solución planteada:

Se elaborará un prototipo completo y funcional del timbre, el cual consiste en una placa electrónica adaptada a un módulo de Xbee y la respectiva creación de la malla de los demás dispositivos de Xbee enlazados mediante el protocolo de Zigbee, el Xbee adaptado a la placa electrónica toma el nombre de coordinador, el cual se enciende cuando alguien acciona el botón del timbre, del lado de la malla a una distancia de aproximadamente 5 -10 metros estará el Xbee router que será el encargado de encender los demás nodos que conforman la malla esto sucede mediante mensajes de difusión, cabe recalcar que este módulo estará siempre encendido esperando una señal por parte del coordinador(timbre), una vez encendida la malla, la señal habrá llegado a un dispositivo terminal el cual se encarga de conectarse al servidor MQTT implementado en una edición anterior de la resolución del problema, el cual permite notificar la presencia de un visitante.

ANEXO J. Validación del prototipo de bajo nivel



Nombre: Leonardo David Eras Delgado

Fecha: 16/11/2018

1. ¿Considera que el prototipado de bajo nivel propuesto cumple con los requerimientos del proyecto?

Si X No _____

2. ¿Considera que se debe agregar o mejorar algo en la propuesta de la solución escrita?

Si _____ No X

En caso de marcar el sí, Explique ¿Por qué?:

3. ¿Considera que el diseño inicial planteado para el proyecto es el adecuado?

Si X No _____

¿Por qué?

Se necesita transmitir una señal desde la puerta, un punto sin acceso a redes Wi-Fi.

4. ¿Considera necesario el uso de tecnología IoT para el desarrollo de este prototipo de bajo nivel?

Si X No _____

5. ¿Considera la solución planteada para el prototipo de bajo nivel muy sencilla?

Si X No _____

¿Por qué?

Zigbee es un protocolo bastante sencillo de entender y el servidor MQTT ya está levantado

Leonardo Eras
Firma

leoded94@hotmail.com
Correo