
**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE CUENCA**

**CARRERA DE INGENIERÍA ELECTRÓNICA**

# "IMPLEMENTACIÓN DE UNA PLATAFORMA BASADA EN IOT Y SOFTWARE LIBRE PARA EL DESARROLLO DE UN SISTEMA DOMÓTICO"

Trabajo de titulación previo a la obtención
del título de Ingeniero Electrónico

AUTOR: FERNANDO DARÍO PESÁNTEZ PICÓN
TUTOR: ING. ESTEBAN FERNANDO ORDÓÑEZ MORALES, Ph.D.

Cuenca - Ecuador

2022

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN**

Yo, Fernando Darío Pesántez Picón con documento de identificación N° 1400636062, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 30 de agosto del 2022.

Atentamente,

Fernando Darío Pesántez Picón
1400636062

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Fernando Darío Pesántez Picón con documento de identificación N° 1400636062, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: "Implementación de una plataforma basada en IoT y software libre para el desarrollo de un sistema domótico", el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 30 de agosto del 2022.

Atentamente,

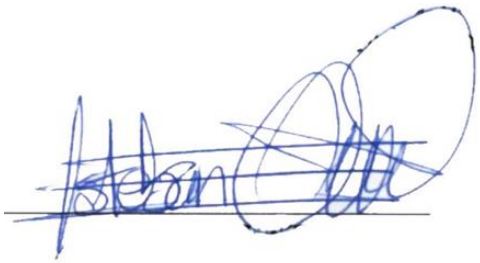
Fernando Darío Pesántez Picón
1400636062

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Esteban Fernando Ordóñez Morales con documento de identificación N° 0102545381, docente de la Universidad Politécnica Salesiana , declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: "IMPLEMENTACIÓN DE UNA PLATAFORMA BASADA EN IOT Y SOFTWARE LIBRE PARA EL DESARROLLO DE UN SISTEMA DOMÓTICO", realizado por Fernando Darío Pesántez Picón con documento de identificación N° 1400636062, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 30 de agosto del 2022.

Atentamente,



Ing. Esteban Fernando Ordóñez Morales, Ph.D.
0102545381

# AGRADECIMIENTOS

Quiero agradecer a todas las personas que estuvieron a mi lado durante el proceso de mi formación profesional, especialmente a mis padres, docentes y compañeros. De manera especial a mi tutor Estaban Ordoñez quien me asesoró y colaboró en el desarrollo de mi proceso de titulación.

**Fernando Darío Pesántez Picón**

# DEDICATORIAS

Este proyecto de titulación va dedicado a toda mi familia, mis padres Hernán Pesántez y Catalina Picón de manera especial a mi abuela Sabina Espinoza y a mis hermanos que han que han sido un apoyo fundamental en mi vida.

**Fernando Darío Pesántez Picón**

# ÍNDICE GENERAL

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS

# RESUMEN

El presente proyecto está dirigido a la automatización de bajo coste de un hogar de clase media, integrando conocimientos programación de placas de desarrollo, internet de las cosas y telecomunicaciones.

En la primera etapa se realizó un análisis de las plataformas existentes con el fin de seleccionar una que cubra las necesidades solicitadas, posteriormente se diseñó un circuito capaz de reconocer un sistema ON/OFF, además se programaron placas de desarrollo NodeMcu para interactuar con la plataforma seleccionada y que funcione conjuntamente con cámaras de seguridad. Se configuró la plataforma para que todo el sistema pueda ser comandado vía web, de forma física, mediante una aplicación móvil, o comandos de voz gracias a un asistente virtual que podría ser *Google* o *Amazon*.

Se realizó el análisis de tráfico de la red con la finalidad de observar el consumo del medio de transmisión, ancho de banda y cantidad de espacio utilizado en la base de datos ubicada en la nube de Internet. Todo este análisis se lo efectuó con el sistema en funcionamiento normal y en diferentes días de la semana.

Además, se realizó un breve estudio de factibilidad financiera del proyecto con la finalidad de estimar un valor aproximado del costo del sistema y también determinar la viabilidad futura de su implementación en una vivienda promedio.

# INTRODUCCIÓN

El desarrollo de la tecnología del internet de las cosas ha tenido un impacto considerable desde sus inicios, en los últimos años es notorio el incremento de la variedad de productos creados bajo este concepto, generándose un nicho de mercado en crecimiento que, además de ser novedoso, es muy útil para facilitar algunas tareas cotidianas en el hogar. El atractivo de esta tecnología crece por la capacidad que tiene al conseguir interactuar con cosas que, comúnmente, antes no podíamos o simplemente lo hacíamos de diferente manera. Por ejemplo, poder controlar el accionar de persianas, tomacorrientes e interruptores mediante el teléfono celular; de esta forma, cada vez más las cosas impensables pueden integrarse al internet de las cosas. El conseguir hacerlo de manera remota, sin ser necesario estar en casa, aumenta el atractivo a la tecnología; pues es posible, monitorear y realizar cambios desde la web. Conocer el estado de nuestros hogares cuando no estamos en casa nos da tranquilidad de que todo está bien, brindando una sensación de seguridad a los consumidores de esta tecnología, además de agilizar tareas comunes, ahorrar tiempo y evitar esfuerzos innecesarios.

# ANTECEDENTES DEL PROBLEMA DE ESTUDIO

La domótica nace en los años 70 cuando en Estados Unidos aparecieron los primeros dispositivos de automatización de edificios basados en la tecnología X-10 [1]. Desde entonces, la Domótica ha tenido gran interés económico y académico, puesto que en ella convergen varias disciplinas como son la electricidad, electrónica, robótica, informática y telecomunicaciones.

Hoy en día, según el estudio *Statista Smart Home Report 2017,* presentado en la feria tecnológica de consumo en Las Vegas en el año 2018, el mercado de la Domótica crecerá a un ritmo del 27,5% anual entre 2017 y 2022.

A lo largo de la historia el ser humano ha tenido la necesidad de mejorar su estilo de vida y facilitar sus tareas cotidianas. Las personas buscan mayor confort en su vivienda lo que aumenta la necesidad de nuevas tecnologías que cumplan con los requisitos y exigencias de los usuarios. Es aquí, donde entra en juego la domótica; sin embargo, se presenta un gran problema a la hora de implementar cualquier sistema domótico en el hogar debido a los altos costos que estos implican, por cada acción que requiera controlar un usuario son necesarios sensores, actuadores o módulos específicos de un determinado fabricante. Estos dispositivos se pueden conseguir en el mercado, pero siempre hay que tener presente la compatibilidad con el sistema domótico que se haya adquirido por parte del usuario y que lo esté utilizando en el hogar. Las empresas que brindan estas soluciones domóticas optan por desarrollar sistemas y plataformas cerradas, con licencias que no permiten la instalación de dispositivos de diferentes marcas que dependen de una plataforma de control y monitoreo que son poco flexibles en el momento de integrar diferentes tecnologías. Si el usuario requiere alguna acción en particular y la empresa dueña del sistema domótico no ofrece ese servicio, se ve obligado a conformarse con las funcionalidades que brinda la empresa o comprar un sistema domótico nuevo de otra empresa que cubra dicho requerimiento; de no ser así, se eleva considerablemente el costo y es necesario

tener varias aplicaciones en el teléfono para conseguir controlar todos los dispositivos instalados. Todo esto genera desagrado y molestias, repercutiendo en una mala experiencia para el usuario; de esta forma, se reduce el interés de querer adquirir o aumentar los dispositivos inteligentes en su hogar.

En estos últimos años grandes empresas han desarrollado sistemas automatizados e inteligentes como Google Home, el cual posee un sistema que obedece a comandos de voz y también integran varios dispositivos del mercado actual, pero muy pocas son las personas que se ven atraídas por esta tecnología debido al desconocimiento de la capacidad que tienen y el costo de implementación.

En el transcurso de la carrera universitaria se han desarrollado proyectos aislados como el reconocimiento de comandos de voz o para controlar actuadores remotamente mediante protocolos IP. Todo ha conllevado a un gran interés en el desarrollo del presente proyecto, al migrar todos estos conocimientos previos a una sola plataforma de software libre que permita vincular varios aspectos en un solo sistema que con soluciones de pago serían prácticamente incompatibles.

# JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)

El ser humano ha tenido la necesidad de mejorar su estilo de vida y facilitar sus tareas cotidianas. Hoy en día esa necesidad sigue siendo prioritaria para la humanidad, es una tendencia en crecimiento puesto que las personas buscan un mayor confort en su vivienda; lo que aumenta la necesidad de nuevas tecnologías que cumplan con los requisitos y exigencias emergentes de los usuarios. Es aquí donde entra en juego la domótica; sin embargo, se presenta un gran problema a la hora de implementar cualquier sistema domótico en el hogar con respecto al precio de los sensores y actuadores, esto debido a que dependen de una plataforma de control y monitoreo que son de costos elevados y poco flexibles en el momento de integrar diferentes tecnologías. Además, por cada acción que se requiera controlar se necesita un módulo específico del fabricante escogido por el usuario y su correspondiente sensor y actuador. Este problema se visualiza claramente al consultar los precios de las principales empresas en el área de la domótica. Por mencionar algunos, tenemos a Apple que ofrece un sistema de seguridad básico a no menos de los 150 dólares, también tenemos a Phillips que ofrece un sistema de iluminación básico aproximadamente a 185 dólares, todos estos sistemas funcionan bajo su propia plataforma siendo imposible combinar los diferentes dispositivos.

Además, las empresas que brindan estas soluciones domóticas optan por desarrollar sistemas y plataformas cerrados y con licencias, en los cuales son prácticamente imposible implementar dispositivos de diferentes marcas. Esto ocasiona que, si el usuario requiere alguna acción en particular y la empresa dueña del sistema domótico no la tiene, el usuario tiene dos opciones, conformarse con las funcionalidades que brinda la empresa o comprar un sistema domótico nuevo de otra empresa que si cubra dicho requerimiento.

Por estos motivos se han buscado posibles soluciones para estos problemas que presenta los sistemas domóticos. Una de estas soluciones se basa en nuevos conceptos como de IoT y sus plataformas. Estas plataformas son muy flexibles a la hora de admitir diferentes tipos de dispositivos de diferentes marcas, también son fácilmente escalables debido a que se basan en el protocolo IP.

En virtud de lo expuesto, se pretende utilizar una plataforma IoT que tenga las siguientes características (i) abierta, (ii) permita la integración de diferentes tecnologías en el mercado ya existentes y futuras, (iii) flexibilidad y (iv) permita la adaptación de distintos dispositivos de diferentes fabricantes dentro de un mismo sistema domótico.

# OBJETIVOS

## OBJETIVO GENERAL

Implementación de una plataforma domótica basada en IoT y software libre para la integración de módulos electrónicos al sistema.

## OBJETIVOS ESPECÍFICO

- Realizar un estado del arte sobre los diferentes sistemas y plataformas domóticas, dispositivos y sus tecnologías, para determinar la más idónea; así como también, de los conceptos de IoT orientados a la domótica.
- Configurar la plataforma con la finalidad de monitorizar y controlar dispositivos/actuadores domóticos.
- Integrar a la plataforma un asistente virtual, un control inteligente, dos módulos electrónicos que controlarán dos luminarias, la apertura de una puerta, y el sistema de riego inteligente de un jardín; desarrollar

una interfaz móvil para mejorar la experiencia del usuario y la interacción con el sistema domótico.

- Realizar pruebas para verificar el correcto funcionamiento de todo el sistema.
- Realizar un análisis económico de factibilidad para la implementación de la plataforma IoT.

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA O ESTADO DEL ARTE

## 1. INTRODUCCIÓN

En esta sección se presenta el estado del arte, el mismo que abarca una investigación documental para considerar los estudios y resultados generados por otros autores, esto con la finalidad de comprender el funcionamiento más detallado de una red IoT; como por ejemplo, la estructura de comunicación y los protocolos utilizados en estas mencionadas redes; además de las diversas plataformas existentes a nivel mundial que facilitan el control de los dispositivos, la visualización de los datos de los sensores y el estado de cada uno de ellos. Al tener claro estos conceptos es posible una óptima selección de una plataforma que permita cubrir las necesidades requeridas en el presente proyecto.

### 1.1. IOT.

La evolución constante y el desarrollo acelerado de la tecnología de información y comunicación posibilita automatizar e interconectar todo lo que las personas tienen a su alrededor, mejorando su calidad de vida y comodidad en todos los ámbitos de su diario vivir donde todo está conectado a Internet. Una parte de las necesidades tecnológicas presentadas pueden ser solucionadas mayormente con el concepto del Internet de las Cosas (de sus siglas en inglés *Internet of Things*, IoT), que en los últimos años ha presentado avances considerables en su desarrollo; su función es hacer que los objetos escuchen, procesen y ejecuten trabajos por sí solos o mediante el control humano [2]. En otras palabras, este avance tecnológico convierte objetos tradicionales en “inteligentes” [3], consigue que los dispositivos electrónicos sean capaces de comunicarse entre sí, lo que permite que compartan información mediante diversas estructuras de red y protocolos [4].

Muchos objetos se incorporan cada vez más a la amplia gama de artículos IoT disponibles en el mercado actual, pues al hablar del internet de las cosas no nos referimos solamente a dispositivos electrónicos que comúnmente conocemos como laptops, celulares, PC's; sino también, a cosas que no asumimos como electrónicos: ropa, muebles, llaves, puertas, cortinas, lentes, obras de arte, entre otros. IoT también se puede ver como una red global que permite a las personas comunicarse con objetos de diversa índole en el mundo ya que cada uno ofrece una identidad única [2] . Por su parte en [3] se indica que existen varias áreas en las que está presente esta tecnología; por ejemplo, transporte, industria, tiendas, escuelas, vehículos, agricultura, domótica, cuidado de la salud, aplicaciones industriales, militares, etc, en la gráfica Figura 1.1 se representan las áreas donde tiene aplicación esta tecnología.


***Figura 1.1*** *Panorama general de IoT* ***[3]****.*

Algunas de las áreas que más destaca IoT son las siguientes:

- Casas inteligentes que mejoran el estilo de vida personal facilitando la monitorización de los aparatos y sistemas domésticos de forma remota.

- Ciudades inteligentes, cuyo concepto está orientado a mejorar la habitabilidad al facilitar a los residentes de una ciudad, la búsqueda de información interesante y necesaria enfocada a la prestación de servicios que se puedan brindar.
- Redes inteligentes que ayudan a los proveedores a controlar y administrar los recursos para ahorrar energía en respuesta al crecimiento de la población.
- Cuidado de la salud, se esta área IoT sustenta las comunicaciones de diversos sensores y actuadores en los pacientes, de esa manera se lleva un control y seguimiento más exhaustivo.
- Agricultura inteligente orientada a la gestión cuantitativa de la producción, el control medioambiental, la gestión de la calidad, la seguridad de los cultivos o la trazabilidad.

IoT está presente en varios aspectos de nuestra vida desde las rutinas diarias, en las finanzas y hasta el estado de nuestra salud. Debido a esto, un punto importante dentro de IoT es la seguridad en la información, debido a alta sensibilidad de algunos datos de los usuarios que pueden llegar a transitar por una red con esta tecnología. En este sentido, es necesario una política de seguridad que controle el acceso a ese tipo de datos [3].

El crecimiento de esta tecnología no se ha visto afectada por la pandemia del Covid-19; por lo contrario, ha seguido desarrollándose existiendo más conexiones de IoT día a día. Se estima que para el año 2025 habrán más de 30.000 millones de dispositivos IoT conectados, lo que equivaldría a 4 dispositivos por cada persona en el mundo [5] como se aprecia en la Figura 1.2.

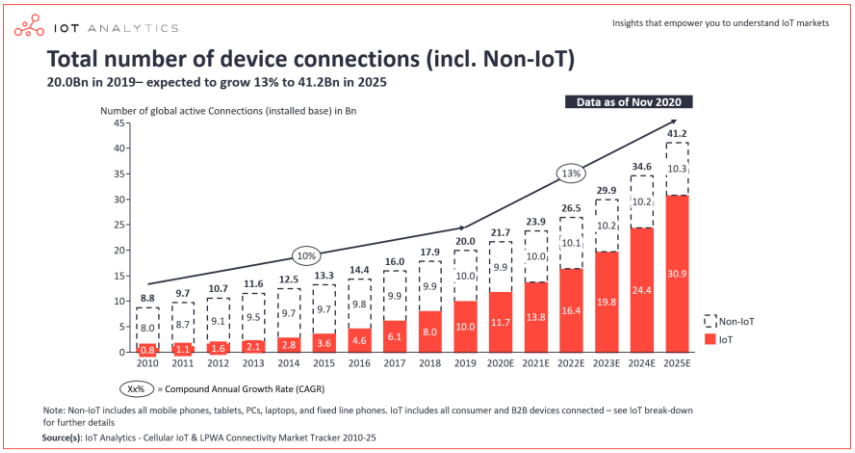



***Figura 1.2 Predicción de la evolución de dispositivos IoT. [4].***

En [6] se mencionan que las grandes empresas no les interesa que los usuarios o la competencia tengan acceso ni conocimiento a los aspectos técnicos del funcionamiento de su sistema IoT. Las empresas, en su mayoría, crean sistemas listos para su utilización obviando por completo la comprensión del manejo de éstos. Los autores de [6] proponen una solución basada en tecnologías de código abierto y las divide en tres capas detección, datos y servicio.

La capa de detección se orienta a la toma de información mediante sensores, mientras que la capa de datos propone la lógica de *back-end* que es el almacenamiento y procesamiento de la información que se va a almacenar para después ser utilizada en otras capas; de tal manera que los usuarios tengan acceso a ella. En cuanto a la capa de servicio, ésta representa el conjunto de aplicaciones que permiten interactuar al sistema IoT con el usuario final de manera mucho más amigable. El método de conexión y comunicación que utilicen los sensores para conectarse a una base de datos y el envío de la información es fundamental para el correcto funcionamiento de un sistema.

El incremento de la cantidad de dispositivos IoT ha fomentado la creación y el desarrollo de plataformas para gestionar estos dispositivos. Comúnmente, cada fabricante tiene su plataforma que gestiona los dispositivos vinculados, el problema de éstos radica en que no todos tienen los mismos costos y sus artículos no ofrecen todos

los servicios requeridos por los usuarios, esto ocasiona que en casa existan distintos dispositivos IoT de varios fabricantes. Normalmente, para conseguir administrar todos estos es necesario tener una aplicación diferente establecida por cada fabricante, según sea el caso. Esto genera desagrado a la hora de gestionar la diversidad de sistemas, debido a que es engorroso tener que abrir una aplicación específica para cada una de las distintas funciones que necesitemos; una solución para estos problemas serían las plataformas IoT. Actualmente, existen varias plataformas disponibles en las cuales podemos vincular dispositivos IoT de distintos fabricantes o creados (según las necesidades del usuario) en una sola aplicación. Con esto podemos tener el control de todos los dispositivos IoT existentes en el hogar y además añadir dispositivos desarrollados por el usuario, resultando ser muy flexibles la implementación de estas plataformas. A continuación, se da conocer una breve descripción de algunas de las plataformas existentes:

## 1.2. **PLATAFORMAS IOT**

### **1.2.1. Ubidots**

UBIDOTS es una plataforma que permite administrar información en proyectos de IoT. Algunas aplicaciones o proyectos de IoT necesitan procesar grandes cantidades de datos, ya sea por la cantidad de sensores que se procesan o por la información que se extrae de ellos [6]. UBIDOTS facilita este proceso, pues permite transferir datos desde los sensores a una nube en Internet, configurar paneles y alertas, conectarse a otras plataformas, usar herramientas de análisis y ver mapas de datos en tiempo real [8], como se observa en la Figura 1.3.

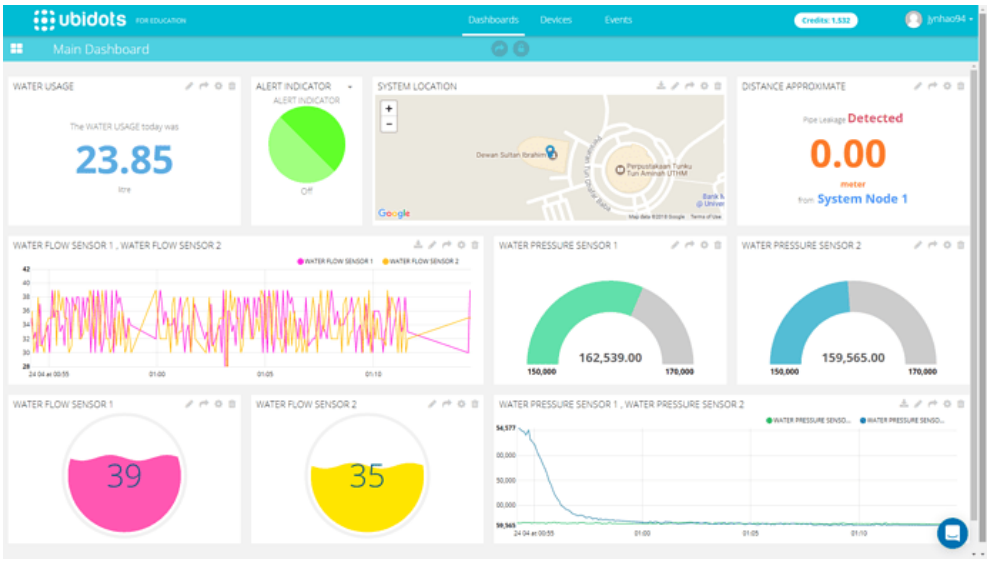



***Figura 1.3 Visualización de la plataforma Ubidots [9].***

### 1.2.2. Node red

Es una plataforma muy potente para comunicar hardware y servicios de forma muy rápida y sencilla. Simplifica enormemente las tareas de desarrollo del lado del servidor con una programación intuitiva [10]. Node-RED fue creado en 2013 por Nick O'Leary y Dave Conway-Jones de IBM *Emerging Technology Services*, su finalidad es solucionar las complejidades que surgen cuando queremos integrar el dispositivo con otros servicios, permitiendo a los usuarios programar sin escribir [11]. Node-RED es un editor de flujo basado en navegador donde se puede agregar o eliminar nodos y conectarlos para comunicarse entre sí (ver Figura 1.4) [12].

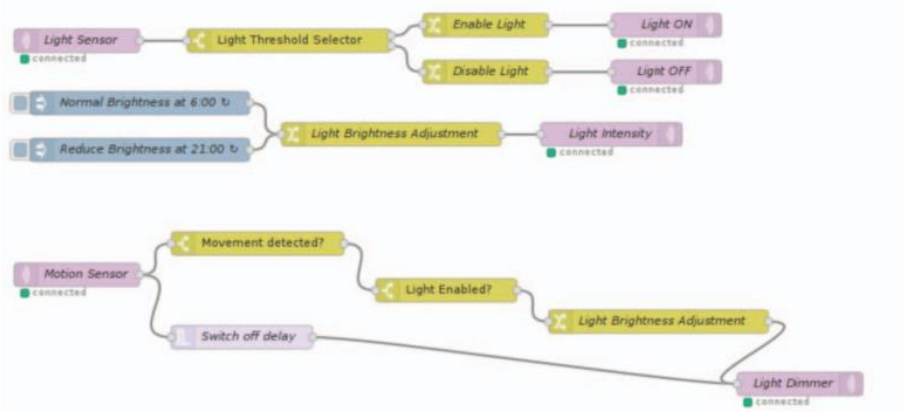



***Figura 1.4 Comunicación Node-red [10].***

### 1.2.3. Home assistant

Home *Assistant* es un software orientado a la automatización del hogar, escrito en Python 3 que es un lenguaje de programación de código abierto. Cualquier persona puede descargar y utilizar todos sus beneficios. *Home Assistant* está diseñado para tomar el control total de su hogar a través de la automatización [13]. Los creadores de esta plataforma son organizaciones sin fines de lucro que colaboran en su tiempo libre, contribuyendo al desarrollo del sistema. Todo su código es gratuito y es posible descargarlo en su página de Github [13] [14].

*Home Assistant* dispone de módulos para gestionar actuadores y datos que ingresen a la plataforma, con la posibilidad de disponer todos los dispositivos como se desee, de manera que se puedan encontrarlos rápidamente cuando los necesitemos; sin necesidad de tener que buscarlos en una interfaz desordenada como sucede en otras plataformas similares (ver Figura 1.5) [15].


***Figura 1.5 Visualización de la plataforma Home Assitant [16].***

### 1.2.4. OpenHAB

Bus de automatización de casa abierta (de sus siglas en inglés *Open Home Automation Bus*, OpenHAB) es una plataforma o sistema IoT de código abierto enfocada en la automatización de hogares [18]. OpenHAB es capaz de gestionar o controlar todos los dispositivos disponibles dentro del hogar. Se caracteriza por ser capaz de coexistir con varios protocolos de comunicación de dispositivos de muchos fabricantes, estos protocolos de comunicación se gestionan desde un único sistema sin ser necesario tener varias aplicaciones para controlar los distintos dispositivos en un hogar [19]. Este software puede proporcionar muchas soluciones predeterminadas fáciles de configurar; sin embargo, cuanto más queremos que las cosas se vean y funcionen de forma personalizada, más trabajo tendremos que poner en ello. OpenHAB consigue realizar cualquier función que se requiera, para esto es necesario tener sólidos conocimientos en programación [18].

Una de las funcionalidades importantes que resalta de OpenHAB es el uso de los canales, éstas son funciones concretas de una cosa (*thing*). Es decir, podemos tener varias funciones en un solo dispositivo y a éstas las vamos a clasificar por canales, de esa manera se tiene acceso por separado a todas las acciones disponibles de dicho dispositivo; de esta forma se consigue controlar varias propiedades de una cosa como se muestra en la Figura 1.6.

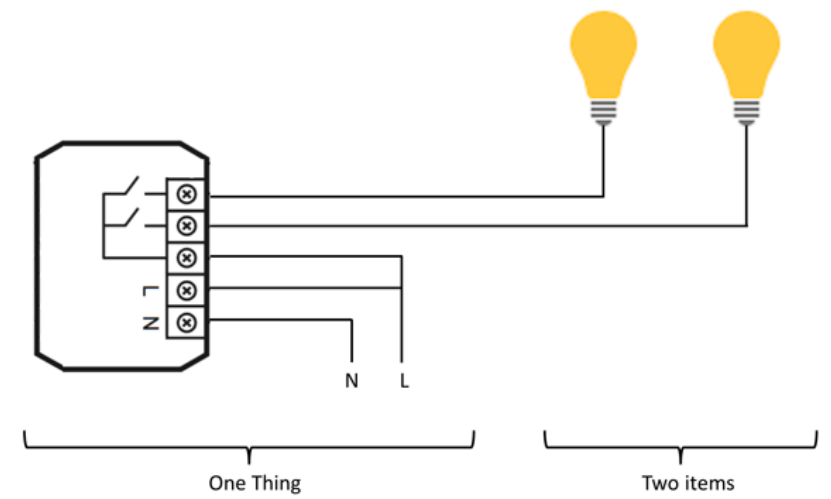



***Figura 1.6 Things e Items [17].***

La característica que destaca en esta plataforma es la interoperabilidad entre varios sistemas domóticos. Sus dispositivos de hardware y sus protocolos se conectan a través de los llamados *Binding*. Los *Bindings* se los puede asemejar a un adaptador de software, estos hacen que todas las cosas asociadas a este *Binding* estén disponibles en el sistema domótico, siendo posible asociar varios de éstos en un solo panel de control, como se muestra en la Figura 1.7. Esta arquitectura permite la creación de una interfaz de usuario centralizada que vincula sistemas independientes, permitiendo el uso de dispositivos que implementan una gran cantidad de tecnologías diferentes según sea su propósito, de esta manera OpenHAB permite utilizar un mismo controlador domótico para diferentes tecnologías.

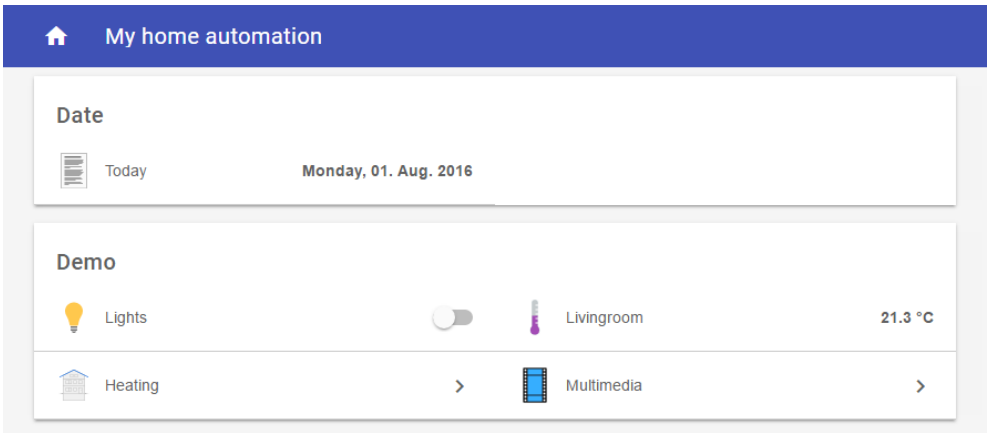



***Figura 1.7 Control y monitoreo OpenHAB [17].***

Para llegar a tener una mejor comprensión acerca de las plataformas analizadas, se presentan en la Tabla 1.1 y Tabla 1.2, donde están las características más considerables de cada una de ellas, así como también sus ventajas y desventajas.

*Tabla 1.1 Resumen de plataformas analizadas pt1.*

| Plataforma | Logotipo | Características | Ventajas | Desventajas |
|---|---|---|---|---|
| Ubidots | ubidots | • Manejo de gran cantidad de datos<br>• Variedad de Dashboards<br>• Muestra los datos en tiempo real<br>• Soporta una Variedad extensa de tarjetas electrónicas.<br>• Soporte y actualizaciones constantes.<br>• Servicio en la nube no necesariamente se tiene que instalar en algún dispositivo para la utilizarlo.<br>• Los datos permanecen en una nube.<br>• Soporta varios protocolos de comunicación: HTTP, MQTT, TCP, UDP.<br>• La plataforma tiene aplicación móvil de uso fácil e intuitivo. [8] | • Es posible utilizar varias tarjetas, debido a la amplia compatibilidad de estos<br>• Manejo simple de plataforma al mostrar datos requeridos<br>• Fácil manejo de extensa cantidad de datos.<br>• Factibilidad de trabajar con MQTT.<br>• No es necesario instalar el software para tener acceso a él. [8] | • Es posible utilizar varias tarjetas, debido a la amplia compatibilidad de estos<br>• Manejo simple de plataforma al mostrar datos requeridos<br>• Fácil manejo de extensa cantidad de datos. [8]<br>• Factibilidad de trabajar con MQTT.<br>• No es necesario instalar el software para tener acceso a él. |
| Node-Red | Node-RED | • Configuración intuitiva y didáctica de la plataforma.<br>• Variedad de Dashboards para la muestra de datos.<br>• Fácil manejo de plataforma.<br>• Funcionamiento basado en flujos, mediante bloques.<br>• Vinculación con varios protocolos incluyendo MQTT, REST, AMQP.<br>• Posibilidad de trabajar con datos de otras Apps conocidas como Instagram, Facebook, Twitter, etc.<br>• Trabaja en dispositivos de capacidad limitada como la Raspberry Pi. [11] | • Para que la plataforma trabaje con sus funciones básicas no es necesario tener un amplio conocimiento de programación debido a que dispone de una interfaz gráfica mediante flujos fácilmente comprensible consiguiendo una agradable experiencia la hora de configurarla.<br>• Extensa información, tutoriales y aportes en la web por parte de los usuarios de la plataforma, facilitando la comprensión de configuración y a su vez del uso de la misma.<br>• Plataforma de código abierto, de modo que no es necesario pagar un rubro mensual por la utilización de esta.<br>• Es posible instalar en la Raspberry Pi y además no ocupa tantos recursos de esta. [11] | • Cuando existen muchos hilos o información la plataforma no responde a la velocidad necesaria.<br>• Cuando se requiere funciones especiales es necesario tener un amplio conocimiento de JavaScript para conseguir que la aplicación funcione correctamente.<br>• No es aconsejable para un amplio manejo de dispositivos<br>• Existen muchos cambios cuando es lanzada una nueva versión, haciendo que los proyectos anteriores no sean actualizados con facilidad. [11] |

*Tabla 1.2 Resumen de plataformas analizadas pt2.*

| Plataforma | Logotipo | Características | Ventajas | Desventajas |
|---|---|---|---|---|
| Home assistant |  | • Esta plataforma fue diseñada en Python.<br>• La plataforma se basa en código abierto.<br>• Para facilitar la configuración de los dispositivos vinculados dispone de módulos para administrar actuadores y datos.<br>• Es necesario instalarlo localmente para su utilizarlo.<br>• Variedad de Dashboards para la muestra de datos.<br>• Dispone de App para la gestión desde dispositivos móviles[12], [14]. | • Es de código abierto.<br>• Manejo simple de plataforma al mostrar datos requeridos.<br>• Variedad de Dashboards para la muestra de datos.<br>• Se puede vincular fácilmente con dispositivos IoT que existen actualmente en el mercado.<br>• Plataforma de código abierto, de modo que no es necesario pagar un rubro mensual por la utilización de esta.<br>• Compatible con varios protocolos de comunicación.<br>• Dispone de una App para teléfonos móviles lo cual no es necesario desarrollar una [12], [14]. | • Es necesario tener un dispositivo siempre para el funcionamiento del sistema.<br>• Es necesario abrir puertos en el Router para tener acceso a las funciones, actuadores y valores de sensores.<br>• Si requiere de Dashboards especiales es necesario sólidos conocimientos de programación en JavaScript.<br>• Uso limitado para un hogar no es proyectado hacia la industria.[12], [14]. |
| OpenHab |  | • La plataforma se basa en código abierto.<br>• Dispone de una App para teléfonos móviles lo cual no es necesario desarrollar una.<br>• Compatible con varios protocolos de comunicación.<br>• Soporte y actualizaciones constantes.<br>• La plataforma está enfocada a la automatización de hogares.<br>• Es necesario instalarlo localmente para utilizarlo [17]–[19]. | • Es posible acceder a los actuadores, datos de sensores, imágenes de forma remota sin ser necesario abrir puertos en el Router ni tener una IP fija.<br>• Es de código abierto.<br>• Manejo simple de plataforma al mostrar datos requeridos.<br>• Variedad de Dashboards para la muestra de datos.<br>• Se puede vincular fácilmente con dispositivos IoT que existen actualmente en el mercado.<br>• Plataforma de código abierto, de modo que no es necesario pagar un rubro mensual por la utilización de ésta.<br>• Compatible con varios protocolos de comunicación.<br>• Dispone de una App para teléfonos móviles, por lo cual no es necesario desarrollar una [17]–[19]. | • Si requiere de Dashboards especiales es necesario sólidos conocimientos de programación en JavaScript [17]–[19]. |

## 1.3. **PROTOCOLOS DE COMUNICACIÓN**

El hardware requiere transmitir los datos que se recolectan del usuario y de su entorno, para realizar esta tarea se usan protocolos de comunicación especialmente desarrollados para IoT. Entre los principales protocolos de comunicación se tiene:

- Bluethooth de bajo consumo de energía (de sus siglas en inglés, *Bluetooth Low-Energy*, BLE): BLE es un protocolo Bluetooth que funciona en la banda de radio industrial, científica y médica (de sus siglas en inglés, Industrial, *Scientific and Medical*, ISM) Trabaja a una frecuencia de 2.4Ghz y es de bajo consumo de energía debido a que el bluetooth está en permanente estado de suspensión mientras éste no sea utilizado. Está basado en el estándar de Bluetooth 4.0 que mejora el radio y la velocidad de transferencia, es una tecnología ampliamente utilizada en el área de IoT por sus prestaciones que ofrece referente a la transmisión de datos y al ahorro de energía.
- Protocolo de cola de mensajes avanzado (de sus siglas en inglés, *Advanced Message Queuing Protocol*, AMQP), Es un protocolo que opera en la capa de aplicación, creado por la necesidad de procesar un gran número de mensajes sin perder ninguno por la importancia que estos tienen, inicialmente basado en el protocolo de control de transmisión (de sus siglas en inglés, *Transmission control protocol*, TCP) es capaz de almacenar mensajes en cola para no perderlos. Además, tiene la cualidad de funcionar de manera asíncrona entre el transmisor y el receptor, sin ser necesario una confirmación de recepción por parte del emisor, consiguiendo seguir con su funcionamiento normal atendiendo a solicitudes de otros y, en el momento que esté disponible el receptor solicitado, enviar los mensajes almacenados en cola, de esa manera se optimiza los tiempos de inactividad de la red.
- Servicio de distribución de datos (de sus siglas en inglés, *Data Distribution Service*, DDS), es un protocolo diseñado para facilitar el intercambio de datos

entre aplicaciones, es de tipo *publicación/suscripción* opera en la capa de sesión y es comúnmente utilizado para aplicaciones de comunicación máquina a máquina (de sus siglas en inglés, *Machine to Machine*, M2M)

Una vez mencionados estos protocolos se puede observar que existen varios estándares bajo los cuales los dispositivos consiguen comunicarse entre ellos, ya sea entre *software* y *hardware* o únicamente entre *software*. Estos protocolos han facilitado el desarrollo de aplicaciones más detalladas y complejas al no ser necesario profundizar en la transmisión de datos entre aplicaciones o bases de datos [18].

### 1.3.1. MQTT

Transporte de telemetría *MQTT* (de sus siglas en inglés, *Message Queuing Telemetry Transport*, MQTT). *MQTT*, al igual que los anteriores, es también un protocolo de comunicación entre dispositivos *M2M* creado por Andy Stanford-Clark de la compañía International Business Machines más conocida como IBM y Arleen Nipper en 1999, para conseguir la comunicación de telemetría de oleoductos por satélite. En el año 2010 este protocolo fue liberado a todo el mundo, siendo uno de los pilares fundamentales para la domótica en los últimos años [19].

Una red MQTT opera bajo TCP/IP u otros protocolos, está diseñada para ser utilizado en IoT, funciona como un modelo publicación/suscripción que comúnmente es utilizado para las comunicaciones *M2M*, permitiendo la comunicación bidireccional entre los dispositivos, enviando la información de manera ordenada y sin pérdida [20]. *MQTT* es un protocolo de mensajería simple, especialmente diseñado para trabajar con dispositivos con un ancho de banda limitado; es decir, se enfoca en la transmisión de datos en bajo rendimiento en redes que presenten alta latencia o que se encuentren con una eficiencia limitada lo que posibilita usar paquetes de control pequeños y livianos [21]. Su estructura de red es de tipo estrella ya que todos los clientes se conectan directamente al servidor llamado "*Broker*". La estructura principal de una red MQTT está compuesta por:

1. $Broker$: en éste se gestiona o monitorea la función sobre el mensaje publicado o suscrito [20], constantemente se envían mensajes (*PINGREQ*) para mantener el canal activo, responde con un mensaje (*PINGRESP*), tiene la capacidad de gestionar la comunicación de hasta 10.000 clientes [10] como se ve en la Figura 1.8. Las funciones del $broker$ son las siguientes:

   - Acepta las solicitudes de los clientes.
   - Recibe los mensajes publicados por los clientes.
   - Procesa las solicitudes de suscripción o dar de baja a un usuario.
   - Envía los mensajes a los clientes destinados.

2. *Clientes:* los clientes pueden tener dos funciones como $publicador$ y $suscriptor$.

   - $Publicador$: El cliente en modo $publicador$ transmite información al broker con un $topic$ definido.
   - $Suscriptor:$ En el modo $suscriptor$ el cliente recibe información del $broker$ de igual manera con un $topic$ definido.

3. $Mensaje:$ es la información que comparten los clientes con el $broker$ mediante un $topic$ definido.

4. $Topic:$ es una etiqueta donde los clientes pueden suscribirse para enviar o recibir mensajes, éstos pueden ser almacenados para una cola de mensajes y pueden ser utilizados por varios clientes [21].

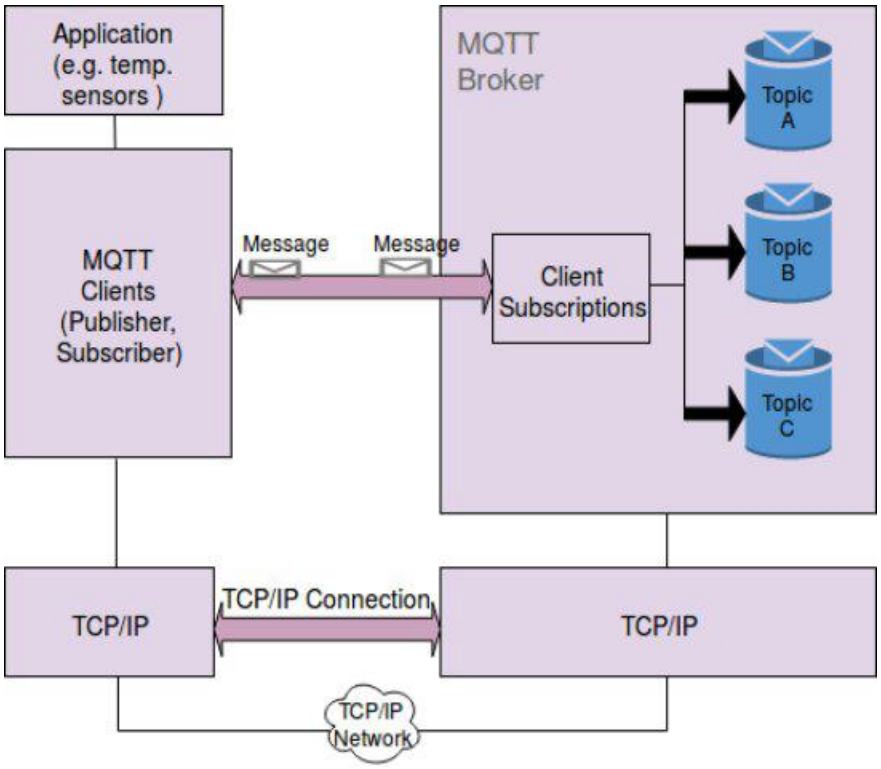



***Figura 1.8 Funcionamiento MQTT [23].***

Antes de conseguir una comunicación entre los *clientes* y el $broker$, este último establece una sesión y la cierra al finalizar la comunicación. El $broker$ como intermediario de comunicación entre los clientes ofrece ciertas ventajas como por ejemplo [21]:

- El $publicador$ solo necesita conocer la IP y el puerto utilizado del $broker$ para lograr compartir y recibir información.
- El $publicador$ y el $suscriptor$ no necesariamente tienen que estar conectados al mismo tiempo, puesto que, el $broker$ es capaz de almacenar la información para cuando ésta esté disponible.
- No es necesaria una sincronización entre el $publicador$ y el $suscriptor$.

Como se mencionó, un escenario común de una red MQTT se presenta de la siguiente manera: funciona en tipo estrella como se muestra en la Figura 1.9, siendo

el $broker$ el centro de gestión de mensajes. Los clientes se encuentran a los extremos enviando o recibiendo información mediante $topics$ como $publicador$ o $suscriptor$.

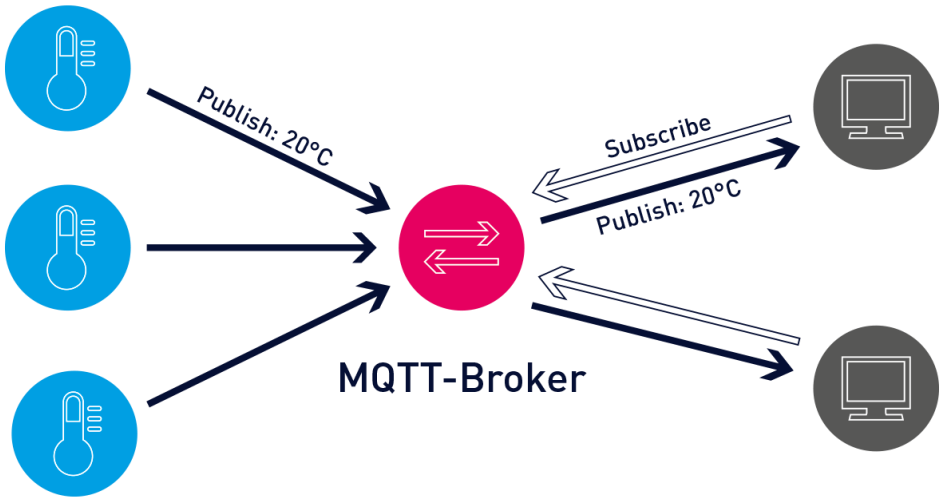



***Figura 1.9 Funcionamiento Típico de MQTT [22].***

Una de las principales características de MQTT es que cuenta con un mecanismo de calidad del servicio (de sus siglas en inglés, *Quality of service*, QoS). Esta característica brinda la seguridad de entrega de un mensaje específico. El nivel de calidad de servicio es definido por el cliente que publica un mensaje en el $broker$ en el momento de la suscripción. El $broker$ transmite el mensaje a los clientes suscriptores utilizando el QoS que cada cliente definió al momento de suscribirse con el mismo; en el caso de que el cliente definiera un QoS más bajo que el cliente $publicador$, el $broker$ inmediatamente transmite el mensaje con el QoS más bajo. Existen tres niveles de calidad de servicio que son:

1. QoS0 (como máximo una vez)

Este es el nivel más bajo de QoS, no existe garantía de entrega del mensaje, el $publicador$ envía el mensaje una sola vez sin recibir un mensaje de confirmación

de llegada. Este nivel de calidad de servicio se emplea cuando existe una red estable entre el cliente y el $broker$ y cuando los datos no son tan importantes.

2. QoS1 (Al menos una vez)

En este nivel de calidad se garantiza que el mensaje llegue al menos una vez. El remitente guarda el mensaje hasta que recibe un paquete de confirmación llamado $PUBACK$ del receptor. Con este nivel de QoS es probable que el mensaje se envié más de una vez hasta que se reciba el mensaje de confirmación, en el caso de no recibir el paquete de confirmación $PUBACK$ en un tiempo prudente, el mensaje es reenviado. Este nivel de calidad de servicio se utiliza cuando es necesario recibir todos los mensajes y el receptor puede manejar mensajes duplicados.

3. QoS2 (Exactamente una vez)

En este nivel de calidad de servicio el mensaje se envía una vez y se garantiza que sea recibido una y solo una vez; este QoS es el más seguro, pero a su vez el más lento. Utiliza un identificador de paquetes para verificar que el mensaje se envíe una sola vez, en cuanto el receptor recibe un mensaje $PUBLISH$ éste responde con un mensaje $PUBREC$. En el caso de que el receptor no reciba el mensaje de confirmación de recibido $PUBREC$, se envía un mensaje $PUBLISH$ nuevamente con un indicador de duplicado $DUP$.

### 1.3.2. Paquete de control MQTT

Este protocolo se comunica con otros dispositivos mediante el intercambio de paquetes de control, su estructura está definida por tres partes como se ve en la Tabla 1.3.

*Tabla 1.3 Estructura del paquete de control MQTT [23].*

| |
|---|
| Cabecera fija (presente en todos los paquetes de control MQTT) |
| Cabecera variable (presente en algunos paquetes de control MQTT) |
| Carga de paquete (presente en algunos paquetes de control MQTT) |

### 1.3.3. Encabezado de paquete de control MQTT

Todos los paquetes de control están conformados por un encabezado fijo que contiene el siguiente formato como se muestra en la Tabla 1.4, los paquetes también están definidos según su funcionamiento y se los puede separar por su nombre como en se puede apreciar en la Tabla 1.5.

*Tabla 1.4 Encabezado de paquete de control [23].*

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 1 | Tipo de paquete de control MQTT | | | | Bandera específica para cada tipo de paquete de control MQTT | | | |
| 2 | Longitud Restante | | | | | | | |

*Tabla 1.5 Tipos de paquetes de control de MQTT [23].*

| **Nombre** | **Valor** | **Dirección de flujo** | **Descripción** |
|---|---|---|---|
| Reservado | 0 | Prohibido | Reservado |
| CONNECT | 1 | Cliente->Servidor | Petición desconexión |
| CONNACK | 2 | Servidor->Cliente | Confirmación de conexión |

| | | | |
|---|---|---|---|
| PUBLISH | 3 | Cliente->Servidor O Servidor->Cliente | Mensaje publicado |
| PUBACK | 4 | Cliente->Servidor O Servidor->Cliente | Confirmación de mensaje publicado |
| PUBREC | *s* | Cliente->Servidor O Servidor->Cliente | Mensaje publicado recibido |
| PUBREL | 6 | Cliente->Servidor O Servidor->Cliente | Confirmación mensaje publicado recibido |
| PUBCOMP | 7 | Cliente->Servidor O Servidor->Cliente | Publicación completa |
| SUBSCRIBE | 8 | Cliente->Servidor | Petición de suscripción |
| SUBACK | 9 | Servidor->Cliente | Confirmación de petición de suscripción |
| UNSUBSCRIBE | 10 | Cliente->Servidor | Petición de cancelación de suscripción |
| UNSUBACK | 11 | Servidor->Cliente | Confirmación de petición de cancelación de suscripción. |
| PINGREQ | 12 | Cliente->Servidor | Petición PING |
| PINGRESP | 13 | Servidor->Cliente | Respuesta PING |
| DISCONNECT | 14 | Cliente->Servidor O Servidor->Cliente | Notificación de desconexión |
| AUTH | 15 | Cliente->Servidor O Servidor->Cliente | Intercambio de autenticación |

El protocolo MQTT tiene distintas medidas de seguridad. Los mensajes pueden ser cifrados mediante una capa de sockets seguros (de sus siglas en inglés, *Secure Sockets Layer,* SSL) o también mediante la seguridad de la capa de transporte (de sus siglas en inglés, *Transport Layer Security*, TLS). Esta tecnología mantiene segura la comunicación, es decir, cumple con la función de proteger la información enviada, en este caso por cualquier dispositivo vinculado a la red MQTT; además, cuenta con autentificación por usuario y contraseña o mediante certificado. Este protocolo fue diseñado para la comunicación entre sensores, esta característica hace que los dispositivos ocupen un ancho de bando reducido; además, puede ser ejecutado con pocos recursos (memoria, RAM, procesador).

En [20] se comparan dos protocolos de comunicación para IoT MQTT y HTTP, mediante la simulación de una red en el programa packet tracer 7.2. En esta simulación se envía un comando para encender un ventilador mediante ambos protocolos, denotando la gran ventaja que representa MQTT frente a HTTP. El protocolo tiene un encabezado en la estructura de paquete de datos de 2 bytes representando una gran ventaja frente a los 26 bytes como mínimo ocupados en el protocolo HTTP. El autor indica que estas ventajas se ven reflejadas en la rapidez del protocolo a la hora de enviar y recibir un comando siendo MQTT 6.5 veces más rápida que HTTP por lo que es ideal aplicarlo en IoT orientado a casas inteligentes, denotando que MQTT representa una gran ventaja y es un buen protocolo frente a HTTP [20].

## 1.4. **DISPOSITIVOS COMPLEMENTARIOS**

Son dispositivos electrónicos que permiten recopilar datos o que sirven para alojar a los servidores que son necesarios para el control y el almacenamiento de la información recopilada.

### 1.4.1. Node MCU

El dispositivo que facilita la recolección de datos de distintos sensores y el accionar de varios actuadores es el módulo NodeMcu (ver Figura 1.10 NodeMcu es

una placa de desarrollo de código abierto basado en el chip ESP8266 (ESP-12E) como se puede observar en la Figura 1.10 [24]. En un principio la programación de este kit se realizaba mediante el lenguaje de programación LUA [25], pero hoy en día se lo puede programar desde el ID de Arduino [26], [27]. Estos módulos ofrecen soluciones completas cuando se necesitan dispositivos que trabajen bajo el estándar WiFi y que requieran un consumo de energía bajo, puesto que estos dispositivos trabajan a 3.5 voltios [26]. El ESP8266 es capaz de recibir información de sensores conectados a sus periféricos y enviarla a Internet mediante un enlace WIFI a un punto de acceso [28].

***Figura 1.10 NodeMcu [29].***

Este dispositivo cuenta con ocho puertos digitales y un analógico como se puede observar en la Figura 1.11. Existen 17 pines digitales, pero solo podemos utilizar ocho de ellos pues el resto son usados para la comunicación entre el microcontrolador y la memoria flash.

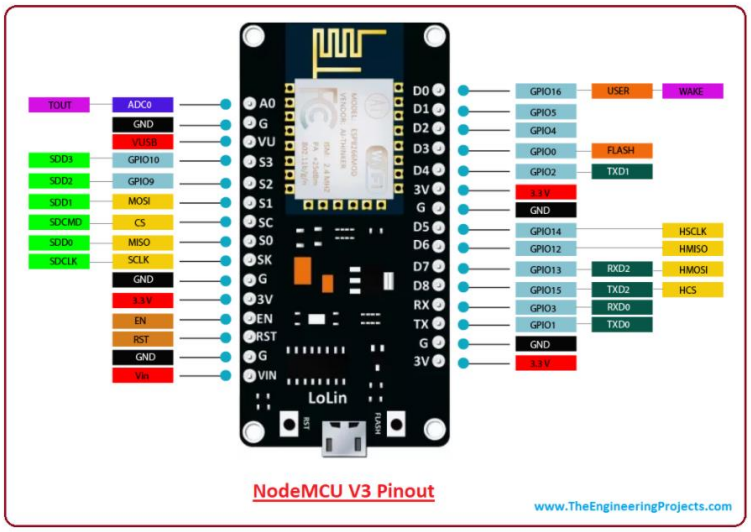



***Figura 1.11 Diagrama NodeMCU [29].***

### 1.4.1. Raspberry Pi.

Este dispositivo fue un proyecto ideado en 2006 con el objetivo de que las personas de bajos recursos puedan acceder a una computadora, ya que esta placa es de muy bajo costo en comparación a las computadoras existentes en el mercado [1]. Su aspecto físico se muestra en la Figura 1.12.

La Raspberry Pi está compuesta por la tecnología chip sobre un sistema (de sus siglas en inglés, *System on Chip*, SoC ) que posee una CPU, memoria RAM, puertos de entrada y salida de audio y vídeo, conectividad de red, ranura SD para almacenamiento, reloj, una toma para la alimentación, conexiones para periféricos de bajo nivel, GPIO, entre otros [30].

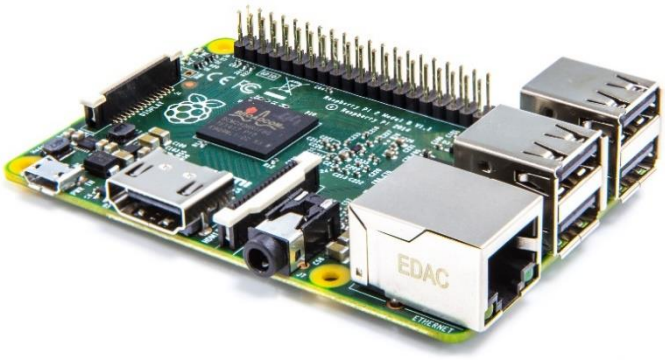
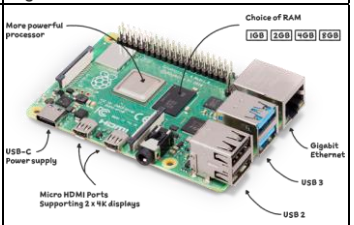



***Figura 1.12 Raspberry Pi 3 Modelo B [10].***

La Raspberry se basa en la arquitectura de un microprocesador llamado máquina avanzada (de sus siglas en inglés *Advanced RISC Machine*, ARM) que utiliza un "grupo reducido de instrucciones para computadoras" (de sus siglas en inglés, *Reduced Instruction Set Computer*, RISC) [29], que ofrece un procesamiento simplificado, bajo consumo de energía y menor costo de fabricación.

En el año de 2011 lanzaron al mercado las primeras placas Beta de las Raspberry pi, demostrando sus capacidades al reproducir videos de 1080p Full HD. Debido a su capacidad por el bajo costo que estos tienen. Existen varias versiones de Raspberry Pi en el mercado como se distingue en la Tabla 1.6, éstas se distinguen por su capacidad y diseño.

***Tabla 1.6 Especificaciones técnicas***
***Raspberry pi A, B, Pi 400, Pi 4.***

| Modelo | Model A+ | Model B | Model B+ | Pi 400 | Pi 4 |
|---|---|---|---|---|---|
| Cpu | Broadcom BCM2837B0 | Broadcom BCM2837 64bit CPU | Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz | Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz | Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz |
| Ram | 512MB LPDDR2 SDRAM | 1GB RAM | 1GB LPDDR2 SDRAM | 4GB LPDDR4-3200 | 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model) |
| USB PORTS | Single USB 2.0 ports | 4 USB 2 ports | 4 USB 2.0 ports | 2 × USB 3.0 and 1 × USB 2.0 ports | 2 USB 3.0 ports; 2 USB 2.0 ports. |
| Tarjeta de red inalámbrica | 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN | BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board | 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE | Dual-band (2.4GHz and 5.0GHz) IEEE 802.11b/g/n/ac wireless LAN | 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE |
| Ethernet Ports | No tiene | 100 Base Ethernet | Gigabit Ethernet | Gigabit Ethernet | Gigabit Ethernet |
| Tarjetas |  |  |  |  |  |

El modelo Raspberry Pi Model 3 B+ tiene una dimensión de 85mm por 56mm, posee, como todas las Raspberry Pi, un microprocesador ARM que en este caso es el Broadcom BCM2837 con una unidad central de proceso de 1.4GHz 64-bit quad-core ARMv8, un RISC de 64 bits, una RAM de 1 GB (compartidos con la GPU). Su fuente de alimentación es de 5V a una corriente máxima de 2.5 A y posee 4 USB 2.0. Una de las características que distinguen a este dispositivo son las entradas y salidas de propósito general (de sus siglas en inglés, *General Purpose Input Output*, GPIO) que consiste en pines con la capacidad de ser controlados mediante una programación previa y que pueden funcionar como entradas o salidas. En el caso de la Raspberry pi Model 3 B+ dispone de 40 pines, estos no se encuentran protegidos contra sobre voltajes por lo que es preciso ser cuidadoso a la hora de trabajar con ellos, pues de ser el caso, un exceso de voltaje de entrada o elevado amperaje de salida por estos pines pueden a llegar a quemar la Raspberry pi.

### 1.4.2. Sensor DHT22

El DHT22 es un sensor digital de temperatura y humedad funciona con voltajes de entre 3,3V a 6V cuando el sensor se enciende tarda un segundo en enviar la información. Es un sensor de tipo capacitivo de bajo consumo de energía que comúnmente es utilizado en proyectos con placas “Arduino” por su fácil conexión de 4 pines como se aprecia en la Figura 1.13, [31].

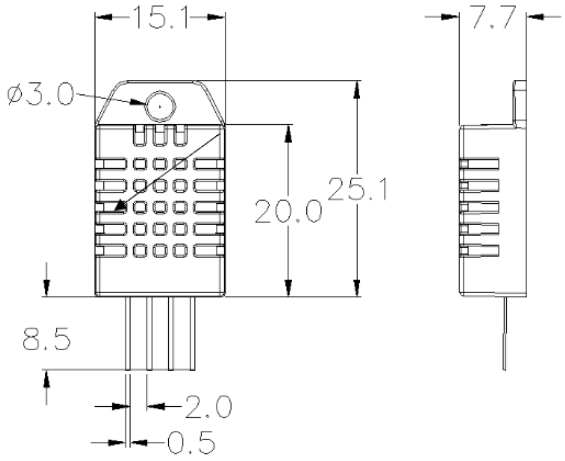



***Figura 1.13 Dimensiones y pines DHT22 [31].***

Los números de pines se los lee de izquierda a derecha siendo estos 4, 1 2 3 4 como se muestra en la Tabla 1.7.

***Tabla 1.7. Detalle pines sensor DHT22 [31].***

| Pin | Función |
|---|---|
| 1 | VDD (voltaje de alimentación) |
| 2 | Datos |
| 3 | NULL |
| 4 | GND |

Las especificaciones técnicas están detalladas en la Tabla 1.8.

***Tabla 1.8. Especificaciones técnicas sensor DHT22 [31].***

| | |
|---|---|
| Model | DHT22 |
| Power supply | 3.3-6V DC |
| Output signal | digital signal via single-bus |
| Sensing element | Polymer capacitor |
| Operating range | humidity 0-100%RH; temperature -40~80Celsius |
| Accuracy | humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius |
| Resolution or sensitivity | humidity 0.1%RH; temperature 0.1Celsius |
| Repeatability | humidity +-1%RH; temperature +-0.2Celsius |
| Humidity hysteresis | +-0.3%RH |
| Long-term Stability | +-0.5%RH/year |
| Sensing period | Average: 2s |
| Interchangeability | fully interchangeable |
| Dimensions | small size 14*18*5.5mm; big size 22*28*5mm |

### 1.4.3. Módulo relé 4 canales.

Este es un módulo compuesto por cuatro relés electromagnéticos, como se aprecia en la Figura 1.14, protegidos por optoacopladores, controlados mediante una señal digital de 5v/20mA que al ser estimulados cierra o abre un contacto de un circuito que contiene una corriente distinta al circuito que la acciona. La potencia máxima soportada por estos relés es de 10A/250V [32].

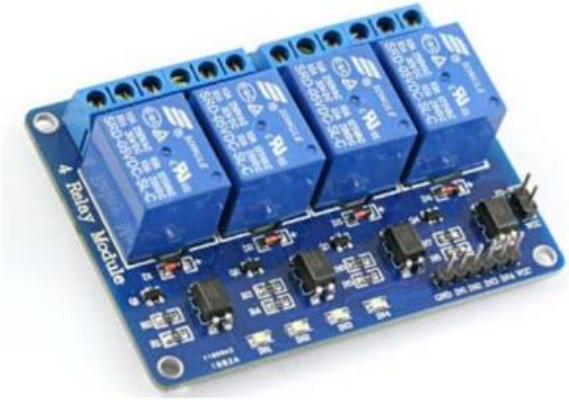



*Figura 1.14 Módulo de relés de cuatro canales [32].*

### 1.4.4. Optoacoplador PC817

Un optoacoplador es un circuito electrónico capaz de aislar circuitos de elevados voltajes con circuitos de control de bajo voltaje; éste identifica voltajes mediante un diodo el cual emite una luz infrarroja hacia un fototransistor como se puede observar en la Figura 1.15, consiguiendo accionarlo sin ser necesario estar en contacto directo con este circuito [33].

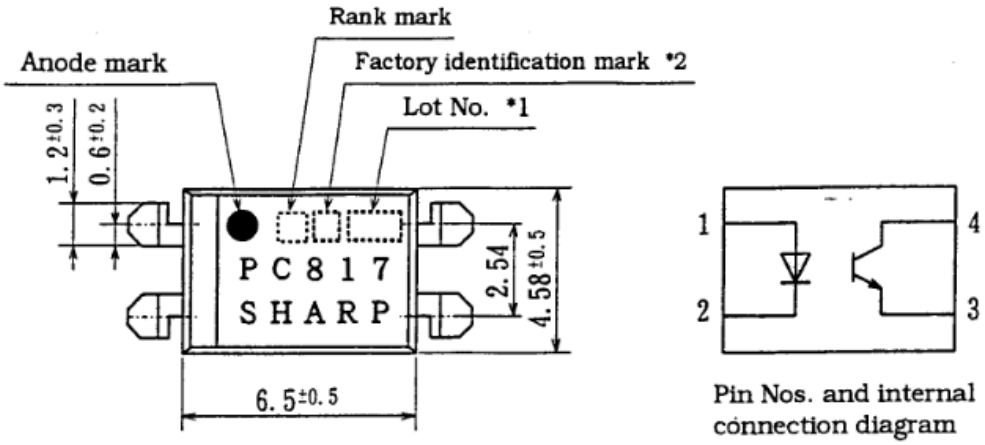



*Figura 1.15 Esquema optoacoplador PC817 [34].*

### 1.4.5. Diodo Zener

Un diodo Zener como se aprecia en la Figura 1.16, es un componente electrónico que permite el paso de corriente en un solo sentido cuando éste alcanza un voltaje determinado.

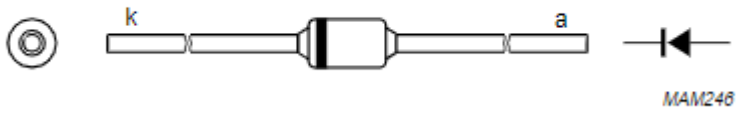



***Figura 1.16 Diodo Zener [33].***

### 1.4.6. Access Point CISCO 1242AG

Un punto de acceso es un dispositivo electrónico que proporciona la capacidad de crear una conexión inalámbrica a una red local, normalmente se utilizan para reducir las conexiones cableadas. Esta tecnología usa bandas libres que están en el rango de 2.4 Ghz y 5 Ghz. Para la banda de 2.4 GHz existen 14 canales de comunicaciones y en la banda de 5 GHz existen 21[35], el dispositivo ocupado es dual band por lo que dispone de ambas bandas, las antenas delanteras son las de 2.4Ghz y las de atrás trabajan a 5 Ghz como se observa en la Figura 1.17.

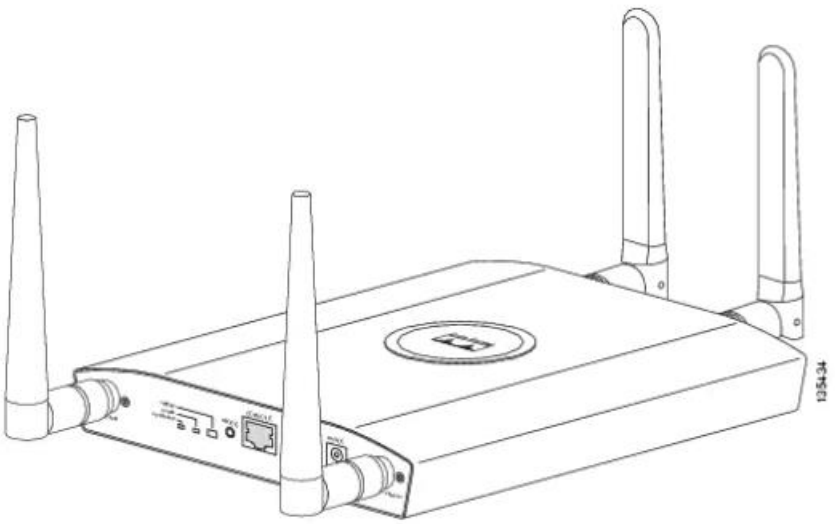


***Figura 1.17 Punto de acceeso Cisco 1242AG [36].***

### 1.4.7. CCTV Hikvision

El circuito cerrado de cámaras de vigilancia o CCTV es una tecnología compuesta por varios dispositivos como videocámaras y dispositivos de almacenamiento, como se observa en la Figura 1.18, éstos captan videos y lo almacenan para, posteriormente, poder acceder a ellos. [37]


***Figura 1.18 CCTV Hikvision funcionamiento [36].***

# CAPÍTULO II:

## 2. DISEÑO Y CONFIGURACIÓN DE LA PLATAFORMA.

El proyecto tiene como finalidad investigar y configurar una plataforma idónea de domótica basada en software libre e IoT, que permita implementar un sistema domótico capaz de reconocer dispositivos de varios fabricantes y que puedan ser comandados por medio de una interfaz web, móvil y de forma manual (esta última dentro del hogar).

A la plataforma escogida se pretende implementarle:

- Un asistente virtual desarrollado por terceros que puede ser de Amazon o Google. A través de este asistente será posible controlar mediante la voz las diferentes operaciones de la plataforma.
- Módulos electrónicos de propósito general que permitan controlar cualquier dispositivo o actuador (se controlará dos luminarias y la apertura de una puerta).
- La visualización de dos cámaras de seguridad.

A modo de aplicación, a través del módulo de control inteligente, se va a:

- Desarrollar y controlar un sistema de riego para jardines que sensará temperatura ambiente, humedad del aire y del suelo; en base a estos parámetros el sistema tomará la decisión autónoma de realizar el riego o no.

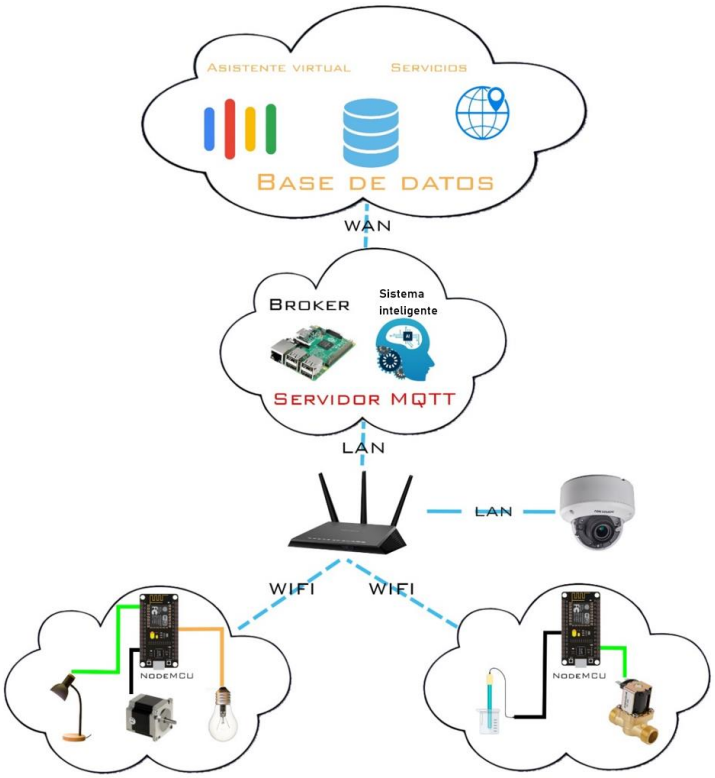



***Figura 2.1 Arquitectura de la plataforma.***

El diseño y construcción de la plataforma del sistema domótico se encuentra estructurada de la siguiente manera, se puede observar en la Figura 2.1, en la parte inferior de esta figura, observamos distintos dispositivos eléctricos y sensores, los mismos que se encuentran conectados a dispositivos electrónicos conocidos como NodeMcu que permiten la comunicación vía WiFi hacia un router y, de éste, hacia el $broker$ (otros dispositivos que tienen la conexión integrada de WiFi, como las cámaras de vigilancia, se conectarán directamente al $broker$ vía el router).

El $broker$ está constituido por una Raspberry pi, la cual se establece como un nodo central de comunicaciones entre todos los dispositivos que se encuentran conectados a la red. En la Raspberry pi se instalará y configurará según la plataforma elegida en el capítulo I. La plataforma instalada en la Raspberry pi, permitirá el control de los distintos dispositivos eléctricos y electrónicos conectados a la red; así como también, conocer los valores de los sensores.

Mediante una conexión de red de área amplia energía (de sus siglas en inglés, *Wide Area Network*, WAN), será enviada la información obtenida de los sensores y configuraciones del usuario a una base de datos ubicada en la nube de Internet, para que éstos estén disponibles para el usuario con un respaldo.

Como se mencionó, el sistema también permite ser gestionado mediante comandos de voz que son receptados por un asistente virtual desarrollado por terceros, como pueden ser *Alexa* o *Google Assistant*; el asistente virtual es accedido gracias al enlace a Internet. La intención de estas prestaciones es mejorar la interacción entre la plataforma y el usuario.

La implementación del proyecto puede dividirse en secciones para su mejor comprensión y desarrollo, por lo que se ha propuesto los siguientes pasos:

1. Reconocimiento del lugar donde se va a realizar la instalación del sistema domótico.

2. Instalación del servidor y de la plataforma.
3. Diseño, programación y desarrollo de dispositivos electrónicos.
4. Integración de servicios a la plataforma.
5. Diseño de interfaz para el usuario.
6. Integración de Asistentes virtuales.

## 2.1. RECONOCIMIENTO DEL LUGAR DONDE SE VA A REALIZAR LA INSTALACIÓN DEL SISTEMA DOMÓTICO.

Para iniciar el estudio de la implementación de la plataforma domótica, se considera el plano que se muestra en la Figura 2.2, el cual pertenece a la mayoría de hogares de clase media según el Instituto Nacional de Estadística y Censos, generalmente está constituida por dos plantas con 4 habitaciones, dos baños, sala, comedor, patio, lavandería y garaje [38].

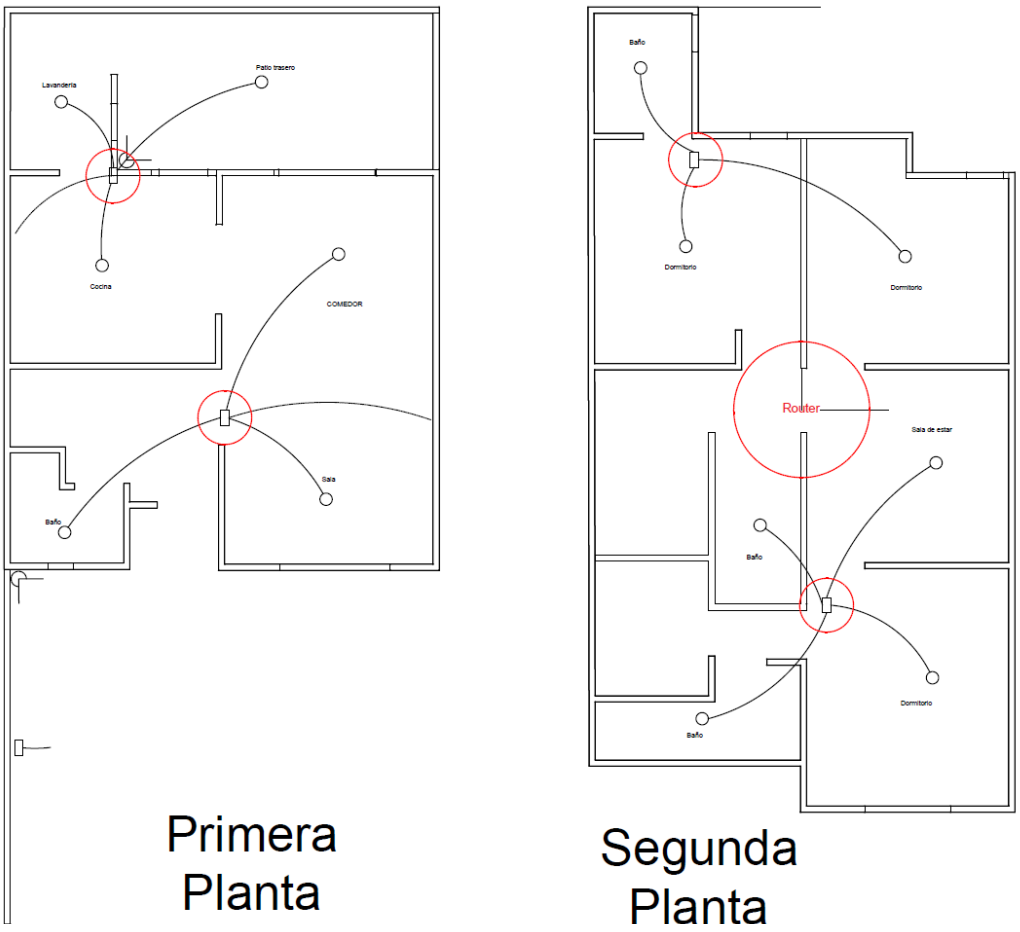



***Figura 2.2 Plano casa de clase media.***

Una vez definido el espacio donde va a ser implementado el proyecto, se elabora el diseño de la ubicación de cada uno de los dispositivos electrónicos, NodeMcu. Es preciso mencionar que se tiene que tomar en cuenta la separación que existe entre el NodeMcu y el dispositivo o luminaria a ser controlada, como se puede observar en la Figura 2.2. Estos dispositivos pueden adquirir distintas funcionalidades de acuerdo a las necesidades presentadas por el usuario, como, por ejemplo, el cambio de estado encendido/apagado de un equipo, o enviar información adquirida de un sensor, según sea el caso.

Para empezar, es necesario tener una red local que servirá de medio de comunicación entre el servidor y los dispositivos, generalmente esta red ya está establecida en la mayoría de los hogares, gracias al servicio de Internet brindado por distintos proveedores de servicio.

En el caso de la presente implementación, se hace uso de un punto de acceso *Cisco 1242AG* a la red para poder gestionar los paquetes inalámbricos de manera eficiente. Una vez definido un nombre de red y una contraseña a la red inalámbrica del AP, se procede a configurar la *Raspberry Pi* para que trabaje como servidor.

## 2.2. **INSTALACIÓN DEL SERVIDOR Y DE LA PLATAFORMA.**

Los dispositivos inteligentes necesitan poder comunicarse entre sí, para esto es indispensable un protocolo de comunicación que consiga el intercambio de información por cualquier medio de transmisión en este caso es MQTT (1.3.1 en el Capítulo I). Para poner en marcha este protocolo es necesario alojarlo en la Raspberry Pi que, como se mencionó, será el servidor central del sistema domótico.

El primer paso es configurar la Raspberry pi e instalar un sistema operativo, en este caso es el *Raspberry pi OS.* A este software lo podemos encontrar en su página oficial *https://www.raspberrypi.com/,* como se muestra en la Figura *2.3*. Una vez con el software descargado se procede a instalarlo en una microSD con la ayuda del programa llamado Raspberry Pi Image.

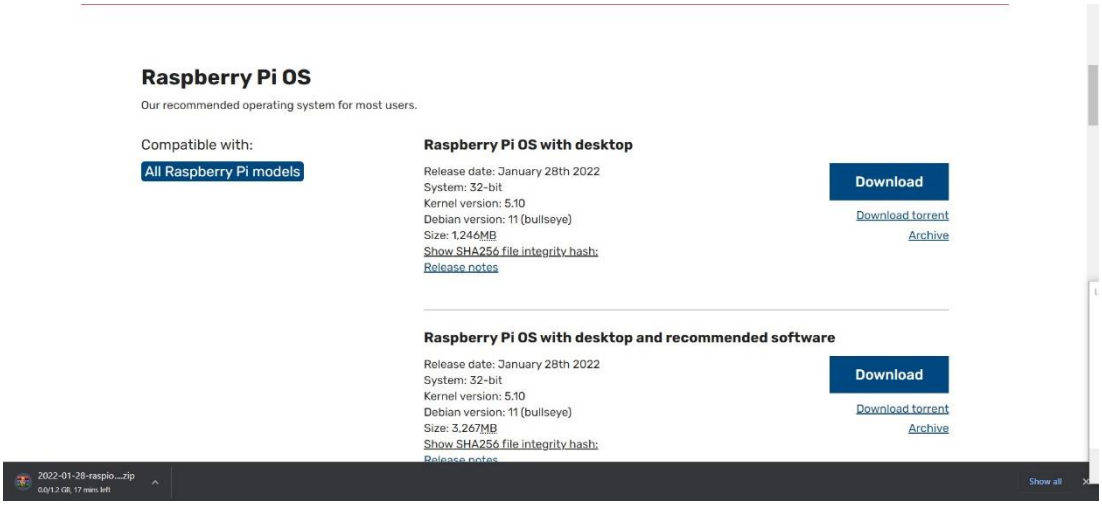



***Figura 2.3 Página oficial Raspian (descarga).***

Es preciso mencionar que se recomienda tener una dirección IP definida en la Raspberry Pi, puesto que los dispositivos que posteriormente se configurarán llevarán en su código dicha dirección.

Para conseguir visualizar la información que los dispositivos comparten y además tener el control de éstos de una manera mucho más didáctica, se instala en la Raspberry pi la plataforma orientada a la domótica llamada OpenHab (ver sección 1.2.4 en el Capítulo I).

Para poder ingresar a la plataforma es necesario un navegador web, el link que nos dará acceso a la configuración es el número de IP del servidor más el puerto *8080,* de esta manera *IP:8080*. Al abrir este link, nos solicita un nombre de usuario y contraseña como se muestra en la Figura 2.4, estas credenciales nos servirán como acceso único a la plataforma.

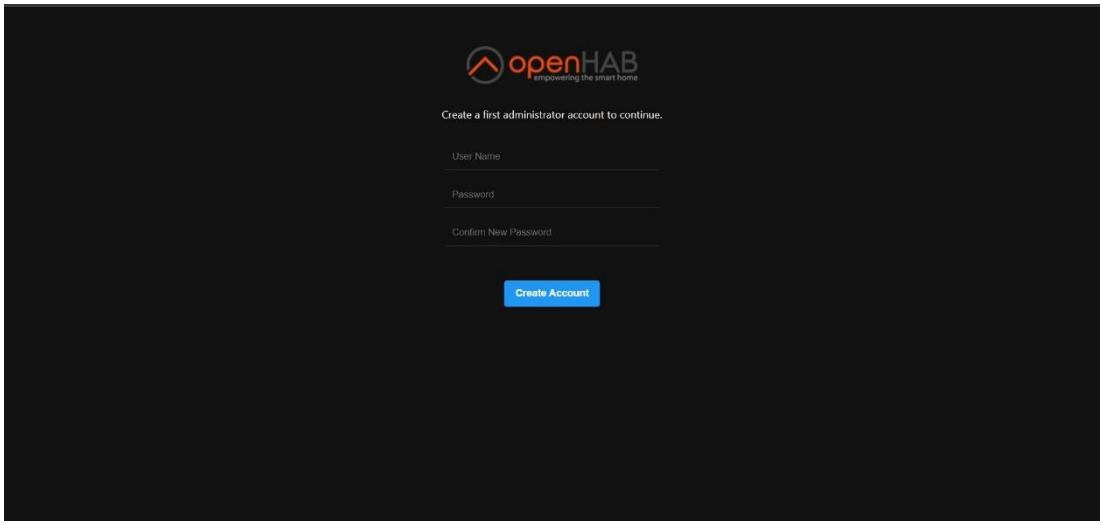



***Figura 2.4. Primer ingreso a plataforma.***

Se ingresa nuevamente las credenciales y se observa el panel principal de configuración, como se observa en la Figura 2.5.

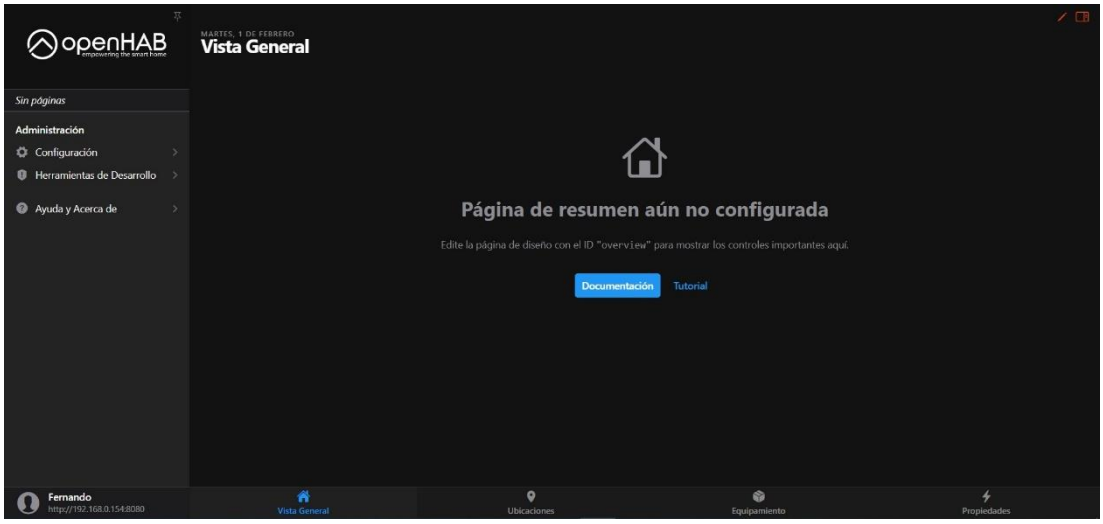



***Figura 2.5. Panel principal OH3.***

### 2.2.1. Instalación de MQTT

El siguiente paso es la instalación del protocolo MQTT, aquí se define también un nombre de usuario y contraseña que serán utilizados por los dispositivos para poder ingresar al servidor, que a su vez hará la función de $broker$.

Para conseguir que la plataforma pueda comunicarse con el servidor, se instala la herramienta MQTT *Binding* que se encuentra en el apartado de *Configuración>Bindings* como se muestra en la Figura 2.6.

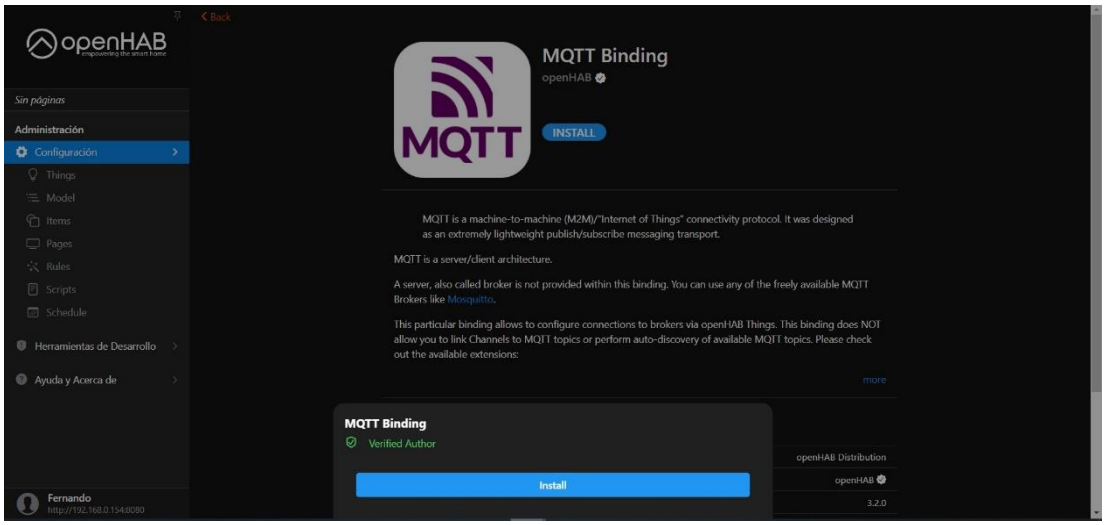



***Figura 2.6 Instalación MQTT Binding.***

Posteriormente se sincroniza la plataforma OpenHAB con el servidor MQTT colocando la IP definida y las credenciales previamente configuradas. Una manera de corroborar la sincronización del servidor MQTT en la plataforma, son los indicadores de estado, si éstos se colocan en verde significa que el servidor MQTT está ya asociado con éxito, como se aprecia en la Figura 2.7. A partir de este punto toda la información que se encamine a través del servidor será reconocida también por la plataforma.

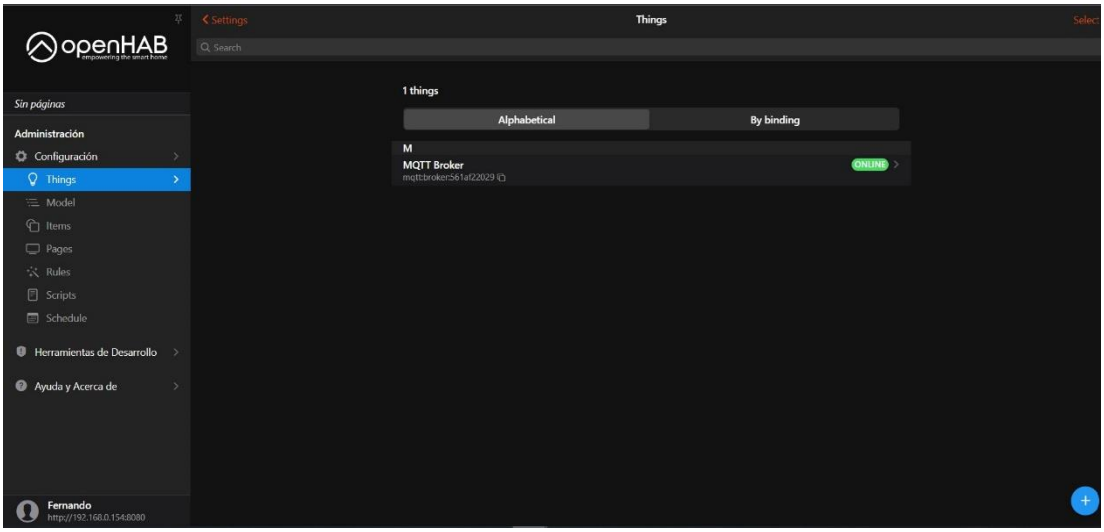



***Figura 2.7 Indicador de plataforma MQTT activo conectado al broker, online.***

## 2.3. DISEÑO, PROGRAMACIÓN Y DESARROLLO DE DISPOSITIVOS ELECTRÓNICOS.

Una vez se comprueba el funcionamiento del servidor MQTT se procede al diseño y configuración de los distintos dispositivos NodeMcu. Para la demostración de la funcionalidad se han diseñado tres dispositivos, uno se encargará del encendido de 2 luminarias y la apertura de una puerta, el segundo obtendrá datos de sensores de humedad del suelo, índice de calor y temperatura ambiente y, por último, el encendido de una luminaria.

Para que estos dispositivos funcionen con estas características es necesario el programa llamado "*Arduino*", donde se programará las distintas funcionalidades que tendrán cada uno de ellos.

Como ya se mencionó, los dispositivos NodeMcu tienen que estar dentro de la misma red en donde se encuentra vinculada la *Raspberry Pi.*

Los dispositivos han sido programados para tener la capacidad de reconocer cuando se enciende o apaga una luminaria asociada a ellos, esto se consigue mediante la ayuda de un circuito electrónico, que al detectar su estado lógico ON/OFF envía mediante $Topic's$ órdenes que el $broker$ reconoce e interpreta.

Para el reconocimiento de los dispositivos físicos (Conmutadores) se ha propuesto el siguiente circuito electrónico que se aprecia en la Figura *2.8,* éste facilita la identificación de cruce por cero es decir reconoce la tensión en la fase, y envía esa información a los puertos digitales del NodeMcu. Consiste en un optoacoplador para el reconocimiento de fase y de un diodo Zener para evitar errores de lectura por corrientes parásitas que puedan aparecer cuando están conectados a la red o a algún aparato eléctrico.

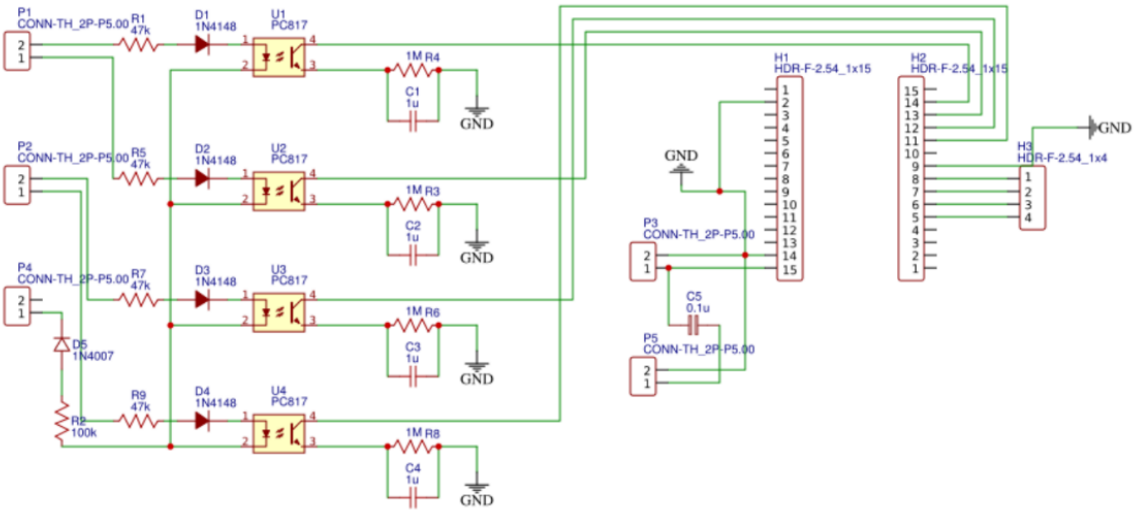



***Figura 2.8 Diagrama de reconocimiento de fase para NodeMcu.***

En la Figura 2.8, se observan cuatro optoacopladores con su respectivo diagrama los cuales se ha utilizado y conectado a los puertos digitales del NodeMcu para leer su estado. El diseño en 3d de esta placa se puede apreciar en la Figura *2.9*. Los puertos que van conectados con la fase que entrega energía a la luminaria son los P1-CONN-TH y P2-CONN-TH como se observa en la Figura 2.8 .

Las borneras facilitan la conexión de los cables de sección más grande que serán conectados a los distintos dispositivos que en este caso son las luminarias y la chapa eléctrica.

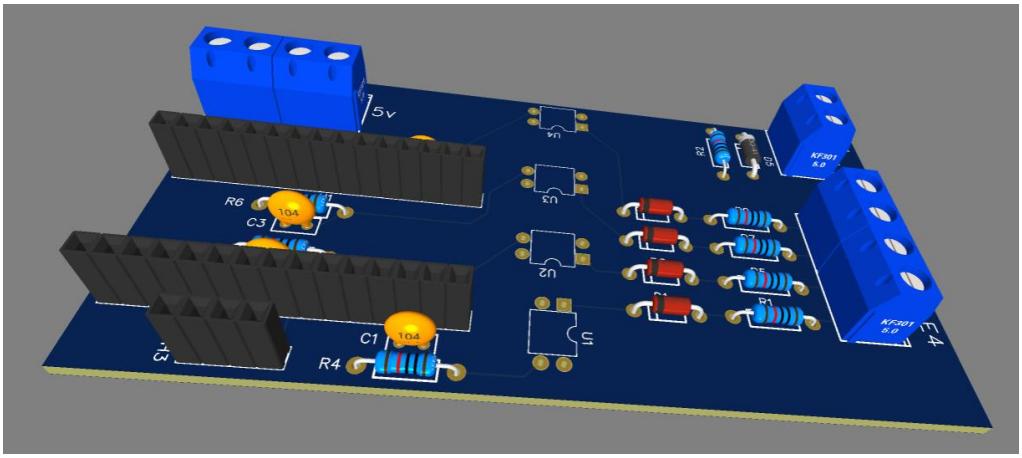



***Figura 2.9 Vista 3D del módulo para cuatro actuadores.***

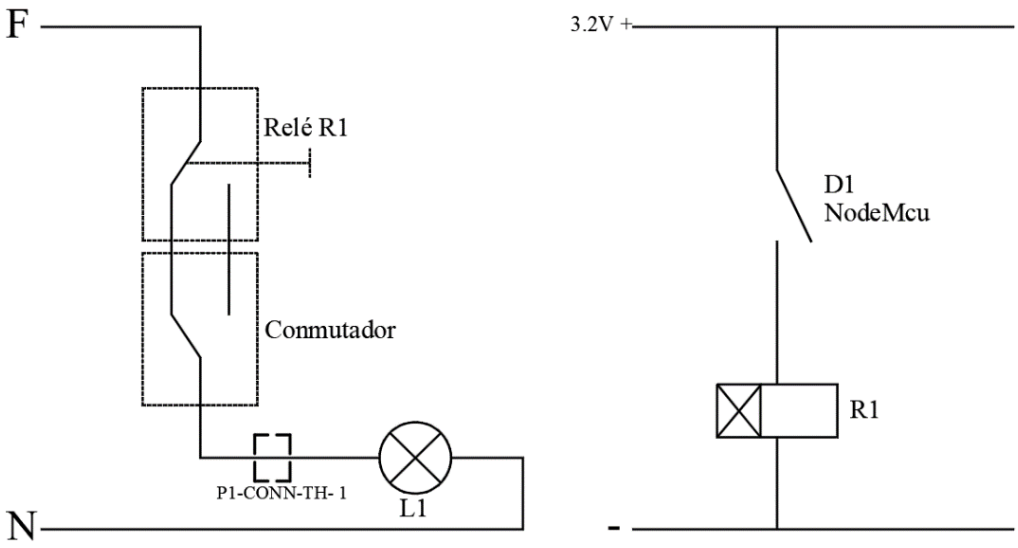



***Figura 2.10 Conexión eléctrica Relé con conmutador.***

La conexión eléctrica que comparte cada relé con los conmutadores se puede apreciar en la Figura 2.10, donde la conexión realiza la conmutación entre el Switch común del hogar y el sistema domótico. En el caso de que se encuentre un sistema conmutado se realiza una conmutación triple para conseguir el control remoto de la luminaria.

La programación realizada se graba en los NodeMcu mediante la plataforma de programación "*Arduino*". Dentro de estas líneas de código se define el nombre de un *Topic*, este sirve cuando el servidor necesita comunicarse con un dispositivo definido; en ese caso, éste responde mediante esta etiqueta. El dispositivo estará escuchando constantemente cualquier orden que se le indique realizar; por ejemplo, en este caso, se ha seleccionado las letras del abecedario para distintas órdenes como por ejemplo la letra "*a*" realiza la acción de encender y la letra "*b*" de apagar, mediante el *topic* llamado "*inTopic*". Para conseguir realizar estas acciones desde la plataforma, se define el nombre del *topic* y los comandos asignados para cada acción.

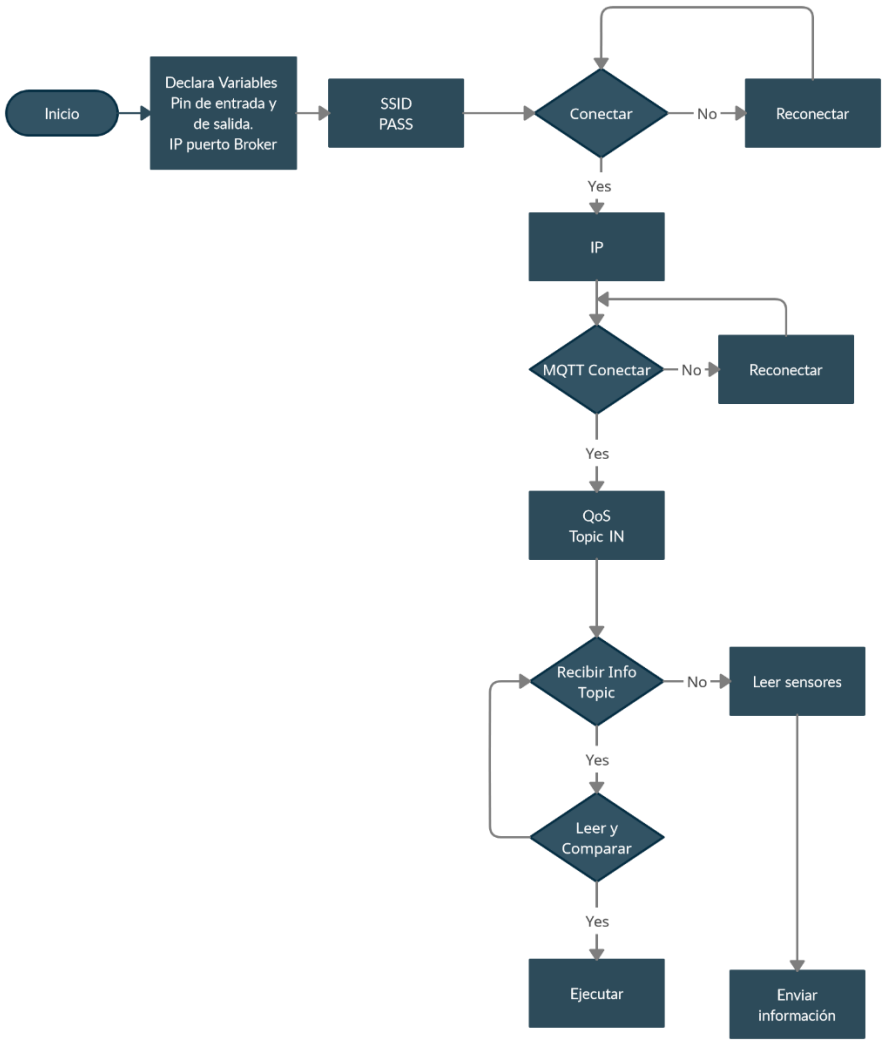



***Figura 2.11 Diagrama de bloques de NodeMcu.***

En la Figura 2.11 se observa el funcionamiento de un nodeMcu mediante un diagrama de bloques donde se indica el proceso de recibir y de enviar información hacia el broker. El nodeMcu primero pasa por un proceso de conexión para vincularse a la red wifi, luego a la red MQTT, una vez establecida la conexión se negocia el QoS con el que va a operar para luego declarar el Topic con el que recibirá información, a partir de este punto recibirá los datos que estén asignado al Topic declarado y podrá enviar información al broker, en el caso de no poder vincularse a la red wifi o a la red MQTT siempre intentará reconectarse.

La automatización de riego con notificaciones a la aplicación requiere que se realice un *Script* dentro de la plataforma. En el apartado del mismo nombre, nos permitirá tomar decisiones en base a los parámetros de los sensores y de la hora en la

que se encuentre operando, ya que se estima un riego por las mañanas y en el caso de hacer falta, también por las tardes, esto todos los días. Para este caso, se considera el valor del sensor de humedad del suelo y la hora, en base a estos resultados se envía una orden para realizar el riego de las plantas y, de inmediato, se procede a notificar al usuario que esta orden ha sido ejecutada. Las notificaciones son enviadas al teléfono celular mediante notificaciones tipo *push*, como se puede observar en la Figura 2.12.


***Figura 2.12 Notificación push OpenHab.***

## 2.4. **INTEGRACIÓN DE SERVICIOS A LA PLATAFORMA.**

Para conseguir que varios servicios se puedan visualizar e interactuar entre sí, la plataforma incluye a éstos mediante *Bindings* que, en este caso en particular, se establecen en el servidor MQTT junto con las cámaras de seguridad. Con esto se procede a configurar la plataforma de la siguiente manera:

- Integración de los módulos electrónicos.
- Integración de servicio de visualización de las cámaras de seguridad.

### 2.4.1. Integración de los módulos electrónicos.

Para incluir los módulos electrónicos y tener el control de ellos mediante la plataforma se crea las llamadas “cosas”, en el apartado de MQTT genéricas se separarán las características especiales de cada dispositivo previamente programados. Para configurarlos nos dirigimos a *Things*> > *MQTT Binding* > *Generic MQTT Thing*, posteriormente se asocia al $broker$ como se aprecia en la Figura 2.13.

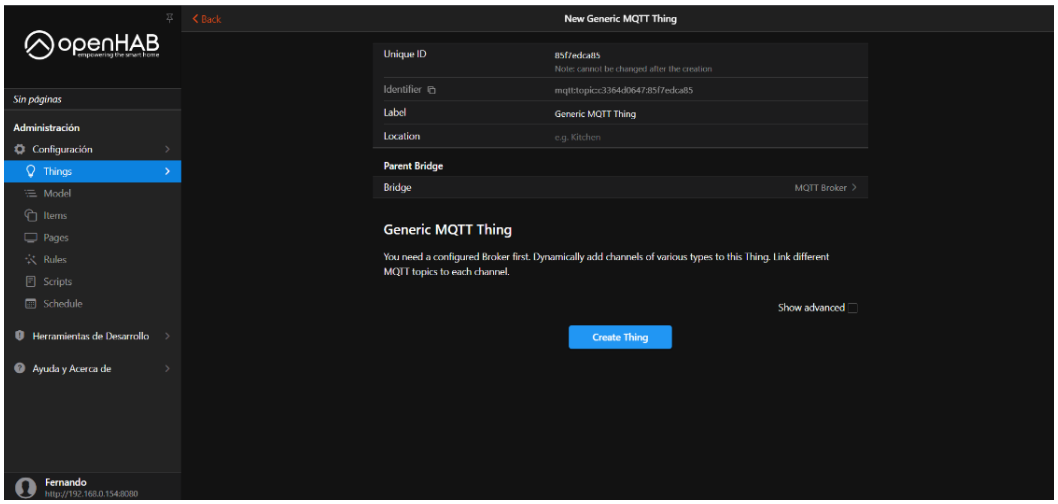



***Figura 2.13 Creación de cosas MQTT genéricas.***

Una vez creado aparecerá dentro de los “*Things*” en donde se puede manipular el estado del mismo, mediante el interruptor que se encuentra en la parte superior, como se muestra en la Figura 2.14.

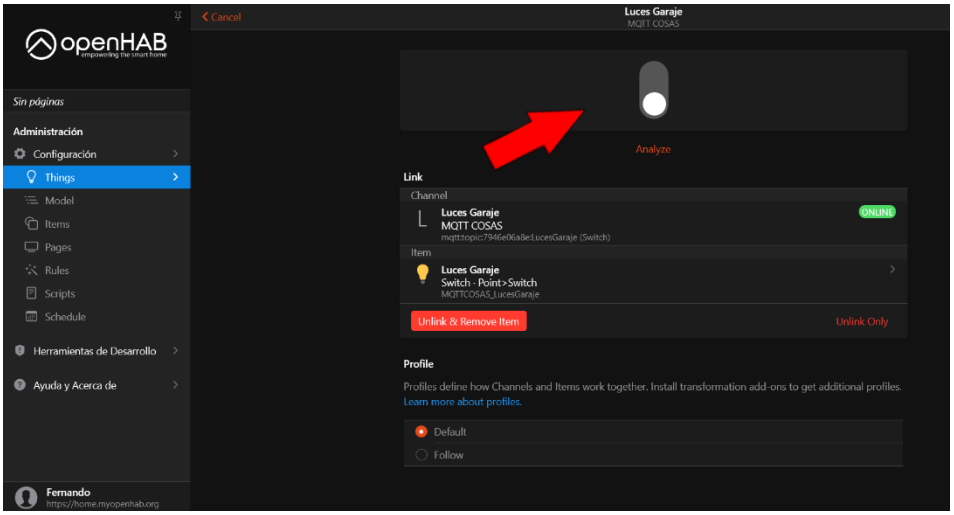



***Figura 2.14 Interacción de “cosas” en la plataforma.***

Una vez añadidos todos los dispositivos, se instala el *openHAB connector* como se observa en la Figura 2.15, esta herramienta permite el acceso remoto sin ser necesario tener una IP pública o una VPN. Recordemos que esta característica es una ventaja muy importante frente a otras plataformas domóticas revisadas en el capítulo I.

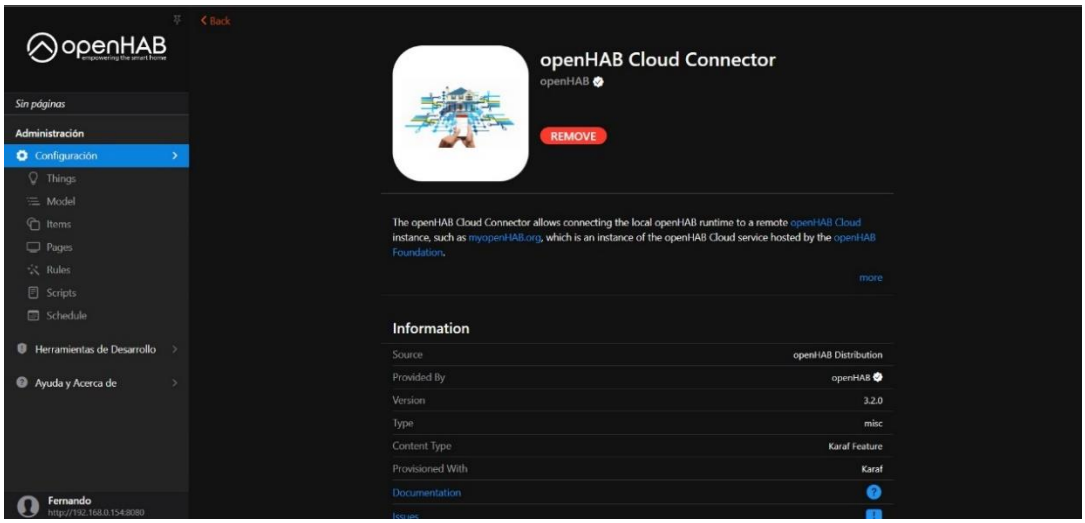



***Figura 2.15 Instalación de OpenHAB connector.***

Para tener acceso a la plataforma desde un teléfono móvil es necesario instalar la aplicación OpenHab y vincularlo a una cuenta de *myopenhab.com*, para esto se obtienen ciertos parámetros que se encuentran en la Raspberry Pi y se los puede encontrar en la siguiente ubicación *"/var/lib/openhab/uuid*". Aquí se obtiene el identificador único universal, llamado "UUID", el formato de este código está representado como se puede observar en la Figura 2.16


***Figura 2.16 Dirección de UUID.***

Ya con el UUID copiado se procede a obtener la contraseña de acceso de la siguiente ubicación "*var/lib/openhabcloud/secret*". El código vendrá dado por la misma plataforma y el formato es parecido al de la Figura 2.17.


***Figura 2.17 Contraseña de acceso a la plataforma.***

Con el *UUID* y la clave se ingresa al link *myopenhab.com* mediante cualquier navegador web y se crea una cuenta que podrá ser asociada a la plataforma con los datos obtenidos de ella. Una vez con la cuenta creada se llenan los espacios que se muestran en la Figura 2.18 con la información que previamente se copió.

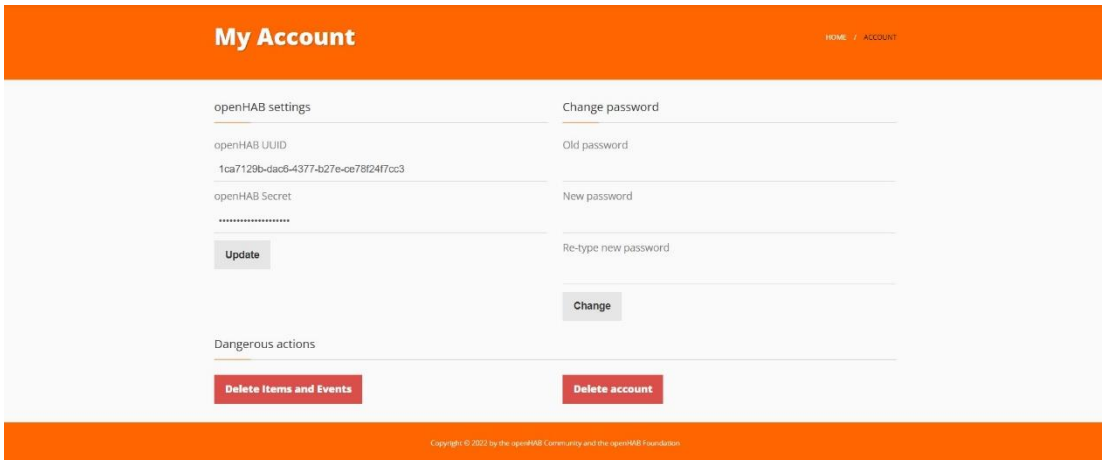



***Figura 2.18 Ingreso de información en la página de OpenHAB.***

Los dispositivos reconocidos online dependen directamente de la configuración de la plataforma local, para esto es necesario configurar los ítems y el sitemap.

La configuración de los ítems está dada en la dirección: "*/etc/openhab/items/Tesis.items*", en este apartado se coloca el nombre de cada dispositivo y se asocia a un canal definido previamente en el Generic MQTT things, como se puede apreciar en la Figura 2.19. Con esto se termina la configuración de los dispositivos a la plataforma.

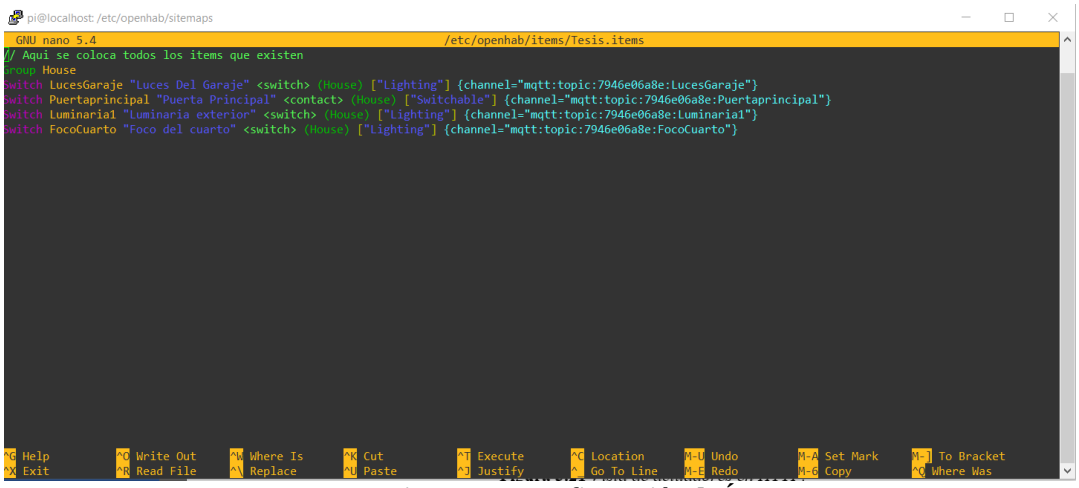



***Figura 2.19 Configuración de Ítems.***

### 2.4.2. Integración del servicio de visualización de las cámaras de seguridad.

Para tener acceso a las cámaras de seguridad se instala el *Binding* llamado "*IPCamera Binding*", como se muestra en la Figura 2.20.

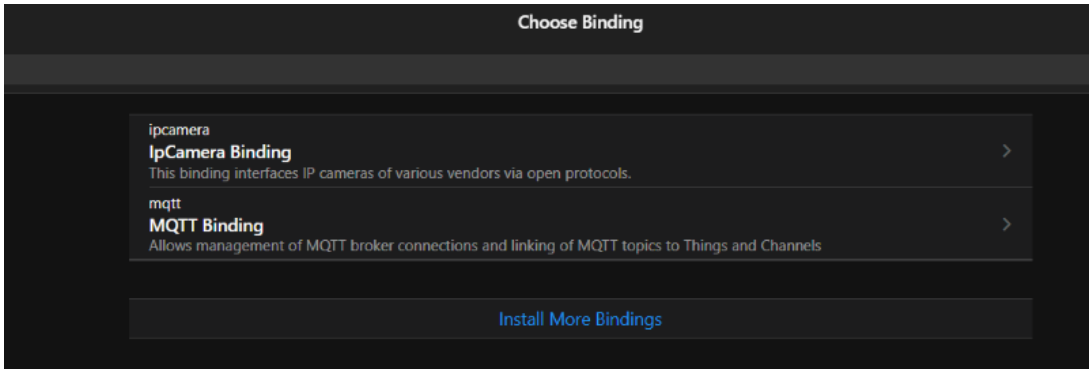



***Figura 2.20 Instalación IPCamera Binding.***

Una vez instalada esta herramienta llenamos los parámetros de configuración que consta en la IP del DVR de las cámaras de seguridad, usuario y contraseña utilizados para ingresar al sistema de las cámaras y el puerto que será utilizado para la visualización de cada una de las cámaras, cabe mencionar que es necesario agregar una por una y configurarla en el sitemap como se muestra en la Figura 2.21.

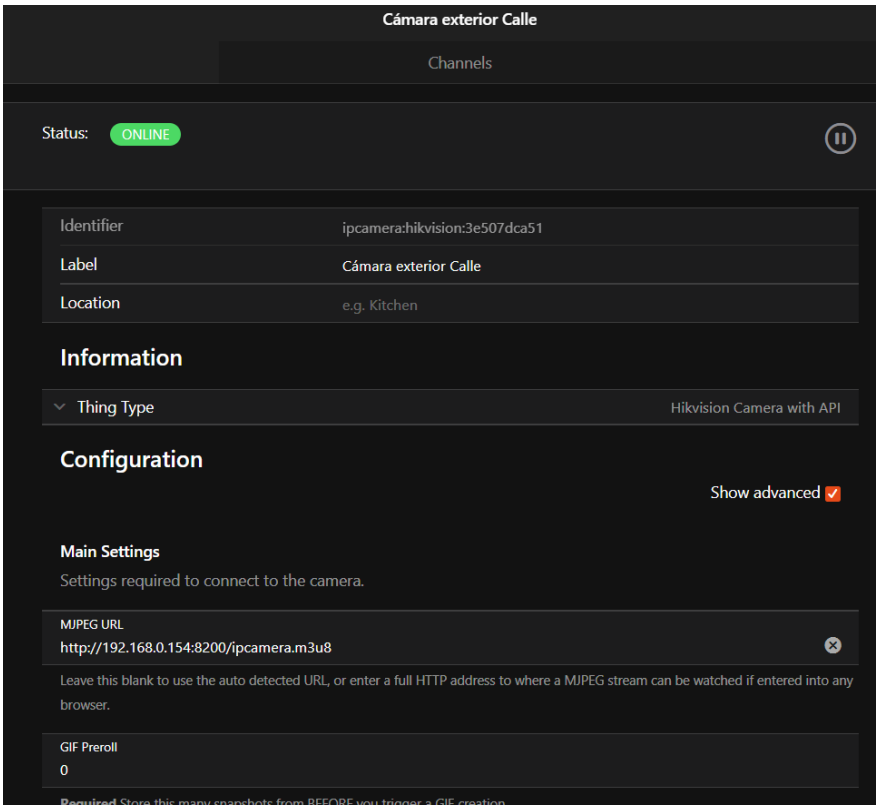



***Figura 2.21 Configuración Things Cámaras.***

## 2.5. **DISEÑO DE INTERFAZ PARA EL USUARIO.**

Una vez integrados los servicios que van a ser utilizados en la plataforma es necesario crear un entorno donde se consiga interactuar con ellos en un solo sitio. En la plataforma esto se consigue creando un archivo llamado sitemap que llevará consigo la programación que será capaz de ordenar por botones, graficas, áreas o lo que desee el usuario como se puede observar en la Figura 2.22. Para el código descrito en la Figura 2.22 se visualiza el siguiente resultado donde existe una interfaz separada por actuadores y lugares a los cuales pertenecen, como se ve en la Figura 2.23. El archivo sitemap fue creado en la siguiente dirección *"/etc/openhab/sitemaps/Tesis.sitemap"*.

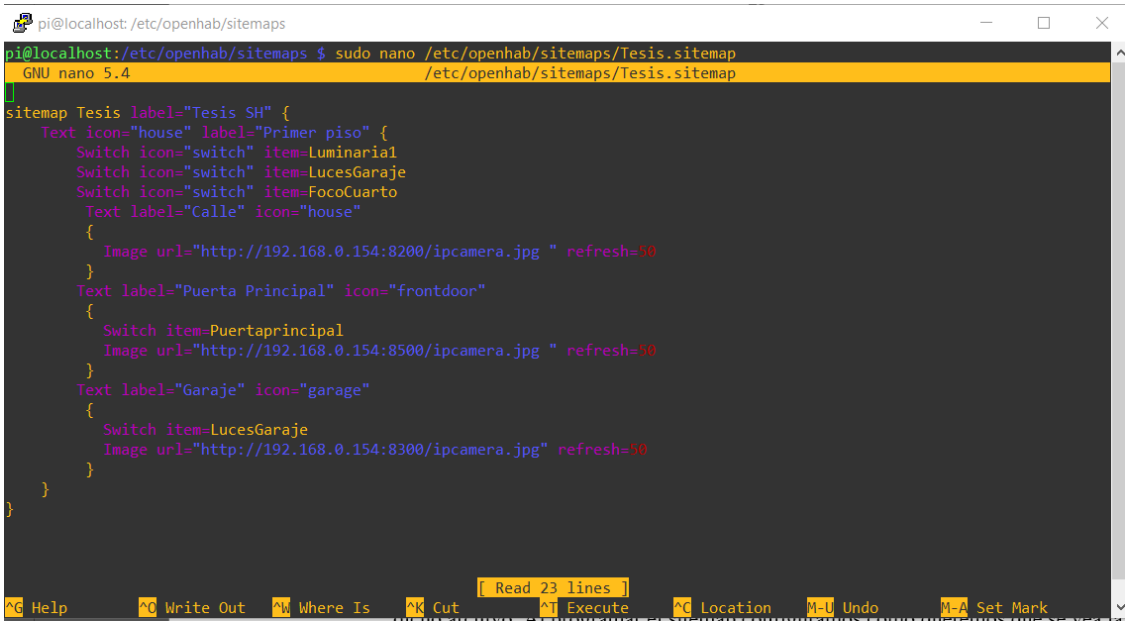



*Figura 2.22 Programación de sitemap.*

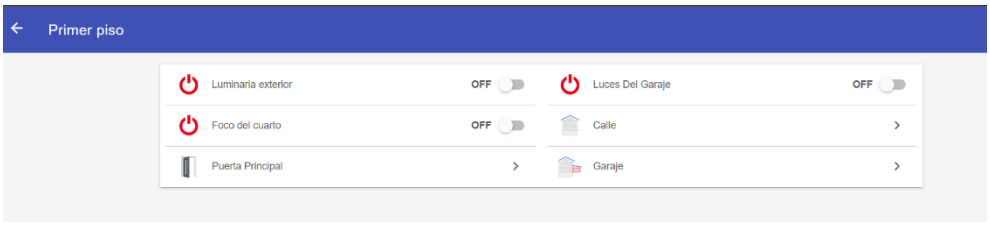



***Figura 2.23 Vista de actuadores en HTTP.***

Para visualizar los cambios realizados desde el teléfono móvil en la aplicación Openhab se ingresa con la cuenta creada, posteriormente aparecerá en pantalla lo programado desde el sitemap como se muestra en la Figura 2.24. Es preciso mencionar que los cambios que realicemos, tienen efecto de igual manera en la aplicación móvil

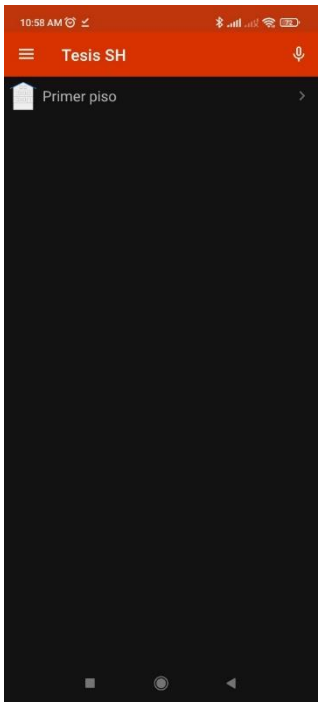



***Figura 2.24. Aplicación móvil Openhab.***

Como se aprecia en la Figura 2.25, los botones configurados en la plataforma se acoplan perfectamente al tamaño de la pantalla del teléfono móvil, se observa también el estado de los actuadores que, en el caso de cambiar, se verá reflejado en la parte de ON (color verde del botón) y OFF (color rojo del botón). En la imagen Figura 2.26, se puede apreciar la división por secciones de la casa. Al ingresar en el apartado de puerta principal se observa la cámara instalada en el punto similar a un video portero y también el actuador para poder abrir la puerta; con esto se consigue tener la función de un portero eléctrico sin la necesidad de abrir otras aplicaciones.

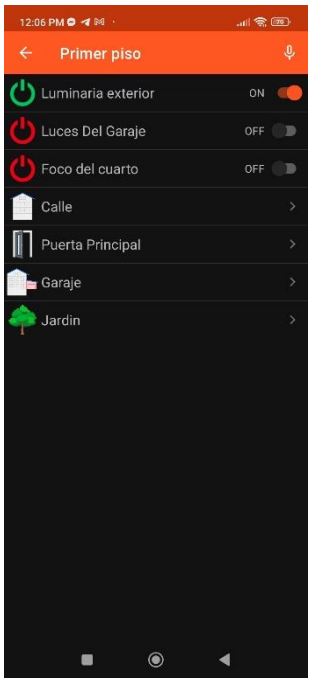



***Figura 2.25. Vista de actuadores y habitaciones en aplicación móvil.***


***Figura 2.26. Vista de cámara de puerta principal en aplicación móvil.***

Si se observa la Figura 2.27 en el apartado de Garaje, la cámara que pertenece al garaje tiene el botón que enciende la luminaria del lugar, cabe recalcar que es posible

colocar muchos más actuadores o sensores dependiendo de las necesidades de los usuarios.


***Figura 2.27. Vista de cámara garaje en aplicación móvil.***

## 2.6. **INTEGRACIÓN DE ASISTENTES VIRTUALES.**

Culminado el registro de los actuadores, sensores y cámaras, se vincula la cuenta de Openhab con *Google Assitant* para poder tener acceso a los dispositivos mediante comandos de voz. Para esto, se inicializa la aplicación de Google Home, en el apartado de *configuración>Servicios>Funciona con Google*, en Openhab, aquí se agrega la cuenta registrada de la plataforma, si toda la configuración es correcta Google Home reconocerá automáticamente los dispositivos y servicios disponibles; en este apartado pueden ser separados por áreas; en la Figura 2.28 se observan diversos ambientes como *dormitorio, entrada y jardín.*


***Figura 2.28 Visualización de separación de áreas en la aplicación.***

Ya realizada la vinculación de la cuenta de Openhab con la de *Google Home* puede hacerse uso de los comandos de voz mediante el asistente de *Google*, diciendo las palabras "*Ok Google*" los distintos actuadores se activan según el nombre asignado en la plataforma, en los ítems y *sitemap*. Por ejemplo, si el comando de voz es "Ok Google enciende la luz del garaje" este reconocerá el comando encender (ON) y encenderá, en este caso, la luz del garaje comandando el actuador vinculado con esta luminaria. De esta forma, se tiene acceso a todos los actuadores por separado o en conjunto en el caso de decir "apaga todas las luces de la casa" o "apaga todas las luces de la sala".

Los comandos de voz pueden ser activados desde el teléfono, Google home o Google mini. En el caso de disponer un dispositivo *Alexa* el proceso es similar solo es necesario vincular la cuenta de *Alexa* con Openhab.

A continuación, el sistema está listo para poder responder y funcionar de cualquier manera, ya sea físicamente encendiendo o apagando una luminaria como regularmente se hace, mediante un computador desde la plataforma web, a través de un teléfono celular con la aplicación móvil o mediante comandos de voz con la ayuda de cualquier asistente de voz configurado. Esto funciona de la siguiente manera, la plataforma recibe el comando mediante los botones o comandos de voz esta la procesa y actúa enviando los comandos al dispositivo que corresponda, en cuanto a los sensores estos están recibiendo información todo el tiempo enviando esos datos a la plataforma en el caso de requerir una acción según los valores de los sensores, se envía la orden desde la plataforma al correspondiente dispositivo.

Toda la información recibida por el $broker$ será almacenada en la nube local de OpenHab (ver Figura 2.29) y también en la nube online de *MongoDB* (ver Figura 2.30), de esta manera podemos consultar el valor de los sensores ya sea por día, semana, mes o año, estos datos pueden ser representados mediante gráficas.

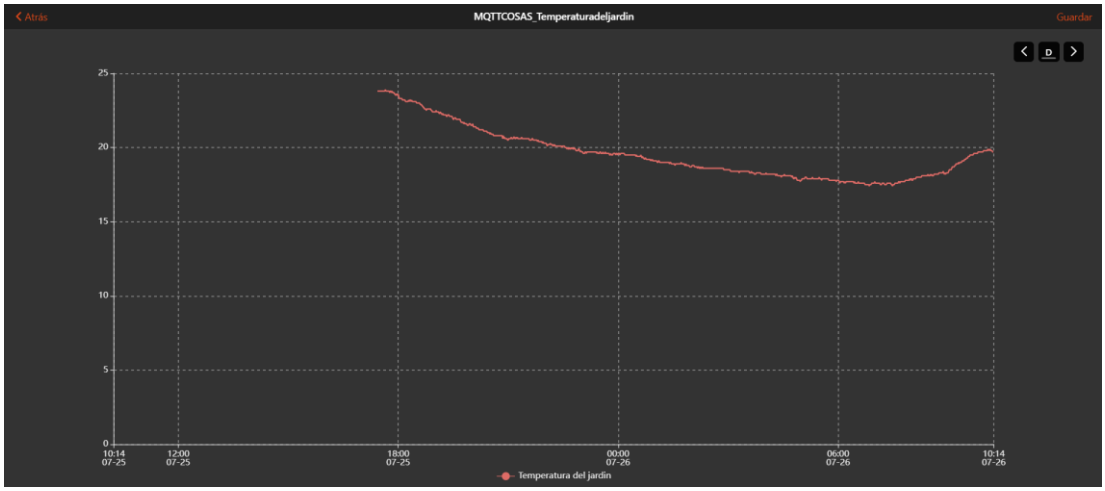



***Figura 2.29 Gráfica sensor de temperatura.***

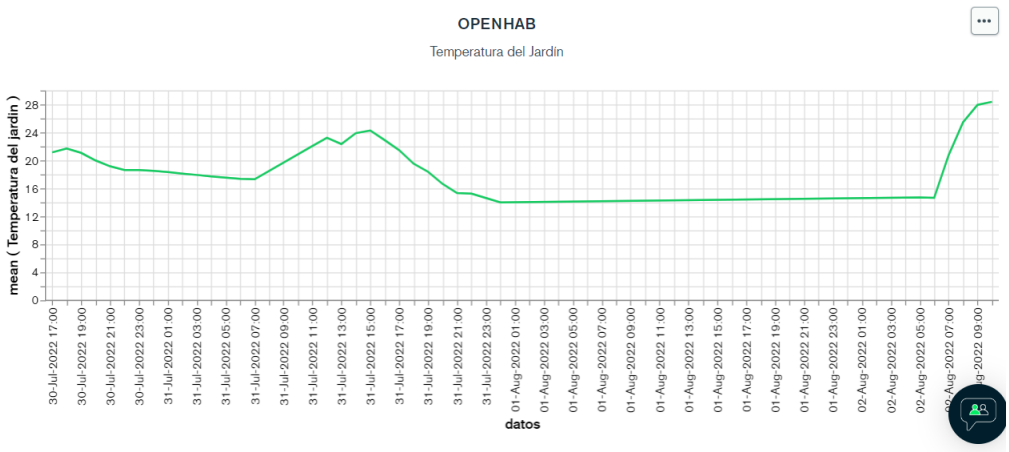



***Figura 2.30 Gráfica de sensor de temperatura en MongoDB.***

El almacenamiento de información en la nube brinda datos importantes al realizar un análisis de comportamientos y rutinas de los valores entregados por los sensores y actuadores, además es importante mencionar que es posible acceder a estos datos de forma remota, haciendo énfasis nuevamente en que no es necesario estar dentro de la red local para observar los datos entregados por el sistema domótico.

Finalmente se ha realizado la vinculación de los sistemas de cámaras de seguridad, actuadores encargados de tener el control de las luminarias y sensores que recogen información, en una sola plataforma que además de darnos distintas opciones para el acceso a estos, guarda los datos en la nube donde es posible observar el comportamiento de cada uno de estos.

# CAPÍTULO III:

## 3. IMPLEMENTACIÓN Y ANÁLISIS DE RESULTADOS

### 3.1. INSTALACIÓN DEL SISTEMA

La implementación del sistema se ha realizado en un hogar de clase media con la siguiente distribución, en la Figura 3.1, se puede apreciar las áreas que van a ser monitoreadas o controladas; son las siguientes:

- Garaje.
- Puerta principal.
- Dormitorio.
- Jardín.

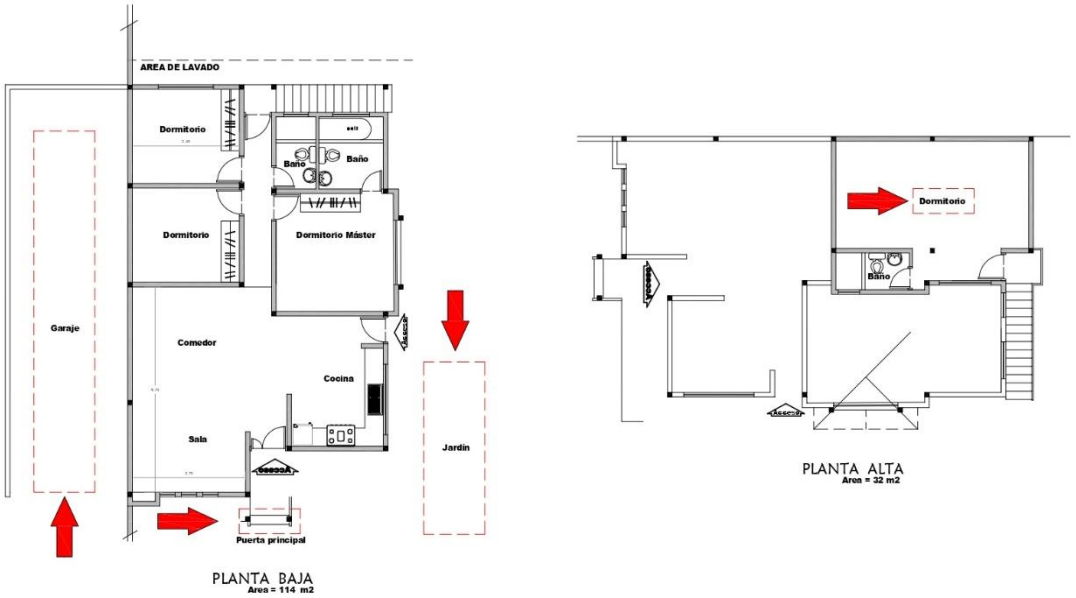



***Figura 3.1 Zonas monitoreadas y/o controladas.***

Los dispositivos a ser controlados son los siguientes:

- 3 luminarias.
- 1 chapa eléctrica.
- 1 sensor de temperatura.

- 1 sensor de humedad de suelo, humedad del aire, índice de calor.
- 1 electroválvula
- 4 cámaras de seguridad.

Una vez realizada la instalación de los dispositivos en las zonas indicadas, se prosiguió con la disposición de las cámaras de seguridad (ver Figura 3.2), desde ellas es posible notar los cambios en los actuadores a controlar, con los que posteriormente se configura la plataforma definiéndolo en función de las zonas establecidas.

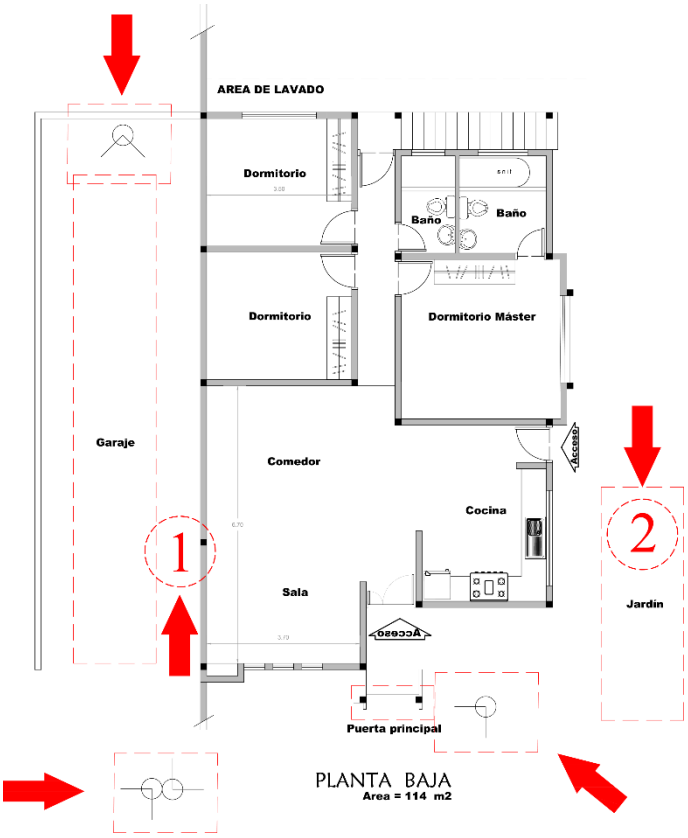



***Figura 3.2 Posición de cámaras de seguridad y NodeMcu´s.***

Los dispositivos NodeMcu van a ser instalados en los círculos rojos que se muestra en la Figura 3.2, el primero, indicado como número 1 en la Figura 3.2, va a realizar el control de:

- Luces del garaje.
- Chapa de la puerta principal.
- Control de la luminaria exterior.

El segundo NodeMcu va a sensar los datos requeridos en el jardín del hogar y controlar la electroválvula que servirá para el riego, el tercero se instaló en la planta alta como se muestra en la Figura 3.3, éste va a controlar una luminaria. Todos los nodeMcu´s han sido colocados dentro del rango de cobertura del Router principal que se puede observar en la Figura 3.3.

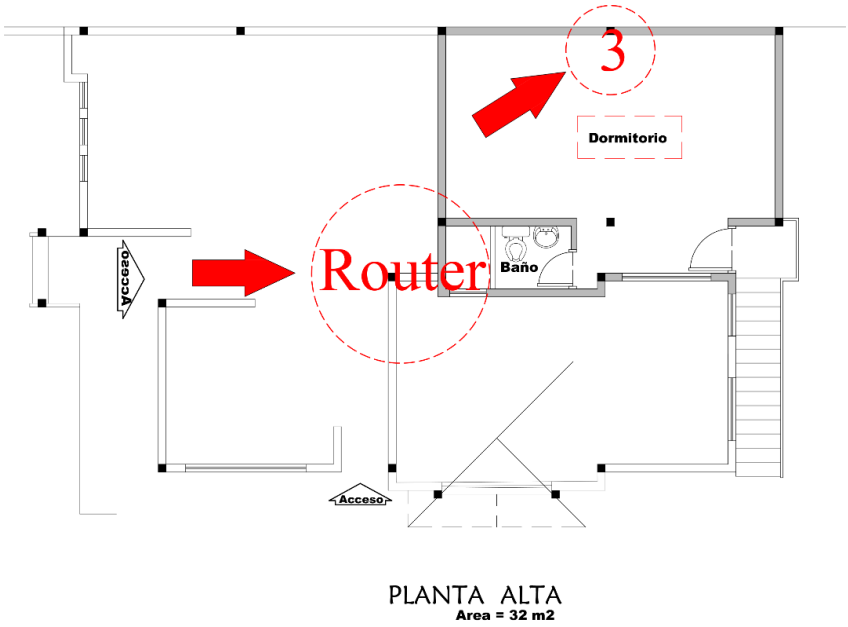



***Figura 3.3 Posición NodeMcu y Router.***

En la Figura 3.4 se puede apreciar la conexión del primer NodeMcu donde se elaboró el sistema de reconocimiento de fase para dos puertos, en este caso se censará los focos del garaje y la luminaria exterior, el módulo de relé tendrá la función de switch de las funciones ya mencionadas.

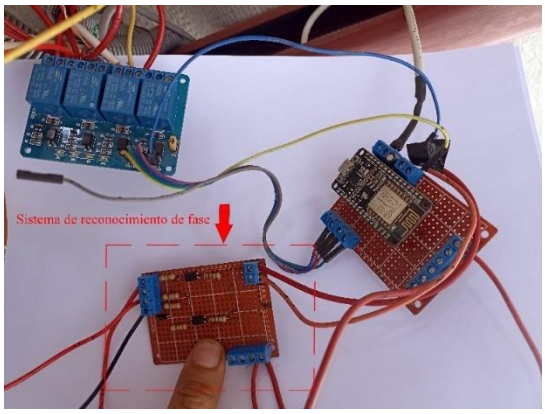



***Figura 3.4 Conexión física de NodeMcu.***

***Figura 3.5 NodeMcu ubicado en el jardín.***

El segundo NodeMcu está ubicado en la jardinera como se muestra en la Figura 3.5, éste recibe los datos de los sensores de humedad del suelo, humedad del aire, temperatura e índice de calor; además, de activar una electroválvula cuando las condiciones establecidas en el programa se cumplen.

Una vez instalados todos los dispositivos, sensores y actuadores se procede a realizar las pruebas de funcionamiento y del tráfico generado en la rede por todo el sistema.

## 3.2. ANÁLISIS DEL TRÁFICO GENERADO POR EL SISTEMA DOMÓTICO

En una red MQTT todo el tráfico tiene que necesariamente pasar por el broker para que este procese los datos y, en base a éstos, envíe mensajes de comunicación a los dispositivos vinculados (recordar Capítulo 2, sección 1.3.3). El sistema funciona

con una topología tipo estrella, siendo el centro de todo el $broker$ que, en el sistema domótico, es la Raspberry pi. El $broker$ además de procesar todos los datos envía mensajes por broadcast para todos los dispositivos nuevos que necesiten vincularse con él; además, envía mensajes de $keep\ \ alive$ de parte de todos los dispositivos vinculados a la red, éstos generan tráfico que, aunque son mensajes con una cantidad mínima de Bytes a gran escala, llegan a ser un problema debido a la inundación de solicitudes e información transmitida por la cantidad de dispositivos conectados. Estos problemas se presentan cuando la red es demasiado extensa y existe un solo medio de acceso al broker. En la mayoría de los casos los dispositivos se comunican vía Wifi y, en un entorno común, la red Wifi del hogar casi siempre va a ser utilizada por las personas que ahí viven. Normalmente, cada persona tiene al menos un dispositivo conectado a la red; por ejemplo, en una reunión familiar el número de dispositivos conectados se eleva considerablemente, ocasionando que la respuesta de los dispositivos MQTT se ralentice.

En otras palabras, el tiempo de respuesta de los dispositivos IoT está directamente relacionada con la calidad de conexión que exista en la red local. Este tipo de inconvenientes no son muy relevantes en estas redes, donde el número de personas promedio conectadas a la red es insignificante, pues se comprende de 4 habitantes por familia, según el INEC [38]. Cabe recalcar que es posible dar prioridad a la entrega de mensajes mediante la configuración del nivel de QoS definido entre los dispositivos y el $broker$, como previamente se explicó en el capítulo 1 en el apartado de MQTT (sección 1.3.1).

El sistema en reposo envía constantemente datos de los sensores instalados y también mensajes que utiliza para tener la certeza de que los dispositivos aún están vinculados, estos mensajes son más conocidos como mensajes $keep\ \ alive$. Al realizar un análisis del tráfico de red, se observa en la Figura 3.6, la cantidad de bytes ocupados en el medio de transmisión, los 95 bytes/s corresponden a los datos de los

sensores; mientras que, los 4 bytes/s de cada dispositivo son los mensajes de $Keep\ alive$ que el sistema envía automáticamente como parte del protocolo MQTT. En la Figura 3.7 se puede observar el consumo del ancho de banda en un determinado tiempo y el puerto que utiliza.

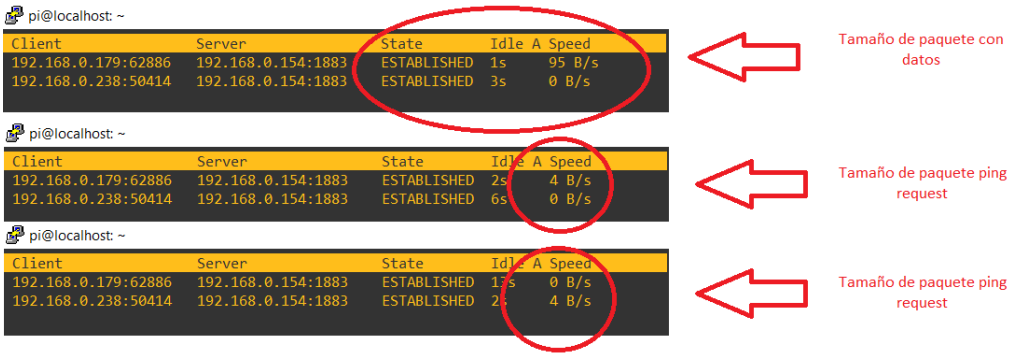



***Figura 3.6 Tamaño de mensajes publicados y Keep alive.***

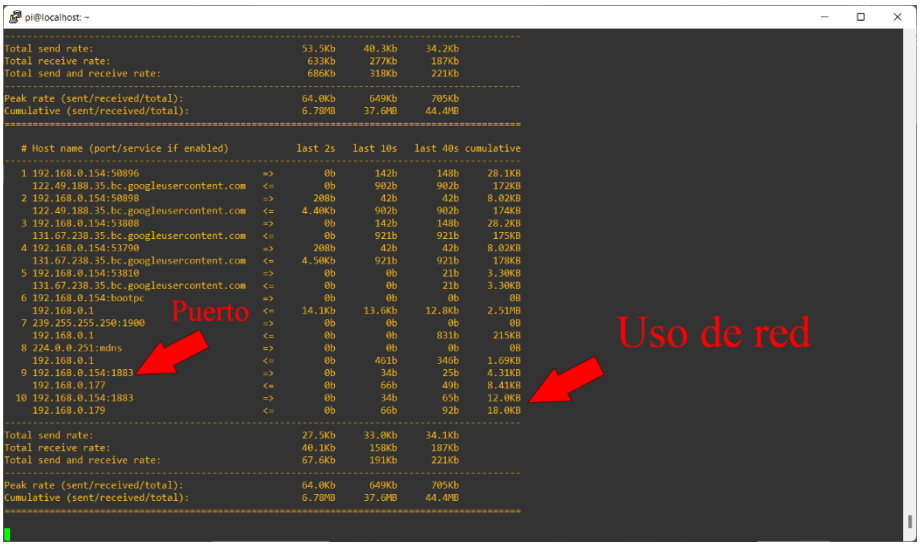



***Figura 3.7 Uso de red MQTT.***

Con la ayuda del programa de análisis de paquetes de tráfico *wireshark* que se puede observar en la Figura 3.8, el tamaño que tienen los mensajes *keep alive* que son el Ping request y el Ping response, cada uno de ellos pesan 2 bytes. En la Figura 3.9 se muestran los datos de los *Topics* correspondientes a los datos de los sensores; así como su tamaño.

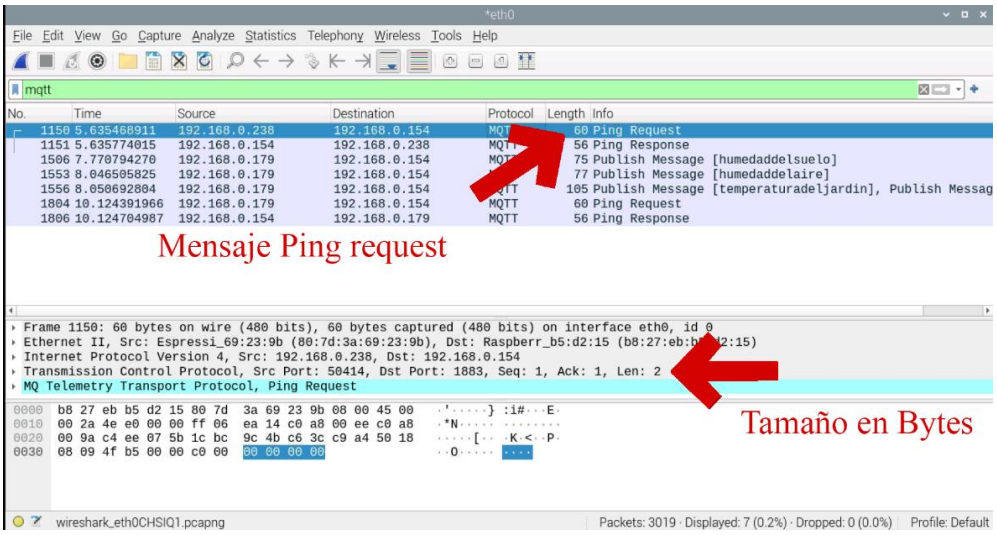



***Figura 3.8 Tamaño de mensajes keep alive.***

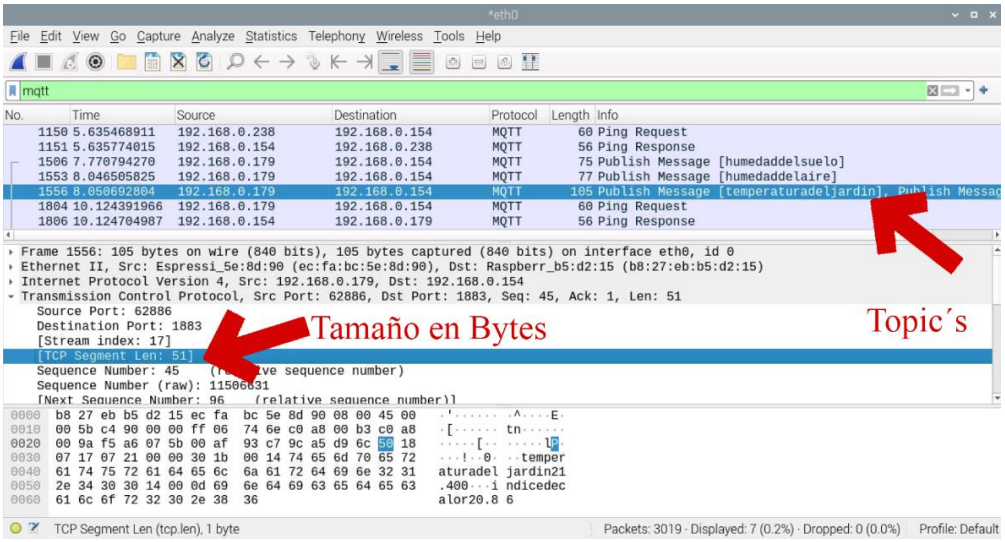



***Figura 3.9 Mensaje Publish mesasage.***

La plataforma en reposo ocupa 4 bytes por cada dispositivo vinculado, en el sistema domótico que hemos instalado, son 3 NodeMcu, por lo que cada 15 segundos se enviará mensajes de $keep\ \ alive$ (por partes del $broker$) y su respuesta $ping\ \ response$ (por parte del nodo). Estos mensajes representan 4 bytes de carga en el medio de transmisión, la longitud del mensaje con los datos de los sensores tiene una carga total de 95 bytes, el censado tiene una frecuencia de 15 segundos; por lo tanto, la suma de estos nos da como resultado 4+4+4+95=107 bytes, cada 15 segundos. Al calcular este valor mediante la Tabla 3.1, se puede observar que el

consumo de cada hora, diario y mensual del sistema en estado de reposo, no representan un gran consumo en el uso del medio de transmisión.

***Tabla 3.1 Consumo de ancho de banda (sistema en reposo).***

| Periodo de tiempo | Unidad |
|---|---|
| Hora | 12840 bytes |
| Diario | 0,308160 MB |
| Mensual | 9,2448 MB |

Para realizar la estimación del consumo de ancho de banda, se considera el servicio que se ocupe, debido que no es igual encender una luminaria que observar las imágenes de una o múltiples cámaras.

Para conseguir observar el consumo de ancho de banda debido a la carga de los datos hacia la base de datos *MongoDB*, se ingresó a su página online y se obtuvo la gráfica de la tasa promedio de bytes ocupados (ver Figura 3.10), se muestra que ocupa 5.82 bytes/s.


***Figura 3.10 Tasa promedio de bytes de subida.***

La base de datos ocupa un espacio de almacenamiento para guardar la información ingresada, para observar el uso se ha obtenido la muestra de 6 horas del sistema en funcionamiento como se muestra en la Figura 3.11, entregando un total de

52.88MB - 52.59MB = 0.29MB, los 52.59MB son los megas ocupados antes de iniciar la medición y el valor de 52.88MB son los megas luego de transcurrir las 6 horas de funcionamiento del sistema, con estos valores es posible aproximar el consumo de cada hora, diario y mensual de espacio de la base de datos como se muestra en la Tabla 3.2.

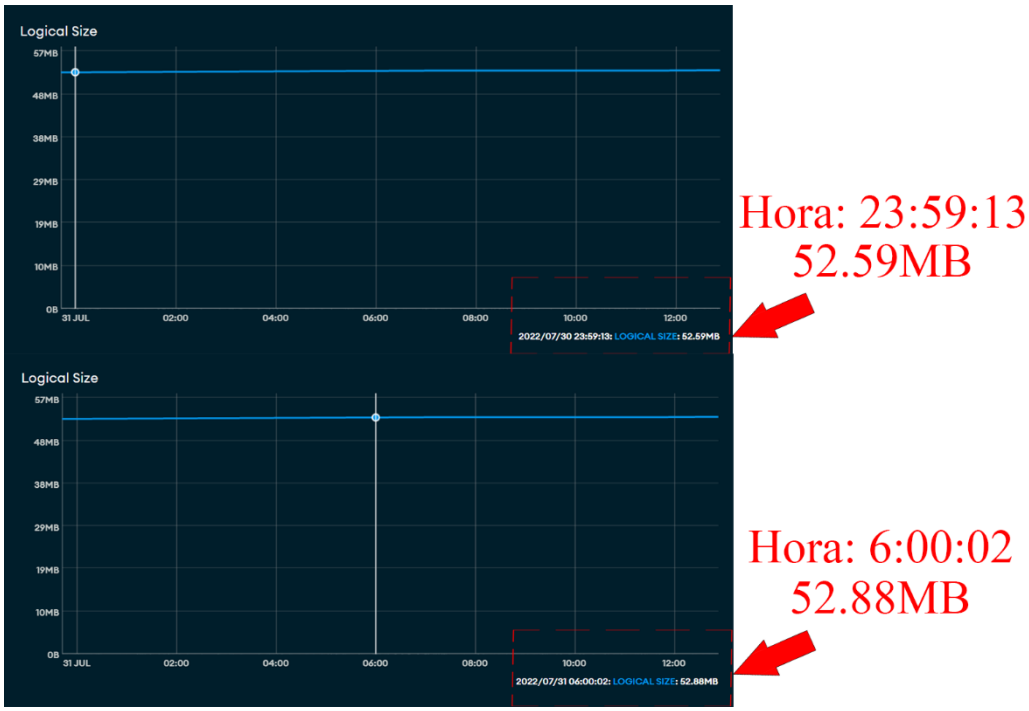



***Figura 3.11 Consumo de espacio de almacenamiento en base de datos online.***

***Tabla 3.2 Cálculo de consumo de espacio de almacenamiento de base de datos.***

| Periodo de tiempo | Unidad |
|---|---|
| Hora | 0,0483 MB |
| Diario | 1,16 MB |
| Mensual | 34,8 MB |

Cabe mencionar que la base de datos tiene un espacio de almacenamiento gratis de 512 MB siendo capaz de guardar datos de aproximadamente 14,71 meses sin costo alguno; además, es posible borrar los datos que ya no necesitemos que sigan almacenados en la nube.

Para el análisis del uso de ancho de banda local se ha ingresado a observar una cámara de seguridad, con la ayuda de analizador de tráfico de red llamado *Bmon*, con el que se obtiene una lectura de Rx:436 kB/s Tx:454.42 kB/s como se muestra en la Figura 3.12.

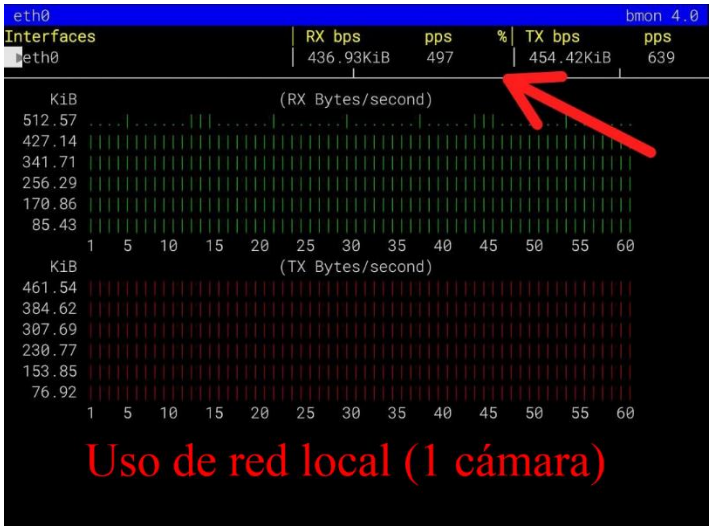



***Figura 3.12 Uso de red local al observar una cámara.***

Cuando se ingresa a la plataforma mediante la aplicación del celular, el uso de la red ocupa menos datos, a diferencia que cuando se lo hace de manera local, teniendo un tasa de tranferencia de Rx:53.82 kB/s y Tx: 52.83 kB/s como se muestra en la Figura 3.13. Esto representa aproximadamente 9 veces menos, esto se debe a que la plataforma realiza un muestreo de las imágenes que se pueden ver de las cámaras para reducir el consumo de ancho de banda de sus servidores.


***Figura 3.13 Uso de red con plataforma online.***

En base a este análisis el uso de la red local depende del ingreso de los usuarios hacia la plataforma y de las interacciones que se tenga con el sistema dentro del hogar. No obstante, el consumo de ancho de banda no es considerable si la aplicación permanece en reposo.

## 3.3. **COSTO DE IMPLEMENTACIÓN EN UN HOGAR DE CLASE MEDIA**

El presente proyecto está pensado para un hogar de clase media o media alta, donde conviven un promedio de 4 personas. Una casa promedio está conformada de dos plantas, en la planta baja una sala, comedor, cocina, patio, baño y, en la planta alta, constituida por un cuarto de padres, dos cuartos de hijos y dos baños. Según las estadísticas del INEN, en función del nivel socioeconómico de los hogares los ingresos económicos de un hogar de clase media alta y media típica en el Ecuador son de entre 212.50 y 1062.50 dólares mensuales por persona [38].

Para realizar una aproximación del costo de implementación de un sistema domótico como el diseñado y desarrollado en el presente trabajo para un hogar de clase media promedio, serán basadas en las encuestas desarrolladas en el año 2011 por el INEC [38]. Estas encuestas demuestran que una casa de clase media en Ecuador se establece como la mostrada en la Figura 3.14.

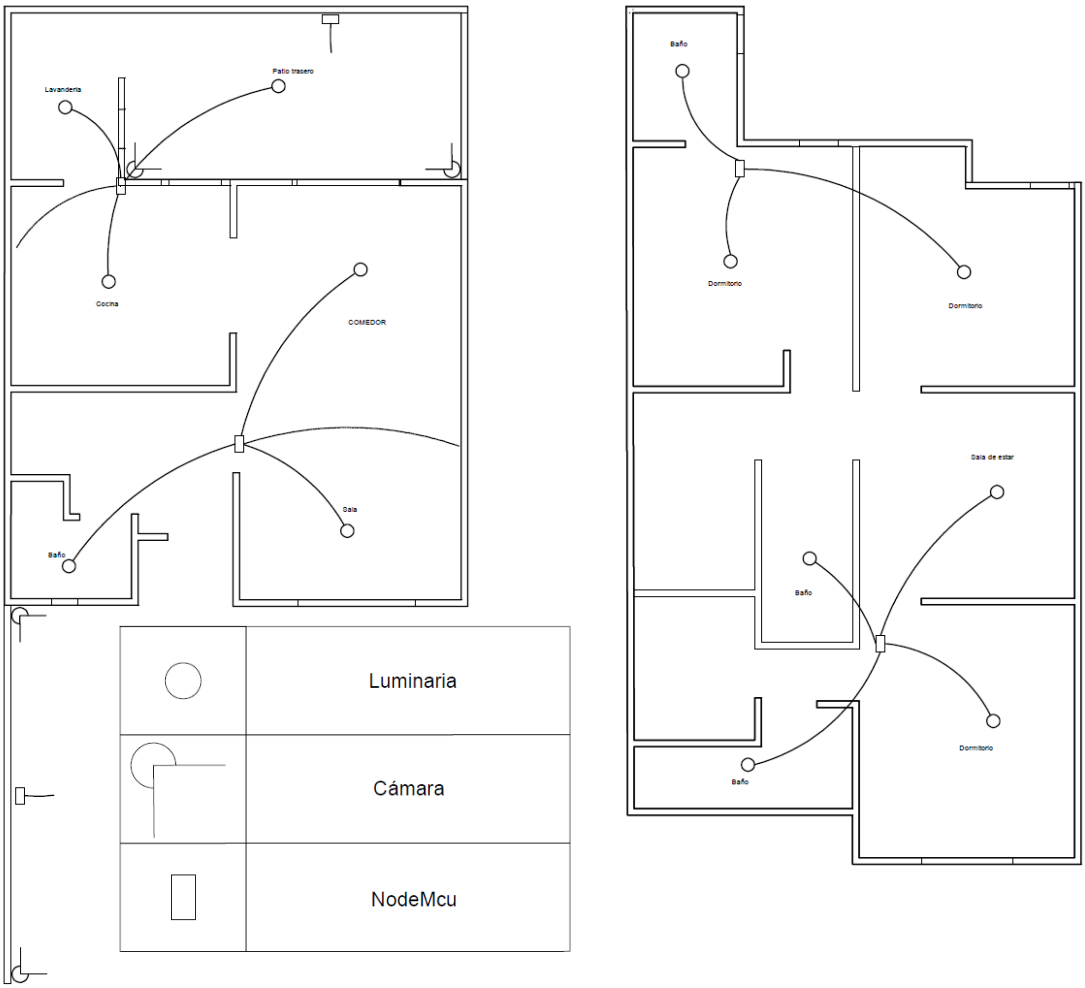



***Figura 3.14 Plano casa media promedio.***

Para la automatización del hogar de clase media promedio se considera los siguientes materiales que se indican en la Tabla 3.3.

***Tabla 3.3 Materiales y dispositivos necesarios para la instalación del sistema domótico en una casa promedio.***

| Descripción | Cantidad | Costo Unitario | TOTAL |
|---|---|---|---|
| diodos Zener 1n4148 | 16 | 0.15 | 2.4 |
| optoacopladores pc817 | 16 | 0.45 | 7.2 |
| resistencias 1m 1/4 w | 16 | 0.03 | 0.48 |
| 4 resistencias de 100k 1/4 w | 16 | 0.03 | 0.48 |

| | | | |
|---|---|---|---|
| resistencias de 47k 1/4w | 16 | 0.03 | 0.48 |
| Diodo 1n4007 | 4 | 0.1 | 0.4 |
| Borneros azules dobles para placa perforada. | 20 | 0.2 | 4 |
| Capacitores cerámicos 1uF | 16 | 0.1 | 1.6 |
| Cap electrolíticos de 0.1uF | 16 | 0.1 | 1.6 |
| Chapa eléctrica | 1 | 45 | 45 |
| Módulo de 4 relés | 5 | 4.4 | 22 |
| Módulo 1 relé | 1 | 2 | 2 |
| Placa perforada 10x7 | 4 | 0.45 | 1.8 |
| Estaño | 1 | 3.5 | 3.5 |
| Pasta | 1 | 3 | 3 |
| Sensor de humedad dht12 | 1 | 6.2 | 6.2 |
| Sensor de humedad de la tierra | 1 | | 0 |
| Sensor magnético | 4 | 2.5 | 10 |
| Kit camaras 8 dvr Hikvision | 1 | 235 | 230 |
| Micro SD de 64 GB | 1 | 12 | 12 |
| NodeMcu | 5 | 8 | 40 |
| Fuentes de alimentación de 5 o banco de condensadores | 6 | 5 | 30 |
| electro válvula | 1 | 10 | 10 |
| módulo Google mini | 1 | 47 | 47 |
| CABLE UTP CAT6 8HILOS (METRO) | 1 | 50 | 50 |
| KIT Raspberry Pi model B | 1 | 250 | 250 |
| TOTAL | | | 755 |

También debemos considerar los costos de instalación y otros gastos para la implementación del sistema domótico en un hogar promedio. Estos costos se presentan en la Tabla 3.4.

*Tabla 3.4 Costo de instalación técnico y ayudantes.*

| Costo de instalación técnico y ayudantes | | | |
|---|---|---|---|
| **Descripción** | **Cantidad** | **Costo/Hora** | **Costo diario** |
| Costo de Técnico | 2 | $3,43 | $54,88 |
| Costo personal de apoyo | 2 | $2,81 | $44,96 |

La instalación de las cámaras de seguridad y las horas de programación de los dispositivos que incluye el ensamblaje de los distintos nodeMcu´s y se pueden contemplar en la Tabla 3.5.

*Tabla 3.5 Costo de programación, ensamble de NodeMcu´s e instalación de cámaras.*

| Costo de programación y ensamble de NodeMcu´s y cámaras de seguridad | | | |
|---|---|---|---|
| **Descripción** | **Cantidad** | **Costo unitario o por Hora** | **Costo total** |
| Costo de programación | 4 | 7,5 X Hora | $30,00 |
| Costo de ensamblaje | 6 | 8 | $48,00 |
| Costo instalación de cámaras | 1 | $120,00 | $120,00 |

El costo de instalación del sistema domótico propuesto con los 6 NodeMcu´s se podrá realizar en 8 horas con un ayudante y un técnico, para mejorar y optimizar el tiempo se propone 2 grupos de trabajo consiguiendo aumentar el número de instalaciones diarias como se observa en la Tabla 3.6.

*Tabla 3.6 Tiempo de instalación.*

| Tiempo de instalación | |
|---|---|
| | Tiempo en horas |
| 1 técnico y 1 ayudante | 16 |

Después de obtener el tiempo y el costo de instalación del sistema domótico se puede calcular total de gastos que representa la implementación de dicho sistema.

$$total\ de\ gasto\ de\ implementación = (3{,}43 + 2{,}81 + 2{,}81) * 16 + 30 + 48 + 120$$

$$total\ de\ gasto\ de\ implementación = 397.68$$

Para el realizar el cálculo total de implementación del sistema domótico propuesto se suma todos los valores de instalación y materiales ocupados en el proceso como se indica en la Tabla 3.7.

***Tabla 3.7 Costo total de implementación.***

| **Costo total** | |
|---|---|
| Instalación | $397,68 |
| Materiales | $936,20 |
| Total | $1.333,88 |

En la inversión inicial se contempla las herramientas, equipos de seguridad, costo de un medio de trasporte y publicidad inicial como se muestra en la Tabla 3.8.

***Tabla 3.8 Inversión inicial.***

| **Inversión inicial** | |
|---|---|
| Equipo de seguridad, herramientas, publicidad. | $5.000,00 |
| Vehículo | $8.900,00 |

La proyección de ventas ha sido basada en estadísticas de las encuestas nacionales de edificaciones, donde indican que para el año del 2019 hubo un decrecimiento de futuras edificaciones debido a la pandemia por lo cual el proyecto será tomado en cuenta para las proyecciones de un año antes. En el 2019 las proyecciones de edificaciones de viviendas son de 4519 viviendas con un promedio de 50 mil dólares de construcción por lo que en su mayoría son casas de clase media [39], tomando estos datos se estima que se instalará al 3% de ellos, siendo eso 4519*0.03= 135 viviendas. Cabe recalcar que el sistema no solamente es para viviendas nuevas, por lo que el mercado objetivo es mucho más viable considerando que el número de viviendas en la cuidad es de 174573 según el censo realizado por el INEC en el año 2010 [40].

El costo de venta del sistema domótico es el 24% del costo de los materiales originalmente dando como resultado $755 * 0.24 = 181.20$ a esto se suma el costo de instalación sin considerar los materiales como se aprecia en la Tabla 3.9.

***Tabla 3.9 Ingreso de venta de sistemas domóticos.***

| Ingreso de venta de sistemas domóticos | | | | | | |
|---|---|---|---|---|---|---|
| **Descripción** | **Cantidad** | **Costo** | **Diario** | **Ventas Mensuales** | **Ingreso Mensual** | **Anual** |
| Costo de instalación | 1 | $397,68 | 0,50 | 10,00 | $3.976,80 | $47.721,60 |
| Venta de Sistema domótico | 1 | $181,20 | 0,50 | 10,00 | $1.812,00 | $21.744,00 |
| | | | | | Total | $69.465,60 |

Los egresos de salarios a los funcionarios que conformarán el grupo de trabajo se encuentran detallado en la Tabla 3.10, donde se asigna un ingeniero electrónico a cargo de la programación y supervisión de ensamble de dispositivos, así también la contratación de tres personas en el área de ventas que serán encargadas de promocionar el producto y cerrar contratos, para el área de campo se asigna dos grupos de trabajo un técnico y un ayudante para cada grupo.

***Tabla 3.10 Egresos de funcionarios.***

| Egresos de funcionarios | | | | |
|---|---|---|---|---|
| **Descripción** | **Cantidad** | **Cantidad de funcionarios** | **Remuneración mensual** | **Remuneración anual** |
| Ingeniero electrónico | $1.200,00 | 1 | $1.200,00 | $14.400,00 |
| Décimo cuarto | $425,00 | | | $425,00 |
| Décimo tercero | $1.200,00 | | | $1.200,00 |
| Fondos de reserva 8,33% | $99,96 | | $99,96 | $1.199,52 |
| Vendedora | $525,00 | 3 | $1.575,00 | $18.900,00 |
| Décimo cuarto | $425,00 | | | $1.275,00 |
| Décimo tercero | $525,00 | | | $1.575,00 |
| Fondos de reserva 8,33% | $43,73 | | $131,20 | $1.574,37 |
| Técnico | $600,00 | 2 | $1.200,00 | $600,00 |
| Décimo cuarto | $425,00 | | | $425,00 |
| Décimo tercero | $600,00 | | | $1.200,00 |
| Fondos de reserva 8,33% | $49,98 | | $99,96 | $1.199,52 |
| Ayudante | $450,00 | 2 | $900,00 | $10.800,00 |
| Décimo cuarto | $425,00 | | | $850,00 |

| | | | | |
|---|---|---|---|---|
| Décimo tercero | $450,00 | | | $900,00 |
| Fondos de reserva 8,33% | $37,49 | | $74,97 | $899,64 |
| | | | Total | $57.423,05 |

Se tiene que considerar también el costo de un local comercial donde los clientes podrán acercarse a solicitar instalaciones o cualquier tipo de soporte, en la Tabla 3.11 se puede observar el rubro del costo de arrendamiento considerado ya los gastos por concepto de servicios básicos.

***Tabla 3.11 Costo de arriendo de local comercial.***

| **Arriendo de local** | | | |
|---|---|---|---|
| **Descripción** | **Cantidad** | **Mensual** | **Anual** |
| Local comercial | 1 | $600,00 | $7.200,00 |

El cálculo de los egresos totales se la realiza sumando todos los egresos tanto los de pagos de salarios como los de arriendo de local dando un total de $64.623,05 como se puede observar en la Tabla 3.12.

***Tabla 3.12 Total de egresos***

| **Total de egresos** | |
|---|---|
| Egresos de funcionarios | $57.423,05 |
| Arriendo de local | $7.200,00 |
| Total | $64.623,05 |

En la Tabla 3.13 se observa la ganancia con la venta del sistema.

***Tabla 3.13 Ganancia o utilidad.***

| **Ganancia o utilidad** | |
|---|---|
| Total, ingresos | $69.465,60 |
| Total, egresos | $64.623,05 |
| Ganancia Total | $4.842,55 |

### 3.3.1. Cálculo de VAN y TIR

El valor actual neto VAN y la tasa interna de retorno TIR nos indican la rentabilidad del proyecto y el tiempo que tardará en recuperar la inversión inicial.

La depreciación del proyecto en un plazo de 5 años representa toda la inversión realizada esto con el objetivo de renovar todas las herramientas y el vehículo ocupado para las actividades como se puede observar en la Tabla 3.14.

***Tabla 3.14 Cálculo de depreciación en 5 años.***

| | |
|---|---|
| Ingresos | $69.465,60 |
| Egresos | $64.623,05 |
| Inversión | $15.233,88 |
| Vida útil | 5 |
| Valor residual | $3.046,78 |
| Depreciación | $15.233,88 |

Para calcular el valor actual neto del proyecto obtenemos el flujo neto de dinero en base a los ingresos y egresos se obtiene el flujo de caja.

Los valores calculados en la Tabla 3.15 muestra los valores de ingresos y egresos en un periodo de 5 años, el crecimiento financiero se proyecta en un 5% anual gracias al creciente interés en el mercado IoT, los egresos también se aumentan en un 3.5% ocasionado por el incremento anual del salario básico unificado en el país.

***Tabla 3.15 Ingresos, egresos y flujo neto en 5 años.***

| Periodo | Ingresos | Egresos | Flujo neto |
|---|---|---|---|
| 0 | 0 | -$15.233,88 | -$15.233,88 |
| 1 | $69.465,60 | -$64.623,05 | $4.842,55 |
| 2 | $76.412,16 | -$66.884,86 | $9.527,30 |
| 3 | $84.053,38 | -$69.225,83 | $14.827,55 |
| 4 | $92.458,71 | -$71.648,73 | $20.809,98 |
| 5 | $101.704,58 | -$74.156,44 | $27.548,15 |

La tasa de rentabilidad propuesta es del 16% anual, tasa referencial en función al crédito de consumo, este valor será el parámetro fundamental para calcular el VAN y TIR, como se puede observar en la Tabla 3.16.

***Tabla 3.16 Periodo de recuperación de capital.***

| | |
|---|---|
| Tasa anual | 16% |
| VAN | $30.129,65 |
| TIR | 63,33% |
| Periodo de recuperación | 0,82 |

El periodo de recuperación de capital que se puede observar en la Tabla 3.16, en la que se indica que en menos de un año se recupera la inversión inicial y al cabo de 5 años se obtiene una utilidad de 30 mil dólares, recordando el crecimiento de ingresos anual del 5%. Estos datos nos indica que el proyecto viable.

# CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

OpenHAB llega a ser un software que ofrece muchas ventajas frente a otras plataformas IoT existentes, gracias a su administración remota mediante el dominio *myopenhab.org*. Permite a los usuarios acceder a su servidor y la interfaz web a través de servicios en la nube desde cualquier lugar, también admite la integración con Google Home, Amazon Alexa e IFTTT. Gracias a esta considerable ventaja no requiere que los usuarios abran puertos en su conexión a internet domiciliaria, esto puede conducir a una violación de seguridad de carácter crítico.

Luego de haber diseñado, configurado e implementado el sistema domótico propuesto en el presente trabajo, se concluye que es posible construir un sistema inteligente para un hogar de bajo coste, integrando dispositivos de varios fabricantes. Esto reduce el costo de implementación de sistemas IoT en los hogares, y a su vez tiene las mismas prestaciones de un sistema avanzado de cualquier empresa comercial.

Se pudo evidenciar que el protocolo MQTT es totalmente funcional para este tipo de proyectos, debido a su mayor eficiencia si lo comparamos con otros protocolos como TCP/IP; además de que utiliza un bajo ancho de banda.

Con respecto al análisis de paquetes de datos transmitidos, es importante destacar también que el protocolo MQTT es muy sensible y vulnerable en el caso de no activar la seguridad TLS, debido a la facilidad de obtener las credenciales como el nombre y contraseña del broker; así como también, los nombres de los *Topic*. Estos datos son muy importantes por lo que se ve comprometida la seguridad a nivel local en el caso de que una persona mal intencionada, con el conocimiento suficiente, llegue a escanear la red local; y de esa manera obtener las credenciales o a su vez también colocarse en medio de la conexión y enviar sentencias falsas consiguiendo así el

control del hogar. En virtud de lo mencionado, es recomendable configurar la seguridad TLS que este protocolo ofrece.

Se recomienda conectar el servidor Raspberry PI a la red local mediante cable ethernet para mejorar el medio de transmisión; debido al uso de varios servicios como el de las cámaras, se puede ver afectado el rendimiento de la red provocados por cuellos de botella que se pueden generar en una red inalámbrica deficiente.

El uso de las notificaciones push son ideales a la hora de requerir estar pendientes de algún sensor o servicio que se tenga en la vivienda. Es importante conocer que es posible personalizarlos con condiciones especiales que coloquemos en el apartado *reglas* o en los *Scripts* creados para hacer más amistosa la interacción con el sistema domótico.

La ventaja que representa combinar las tecnologías existentes abre una amplia gama de posibilidades para crear nuevos productos en base a otros, aprovechando sus características y adaptándola a nuestras necesidades. En el presente trabajo se creó un video portero con la ayuda de cámaras de seguridad y un nodeMcu, así como también la automatización de riego del jardín. También podrían crearse incluso productos que no existen pero que los necesitamos en nuestros hogares, de esta manera se abre una un amplio abanico de posibilidades para la creación y desarrollo de nuevas aplicaciones para el hogar.

IoT crea un nicho de mercado explotable en varias áreas, por ejemplo, en el área de investigación que es la domótica, sobre todo si consideramos que el mercado actual no está muy explotado en este campo. Gracias al desarrollo de la tecnología es posible implementar nuevos sistemas que se adapten a las necesidades de los usuarios, reduciendo los costos que demandaría al cubrir todos sus requerimientos con productos de empresas reconocidas en el ámbito comercial, esto representa una ventaja competitiva en el mercado; además, en base al análisis del TIR y VAN se evidencia de que el proyecto es rentable, donde el capital es recuperable en menos de un año.

Finalmente, si consideramos la situación económica del país, situados desde la clase media, el costearse un sistema domótico avanzado puede tornarse imposible, debido a su alto costo que éste representa. Por lo que, la implementación de un sistema como el propuesto en el presente trabajo, viene a ser muy atractivo para los usuarios que requieren cubrir la necesidad de irse actualizando con las comodidades que se pueden llegar a tener actualmente. Esto no solo mejora su calidad de vida si no también la seguridad y confianza en los hogares, por lo que estos tipos de sistemas ya no son un lujo si no que se están convirtiendo en una necesidad y en un beneficio para el propietario de un hogar. En este sentido, pensamos que la propuesta del sistema presentado en este trabajo puede llegar a democratizar el acceso a poseer un hogar automatizado e inteligente por parte de los hogares de clase media. Además, conociendo que nuevos productos más baratos de IoT constantemente están saliendo al mercado, se allana el camino hacia la masificación de este tipo de sistemas domóticos.

# REFERENCIAS BIBLIOGRÁFICAS


[1] V. Miori y D. Russo, “Domotic Evolution towards the IoT”, en *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, BC, Canada, may 2014, pp. 809–814. doi: 10.1109/WAINA.2014.128.

[2] S. Madakam, R. Ramaswamy, y S. Tripathi, “Internet of Things (IoT): A Literature Review”, *J. Comput. Commun.*, vol. 03, núm. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, y M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”, *IEEE Commun. Surv. Tutor.*, vol. 17, núm. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[4] F. Yalçinkaya, H. AydiLek, M. Y. Erten, y N. İNanç, “IoT based Smart Home Testbed using MQTT Communication Protocol”, *Uluslar. Muhendislik Arastirma Ve Gelistirme Derg.*, p. 317, ene. 2020, doi: 10.29137/umagd.654056.

[5] “State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time”, *IoT Analytics*, el 19 de noviembre de 2020. https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/ (consultado el 27 de febrero de 2022).

[6] L. Calderoni, A. Magnani, y D. Maio, “IoT Manager: a Case Study of the Design and Implementation of an Open Source IoT Platform”, en *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, abr. 2019, pp. 749–754. doi: 10.1109/WF-IoT.2019.8767304.

[7] E. Raghuveera, P. Kanakaraja, K. H. Kishore, C. Tanvi Sriya, D. P. B, y B. Sai Krishna Teja Lalith, “An IoT Enabled Air Quality Monitoring System Using

LoRa and LPWAN", en *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, abr. 2021, pp. 453–459. doi: 10.1109/ICCMC51019.2021.9418440.

[8] S. Meivel, K. Indira Devi, T. Muthamil Selvam, y S. Uma Maheswari, "Real time analysis of unmask face detection in human skin using tensor flow package and IoT algorithm", *Mater. Today Proc.*, p. S2214785320405826, feb. 2021, doi: 10.1016/j.matpr.2020.12.864.

[9] "IoT platform | Internet of Things | Ubidots". https://ubidots.com/ (consultado el 5 de marzo de 2022).

[10] A. Rajalakshmi y H. Shahnasser, "Internet of Things using Node-Red and alexa", en *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, Cairns, Australia, sep. 2017, pp. 1–4. doi: 10.1109/ISCIT.2017.8261194.

[11] M. Lekic y G. Gardasevic, "IoT sensor integration to Node-RED platform", en *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, mar. 2018, pp. 1–5. doi: 10.1109/INFOTEH.2018.8345544.

[12] "About : Node-RED". https://nodered.org/about/ (consultado el 5 de marzo de 2022).

[13] S. K. Sooraj, E. Sundaravel, B. Shreesh, y K. Sireesha, "IoT Smart Home Assistant for Physically Challenged and Elderly People", en *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, sep. 2020, pp. 809–814. doi: 10.1109/ICOSEC49089.2020.9215389.

[14] IEEE Staff, *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. Piscataway: IEEE, 2019. Consultado: el 5 de marzo de 2022. [En línea]. Disponible en:

http://ezproxy.canterbury.ac.nz/login?url=https://ieeexplore.ieee.org/servlet/opac?punumber=8967328

[15] "¿Qué es Home Assistant?", *Tecnonucleous*, el 19 de julio de 2018. https://tecnonucleous.com/2018/07/19/que-es-home-assistant/ (consultado el 5 de marzo de 2022).

[16] "Cómo instalar central domótica con Home Assistant", *El Output*. https://eloutput.com/productos/domotica/que-es-home-assistant/ (consultado el 5 de marzo de 2022).

[17] "Concepts". https://www.openhab.org/docs/concepts/ (consultado el 5 de marzo de 2022).

[18] S. Team, "Getting Started Guide", p. 86.

[19] A. G. Azwar, R. Haviani Laluma, R. P. Halim, Nurwathi, Gunawansyah, y Gunawan, "Smart Trash Monitoring System Design Using NodeMCU-based IoT", en *2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Bali, Indonesia, oct. 2019, pp. 67–71. doi: 10.1109/TSSA48701.2019.8985517.

[20] K. Pitcheri, K. Murali, y M. Ramprakash R., "Design and Implementation of Smart Home using Cisco Packet Tracer Simulator 7.2", *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, núm. 11S, pp. 107–111, oct. 2019, doi: 10.35940/ijitee.K1024.09811S19.

[21] E. Guerrero y H. Francisco, "Plataforma basada en Internet de las cosas (IOT) para la Medición y Monitoreo Remoto del Nivel de Gas Sulfuro de Hidrógeno Generado por Baterías de Montacargas Eléctricos en entornos logísticos", p. 132.

[22] "MQTT Version 3.1.1". http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc385349261 (consultado el 11 de julio de 2022).

[23] J. de L. Alameda y M. T. A. Gómez, “Aplicación domótica en Android con OpenHAB para el control de los dispositivos del hogar”, p. 130.

[24] S. S. Prayogo, Y. Mukhlis, y B. K. Yakti, “The Use and Performance of MQTT and CoAP as Internet of Things Application Protocol using NodeMCU ESP8266”, en *2019 Fourth International Conference on Informatics and Computing (ICIC)*, Semarang, Indonesia, oct. 2019, pp. 1–5. doi: 10.1109/ICIC47613.2019.8985850.

[25] S. L. Jurj, R. Rotar, F. Opritoiu, y M. Vladutiu, “White-Box Testing Strategy for a Solar Tracking Device Using NodeMCU Lua ESP8266 Wi-Fi Network Development Board Module”, en *2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, Iasi, oct. 2018, pp. 53–60. doi: 10.1109/SIITME.2018.8599250.

[26] S. Rezwan, W. Ahmed, M. A. Mahia, y M. R. Islam, “IoT Based Smart Inventory Management System for Kitchen Using Weight Sensors, LDR, LED, Arduino Mega and NodeMCU (ESP8266) Wi-Fi Module with Website and App”, en *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, Subang Jaya, Malaysia, oct. 2018, pp. 1–6. doi: 10.1109/ICACCAF.2018.8776761.

[27] S. Shakthidhar, P. Srikrishnan, S. Santhosh, y M. K. Sandhya, “Arduino and NodeMcu based Ingenious Household Objects Monitoring and Control Environment”, en *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, Chennai, India, mar. 2019, pp. 119–124. doi: 10.1109/ICONSTEM.2019.8918730.

[28] G. Suprianto y Wirawan, “Implementation of Distributed Consensus Algorithms for Wireless Sensor Network Using NodeMCU ESP8266”, en *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar*

*(EECCIS)*, Batu, East Java, Indonesia, oct. 2018, pp. 192–196. doi: 10.1109/EECCIS.2018.8692952.

[29] A. Aqeel, "Introduction to NodeMCU V3", *The Engineering Projects*, el 11 de octubre de 2018. https://www.theengineeringprojects.com/2018/10/introduction-to-nodemcu-v3.html (consultado el 27 de febrero de 2022).

[30] K. Sharma y R. Nandal, "A Literature Study On Machine Learning Fusion With IOT", en *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, abr. 2019, pp. 1440–1445. doi: 10.1109/ICOEI.2019.8862656.

[31] "DHT22 pdf, DHT22 Description, DHT22 Datasheet, DHT22 view ::: ALLDATASHEET ": https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html (consultado el 16 de junio de 2022).

[32] "4Ch-relay.pdf". Consultado: el 16 de junio de 2022. [En línea]. Disponible en: https://www.handsontec.com/dataspecs/4Ch-relay.pdf

[33] "1N4148 pdf, 1N4148 Descripción Electrónicos, 1N4148 Datasheet, 1N4148 view ::: ALLDATASHEET ": https://pdf1.alldatasheet.es/datasheet-pdf/view/15021/PHILIPS/1N4148.html (consultado el 16 de junio de 2022).

[34] "PC817 pdf, PC817 Description, PC817 Datasheet, PC817 view ::: ALLDATASHEET ": https://pdf1.alldatasheet.com/datasheet-pdf/view/43368/SHARP/PC817.html (consultado el 16 de junio de 2022).

[35]

"https://www.cisco.com/web/ANZ/cpp/refguide/hview/wireless/1240AG.html". Consultado: el 16 de junio de 2022. [En línea]. Disponible en: https://www.cisco.com/web/ANZ/cpp/refguide/hview/wireless/1240AG.html

[36] https://www.cisco.com/web/ANZ/cpp/refguide/hview/wireless/1240AG.html (Consultado el 16 de junio de 2022).

[37] “Productos Turbo HD”, *hiknow*. https://www.hikvision.com/es-la/products/Turbo-HD-Products/ (consultado el 16 de junio de 2022).

[38] “111220_NSE_Presentacion.pdf”. Consultado: el 1 de agosto de 2022. [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec//documentos/web-inec/Estadisticas_Sociales/Encuesta_Estratificacion_Nivel_Socioeconomico/111220_NSE_Presentacion.pdf

[39] “2. 2020_ENED_Principales_resultados.pdf”. Consultado: el 2 de agosto de 2022. [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Economicas/Encuesta_Edificaciones/2020/2.%202020_ENED_Principales_resultados.pdf

[40] “de - La mayoría de las personas beben el agua tal como .pdf”. Consultado: el 2 de agosto de 2022. [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec/wp-content/descargas/Manu-lateral/Resultados-provinciales/azuay.pdf