

Universidad Tecnológica Nacional

Proyecto Final

**Sistema de control vía Mqtt con red
inalámbrica propia expansible**

Autores:

- *Perero, Juan Manuel*

Director:

- *Ing. Gustavo Maggiolo*

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero Electrónico*

en la

Facultad Regional Paraná

Febrero de 2022

Declaración de autoría:

Yo/nosotros declaro/declaramos que el Proyecto Final “Título” y el trabajo realizado son propio/s. Declaro/declaramos:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

-

Fecha:

Agradecimientos:

Profundamente agradecido con el Ingeniero Gustavo Maggiolo, que accedió a ser mi docente director y ejerció su labor aun con el ritmo acelerado con el que trabaje sobre el proyecto. También debe agradecer a compañeros y amigos, que estuvieron para escucharme y darme respuestas, incluso más allá del campo académico. A aquellos docentes que además de compartir sus enseñanzas nos prestan atención y nos dan una mano para solucionar problemas, a veces causados por nosotros mismos. A mi familia y a aquellas personas que me toco conocer e hicieron posible que hoy este escribiendo estas palabras a espera de finalizar mi carrera.

Perero, Juan Manuel

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

Sistema de control vía Mqtt con red inalámbrica propia expansible

Perero, Juan Manuel

Abstract:

This paper deals with the development of a control system via Mqtt with its own extensible wireless network, with what vision is raised its design, as are armed its various stages and the product obtained.

To implement the system, it is necessary to use a netbook as a server to mount the Mqtt Broker, the Web server and the control and management applications. The system's wireless network has the capacity to be extensible and self-configurable, since nodes can be added to the network and these are automatically configured. This is feasible with ESP-12 chips that work with the 802.11 standard, better known as Wi-Fi, and a conventional home router as the base of the network.

The product to be developed consists of a system with development at all levels, communication layer, low level processing or backend and web interface for users making up the frontend. The automatisms within the system are designed to provide an innumerable number of options for applications, such as the use of normal lighting as a deaf alarm triggered by sensors on doors and windows.

Keywords:

Automation, chip ESP, home automation, Industry 4.0, Wi-Fi networks.

Resumen:

El presente escrito, trata del desarrollo de un sistema de control vía Mqtt con red inalámbrica propia extensible, con que visión está planteado su diseño, como se encuentran armadas sus distintas etapas y el producto obtenido.

Para implementar el sistema es necesaria la utilización de una netbook a modo de servidor para montar el Broker Mqtt, el servidor Web y las aplicaciones de control y gestión. La red inalámbrica del sistema posee la capacidad de ser extensible y auto configurable, ya que se pueden agregar nodos a la red y estos se configuran automáticamente. Esto es factible de realizar con los chips ESP-12 que funcionan con el estándar 802.11, más conocido como Wifi, y un router hogareño convencional como base de la red.

El producto a elaborar consiste en un sistema con desarrollo en todos los niveles, etapa de comunicación, procesamiento en bajo nivel o backend e interfaz web para usuarios conformando el frontend. Los automatismos dentro del sistema están pensados con el objetivo de brindar una cantidad innumerable de opciones para aplicaciones, como por ejemplo el uso de luminaria normal como alarma para sordos disparada por sensores en puertas y ventanas.

Palabras Clave:

Automatización, chip ESP, Domótica, Industria 4.0, redes Wifi.

Índice:

Capítulo 1: Introducción	1
Capítulo 2: Desarrollo	4
2.1 – Planteamiento del sistema	4
2.2 – Análisis del sistema por etapas	5
2.2.1 - Tecnología de red inalámbrica	5
2.2.2 - Servidor o Broker en Protocolo MQTT	6
2.2.3 - Equivalencia del sistema de comunicación con OSI.....	7
2.2.4 - Aplicación para gestión de configuraciones de los nodos de red.....	7
2.2.5 - Estructura de las peticiones y envío de configuración para nodos.....	9
2.2.6 - Rutina de funcionamiento de los nodos de red	10
2.2.6.1 - Secuencia de búsqueda de AP e incorporación a la red	11
2.3 – Diseño del dispositivo de accionamientos.....	14
2.3.1 - Tipos de entradas y salidas del dispositivo	14
2.3.2 - Tipos de entradas y salidas del dispositivo	18
2.3.2.1 - Pines de entrada y salida digital	18
2.3.2.2 - Salidas PWM.....	19
2.3.2.3 - Entradas ADC	20
2.3.2.4 - Mensaje de reporte de estado	21
2.3.2.5 - Funciones de sincronización de hora UNIX	21
2.3.3 - Programación de los ESP como dispositivos de acción	22
2.3.3.1 - Diagrama de flujo de la programación.....	22
2.4 – Base de la automatización	25
2.4.1 - Software de gestión	25
2.4.2 - Diseño de la base de datos	25
2.4.2.1 - Definiciones de las tablas.....	26
2.4.2.2 - Relaciones entre las tablas	28
2.4.3 - Operaciones del software	29
2.4.3.1 - Inicialización del programa.....	30
2.4.4 - Definición de los accionamientos en la base de datos	31
2.4.4.1 - Accionamientos basados en tiempo	31
2.4.4.2 - Accionamientos basados en una condición.....	32
2.4.5 - Sitio web del sistema.....	33
2.4.5.1 - Apertura de servidor web	34
2.4.5.2 - Mapa de la web	34
Capítulo 3: Resultados.....	37
Capítulo 4: Análisis de Costos.....	40
Capítulo 5: Discusión y Conclusión.....	42
Capítulo 6: Literatura Citada.....	44
Capítulo 7: Anexo: Hojas de datos de componentes utilizados.....	45

Lista de Figuras:

Figura 1: Esquema del sistema de control vía Mqtt	5
Figura 2: Diagrama de algoritmo de inicialización y conexión de la interfaz estación.....	12
Figura 3: Diagrama de algoritmos para mqtt y escaneo de Access Point externos	13
Figura 4: Esquemático de la adaptación para pines de salida.....	15
Figura 5: Esquemático de la adaptación para pines de entrada	16
Figura 6: Esquemático de la adaptación para el pin de entrada analógica	17
Figura 7: Diagrama de algoritmo de inicialización y conexión de la interfaz estación.....	23
Figura 8: Diagrama de algoritmos para mqtt y escaneo de Access Point externos	24
Figura 9: Esquema de las relaciones entre las tablas en la base de datos.....	29
Figura 10: Tabla de definición de accionamientos	33
Figura 11: Mapa del sitio web	34
Figura 12: Pantalla de ingreso de usuario.....	35
Figura 13: Pantalla inicial con datos del sistema.....	35
Figura 14: Pantalla de accionamientos con controles de edición	36
Figura 15: netbook utilizada como servidor y router principal de la red	37
Figura 16: nodos para ampliación de la red	38
Figura 17: Dispositivos de accionamiento con sensor y luminaria de prueba	38
Figura 18: Mensajes enviados dentro del sistema mediante MQTT	39

Lista de Tablas

Tabla 1: Niveles del modelo OSI.....	7
Tabla 2: Niveles del sistema	7
Tabla 3: Costos de un dispositivo de accionamientos	40
Tabla 4: Costos de un nodo de red.....	40

Lista de Abreviaciones y Símbolos

AP: Access Point o Punto de acceso inalámbrico

STA: Station o estación, interfaz cliente de conexión wifi

ADC: Analog-to-digital converter, traducido como conversor analógico digital

IoT: Internet of Things, traducido como internet en las cosas

ID: Identificador

SoC: System on a Chip, traducido como Sistema en un chip

Dedicado a:

Mi familia, que fueron aquellos que me ayudaron en los primeros pasos de la carrera y en los que siguieron, sin ellos mi camino hubiera sido más difícil.

Capítulo 1: Introducción

A día de hoy, paradigmas como IoT, la casa inteligente o industria 4.0 ya son conocidos en el ámbito tecnológico, pero si bien la ideas que presentan se comprenden pueden tener un problema que impidan su avance, para qué utilizarlas y el cómo realizar estas aplicaciones. No son muchos los sistemas que actúan sobre los enfoques nombrados y tengan un amplio rango de utilidad, algunos simplemente se limitan a la aplicación en si, como la adquisición de datos, pero no contemplan la plataforma de comunicación sobre la que funcionan. Otros simplemente se limitan al uso con el que fueron planeados, desaprovechando muchas de sus posibilidades. Bajo estas ideas es que se trabaja en el presente proyecto.

El objetivo del proyecto es la implementación de un sistema de control o gestión de dispositivos normalmente de forma inalámbrica y automática. Dicho sistema necesitará como base una plataforma o red donde pueda funcionar, para lo cual se pretende diseñar una con la posibilidad de agregar nuevos elementos a esta para ampliar su radio de alcance, conformando una red expansible. Dada la posibilidad de agregar una múltiple cantidad de “extensores” a la red, para que su utilización no resulte compleja y engorrosa se programaran para que sean capaces de reconocer los elementos nodos en la red, conectarse a ellos y autoconfigurarse para formar parte del sistema.

Ya que el planteo del sistema se basa en el protocolo Mqtt para la comunicación, se necesita de un servidor o computadora que funcione como tal, con la finalidad de montar un broker Mqtt. Adicionalmente el servidor precisa de un software o aplicación de control que gestione los dispositivos que conforman la red inalámbrica expansible, para así también proveer los datos necesarios para su autoconfiguración. Para realizar el control de dispositivos se necesita de algún tipo de aplicación que envíe o reciba los mensajes para actuar de alguna forma. Como esta acción variará según el tipo de dispositivo, será requerido que la app realice almacenamiento de datos, control de dispositivos en forma manual y automatizaciones ya sea originadas por sensores como por tiempo. Montado el sistema, tiene la facultad de incorporar cualquier producto que funcione bajo el protocolo Mqtt.

Un ejemplo de dispositivos a controlar es un conjunto de luces, que, con la correcta adaptación, se puede gestionar varias de estas, así como su color y

tiempos de variaciones para crear conjuntos de secuencia. Con los sensores adecuados se puede establecer automatismos varios, ya sea dentro de un hogar o maquinas en un ambiente industrial.

Un beneficio que trae la forma de planteamiento dada al proyecto es que, al tener su propia red inalámbrica, no afecta a redes pre-existentes con saturación de dispositivos, lo cual es una problemática poco vista que se agrava a medida que se masifican los dispositivos wifi de uso cotidiano, ya sea en el ámbito hogareño, lugares de trabajo o espacios libres.

El sistema a implementar posee una gran potencialidad ya que sus limitaciones de uso se encuentran ligadas al fin que uno piense en darle.

Actualmente se pueden encontrar dos focos en donde aplicar el proyecto, lo que se está difundiendo como Industria 4.0 y el campo de IoT con domótica y la realización de diversos automatismos.

Se encuentran diversos productos que funcionan bajo el protocolo Mqtt, ya sea a nivel industrial como hogareño. Estos en si presentan características fijas de conectividad, ya sea cableada o inalámbrica. Además, realizar la preparación del broker propio del protocolo para cumplir con las funcionalidades fijas de dichos productos puede considerarse un desaprovechamiento de la potencialidad del sistema. El proyecto a diseñarse se plantea con la idea de poder incorporar estos dispositivos e implementar otros propios con la particularidad de ejecutar tareas genéricas como activaciones o lectura de sensores.

Abarcando la conectividad inalámbrica y compitiendo con la tecnología WiFi y su aplicación dentro del proyecto, se encuentran la tecnología Zigbee y las redes mesh. Ambas tienen la particularidad de poder armar redes de largo alcance con distintas topologías. Pero también ambas tienen el mismo problema, a comparación de redes WiFi simples, no son tan conocidas en comparación y así como tampoco se comparan en su precio. Al final de cuentas en mucho más fácil encontrar redes de tecnología WiFi que de otro tipo y se pretende que el sistema propuesto sea integrable a infraestructuras existentes.

Haciendo una revisión de los recursos necesarios para la implementación del sistema propuesto, nos encontramos con que necesitamos varios dispositivos de conexión inalámbrica para armar la red para soportar el

sistema. De estos dispositivos existe buena variedad en cuanto a prestaciones, tecnologías y formas de uso, aun así, tanto su adquisición física como información de estos no es de difícil acceso por lo tanto no se ve complejidad en el diseño de la red. Idealmente, para que la red posea un punto de nacimiento confiable, se pretende utilizar un router inalámbrico con buenas prestaciones.

Otro elemento necesario para la red propuesta es el equipo servidor, que puede implementarse en forma de PC o incluso una placa Raspberry, solo se necesita que sea capaz de ejecutar un sistema operativo. En cuanto a software, lo que se necesita es el servidor broker de mqtt y las aplicaciones de control y gestión. El primero ya se encuentran programas que solo necesitan la instalación, así como servidores en la nube, ambas alternativas con opciones pagas o gratuitas. Las aplicaciones de control y gestión necesitarán que se realicen bajo desarrollo propio para que cumplan con el objetivo del proyecto, ya que no se encuentran aplicaciones para fines tan específicos como el propuesto. Por último, también se precisará de sensores y actuadores para que funcionen en el proyecto, así como el dispositivo de control genérico que fácilmente puede ser uno de los módulos inalámbricos utilizados en la red.

Capítulo 2: Desarrollo

2.1 - Planteamiento del sistema

El sistema que se desarrolla puede separarse en dos partes, una que involucra toda la red de comunicaciones que sirve como plataforma para el funcionamiento de la otra parte del sistema que corresponde al control de varios elementos en forma automática.

Dentro de lo referente a la red de comunicaciones se encuentra la red inalámbrica de dispositivos, encargada de brindar la cobertura. Tenemos también el broker MQTT, necesario para el protocolo de mensajería utilizado. Por último la aplicación de control de red, que gestiona las configuraciones de los nodos de la red inalámbrica para que se carguen los parámetros necesarios automáticamente.

Ahora bien, la parte del sistema de control involucra al broker MQTT ya nombrada y su funcionamiento consiste en interactuar con artefactos de distintos tipos a través de lo que se denominará como dispositivos de accionamiento generales. Estos artefactos a controlar pueden ser por ejemplo sensores, actuadores, luminaria o elementos que se ajusten al formato de entradas y salidas de los dispositivos de accionamiento. Se necesita también de programas que manejen las actividades automáticas que se crean fundamentales o útiles en un sistema de este tipo. Adicionalmente y de carácter indiscutible, se incluye aplicaciones y acceso web para que personas puedan trabajar con el sistema que se implementa.

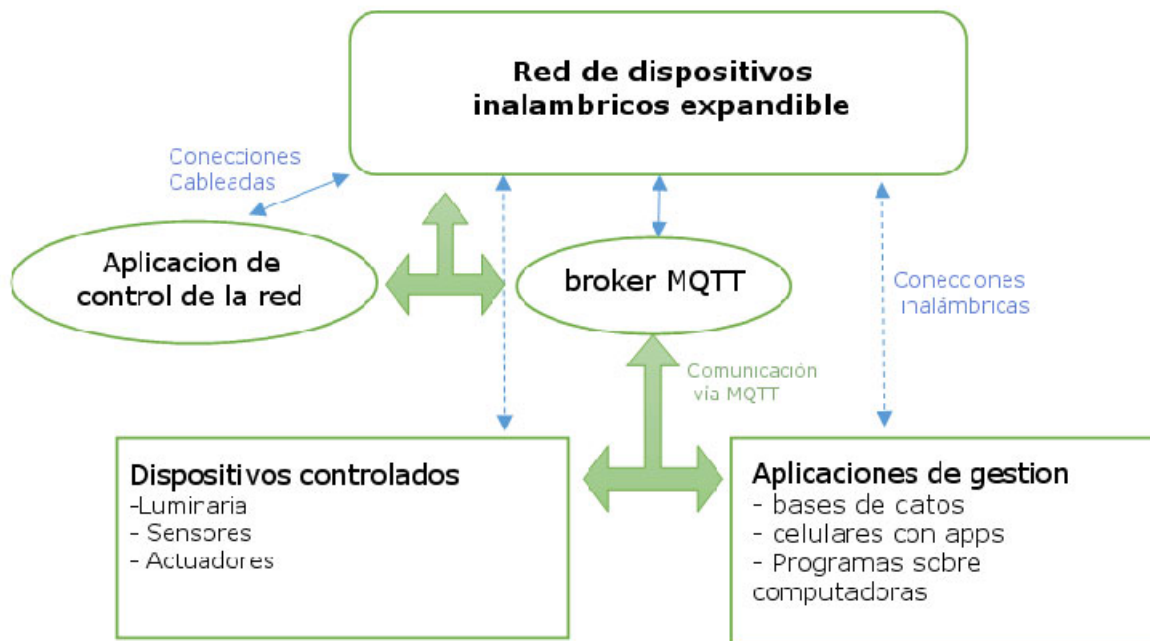


Figura 1: Esquema del sistema de control vía Mqtt

2.2 - Análisis del sistema por etapas

2.2.1 - Tecnología de red inalámbrica

Existen varios dispositivos con los cuales se pueden realizar comunicaciones inalámbricas, cada uno trabajando con su tecnología en particular y características propias de estas. Se pueden nombrar varias de estas, destacándose 2 particularmente en el área de IoT que son LoRa y WiFi, luego tenemos Bluetooth, Zigbee, 3G y 4G, SixFox, WiMax y por supuesto que existen otras, pero no amerita nombrar todas por su cantidad y usos.

Lo que hace que consideremos LoRa y WiFi por las demás, no se limita solamente a que son las preferidas en el campo del proyecto, sino por las propias características de las tecnologías y sus costos.

Para el proyecto se tiene preferencia al uso de tecnología wifi, básicamente porque son prácticamente redes de uso masivo hoy en día. Además, pese a que los módulos para trabajar en ambas tecnologías tienen el mismo costo, para llevar a LoRa a explotar su potencialidad se necesita de un Access Point LoRaWan, el cual si ya posee un precio más elevado y el rango con el

que se dispondría para trabajar superaría al objetivo propuesto, dicho de otra manera, se desaprovecharía el alcance que posee la red LoRaWan cuando se está trabajando para abarcar rangos de nivel hogareño.

Dentro de módulos o chips de tecnología WiFi, encontramos en el mercado los ESP. Módulos considerados como SoC (System on Chip) bastante económicos y disponibles en el mercado en varias versiones y modelos. El chip utilizado por estos módulos es el ESP8266 y se ha encontrado conveniente los montajes de este en las placas ESP-01 y ESP-12. La diferencia entre ambos serán el tamaño y la cantidad de pines disponibles, porque al tratarse del mismo chip, las capacidades y software serán iguales. Estos módulos ya trabajan con el protocolo de señal inalámbrica 802.11 y con el firmware cargado se les dará la facultad de funcionar con paquetes TCP/IP.

2.2.2 - Servidor o Broker en Protocolo MQTT

Pasando a otro nivel del sistema, como protocolo de comunicación se utiliza MQTT, el cual se creó con la consigna de utilizar poco ancho de banda en la transmisión de mensajes. Este protocolo de comunicación M2M (machine-to-machine) se basa en el “push” de mensajes utilizando un sistema de publicación/suscripción a un “tema o topic”. Aclarando, se publica un mensaje con un nombre de tema y todos aquellos a los que estén suscritos a dicho tema se le hará llegar el mensaje.

Este protocolo necesita de un servidor Broker que se encargará de recibir y distribuir los mensajes. Por esta razón, se analizan los softwares existentes para esta tarea, encontrando como alternativas viables a dos de ellos. Una de ellas es el broker “Mosquitto”, bastante popular en el área de mqtt. Aun así, se prefirió dejar de lado para utilizar el servidor EMQ, ya que posee una interfaz gráfica accesible por navegador y cumple con las funcionalidades requeridas. Se ha decidido realizar el montaje de forma local de “EMQ X Broker 3.1.2”. Para montar el servidor se dispuso de una notebook con el sistema operativo Linux, cuya distribución es Ubuntu server 20. La elección se realizó tomando en cuenta que ya se necesita tener un servidor para otras tareas, descartando así la posibilidad de utilizar el broker como servicio en la nube.

2.2.3 - Equivalencia del sistema de comunicación con OSI

Si consideráramos el modelo OSI, el cual es muy utilizado para representar sistemas de comunicaciones, podemos compararlo con lo planteado anteriormente. De esta forma se logra armar un listado de los protocolos de comunicación utilizados por el sistema, descubriendo que se abarca el modelo completo.

Capa	Modelo OSI
7	Aplicación
6	Presentación
5	Sesión
4	Transporte
3	Red
2	Enlace
1	Física

Tabla 1: Niveles del modelo OSI



Capa	
5	Aplicación / MQTT
4	TCP
3	IP
2	802.11
1	

Tabla 2: Niveles del sistema

2.2.4 - Aplicación para gestión de configuraciones de los nodos de red

Para lograr que los nodos se configuren correctamente al momento de incorporarse a la red, se decidió desarrollar un programa que se ejecutara en el servidor y se encargara de gestionar dichas configuraciones. Tanto la petición, como el envío de dichas configuraciones se realizarán por medio de mensajes Mqtt, por lo tanto, el programa desarrollado se comportará como cliente mqtt.

La programación de este gestor se hará en lenguaje C# con NET Core 3.1 utilizando el programa Visual Studio. La ejecución de software debe realizarse en consola para funcionar en la distribución de Linux utilizada y se ejecutará en segundo plano como un servicio o tarea.

En cuanto a lo que es la configuración de los nodos de la red, tiene que plantearse un protocolo de funcionamiento que se basa en las direcciones MAC de las interfaces de los módulos ESP.

Una dirección MAC en general se representa en hexadecimal agrupando los 48 bits en 6 octetos y debería ser única dentro de los dispositivos de una red. Los módulos ESP poseen 2 interfaces de conexión, la interfaz “STATION” o cliente y la “Access Point” o punto de acceso wifi. La dirección MAC de la estación se mantendrá fija y servirá para identificar el dispositivo en el software de gestión. En cambio, la dirección del punto de acceso se ha decidido que modificarla por 2 razones, una es rastrear los dispositivos que conforman la red de nodos y no conectarse erróneamente a otros y lo otra es identificar a que Access point se está conectando el dispositivo y obtener información en base a esto.

La nomenclatura que se decidió establecer, pretende ayudar a la identificación de los distintos elementos, de manera que las distintas redes inalámbricas creadas sean fácilmente identificables independientemente del nombre (SSID) que utilicen. Destinando los primeros 4 octetos para la identificación de nodos, siendo estos “46:53:50:C0”

El 5º octeto se utilizará para determinar el nivel de conexión en el árbol de la red. De igual manera que en el caso anterior, 8 bits sobran para los 4 niveles que se definieron, mientras que el último grupo de bits se utiliza para denominar el dispositivo.

- Nivel 1: “01”
- Nivel 2: “02”
- Nivel 3: “03”

En definitiva, una dirección MAC completa nos queda de la siguiente manera

Dirección MAC de una interfaz AP de nodos N1: 46:53:50:C0:01:05

Dirección MAC de una interfaz AP de nodos N3: 46:53:50:C0:03:02

2.2.5 - Estructura de las peticiones y envío de configuración para nodos

El Tópico o tema de Mqtt sobre el que se trabaja el protocolo de configuración se definió como “notific”. También se estableció la estructura de una serie de mensajes que trabajarán los nodos de la red y la aplicación de control.

- Petición de configuración por parte de los nodos a la aplicación.

"GETCONF-X-IPSTA: + IPstation + -X- + MACsta + -X-"

El mensaje consta de un encabezado y separadores, los datos que poseen son la dirección IP y MAC de la “estación” o cliente del nodo. Los datos del mensaje son todos enviados como caracteres imprimibles.

- Mensaje para enviar configuración desde la aplicación a los nodos.

">' + IPdelared + '/' + MAC + '~' + longitudmqttid + mqttID + ipredtexto^macentexto"

Se establecieron los bytes delimitadores 1º byte = '>', 6º byte = '/', 13º byte = '~', todos como caracteres visibles. "IPdelared" asignará la IP de la nueva red que creará el nodo como Access Point y utilizará la MAC también recibida, esta serie de datos son enviados como bytes debido que de esta forma es más sencillo trabajarlo en los módulos ESP. Luego tenemos un byte que define la longitud del dato mqttID, el cual se utilizará para el nombre del cliente de conexión, será un dato enviado como carácter. Por último, se incorpora los datos de IP y MAC anteriores, pero esta vez en formato de caracteres, esta redundancia se añade para así tener mejor visibilidad de las configuraciones enviadas.

- Petición de estado de la aplicación a los nodos

"RECONF"

Utilizado para que la aplicación de control pida todos los datos de los nodos en la red, en caso de que se necesite recuperarlos o realizar un control de estado. Resulta de gran ayuda al momento de realizar debugs del sistema.

- Respuesta de los nodos a la petición "RECONF"

"STATUS-X-IPSTA: + IPstation + -X- + IPap + -X- + MACap + -X- + MACentexto"

Se transmite el encabezado y los separadores, los datos en el mensaje son la IP de la parte "estación" del dispositivo, la IP del punto de acceso y la dirección MAC del AP. A excepción de la dirección MAC, todos los datos se transmiten como caracteres, por esa razón al final se agregó dicha dirección en formato visible.

2.2.6 - Rutina de funcionamiento de los nodos de red

Los nodos de red son conformados por los módulos ESP-12, los cuales hubo que programar para que realizaran las tareas requeridas. Por una cuestión de prestaciones del firmware para los ESP, se utiliza para la programación el IDE Arduino con las configuraciones para utilizar los módulos en cuestión.

Pasando al algoritmo implementado, este consiste básicamente en iniciar las conexiones y cargar las configuraciones. Las librerías para la programación de la secuencia de trabajo planteado son 2. La librería que nos brinda las funciones para utilizar lo relacionado al control de los ESP es "ESP8266WiFi.h", la cual nos permite ejecutar órdenes del microcontrolador, que en general se limitaron a las funciones relacionadas con la gestión de red wifi. La otra librería utilizada es la "PubSubClient.h" que nos permite la

utilización del protocolo MQTT para mensajería utilizado para la autoconfiguración y control del mapa de red. En lo que se considera significativo profundizar dentro de la programación realizada, es en las secuencias para búsqueda y conexión a la red y los procesos de respuesta a la recepción de mensajes por MQTT que dependerán del “tema” o “topic” de este.

2.2.6.1 - Secuencia de búsqueda de AP e incorporación a la red

Para realizar la conexión de un nodo a la red, se preparó un algoritmo para que realice un escaneo de los Access point disponibles, filtrarlos según algún tipo de parámetros y realizar la conexión como cliente sobre alguno de los elementos escaneados. La secuencia se basa en que el ESP realice un escaneo de APs, obteniendo datos de SSID, MAC y potencia RSSI. Una vez realizada esta operación, se procede a analizar una por una los APs para realizar un descarte por límite de potencia, por el momento establecido como -80 dbm, y por análisis de dirección MAC. En el análisis de MAC se controla si el AP es un router especificado manualmente en la programación o si se trata de otro nodo ya inicializado en la red, cuya MAC iniciará con los 4 bytes "46:53:50:C0". Luego se evalúa en qué nivel se encuentra para agregarla en listas correspondientes a estos.

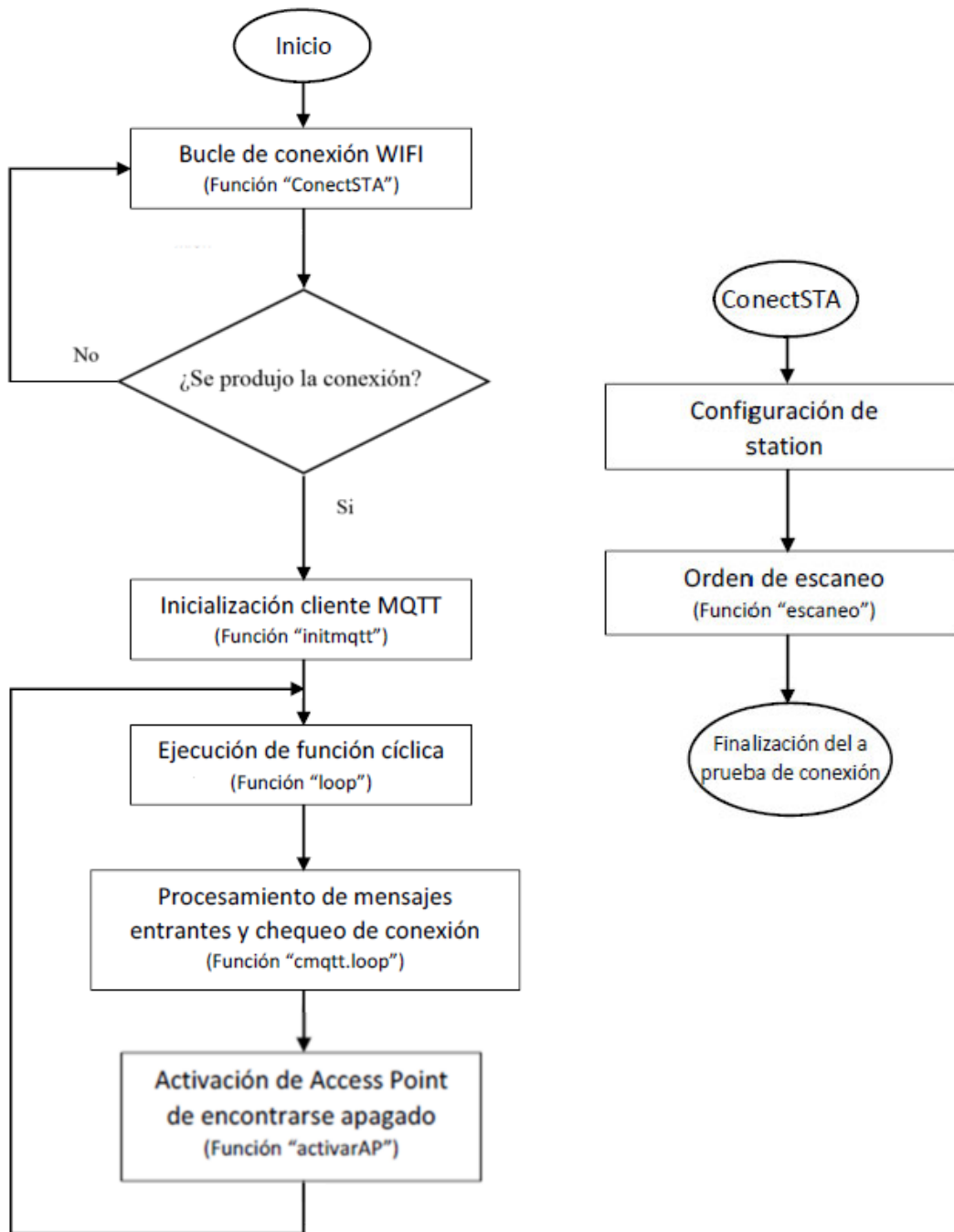


Figura 2: Diagrama de algoritmo de inicialización y conexión de la interfaz estación

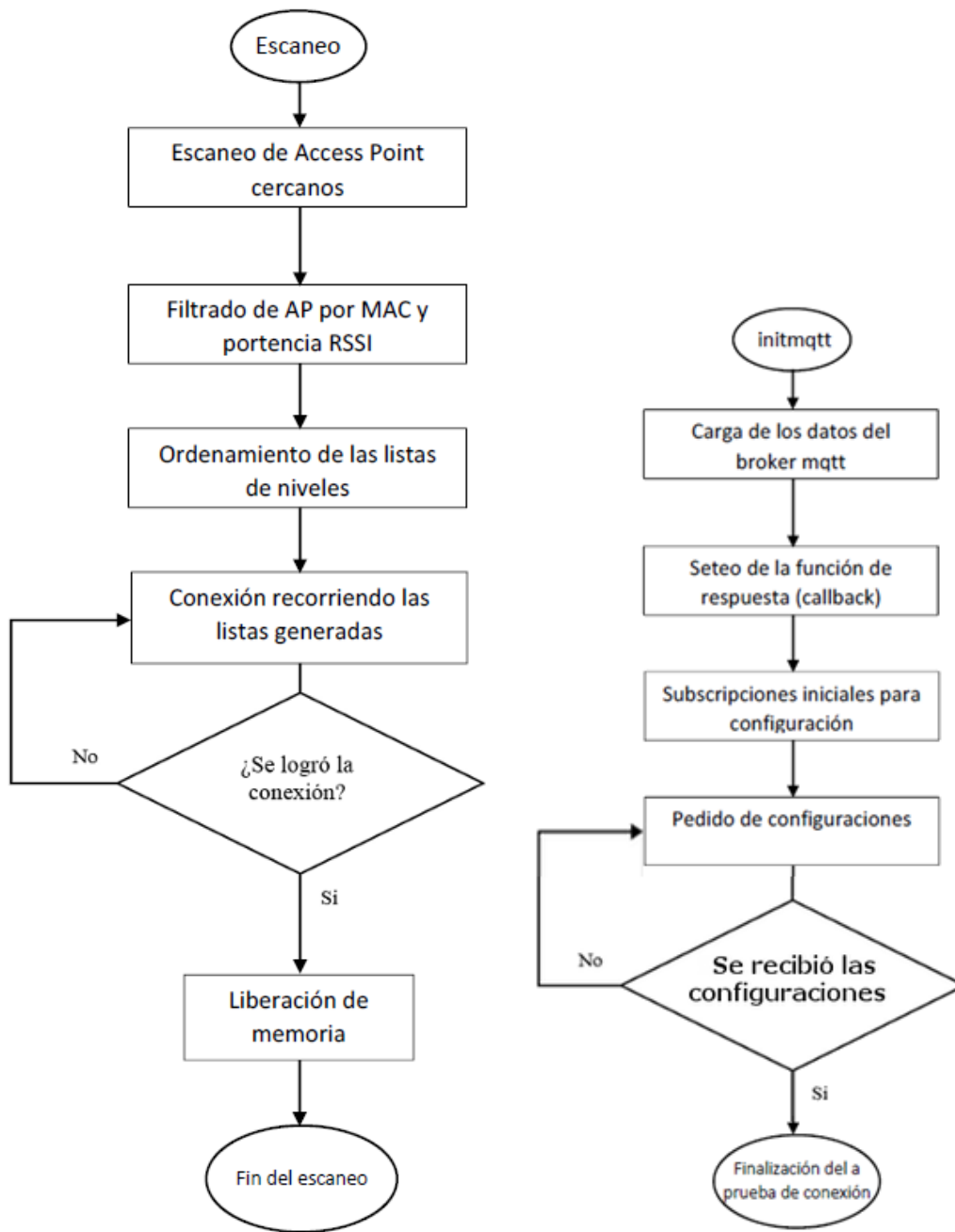


Figura 3: Diagrama de algoritmos para mqtt y escaneo de Access Point externos

Habiéndose finalizado con este rastreo, se procede con la secuencia de conexión, que consiste en recorrer las listas generadas para intentar establecer esta. Si no se logra en un determinado periodo de tiempo establecido de forma arbitraria, se pasará a la siguiente AP en la lista. Si no

se consigue conectar con alguna AP de la lista se comienza un nuevo escaneo y análisis. Este ciclo puede repetirse solo 10 veces, ya que debido a una cuestión de memoria se decidió que llegado a este punto se reinicie el ESP, con lo que en definitiva comienza a inicializarse nuevamente y volvería a entrar en este bucle hasta lograr una conexión.

2.3 - Diseño del dispositivo de accionamientos

Ya habiéndose logrado la red y el sistema de comunicaciones, sería necesario un dispositivo capaz de actuar en función a los mensajes que se le envíen. Tomando en cuenta la finalidad del sistema, el controlador del que se habla deberá interactuar con la mayor cantidad posible de sensores, actuadores u otros elementos de accionamiento.

Justamente los módulos ESP-12 utilizados para formar la red inalámbrica poseen los 2 requerimientos básicos para utilizarse como dispositivos de acciones generales. Los ESP-12 poseen las funcionalidades necesarias para la conexión a la red y también tiene pines de entradas y salidas de distintos tipos, solo debemos definir cuales usar, bajo qué condiciones y que adaptaciones utilizar para que sean funcionales en la mayoría de los casos.

2.3.1 - Tipos de entradas y salidas del dispositivo

A fin de simplificar el uso del dispositivo de acción, se deberá definir algunos parámetros como la cantidad de pines a utilizar y sus funcionalidades. Utilizar todos los pines llevaría a complejizar tanto su uso como su desarrollo, así que se decidió utilizar 3 salidas pwm, con posibilidad de utilizarlos como salidas comunes, 2 salidas digitales adicionales, lo que da un total de 5 salidas de este tipo, un pin de entrada digital y un pin de entrada analógica con ADC.

Si bien algunas de las funciones nombradas podrían combinarse en un mismo pin, es conveniente no hacerlo por varias razones. Una de estas es la necesidad de incorporar circuitos de adaptación en los distintos tipos de pines, los cuales variarán según su funcionalidad, haciendo incompatible por ejemplo que las salidas funcionen como entradas.

Independientemente del diseño del hardware adicional, la decisión de disponer en forma separada los tipos de pines, se ve influenciada por la complejidad que se le sumaría al proyecto en cuanto a desarrollo y también uso del sistema debido a la gran variedad de posibilidades que se presentan. Por dicha razón, se considera que cubrir algunos casos esenciales o comunes es la elección correcta.

Las salidas PWM y las salidas generales son prácticamente las mismas, por lo que llevan el mismo circuito de adaptación. Este consistirá en un transistor mosfet consiguiendo una salida de tipo “open colector” o colector abierto, lo que fija una serie de características a las salidas del dispositivo de control. En primera instancia, se logra aislar al microcontrolador de los aparatos externos que se deseen controlar. Segundo, se podrán manejar magnitudes de potencia mayores, ya que dependerá de las características del transistor en lugar de la corriente y tensión soportadas por los ESP. Por último, el circuito planteado hace que sea necesario utilizar una fuente de alimentación externa a la del microcontrolador, lo que brinda la ventaja de poder elegir sus valores de tensión y potencia según lo requiera el o los aparatos que se quieran controlar.

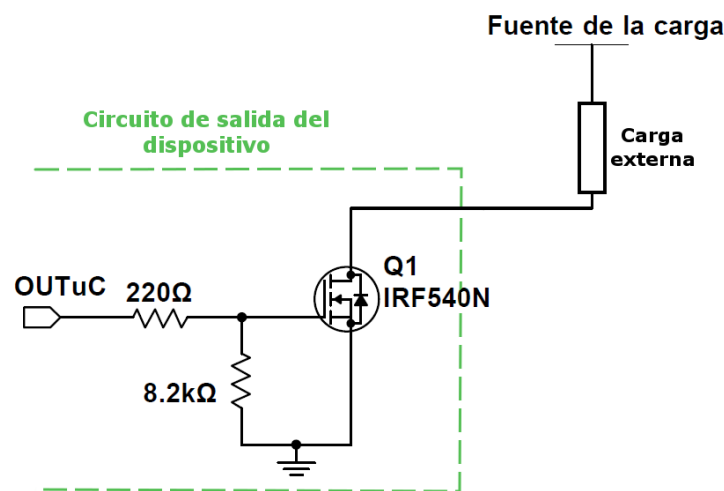


Figura 4: Esquemático de la adaptación para pines de salida

Esto anula la chance de utilizar los pines destinados a la función nombrada como entradas, pero a cambio podemos trabajar niveles de potencia mayores, que ya estarán limitados no por el ESP sino por las características del mosfet.

La entrada digital, se decidió incluir dado que gran cantidad de sensores actúan con formato ON-OFF, en este caso el pin del ESP trabaja con 3.3 [V]. Se incluyó la posibilidad de que el pin trabaje bajo peticiones de lectura y también mediante interrupciones, para así, mejorar la forma de implementar alarmas o controles similares. Para realizar una aislación de dicha entrada, se agrega un optoacoplador a modo de circuito de adaptación. Adicionalmente este servirá para modificar los valores lógicos de funcionamiento, permitiendo que la tensión máxima sea 5 [V], en lugar de la tensión del micro.

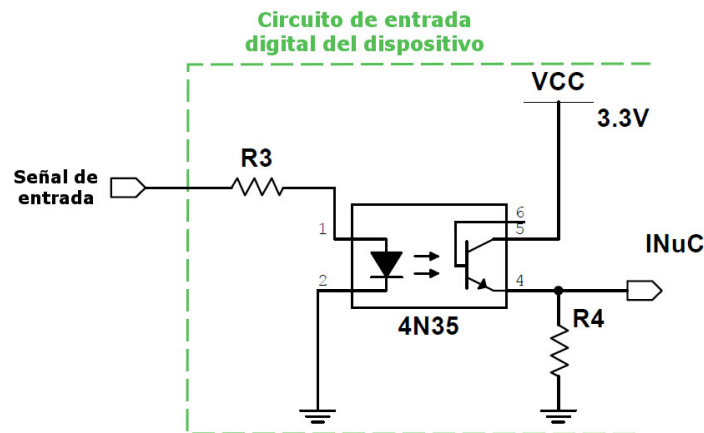


Figura 5: Esquemático de la adaptación para pines de entrada

La entrada analógica consistirá en uno de los pines con un ADC en el ESP12, el límite de tensión que puede trabajar en la placa de desarrollo de NodeMCU es de 3.3[V], mientras que fuera de esta, es decir si se utiliza solo el chip, es de 1[V]. Sumado a esto, los niveles de la señal que se medirán tienen mucha dependencia de los sensores se pretendan utilizar, por lo que se dispondrá de un circuito de acondicionamiento que trabaje entre 0 y 5 [V]. Fuera de este rango se necesitará de un circuito externo como parte del sensor para que adapte la señal a los límites impuestos.

El circuito interno para la entrada analógica está diseñado para realizar la adaptación de tensión máxima hasta 5 [V] y también para aislar los pines del microcontrolador de la propia entrada. Esta aislación consiste en utilizar un optoacoplador, evitando el ingreso directo de la señal. El problema que

posee este dispositivo electrónico es que distorsiona la parte baja de la señal debido a la tensión umbral de funcionamiento del diodo que lo compone.

Para evitar esto, se incluye en el circuito de adaptación un amplificador operacional funcionando como sumador no inversor para sumarle a la señal de entrada una tensión de referencia continua. Dicha tensión proviene de otro amplificador, de preferencia en el mismo chip, que actúa como seguidor de tensión.

La etapa del sumador y el optoacoplador, se encargarán de recomodar los niveles de la señal a los del microcontrolador y la señal continua que se le suma, se compensará por software midiendo la tensión de referencia dada por el amplificador con otro pin del microcontrolador.

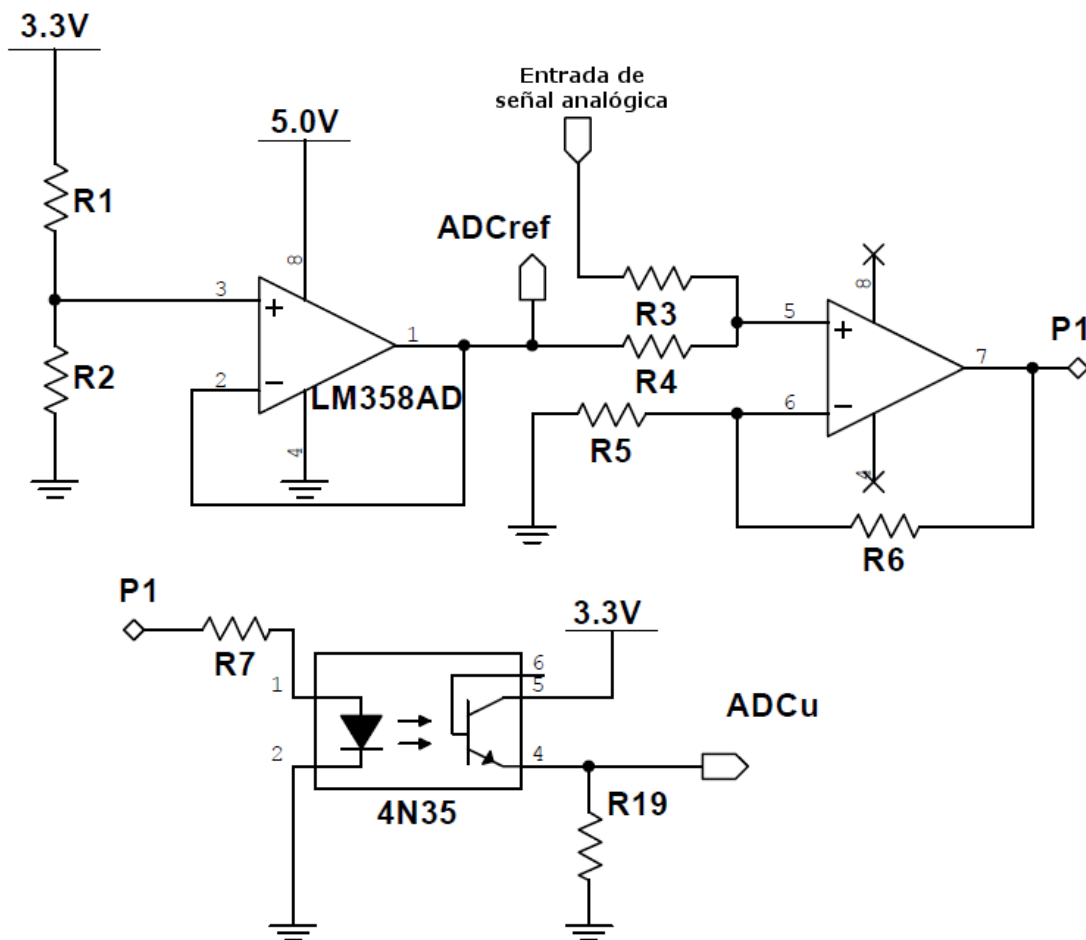


Figura 6: Esquemático de la adaptación para el pin de entrada analógica

2.3.2 Protocolo de utilización de los pines

Al momento de trabajar con los pines, se vuelve necesario implementar una forma de cargar los estados y configuraciones para cada modalidad de salida. Dada la situación se conformó un formato de mensajes por mqtt para las funciones que se consideraron importantes para cada pin.

El tema o “topic” del protocolo de mensajería se definió como la composición de “ctrlG/” y la adición de un identificador único del dispositivo con el formato “CONTROLADORXX” donde las 2 equis al final representan la numeración de dicho identificador. Esta debe cambiar por cada ESP que se programe como dispositivo de acciones generales. Ejemplo del topic final serian “ctrlG/CONTROLADOR00” o “ctrlG/CONTROLADOR25”.

En cuanto a lo relativo al contenido del mensaje, para mantener un formato estructurado en todas las situaciones se utilizan 3 letras para identificar el tipo de salida, un segundo campo para seleccionar pines u configuraciones específicas y de existir un tercero se utiliza para cargar estados o parámetros con un valor dado. Para separar todos los campos se ha utilizado el carácter guion “-” como delimitador. Las distintas variaciones se detallarán a continuación en las explicaciones para cada modalidad de pin.

2.3.2.1 - Pines de entrada y salida digital

Debido a la nomenclatura GPIO (Entrada/Salida de propósito general) utilizada normalmente en el uso de microcontroladores, se utilizó de forma análoga comenzar el contenido de un mensaje destinado a trabajar con estos mismos en el ESP, de la forma “GIO”. Luego del separador, debe incluirse el pin a utilizar. Como ya se mencionó previamente, consideramos para este trabajo 5 salidas y una entrada. Entonces se decidió identificar cada una como P1, P2, P3, P4, P5 y P6, siendo esta ultima la entrada. En el tercer campo es donde se produce la variación entre uno y otro.

En el caso de la entrada, se puede operar de dos maneras, una realizándole peticiones de estado o como se denomina usualmente “polling” y la otra opción es la utilización de la entrada con interrupciones. La petición de consulta de estado, se realiza con el mensaje como se armó hasta el momento, es decir con el tercer campo vacío. Para configurar la función de interrupción, de deben incorporar en la tercera parte las letras “INT”.

Para trabajar con los pines de salida, la tercera parte del mensaje debe armarse utilizando las letras “H” o “L”, con las cuales se indicará la activación o desactivación del pin indicado. Las salidas no poseen configuraciones adicionales, por lo tanto, solo se tienen esas dos opciones. Se proseguirá ahora a mostrar ejemplos de los mensajes para dejar en claro el formato del protocolo implementado.

Ejemplos de mensajes:

GIO-P1-H	→	Poner el PIN1 en estado alto
GIO-P3-L	→	Poner el PIN3 en estado bajo
GIO-P6-	→	petición de entrada
GIO-P6-INT	→	configuración del pin de entrada con interrupción
“PIN6 – Estado: 0”	→	respuesta a petición con entrada en nivel bajo
“PIN6 – Estado: 1”	→	respuesta a petición con entrada en nivel alto

2.3.2.2 - Salidas PWM

Una salida PWM son aquellas que pueden entregar una señal de este tipo, es decir moduladas por ancho de pulso. Estas se suelen utilizar como salidas analógicas variando la potencia que se entrega a un dispositivo, controlando por ejemplo la velocidad de un motor o la intensidad de luz emitida. Justamente por el nombre de la señal, las letras utilizadas en el campo de identificación son “PWM”. Para seleccionar uno de los 3 pines, similar al caso anterior P1, P2, P3 y posteriormente un numero de 0 a 100 que representará el porcentaje del ciclo activo de la señal. También se incorporó la posibilidad de modificar la frecuencia de trabajo de la señal, que puede ir de 100 [Hz] a 40 [KHz], esto se implementó reemplazando el segundo campo donde se utiliza P con “CFG” sin ningún número ya que la configuración corresponderá a las 3 salidas PWM y luego junto con el separador un valor entre 100 y 40000.

Ejemplos de mensajes:

- PWM-P1-5 → Activar salida pwm 1 con un 5% del ciclo activo
- PWM-P2-100 → Activar salida pwm 2 con un 100% del ciclo activo
- PWM-CFG-40000 → Configuración de frecuencia a 40[KHz]

2.3.2.3 - Entradas ADC

Las entradas analógicas funcionan con un conversor del cual se obtiene un valor en byte que hay que convertir para que represente la tensión censada. Por dicha razón se incorporó una secuencia de código con el que se pueda cargar el valor máximo medido por la entrada para poder realizar los cálculos pertinentes y entregar por mensaje mqtt la tensión en su formato ya convertido. Se decidió programar 2 funciones que disparan la medición del adc, una que inicia cuando recibe el mensaje específico y la otra para configurar mediciones automáticas cada cierto tiempo.

Adicionalmente, como el valor de tensión puede necesitar de una referencia temporal dependiendo de la aplicación, dentro del código ejecutado en el ESP, se encuentran algoritmos que permiten llevar un reloj en formato UNIX. La forma de sincronización se verá más adelante, pero el tiempo se incorpora en las respuestas a las peticiones de censado.

El formato utilizado consiste en las letras ADC en el comienzo del mensaje, en el campo siguiente tenemos varias opciones, dos para especificar el modo de disparo y otra para configurar el valor de respuesta. Uno de los modos de disparo, es de ejecución única, especificado al cargar el campo con las letras SIN. El otro modo, se configura con las letras AUT, y en este caso, las conversiones se realizan continuamente según el tiempo que se cargue en la última parte del mensaje.

Por último, tenemos la situación en la que se cargan las letras CNV en el segundo campo. Este sirve para cambiar en el microcontrolador el valor de conversión que se utiliza para el cálculo del valor devuelto por el conversor ADC. Como se fijó el valor máximo de tensión en 5 [V], este será el número de conversión por defecto y en caso de usarse algún circuito de adaptación,

debería modificarse dicho valor. Como el conversor del dispositivo cuenta con 10 bits, los 5 [V] equivalen a los 1023 obtenidos como respuesta.

Ejemplos de mensajes:

ADC-SIN → petición de medición única

ADC-AUT-1000 → comenzar mediciones automáticas cada 1000 [ms]

ADC-AUT-0 → detener adc automáticas

ADC-CNV-5 → carga de valor máximo de tensión 5[V]

ADC: 0 – Tiempo: 16450261552 → resultados de las conversiones del ADC

ADC: 4 – Tiempo: 16450262222

En los mensajes de respuesta tenemos el valor de tensión medido y el tiempo en formato UNIX.

2.3.2.4 - Mensaje de reporte de estado

Otra función a realizar vía los mensajes mqtt, es el reporte de estados y datos del dispositivo. En primera instancia, cuando el dispositivo termina su inicialización, envía un mensaje vía el topic “notific” con sus propios datos de configuración, su ID cliente en mqtt, dirección MAC e IP. Luego, para conocer datos sobre las salidas y entradas se posee la petición “STATUS” bajo el tema “ctrlG/CONTROLADORXX”

2.3.2.5 - Funciones de sincronización de hora UNIX

El formato de hora UNIX es una forma de representar el tiempo que consiste en la cantidad de segundos transcurridos a partir del 1 de enero de 1970 a las 00:00:00, gracias a que se trata de un numero entero facilita varias operaciones en el microcontrolador. La sincronización cuando se reinicia el ESP se realiza con el servidor con un mensaje mqtt bajo el tema “tiempo”. La particularidad que se presenta de realizar un sincronismo vía mqtt es que

cada vez que se envíe el mensaje por este tema con la hora, lo recibirán todos los dispositivos, es decir que todos recibirán la hora para sincronizarse. Esto se producirá cada vez que un microcontrolador se reinicie, por lo tanto, esta operación se produciría con frecuencia.

2.3.3 - Programación de los ESP como dispositivos de acción

La programación cargada en los microcontroladores utilizados como dispositivos de accionamiento consta básicamente de tres partes fundamentales.

La primera es la de conexión automática a la red, la cual es prácticamente la misma operatoria que realizan los nodos de la red cuando buscan incorporarse a ella. La segunda parte corresponde a la conectividad como cliente mqtt, iniciando la comunicación y enviando las notificaciones al gestor para demostrar que existe dentro del sistema y se encuentra iniciado. Por último, se encuentran todos los algoritmos para la interpretación de los mensajes y la utilización de estos datos en las distintas funcionalidades de los pines. Como ya se ha visto, las operaciones que se le puede pedir que realice al dispositivo de accionamiento, tienen formato de comandos y de no respetar su forma de declaración, no funcionarán. Dicho de otra forma, si llega un mensaje al microcontrolador que no respete el formato anteriormente descrito, no existirán acciones a realizar con estos y serán ignorados.

2.3.3.1 - Diagrama de flujo de la programación

En los diagramas de flujos presentados, se muestra la secuencia de las operaciones del software cargado al dispositivo al momento de iniciar.

Siguiendo el algoritmo, lo primero que se realiza al iniciar el ESP es la búsqueda de un punto de acceso a la red del sistema. El método es el mismo que utilizan los nodos de red, reconocen los AP según su dirección MAC y busca conectarse a alguno de ellos. La diferencia con respecto a los nodos es que los dispositivos de acciones generales no necesitarán cargársele una configuración ni utilizarán la interfaz de punto de acceso, por lo tanto, una vez conectado a la red el paso siguiente es la conectividad con el servidor

mqtt. Una vez conectado, el dispositivo envía un mensaje a modo de presentación y comienza su ciclo de funcionamiento donde simplemente espera la llegada de los mensajes de comandos para procesarlos y realizar las acciones.

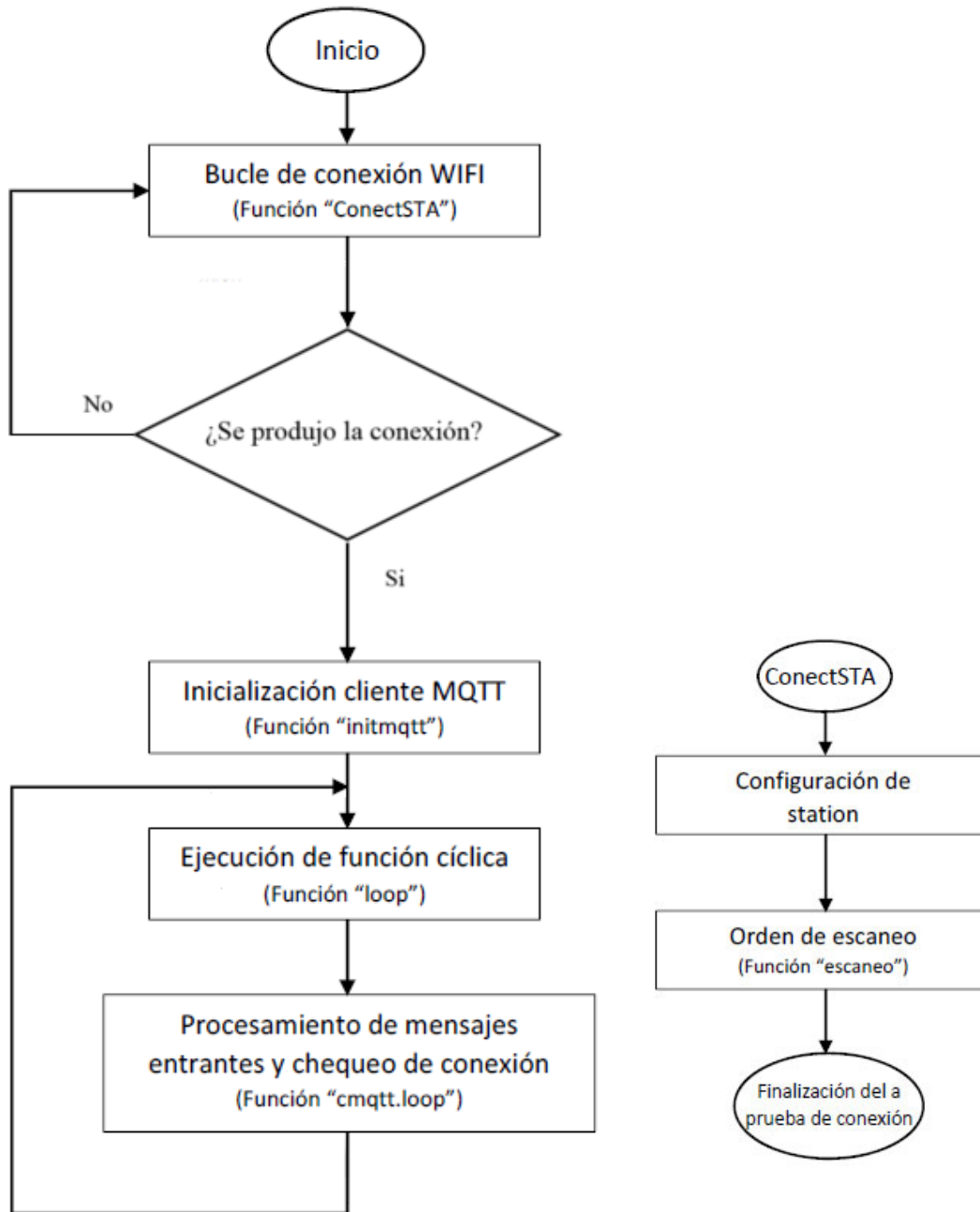


Figura 7: Diagrama de algoritmo de inicialización y conexión de la interfaz estación

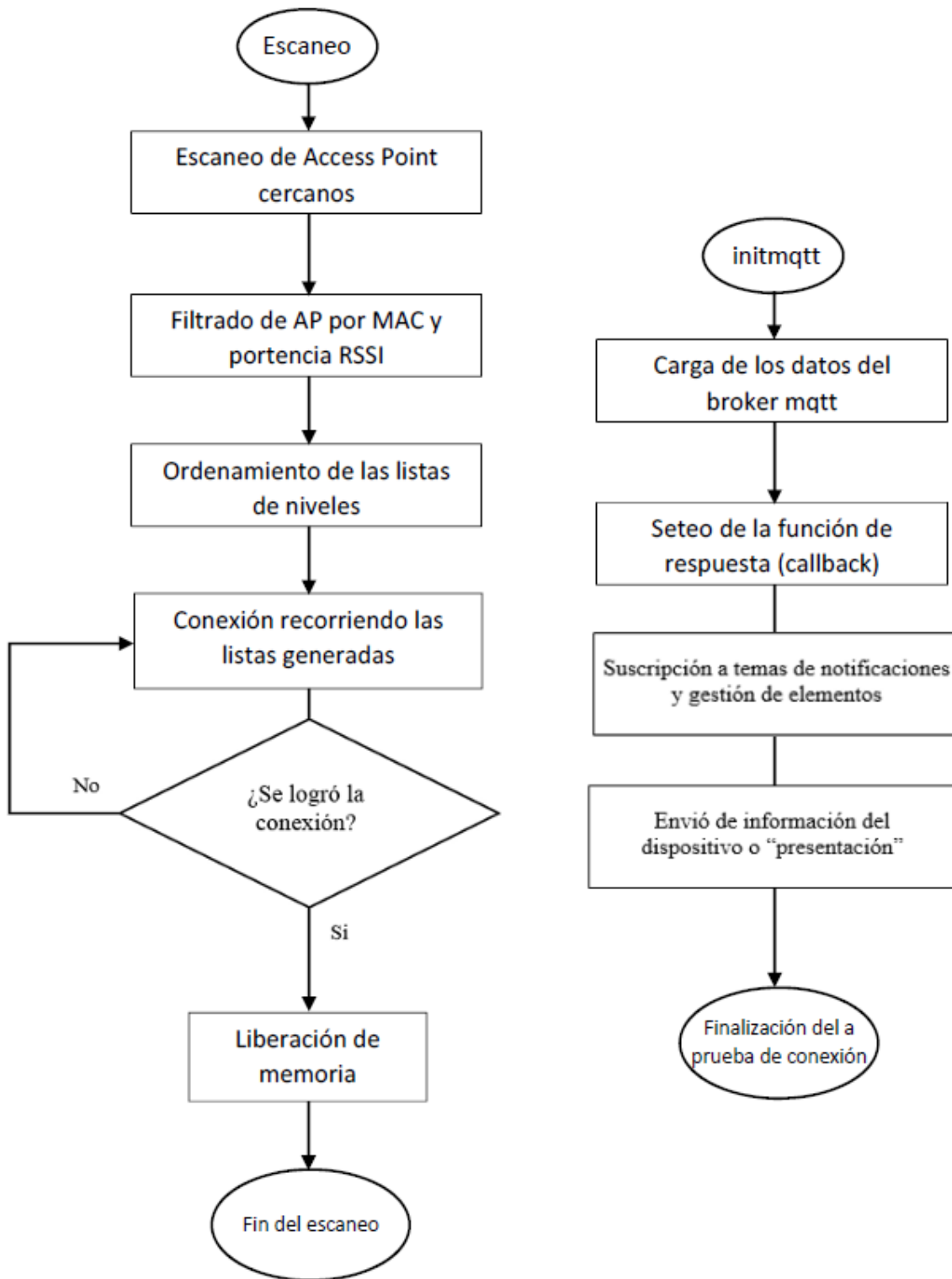


Figura 8: Diagrama de algoritmos para mqtt y escaneo de Access Point externos

2.4 - Base de la automatización

Habiéndose terminado con los dispositivos de acciones generales, se llega al punto en que con un cliente mqtt podríamos controlar sensores y artefactos con los mensajes en el formato dispuesto. El problema es que con esto no se llega al objetivo del sistema propuesto, lo que se busca es lograr que los distintos trabajos o funcionamientos de los aparatos se ejecuten automáticamente bajo las condiciones que se deseen.

Para esto se necesita de una aplicación backend que ejecute las gestiones necesarias. Dicho programa será el que tenga la capacidad de ejecutar las funciones necesarias para distintos accionamientos automáticos. Trabjará en conjunto con una base de datos y ejecutará tareas varias como controlar el estado de los dispositivos y analizar los contenidos de los mensajes para ver si se debe realizar alguna acción, en detalle se explicará posteriormente.

2.4.1 - Software de gestión

El programa de gestión será desarrollado en C# utilizando NET Core 3.1, Se ejecutará el servidor con Linux donde funcionan el resto de aplicaciones de servicios. Para ejecutar la app como un servicio en el sistema operativo debe realizarse en forma de consola, lo cual no es contraproducente ya que no se necesita de interfaz gráfica y el servidor tampoco la posee.

Las funciones de este software involucran la recopilación de la información de los dispositivos, de accionamiento y nodos de red, la actualización sobre la base de datos, administración de las automatizaciones por tiempo y condiciones.

Como la operatividad del programa está ligada a la información dentro de la base de datos, para entrar en detalles sobre distintos algoritmos primero se darán detalles de esta.

2.4.2 - Diseño de la base de datos

Se ha tenido que implementar una base de datos por diversas cuestiones. En principio su uso iba a ser para guardar datos de sensores, pero a medida que se analizaba la asociación de la base de datos con el backend del sistema,

se ampliaron los tipos de datos que podían guardarse e incluso utilizarse con frecuencia.

Para decidir cómo diagramar la base de datos, tenemos que reconocer los elementos del sistema que necesitamos que se almacene su información. Los elementos encontrados en el sistema son los Nodos de red, los dispositivos de accionamiento generales, los distintos sensores y actuadores y los accionamientos o automatismos, que terminan representando un elemento al momento de plantear la consigna al diseñar una base de datos.

Al principio las tablas necesarias con las que se planteó la base de datos fueron una tabla de nodos de red, una de dispositivos de accionamiento, una de elementos de control y una de acciones, luego se agregaron otras tablas a medida que se iba implementando el sistema para agregar y mejorar algunas funciones.

Como ejemplo de tabla agregada, está la de “login”, que puede considerarse como la tabla de datos para identificación de usuario. En esta se almacena la información para el acceso al sistema, nombres de usuario y contraseñas.

Explicaremos ahora las tablas propias del sistema, con su contenido o campos, para posteriormente hablar sobre las relaciones establecidas entre las tablas.

2.4.2.1 - Definiciones de las tablas

Se empezará con la tabla “nodo_red”, la cual representa una lista de todos los nodos de red incorporados al sistema y sus características fundamentales. Como identificador único dentro de la tabla se utiliza la dirección MAC de la interfaz estación. El resto de campos que componen la tabla consiste en las 2 direcciones IP que se le asignan al nodo, la MAC de la interfaz Access Point, el nivel de conexión establecido, los reinicios detectados del nodo y la última hora en la que se detectó activo.

Luego tenemos la tabla de dispositivos de accionamiento, la cual se la llamó “dispgeneral”. De la misma forma que los nodos, los dispositivos de acción se identifican con su dirección MAC, guardado en el campo “MAC_local”. Las características adicionales almacenadas son la IP asignada, la cantidad de reinicios que tuvo el aparato, la última hora activa y el ID de cliente en el

broker Mqtt, denominado “mqttid”. Además, se incorpora otro campo con el cual se vinculará al nodo que se conecte el dispositivo, lo cual se explicará mejor en el apartado de relaciones entre las tablas.

Continuamos con la tabla de sensores y elementos controlados, nombrada “sensyact”. Aquí se le asigna un código de identificación a medida que se incorporan elementos a la tabla. Los demás campos servirán como descripción y serán el nombre que se le asigne, una descripción del elemento y el campo tipo para saber a qué tipo de pin se conecta. Adicionalmente se puede encontrar el campo de vinculo a la tabla de dispositivos de accionamiento.

La tabla “defacc”, posee la definición de los accionamientos definidos en el sistema. Se les asigna un identificador numérico y serán utilizados en la siguiente tabla.

La que utiliza las definiciones es la tabla “acciones” conformada por un identificador y el vínculo al tipo de acción. Esta tabla es fundamental para lo que es la automatización en el sistema, porque cada registro cargado en la tabla representará una acción que se ejecutará si se dan las condiciones. Si bien se considera una tabla importante, aun se necesitan de más parámetros para complementar la acción.

Los parámetros previamente mencionados se encuentran en la tabla intermedia “sya_acc” la cual vincula la tabla “acciones” con la “sensyact”. Gracias a esta relación, se logra la existencia completa del automatismo, ya que se combinándolos se tiene el conjunto acción, dispositivos y condición. Esto se explica junto con los campos que componen esta tabla. Tenemos campos que vinculan los elementos controlados, uno será el que disparan el evento y el otro sobre el que se ejecutará la acción. También tenemos los campos que involucran las condiciones de ejecución de la acción. Aquellas que involucren fecha y hora se guardaran en el campo “hora” de la tabla, mientras que si se trata de un valor específico de tiempo o nivel de señal se guardan en “condición”. Lo que queda por explicar es el campo “mensmqtt”, el cual contiene el mensaje que se enviará para ejecutar la acción determinada.

2.4.2.2 - Relaciones entre las tablas

Respetando el mismo orden con el que se explicó cada tabla, se hablará ahora de las relaciones entre ellas. Como los nodos de red pueden tener conectados varios dispositivos generales, pero dichos dispositivos pueden conectarse solo a un nodo, se dice que la relación es uno a muchos entre la tabla “node_red” y la “dispgeneral”. Este tipo de relación, se refleja en las bases de datos con un campo que apunta a al campo de la otra tabla. En el caso detallado anteriormente, “dispgeneral” posee el campo “MAC_sta1” que guarda el identificador del nodo al que se conecta, es decir el campo “MAC_sta” de la tabla “nodo_red”. A los campos como “MAC_sta1” se los llama usualmente como clave foránea.

La tabla “dispgeneral” también tiene una relación de uno a muchos con “sensyact”. Un dispositivo de accionamiento puede tener muchos sensores o elementos a controlar, mientras que estos últimos solo pueden conectarse a un solo dispositivo de acción. Aquí es donde entra la clave foránea “MAC_local1” de “sensyact”.

Las tablas referentes a los accionamientos se planificaron en forma dividida para mejorar el formato de almacenamiento. Esta es una técnica utilizada para optimizar la base de datos, ocupa menos espacio almacenar un número que apunta a la definición que repetir el texto innumerables veces. La relación queda armada como que un registro de “defacc” puede estar contenida en varios de la tabla “acciones”, pero una acción tendrá solo una definición. De esta forma nos queda la relación una a muchas.

La ultima tabla es la que presenta un poco más de dificultad, ya que se trata de una tabla intermedia. Esta es necesaria porque un sensor o elemento controlado puede ser afectado por muchos automatismos y a la inversa un automatismo involucra varios elementos, por lo tanto, se dice que es una relación de muchos a muchos. Esta tabla intermedia contiene dos claves foráneas hacia la tabla “sensyact” y otra hacia la tabla “acciones”.

Todas las relaciones y como están definidos los campos de cada tabla se podrán ver de mejor manera en el diagrama de la base de datos.

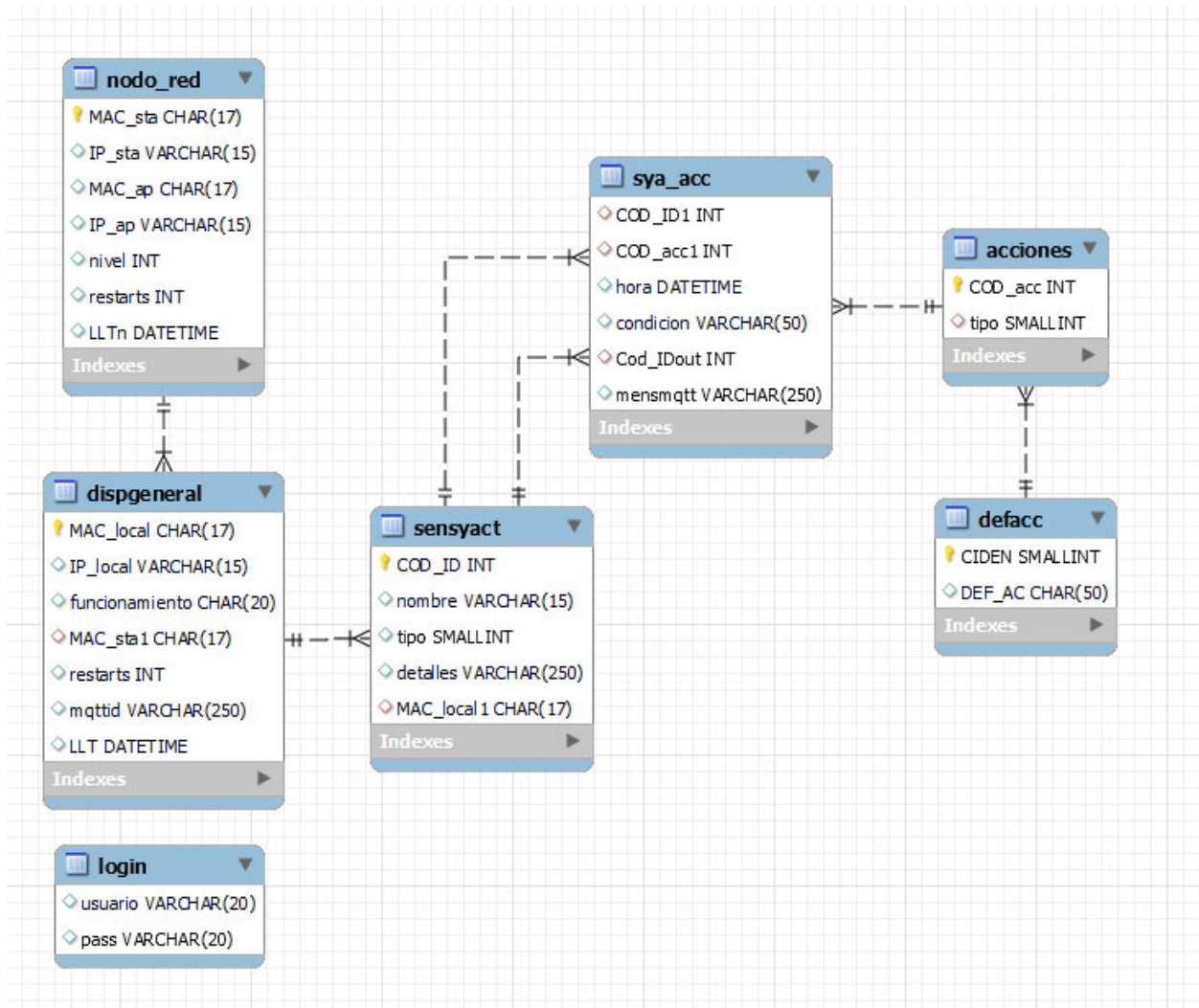


Figura 9: Esquema de las relaciones entre las tablas en la base de datos

2.4.3 - Operaciones del software

La composición del programa de gestión se puede resumir en los algoritmos de inicialización, las funciones de cliente mqtt, temporizadores generados para automatismos por tiempo, funciones de control de mensajes para accionamientos por sensores y código para la apertura de un puerto TCP para poder enviar información desde el servidor web, principalmente para realizar peticiones de actualización de datos.

2.4.3.1 - Inicialización del programa

La primera acción realizada por el software al iniciar es la conexión como cliente al broker mqtt, ya que el sistema se basa en este protocolo, es fundamental que se inicie primero.

Una vez conectado el cliente, se realiza el envío de mensajes para reconocer nodos y dispositivos de acción activos. Esperando un tiempo a que exista respuesta por estos, se arma una lista que posteriormente se comparará con la base de datos.

Mientras se realiza la espera, también se hace la consulta a la base de datos para armar otro listado de dispositivos. Cuando se terminan de recopilar los datos de las 2 fuentes se comparan entre ellos y se realiza una actualización de requerirse.

Una vez recopilada una lista de los dispositivos en el sistema, se realiza la suscripción a los tópicos de control de estos. También se crea una tarea que se ejecutará cada 5 minutos para controlar que los elementos del sistema sigan activos, utilizando los mensajes de reconocimiento.

A continuación, se realizará una consulta a la base de datos para obtener la lista de acciones o automatismos. En cuanto se tengan los datos se procesarán y se generarán los temporizadores para actuar en los momentos configurados y se reconocerán los elementos que disparen otros eventos. Esta lista de eventos se actualizará cada 24hs.

A su vez, cada vez que se actualicen las tareas a realizar debería de reconocerse en el gestor para trabajar con el nuevo listado, por lo tanto, se decidió abrir un puerto por TCP para así poder recibir notificaciones de modificación de la base de datos mediante el servidor web.

Resumen de las operaciones

- Conectividad al broker MQTT como cliente
- Reconocimiento e identificación de nodos y dispositivos de accionamiento activos mediante mqtt
- Petición del listado de dispositivos a la base de datos
- Comparación de las dos listas recopiladas y actualización de datos
- Suscripción a temas de los dispositivos guardados

- Creación de una tarea para controlar si los dispositivos se encuentran activos cada 5 min
- Creación de listado de accionamiento automáticos y carga de disparadores
- Apertura de puerto tcp para recibir por http notificaciones para actualización

2.4.4 - Definición de los accionamientos en la base de datos

En una tabla dentro de la base de datos, estarán definidos los distintos tipos de disparos de eventos con un identificador. Estos serán reconocidos por el gestor, para así aplicar los algoritmos pertenecientes a cada tipo. De esta forma para incorporar un nuevo tipo de disparador es sumamente sencillo, solo se debe agregar este a la base de datos y el algoritmo de procesamiento de disparo al programa.

Dentro de los disparadores, se puede realizar una clasificación que determinará la forma de procesarlos por el gesto. Básicamente se trata de 2, un grupo serán todos aquellos que involucren tiempo y por lo tanto requerirán la creación de temporizadores para controlarlos. El otro grupo, implica los disparadores por condición, ya sea por estado activo o por cruzar un límite referente a un valor, el cual puede ser obtenido por ejemplo mediante un sensor.

2.4.4.1 - Accionamientos basados en tiempo

Como ya se dijo, cuando se procesa un automatismo ejecutado por tiempo se creará un temporizador en el programa para así cuando este termine su ciclo, envíe el mensaje mqtt en cuestión. Debido a la forma de tratar estos temporizadores se puede llevar a cabo los tipos de disparos que se consideraron útiles, con la dificultad de que se necesita una función particular en el software para cada una.

Definición de disparadores en la base de datos del grupo por tiempo:

- Cada X segundos: se trata de disparos periódicos donde el tiempo en segundos estará definido en el campo “condicion” del registro asociado a la acción en la tabla “sya_acc” de la base de datos.
- Unico a la hora X: definimos un disparo único en una fecha y hora determinada. El problema es que una vez disparado, no se vuelve a realizar. La hora y fecha se encuentran en el campo “hora”.
- Cada 24Hs a las Xhs: en este caso si se sigue disparando la acción a la hora fijada todos los días.
- Periódico a partir de Xhs: Ahora nos encontramos con una variación de un disparo periódico a partir de una hora en particular. A partir de este punto los algoritmos necesitan de lo fijado en “condicion” y en “hora” para utilizar como parámetros.
- Hora X con duración...: Disparo a determinada hora durante cierto tiempo
- Periódico dentro de rango horario: se ejecutará la acción periódicamente dentro del rango horario definido

2.4.4.2 - Accionamientos basados en una condición

Aquí el tratamiento es distinto, cuando se recibe un mensaje se controla si el dispositivo origen posee algún accionamiento. Si el contenido del mensaje amerita ejecutar la acción, se enviará el mensaje para hacerlo. En cuanto a procesamiento también se requerirá de una función según el tipo, pero serán más sencillas que las del otro grupo.

Definición de disparadores por condición en la base de datos:

- Si – activación: involucraría lecturas digitales con el estado 1 o true para generar una acción, en general se tratarán de sensores de 2 estados.
- Not – desactivado: mismo caso que el anterior, pero con el estado 0 o false.
- Mayor que...: Se evaluará el valor recibido comparándolo con el campo “condición”, si es mayor se envía el mensaje almacenado en el campo

pertinente. Normalmente se originarán estos disparadores por la lectura del adc de un sensor.

- Menor que...: análogo al caso anterior, pero con el siendo el valor menor.
- Dentro del rango: Se activará si la medición esta entre dos valores.
- Fuera del rango: igual que el anterior, pero a la inversa, si se encuentra fuera.

CIDEN	DEF_AC
1	Si - activacion
2	Not - desactivacion
3	mayor que...
4	menor que...
5	Cada X segundos
6	Unico a la hora X
7	Cada 24Hs a las Xhs
8	Periodico a partir de Xhs
9	Hora X con duracion ms
10	Periodico dentro del las horas...
11	Dentro del valor...
12	Fuera del valor...

Figure 10: Tabla de definición de accionamientos

2.4.5 - Sitio web del sistema

Hoy en día es difícil encontrar cosas que no dispongan de una página web, incluso elementos con funciones totalmente distintas entre sí. Una de las razones de esto es la posibilidad de armar webs con diseño adaptable, que se refiere a que se puede navegar en ellas prácticamente desde cualquier dispositivo gracias a que solo se necesita de un navegador web.

De cualquier forma, esta no es la razón principal por la que se decidió crear una página web, sino que fue por la necesidad de tener una forma de trabajar con el sistema.

La página web será fundamental para la visualización y configuración del sistema. Gracias a esta se podrá tener acceso a la información relevante almacenada en la base de datos y cargar datos de los sensores y las condiciones para sus accionamientos automáticos. Gracias a la página, se logra la interacción entre la aplicación de gestión y la base de datos

permitiendo la carga de información y la actualización del sistema al momento.

2.4.5.1 - Apertura de servidor web

Para armar el servidor web se utilizará la computadora donde está el broker mqtt y las demás aplicaciones del sistema. Se necesitará la instalación de “APACHE” para Linux, que es un servidor web http. Su configuración no es complicada, al menos para las necesidades de nuestro sistema y solo se necesitarán crear las páginas y cargarlas.

La página será creada en código HTML y los scripts para las acciones, parte en JavaScript y parte en PHP. El segundo lenguaje de script se utiliza dado que hay información que no debería salir del servidor, como el diseño de la base de datos, entonces se utiliza PHP que se ejecuta en el servidor a diferencia del otro que se transmite al navegador cliente.

2.4.5.2 - Mapa de la web

La web en si no tiene una distribución complicada, se puede recorrer casi en su totalidad utilizando las opciones en una barra lateral dispuesta en las páginas. Debido a la forma de acceso, prácticamente se puede decir que las paginas se encuentran en un mismo nivel.

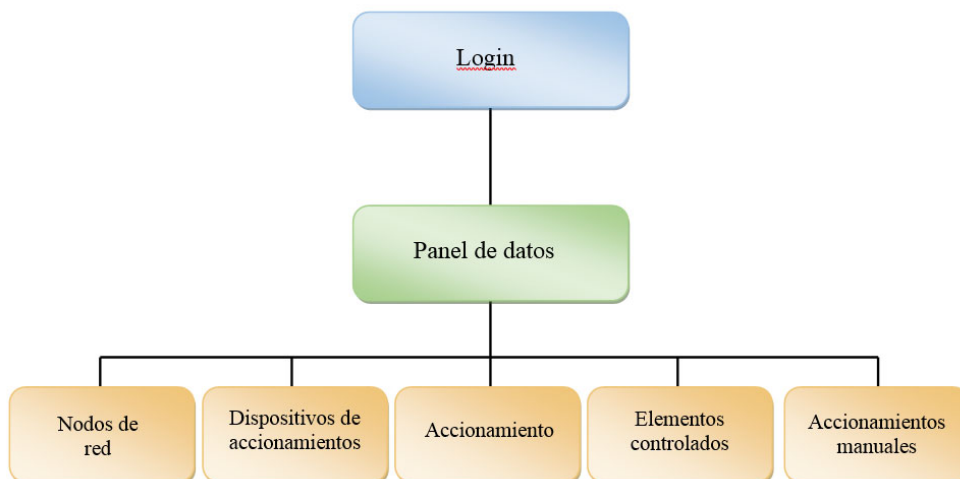


Figura 11: Mapa del sitio web

La primera pantalla que se encuentra al ingresar, es una de identificación de usuario, para así limitar el acceso al sistema y evitar que cualquier persona modifique las configuraciones. La información de la cuenta de usuario, nombre y contraseña, se encuentran almacenados en la base de datos.

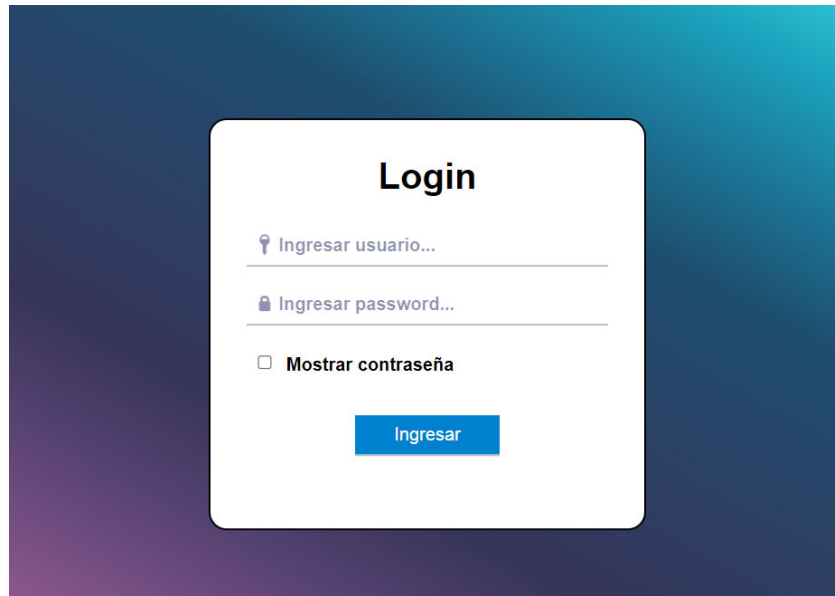


Figura 12: Pantalla de ingreso de usuario

Una vez logueado dentro del sistema, se ingresará a la pantalla principal, en donde se presentará un dashboard con tablas de datos separadas de todos los elementos del sistema. Solamente se tratará de un panel informativo. Las funciones específicas de cada elemento se encuentran en las demás páginas.



Figura 13: Pantalla inicial con datos del sistema

Las páginas de nodos de red y dispositivos de accionamiento general no poseen un uso adicional más que ver los datos, dado que toda la operatividad de estos con el sistema se realiza automáticamente, por lo tanto, no existen operaciones o botones adicionales más que para la navegación en la web.

En la página de accionamientos o automatismos, se muestra la tabla referente a estos. Aquí si se tiene la posibilidad de agregar, editar y eliminar accionamientos trabajando directamente con la base de datos. Al momento de elegir una de estas opciones, se abre una ventana flotante que permitirá elegir las opciones necesarias para agregar o editar una acción.

ID de acción	Dispositivo	Disparador	Tipo de acción	Hora de acción	Condición de acción
48	luces RGB	1	Cada 24Hs a las Xhs	2022-01-29 17:17:00	-
49	luces RGB	1	Cada 24Hs a las Xhs	2022-01-29 17:30:00	20

Figura 14: Pantalla de accionamientos con controles de edición

La página de elementos controlados dispondrá de datos de sensores, actuadores u otro elemento controlado del sistema, al igual que la anterior, posee los controles de agregar, editar y eliminar.

Por ultimo tendremos una página de operaciones manuales, para realizar accionamientos directamente desde la página o incluso enviar mensajes particulares por mqtt distintos a los del formato del sistema. Esto se configuró así para ampliar el rango de uso hacia otros dispositivos que también trabajen con el protocolo.

Capítulo 3: Resultados

Realizando pruebas de uso finales, se logró que los nodos de red ampliaran la red sin notar latencia en la comunicación, lo cual era esperado debido al protocolo utilizado. El ancho de banda utilizado puede considerarse nulo ya que no se transmiten mensajes de forma continua.

En cuanto a hardware, no se considera que las terminaciones finales obtenidas ocupen un espacio significativo, ya que quitando los elementos de mayor tamaño son la notebook servidor y el router. La dificultad puede radicar más en encontrar una toma de alimentación, pero pudiéndose quitar el enchufe y dejar libre el conector USB, alimentarlo por este puerto también es una opción.



Figura 15: netbook utilizada como servidor y router principal de la red



Figura 16: nodos para ampliación de la red

A continuación, se puede ver imágenes con los dispositivos de accionamiento al realizar un test con un sensor magnético y una tira de leds RGB con su fuente externa. También se incluye una serie de mensajes que fueron enviados en el sistema al momento de realizar la prueba.

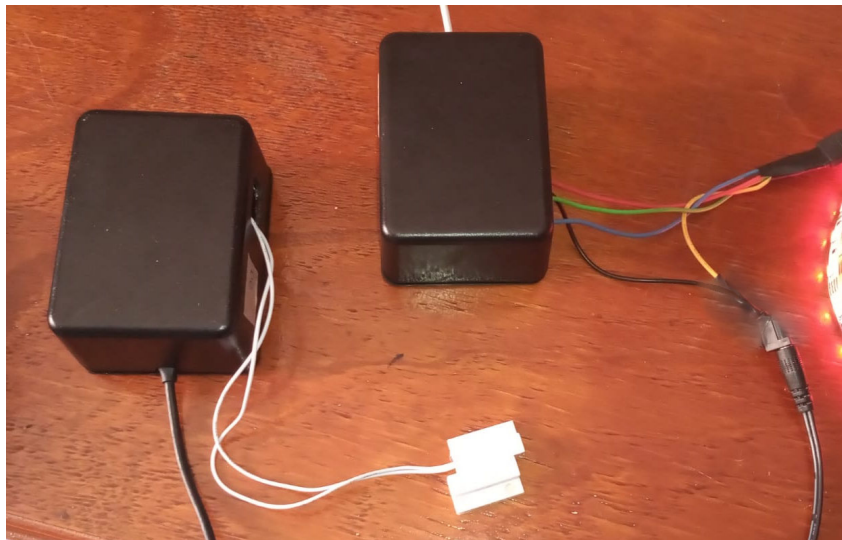


Figura 17: Dispositivos de accionamiento con sensor y luminaria de prueba

Tema	Hora	Fecha	Mensaje
notific	03:39:00 p.m.	21/02/2022	IDENT
notific	03:39:00 p.m.	21/02/2022	RECONF
notific	03:40:52 p.m.	21/02/2022	GETCONF-X-IPSTA:192.168.10.101-X-DC:4F:22:78:3C:44-X-
notific	03:40:52 p.m.	21/02/2022	>♦♦ /FSP♦ ~ 1192.168.30.1^46:53:50:C0:01:01
notific	03:41:29 p.m.	21/02/2022	GETCONF-X-IPSTA:192.168.10.101-X-DC:4F:22:78:3C:44-X-
notific	03:41:29 p.m.	21/02/2022	>♦♦ /FSP♦ ~ 1192.168.10.101^46:53:50:C0:01:01
notific	03:43:55 p.m.	21/02/2022	Iniciacion de dispositivo CONTROLADOR03-84f3:eb:6f:9c:3b-192.168.30.100-
/peticiones/tags	03:43:55 p.m.	21/02/2022	GT
/tiempo	03:43:55 p.m.	21/02/2022	1645469036
notific	03:44:00 p.m.	21/02/2022	IDENT
notific	03:44:00 p.m.	21/02/2022	RECONF
notific	03:44:00 p.m.	21/02/2022	CONTROLADOR03-84f3:eb:6f:9c:3b-192.168.30.100-
notific	03:44:00 p.m.	21/02/2022	STATUS-X-IPSTA:192.168.10.101-X-192.168.30.1-X-FSP♦ ♦-X-DC:4F:22:78:3C:44-X-
ctrlG/CONTROLADOR00	03:45:44 p.m.	21/02/2022	GIO-P6-INT
notific	03:48:22 p.m.	21/02/2022	Iniciacion de dispositivo CONTROLADOR00-84f3:eb:6f:95:16-192.168.30.101-
/peticiones/tags	03:48:22 p.m.	21/02/2022	GT
/tiempo	03:48:22 p.m.	21/02/2022	1645469302
notific	03:49:00 p.m.	21/02/2022	IDENT
notific	03:49:00 p.m.	21/02/2022	RECONF
notific	03:49:00 p.m.	21/02/2022	CONTROLADOR03-84f3:eb:6f:9c:3b-192.168.30.100-
notific	03:49:00 p.m.	21/02/2022	STATUS-X-IPSTA:192.168.10.101-X-192.168.30.1-X-FSP♦ ♦-X-DC:4F:22:78:3C:44-X-
notific	03:49:00 p.m.	21/02/2022	CONTROLADOR00-84f3:eb:6f:95:16-192.168.30.101-
ctrlG/CONTROLADOR00	03:49:51 p.m.	21/02/2022	GIO-P6-INT
notific	03:49:51 p.m.	21/02/2022	GETCONF-X-IPSTA:192.168.10.100-X-DC:4F:22:78:3C:43-X-
notific	03:49:51 p.m.	21/02/2022	>♦♦! /FSP♦ J~ 4192.168.33.1^46:53:50:C0:01:04
ctrlG/CONTROLADOR03	03:50:07 p.m.	21/02/2022	PWM-P2-50
ctrlG/CONTROLADOR03	03:50:22 p.m.	21/02/2022	GIO-P2-L
ctrlG/CONTROLADOR00	03:50:57 p.m.	21/02/2022	8:78:EE- Estado: 1
ctrlG/CONTROLADOR00	03:51:30 p.m.	21/02/2022	GIO-P6-
ctrlG/CONTROLADOR00	03:51:30 p.m.	21/02/2022	- Estado: 1
ctrlG/CONTROLADOR00	03:51:40 p.m.	21/02/2022	GIO-P6-
ctrlG/CONTROLADOR00	03:51:40 p.m.	21/02/2022	- Estado: 0
ctrlG/CONTROLADOR00	03:52:06 p.m.	21/02/2022	GIO-P6-INT

Figura 18: Mensajes enviados dentro del sistema mediante MQTT

Capítulo 4: Análisis de Costos

Para determinar el costo que posee la implementación del sistema completo, primero debemos tener en cuenta la cantidad de elementos que se querrá disponer. A mayor cantidad de nodos implica mayor alcance de la red y el número de dispositivos de acción nos darán las acciones.

Primero se analizará el costo de estos elementos en particular y luego se aproximará el valor de costo para un sistema de 10 dispositivos, con lo que se puede implementar sistema de tamaño medio.

Costos de materiales para la fabricación de un dispositivo de accionamiento			
Materiales	Cantidad	Costo unitario en \$	Costo parcial en \$
Placa pertinax 5x5			\$ 100
ESP-12	1	\$ 700	\$ 700
Borneras dobles	8	\$ 30	\$ 240
Mosfet IRF540	5	\$ 150	\$ 750
Amplificadores LM358	1	\$ 80	\$ 80
Optoacopladores	2	\$ 60	\$ 120
resistencias	20	\$ 5	\$ 100
Caja plástica de cobertura	1	\$ 350	\$ 350
Fuente de alimentación 5[V]	1	\$ 300	\$ 300
Cable USB a micro USB	1	\$ 160	\$ 160
Horas de trabajo	4	\$ 300	\$ 1200
Total			\$ 4200

Tabla 3: Costos de un dispositivo de accionamientos

Costos de materiales para la fabricación de un nodo de red			
Materiales	Cantidad	Costo unitario en \$	Costo parcial en \$
ESP-12	1	\$ 700	\$ 700
Caja plástica de cobertura	1	\$ 250	\$ 250
Fuente de alimentación 5[V]	1	\$ 300	\$ 300
Cable USB a micro USB	1	\$ 160	\$ 160
Horas de trabajo	1	\$ 300	\$ 300
Total			\$ 1710

Tabla 4: Costos de un nodo de red

Si consideramos el sistema de tamaño medio, solamente con nodos y dispositivos de acción tenemos aproximadamente \$30000. A esto se le puede sumar el router que tienen un costo de \$3000 y el equipo que funcione como servidor, donde los precios variarán en función a lo que se elija ya que la gama es amplia, considerando un pc de características decentes, \$40000.

Podemos realizar consideraciones para abaratar este precio, en caso de ya disponerse de una computadora que funcione como servidor o armando un sistema más pequeño, de esta forma los costos se pueden reducir a más de la mitad. Aun así, el total de lo considerado nos da la suma de \$83000.

Tomando en cuenta que no es fácil encontrar en el mercado un sistema completo como el desarrollado y dada la naturaleza versátil del producto logrado, sería conveniente planificar su venta bajo aplicaciones específicas de manera que sea aceptado con mayor facilidad por clientes, si estos pueden observar el funcionamiento del producto sobre sus necesidades.

Capítulo 5: Discusión y Conclusión.

Haciendo un repaso de todo el desarrollo, se puede observar que se ha armado un sistema completo, el cual abarca la conectividad, el procesamiento backend y el frontend. Pensando en la funcionalidad del sistema, vemos que lo logrado posee gran potencialidad en cuanto a su uso, con la desventaja de que este justamente no está definido por completo. Depende de la persona que utilice el sistema el encontrar la aplicación según los objetivos de ella misma. El producto logrado abarca casos genéricos y comunes, por lo que puede trabajar con gran cantidad de elementos con los mismos procesos de configuración.

Comparando con otros sistemas de domótica, como los que utilizan Google Assistant o Alexa, nos encontramos con la característica de necesitar activo siempre el dispositivo servidor, ya que se mantiene en forma local en lugar de tenerlo en la nube, pero esto nos brinda la ventaja de que no necesitamos una conexión de internet o dependencia de una red externa para su funcionamiento.

En cuanto a la red de nodos ESP, no presentan las mismas prestaciones de un routers o repetidores convencionales, por eso se prefirió poseer uno como la raíz de la red. Pese a las características de menor nivel, nos han permitido la disposición de una red bajo condiciones arbitrarias que fueron expuestas en los algoritmos expuestos en el desarrollo previo.

Es importante también tratar el tema de mejoras o actualizaciones que se pueden realizar en el sistema. Incluso realizando modificaciones importantes, al tratarse de un desarrollo por módulos la mayor parte del sistema seguirá siendo funcional. Un aspecto a mejorar es la página web, ya que es básicamente la interfaz de usuario o de alto nivel, se puede modificar aspectos estéticos como funcionales para que su utilización sea más amigable e incluso con más prestaciones (visualización de información en la base de datos, cliente MQTT completo, etc.).

Tratando el firmware de los microcontroladores utilizados, el algoritmo de autoconfiguración de los nodos y dispositivos de acción puede llegar a ser lento si la cantidad de estos es elevada en la red, puede trabajarse sobre esta para mejorar la velocidad de conexión y cantidad de elementos.

También podemos incluir la posibilidad de trabajar con sensores que envían datos por otro tipo de comunicación, como I2C.

Revisando las características de la red inalámbrica, se observa que solo existe un solo router como raíz. La mejora en este aspecto puede tratar sobre la posibilidad de incluir varios routers o incluso la implementación de múltiples redes inalámbricas que funcionen en paralelo.

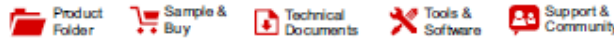
En cuanto a hardware, se puede trabajar sobre el tamaño de los dispositivos de acción y de los nodos de red. Se utilizaron placas con el ESP-12 ya montado, pero se puede reducir el tamaño utilizando ESP-01 o incluso el mismo chip del 12 pero fuera de la placa. Esta opción necesitaría de incorporar circuitería adicional para la alimentación y otros puntos. Se pueden mejorar los circuitos de adaptación para mejorar sus prestaciones, como la utilización de amplificadores de aislamiento en la entrada de pines ADC.

Por otro lado, los automatismos están limitados a funcionar bajo las condiciones que se consideran importantes, pero dependiendo la aplicación que se le quiera dar al sistema puede que no sean suficientes. Por esta razón, para mejorar la versatilidad de este, ayudaría que los algoritmos de procesamiento de condiciones se cargaran como librerías, de manera que fuera sencilla la incorporación de nuevos tipos de automatismos al sistema. En el desarrollo actual, realizar esta actividad implicaría volver a compilar el software del backend y nuevamente instalarlo en el servidor.

Capítulo 6: Literatura Citada

- [1] Espressif. "ESP8266 RTOS SDK user manual". Home - Read the Docs for Business. <https://readthedocs.com/projects/espressif-esp8266-rtos-sdk/downloads/pdf/release-v3.1/>
- [2] Espressif. "NodeMCU Documentation". NodeMCU Documentation. <https://nodemcu.readthedocs.io/en/release/>
- [3] "GitHub - dotnet/mqttnet". GitHub. <https://github.com/dotnet/MQTTnet>
- [4] "GitHub - esp8266/Arduino: ESP8266 core for Arduino". GitHub. <https://github.com/esp8266/Arduino>
- [5] "GitHub - knolleary/pubsubclient". GitHub. <https://github.com/knolleary/pubsubclient>
- [6] L. Marrone, A. Barbieri y M. Robles, Tecnologías Wireless y Movilidad en IPv4/IPv6. Buenos Aires: Editorial de la Universidad Nacional de La Plata, 2011. [En línea]. Disponible: <https://libros.unlp.edu.ar/index.php/unlp/catalog/book/323>
- [7] E. Garcia Villegas, E. López-Aguilera, R. Vidal y J. Paradells, "Effect of adjacent-channel interference in IEEE 802.11 WLANs", Telematics Engineering Dept. Technical University of Catalonia (UPC). Barcelona. 2007.
- [8] B. Mishra y A. Kertesz, "Stress-Testing MQTT brokers: A comparative analysis of performance measurements", Energies, 2021, art. n.º 14-05781.
- [9] IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11, IEEE Computer Society, 2021.
- [10] Information Technology — Open Systems Interconnection — Basic Reference Model: The Basic Model, ISO/IEC 7498-1:1994, Technical Committee : ISO/IEC JTC 1 Information technology, 1996.
- [11] Texas Instrument. "LMx58-N Low-Power, Dual-Operational Amplifiers". <https://www.ti.com/lit/ds/symlink/lm158-n.pdf>
- [12] infineon. ". IRF540N Power MOSFET". Semiconductor & System Solutions - Infineon Technologies. <https://www.infineon.com/dgdl/irf540npbf.pdf?fileId=5546d462533600a4015355e39f0d19a1&redirId=112283>
- [13] Fairchild Semiconductor. "General purpose 6-pin phototransistor optocouplers". <https://www.mouser.com/datasheet/2/149/4N26M-192496.pdf>

Capítulo 7: Anexo: Hojas de datos de componentes utilizados



LM158-N, LM258-N, LM2904-N, LM358-N
 SNOBST31 – JANUARY 2000 – REVISED DECEMBER 2014

LMx58-N Low-Power, Dual-Operational Amplifiers

1 Features

- Available in 8-Bump DSBGA Chip-Sized Package, (See AN-1112, SNVA009)
- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain: 100 dB
- Wide Bandwidth (Unity Gain): 1 MHz (Temperature Compensated)
- Wide Power Supply Range:
 - Single Supply: 3V to 32V
 - Or Dual Supplies: $\pm 1.5V$ to $\pm 16V$
- Very Low Supply Current Drain (500 μA)—Essentially Independent of Supply Voltage
- Low Input Offset Voltage: 2 mV
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Large Output Voltage Swing
- Unique Characteristics:
 - In the Linear Mode the Input Common-Mode Voltage Range Includes Ground and the Output Voltage Can Also Swing to Ground, even though Operated from Only a Single Power Supply Voltage.
 - The Unity Gain Cross Frequency is Temperature Compensated.
 - The Input Bias Current is also Temperature Compensated.
- Advantages:
 - Two Internally Compensated Op Amps
 - Eliminates Need for Dual Supplies
 - Allows Direct Sensing Near GND and V_{OUT} Also Goes to GND
 - Compatible with All Forms of Logic
 - Power Drain Suitable for Battery Operation

2 Applications

- Active Filters
- General Signal Conditioning and Amplification
- 4- to 20-mA Current Loop Transmitters

3 Description

The LM158 series consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc gain blocks and all the conventional op-amp circuits which now can be more easily implemented in single power supply systems. For example, the LM158 series can be directly operated off of the standard 3.3-V power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional $\pm 15V$ power supplies.

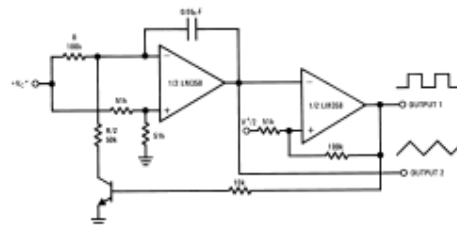
The LM358 and LM2904 are available in a chip sized package (8-Bump DSBGA) using TI's DSBGA package technology.

Device Information⁽¹⁾

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM158-N	TO-CAN (8)	9.08 mm x 9.09 mm
	CDIP (8)	10.16 mm x 6.502 mm
LM258-N	TO-CAN (8)	9.08 mm x 9.09 mm
	DSBGA (8)	1.31 mm x 1.31 mm
LM2904-N	SOIC (8)	4.90 mm x 3.91 mm
	PDIP (8)	9.81 mm x 6.35 mm
	TO-CAN (8)	9.08 mm x 9.09 mm
LM358-N	DSBGA (8)	1.31 mm x 1.31 mm
	SOIC (8)	4.90 mm x 3.91 mm
	PDIP (8)	9.81 mm x 6.35 mm
	TO-CAN (8)	9.08 mm x 9.09 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Voltage Controlled Oscillator (VCO)





GENERAL PURPOSE 6-PIN PHOTOTRANSISTOR OPTOCOUPLERS

4N25
4N37

4N26
H11A1

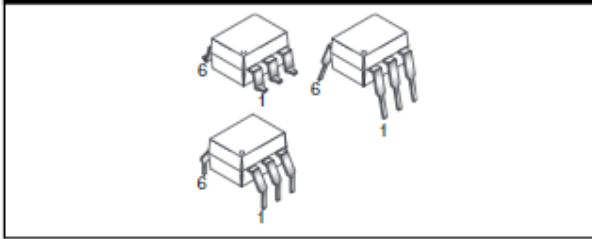
4N27
H11A2

4N28
H11A3

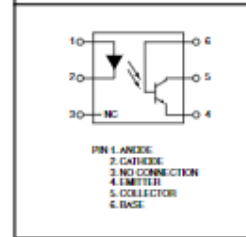
4N35
H11A4

4N36
H11A5

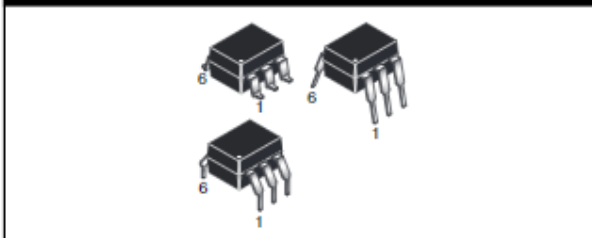
WHITE PACKAGE (-M SUFFIX)



SCHEMATIC



BLACK PACKAGE (NO -M SUFFIX)



DESCRIPTION

The general purpose optocouplers consist of a gallium arsenide infrared emitting diode driving a silicon phototransistor in a 6-pin dual in-line package.

FEATURES

- Also available in white package by specifying -M suffix, eg. 4N25-M
- UL recognized (File # E90700)
- VDE recognized (File # 94766)
 - Add option V for white package (e.g., 4N25V-M)
 - Add option 300 for black package (e.g., 4N25.300)

APPLICATIONS

- Power supply regulators
- Digital logic inputs
- Microprocessor inputs

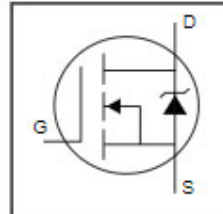
International IOR Rectifier

PD - 91341B

IRF540N

HEXFET® Power MOSFET

- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Fully Avalanche Rated



$V_{DS} = 100V$
$R_{DS(on)} = 44m\Omega$
$I_D = 33A$

Description

Advanced HEXFET® Power MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



Absolute Maximum Ratings

	Parameter	Max.	Units
$I_D @ T_C = 25^\circ C$	Continuous Drain Current, $V_{GS} @ 10V$	33	A
$I_D @ T_C = 100^\circ C$	Continuous Drain Current, $V_{GS} @ 10V$	23	
I_{DM}	Pulsed Drain Current (Ø)	110	
$P_D @ T_C = 25^\circ C$	Power Dissipation	130	W
	Linear Derating Factor	0.87	W/°C
V_{GS}	Gate-to-Source Voltage	± 20	V
I_{AR}	Avalanche Current (Ø)	16	A
E_{AR}	Repetitive Avalanche Energy (Ø)	13	mJ
dv/dt	Peak Diode Recovery dv/dt (Ø)	7.0	V/ns
T_J	Operating Junction and	-55 to +175	°C
T_{STG}	Storage Temperature Range		
	Soldering Temperature, for 10 seconds	300 (1.6mm from case)	
	Mounting torque, 6-32 or M3 screw	10 lbf·in (1.1N·m)	

Thermal Resistance

	Parameter	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case	—	1.15	°C/W
$R_{\theta CS}$	Case-to-Sink, Flat, Greased Surface	0.50	—	
$R_{\theta JA}$	Junction-to-Ambient	—	62	