

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

INGENIERÍA EN SISTEMAS Y COMPUTACIÓN



**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE
INGENIERIA EN SISTEMAS Y COMPUTACIÓN**

**IMPLEMENTACIÓN DE IoT Y CIBERSEGURIDAD CASO DE ESTUDIO
CONJUNTO RESIDENCIAL “CAMPOS DE ARCADIA”**

AUTORES

ANTHONY MIHAEL RIVADENEIRA ERAZO

JAVIER ALEJANDRO ZÚÑIGA SALVADOR

DIRECTOR

FRANCISCO CLAVIJO RODRIGUEZ

QUITO DM, 2022

Agradecimiento Anthony Rivadeneira

Este proyecto de titulación y el avance de mi carrera dentro de la universidad se lo agradezco principalmente a mi familia, a mi papá, a mi mamá, a mis abuelitos, a mi novia, a mi hermano, a mi compañero de proyecto, a mi mejor amigo, a mi mascota y a todos por su apoyo incondicional en todo momento por darme las fuerzas a seguir adelante y seguir luchando por este gran sueño de llegar a ser ingeniero en sistemas y computación. También agradecer a Dios por darme la salud y a toda mi familia porque a pesar de las adversidades he podido salir adelante. Igual a mis amigos de carrera de la banda que fueron un gran soporte tanto de manera académica como dándome su amistad, compañía y por compartir conmigo tan grandes momentos. También agradecer a esas tardes de fútbol por ser un desahogo emocional después de esos duros exámenes y entregas de notas. A mis profesores, en especial a nuestro tutor del proyecto de titulación que siempre trataron de dar su mejor esfuerzo al momento de enseñar tratando de que todo quede claro y forjándome para ser un gran profesional, pero sobre todo por no solo ser profesores, sino también ser amigos con los que se podía conversar de la vida. En fin, el apoyo más importante ha sido mi familia y mi novia que siempre estuvieron para mí, agradecerles infinitamente por ayudarme a convertirme en esta gran persona y por ya casi en un excelente profesional.

Dedicatoria Anthony Rivadeneira

Dedicar este trabajo de titulación principalmente a mi familia, ya que sin ellos esto no sería posible, a mi novia que con su apoyo incondicional y cariño logro que este sueño sea posible, a mi compañero de titulación, que junto a él hemos dedicado varias horas de trabajo para realizar un gran trabajo de titulación y terminar la carrera como la comenzamos con el pie derecho. De igual manera y en especial a mi abuelito, el cual fue un gran apoyo no solo en lo económico, sino por su presión para siempre realizar un buen trabajo y acabar rápido el trabajo y por siempre estar pendiente de mí y ayudarme en lo que fuera posible. Dedicar además a todas las personas cercanas a mí que estuvieron al pendiente de mi esfuerzo y dedicación por lograr un buen trabajo y que siempre estuvieron para solventar cualquier duda y, por último, pero no menos importante dedicárselo a Dios por siempre escuchar mis peticiones y no abandonarme en ningún momento.

Agradecimiento Javier Zúñiga

Quiero agradecer principalmente a Dios, por darme la salud y permitirme terminar una etapa en mi vida, llena de lecciones y enseñanzas que me han forjado como una persona de bien.

A mis padres por ser ese apoyo incondicional y ser mi ejemplo de superación.

A mi madre por ser la mejor madre, ya que ha sacrificado mucho por mí. Su entera confianza y ayuda me han impulsado cada día para hoy estar donde estoy, todo se lo debo a ella y estoy eternamente agradecido por su bondad y paciencia.

A mis abuelitos, que fueron una parte fundamental en mi vida cuando vine a vivir en Quito, porque me ayudaron y me dieron un hogar, sus cuidados cuando enfermaba y por, sobre todo, su cariño y su amor incondicional.

A mi prima que estuvo ahí para escucharme, aconsejarme y acompañarme en mi etapa universitaria.

A toda mi familia que siempre me ha apoyado y ha confiado en mí, incluyendo a los que han tenido que partir.

A mis amigos de Quito y Ambato, que siempre estuvieron ahí cuando los necesitaba, especialmente a David, Anthony y Alejandra, mis mejores amigos.

A mis profesores y a mi tutor del proyecto de titulación, por motivarnos y enseñarnos con paciencia todas las lecciones que nos esperan en la vida laboral.

A mis perritos que fueron ese apoyo emocional en momentos difíciles.

Y, por último, pero no menos importante, me agradezco a mí mismo, por confiar en mí y demostrarme que no existen los límites, por demostrarme que soy capaz de lograr todo lo que me propongo, muchas gracias.

Dedicatoria Javier Zúñiga

El presente trabajo de titulación se lo dedico principalmente a todas las personas que me apoyaron y fueron un pilar fundamental en mi proceso universitario, ya que, sin la ayuda de ellos, no estaría en donde estoy. A su vez, la dedico a mis amigos, que me motivaron y me dieron las fuerzas necesarias para cursar mi carrera universitaria, siendo un buen estudiante y responsable con mis obligaciones.

A mis padres que me apoyaron con mis estudios universitarios en la PUCE, les dedico este trabajo, que es fiel reflejo de cuan agradecido estoy con ellos y que se sientan orgullosos.

Se la dedico a mis amigos de la banda del curso, ya que nos apoyamos mutuamente para aprender y continuamente superarnos.

A mis socios de la empresa Cervecería Vértigo, Hazir y David Gru, fieles amigos, consejeros y catadores de buena cerveza.

Resumen

El presente proyecto de titulación tiene como principal objetivo implementar tecnología IoT y ciberseguridad en el conjunto habitacional “Campos de Arcadia” para que los residentes puedan controlar su hogar de manera remota y segura. La tecnología y el internet han avanzado cada vez más, las personas han optado por adaptarse y convivir con estas a diario en sus hogares, con la finalidad de facilitar procesos y mejorar su calidad de vida. No obstante, la inseguridad está latente y cualquier brecha puede ser explotada por los delincuentes informáticos, ya sea por la propia vulnerabilidad de los dispositivos conectados al internet o por el poco conocimiento que tienen los usuarios sobre ciberseguridad.

Se desarrollaron e instalaron dispositivos IoT enfocados en la seguridad, control y automatización del hogar, los cuales son manejados remotamente mediante el uso de una aplicación móvil para el entorno Android. Estos dispositivos IoT son: controles de luces, aspersores de agua con sensores de temperatura y humedad, sensores de movimiento, sensores de puertas, sensores de ventanas y cámaras de seguridad.

Tomando en cuenta los posibles riesgos a los que se encuentran expuestos los dispositivos IoT al estar conectados al internet, se aplicaron protocolos de ciberseguridad tanto a estos dispositivos como a los routers a los que se encuentran conectados.

El desarrollo planteado en el proyecto de titulación puede ser utilizado como un manual práctico para la implementación de IoT y ciberseguridad en hogares.

Tabla de Contenido

Agradecimiento Anthony Rivadeneira.....	i
Dedicatoria Anthony Rivadeneira.....	ii
Agradecimiento Javier Zúñiga	iii
Dedicatoria Javier Zúñiga	iv
Resumen.....	v
Índice de Figuras	xi
Índice de Tablas	xvi
Introducción	1
Objetivos	3
Objetivo General	3
Objetivos Específicos.....	3
1 Capítulo 1	4
1.1 Historia Conjunto residencial “Campos de Arcadia”	4
1.2 Internet.....	5
1.3 Capas del Modelo OSI.....	6
1.4 Capas del Modelo TCP/IP	8
1.5 Android Studio	9
1.6 Microcontroladores.....	12
1.6.1 Arquitectura de un microprocesador	13

1.6.2	Raspberry Pi.....	14
1.6.2.1	Componentes de Raspberry Pi.....	15
1.7	Sensores.....	16
2	Capítulo 2.....	19
2.2	Arquitectura del Internet de las Cosas.....	20
2.2.1	Capa de aplicación IoT.....	22
2.2.2	Capa de soporte de servicios y aplicaciones IoT	22
2.2.3	Capa de red IoT.....	23
2.2.4	Capa de dispositivos IoT.....	23
2.3	Dispositivos Sonoff	25
2.3.1	Módulo Sonoff	27
2.4	¿Qué es ciberseguridad?.....	28
3	Capítulo 3.....	32
3.1	Arquitectura de Solución.....	32
3.2	Software.....	34
3.2.1	Kanban	34
3.2.1.1	Primera Etapa	36
3.2.1.2	Segunda Etapa	40
3.2.2	Google Cloud Platform	43
3.2.3	MobaXterm	44

3.2.4	DNS.....	45
3.2.5	Apache.....	46
3.2.6	MySQL.....	47
3.2.7	EMQX.....	49
3.2.8	Node.js	52
3.2.9	PlatformIO.....	54
3.2.10	Control Luces	56
3.2.11	Sensores puertas ventanas	57
3.2.12	Sensores de movimiento	59
3.2.13	Aspersores de agua.....	60
3.2.14	ESP32-CAM	62
3.2.15	Ngrok.....	65
3.2.16	Aplicación Android.....	68
3.3	Hardware	74
3.3.1	Placas Dispositivos.....	74
3.3.1.1	Control luces.....	75
3.3.1.2	Sensor puertas y ventanas.....	78
3.3.1.3	Sensor Movimiento	81
3.3.1.4	Aspersores	84
3.3.1.5	Cámara.....	86

3.3.2	Eagle.....	87
4	Capítulo 4.....	93
4.1	Ataque DNS.....	93
4.1.1	Ataque DDOS (Distributed Denial of Service).....	93
4.1.2	Ataques Random Subdomain.....	94
4.2	ARP Spoofing.....	94
4.3	Recomendaciones de seguridad a nivel de router.....	98
4.4	Ciberseguridad aplicada	101
5	Capítulo 5.....	104
5.1	Instalación de los dispositivos	104
5.2	Encuesta de satisfacción	106
5.2.1	Análisis de respuestas	108
5.2.2	Conclusión general de encuestas de satisfacción.....	113
5.3	Entrega del proyecto.....	113
6	Capítulo 6.....	115
6.1	Conclusiones.....	115
6.2	Recomendaciones	116
7	Anexos.....	118
7.1	Anexo A Código Python	118
7.2	Anexo B Código Control Luces	118

7.3	Anexo C Código de Sensores de Puertas y Ventanas.....	129
7.4	Anexo D Código de Sensores de Movimiento	140
7.5	Anexo E Código Aspersores de Agua.....	150
7.6	Anexo F Código Node.js	164
7.7	Anexo G Código ESP32-CAM	185
7.8	Anexo H Código Aplicación Android Studio	191
	Bibliografía	230

Índice de Figuras

Figura 1: Red Informática ARPA. EE. UU.....	6
Figura 2: Las 7 capas del modelo OSI.	8
Figura 3: Capas Modelo Internet.....	9
Figura 4: Android Studio.	11
Figura 5: Microchip.....	13
Figura 6: Arquitectura Harvard.....	14
Figura 7: Raspberry Pi.	15
Figura 8: Partes de la tarjeta Raspberry Pi 4.	16
Figura 9: IoT conectadas.....	20
Figura 10: Modelo de referencia IoT.	20
Figura 11: Aplicaciones IoT.....	21
Figura 12: Dispositivo Sonoff.....	26
Figura 13: Módulos Sonoff Basic.	27
Figura 14: Módulo Sonoff Basic vista posterior.	28
Figura 15: Comparación ciberseguridad Ecuador 2016 vs 2020.	30
Figura 16: Diagrama Arquitectura de solución.....	32
Figura 17: Tablero Kanban Inicio Primera Etapa.	36
Figura 18: Tablero Kanban-To Do Primera Etapa.....	37
Figura 19: Tablero Kanban-Doing Primera Etapa.	37
Figura 20: Tablero Kanban-Done Primera Etapa.....	38
Figura 21: Tablero Kanban Final Primera Etapa.	38
Figura 22: Tablero Kanban Inicio Segunda Etapa.	40

Figura 23: Tablero Kanban-To Do Segunda Etapa.....	40
Figura 24: Tablero Kanban-Doing Segunda Etapa.	41
Figura 25: Tablero Kanban-Done Segunda Etapa.	41
Figura 26: Tablero Kanban Final Segunda Etapa.	42
Figura 27: Google Cloud Platform.....	44
Figura 28: MobaXterm.....	45
Figura 29: Dominio Freenom.....	46
Figura 30: Apache.....	47
Figura 31: Heidi SQL.....	49
Figura 32: Funcionamiento de los dispositivos IoT.....	52
Figura 33: Instalación Node.js.	53
Figura 34: Instalación de MYSQL y MQTT dentro de Node.js.	54
Figura 35: Interfaz PlatformIO.....	55
Figura 36: Código Control Luces.....	57
Figura 37: Código Sensores Puertas y Ventanas.....	58
Figura 38: Código Sensores de Movimiento.....	60
Figura 39: Código Aspersores de Agua.	61
Figura 40: Configuración dominios Ngrok.....	63
Figura 41: Terminal Raspberry Pi desde MobaXterm.	64
Figura 42: Configuración arranque Raspberry Pi.	64
Figura 43: Crontab configuración.	65
Figura 44: Entorno Ngrok.	66
Figura 45: Sección de pago Ngrok.....	67
Figura 46: Versión Pro Ngrok.....	67

Figura 47: Dominios creados.	68
Figura 48: Estructura proyecto Android.	69
Figura 49: Layout activity_login.....	70
Figura 50: Ingreso Aplicación.....	71
Figura 51: Layout activity_main.....	72
Figura 52: Pantalla principal.	72
Figura 53: Placa sin los postes soldados a los pines.	74
Figura 54: Placa con los postes soldados a los pines.	75
Figura 55: Conversor USB a serial TTL.....	76
Figura 56: Conexiones entre la placa y el conversor serial.....	76
Figura 57: Entrada y salida del control de luces.	77
Figura 58: Conexión a GND del sensor de ventana.....	78
Figura 59: Conexión a ESP del sensor de ventana.....	79
Figura 60: Conexiones entre la placa y el conversor serial.....	80
Figura 61: Input y Output del sensor de puertas y ventanas.	80
Figura 62: Conexión sensor de movimiento con la placa y el regulador de voltaje. .	82
Figura 63: Circuito completo sensor de movimiento.	82
Figura 64: Conexiones entre la placa y el conversor serial.....	83
Figura 65: Input y Output del sensor de movimiento.	84
Figura 66: Dispositivo IoT Aspensor.....	85
Figura 67: Sensores de temperatura y humedad, electroválvula.....	85
Figura 68: Módulo ESP32-CAM.....	86
Figura 69: Circuito para módulo de cámara con diodo.....	87
Figura 70: Interfaz Eagle.....	88

Figura 71: Placa aspersores.....	88
Figura 72: Placa cámaras.	89
Figura 73: Planchado.....	90
Figura 74: Placa impresa.....	90
Figura 75: Ácido.	91
Figura 76: Circuito aspersor.....	91
Figura 77: Circuito cámara.....	92
Figura 78: Ataque ARP Spoofing.	96
Figura 79: Configuración ARP.	97
Figura 80: Seguridad en Router.	99
Figura 81: Dispositivos IoT - Sensor de movimiento y cámara.....	104
Figura 82: Dispositivos IoT - Cámara sobre adorno, Sensor de movimiento sobre mesa y Luz debajo de mesa.....	105
Figura 83: Dispositivos IoT - Sensor de Ventana	105
Figura 84: Dispositivos IoT - Sensor de Ventana	106
Figura 85: Plantilla encuesta de satisfacción.	107
Figura 86: Encuesta llena por propietario.	107
Figura 87: Gráfica de pastel pregunta 1.	108
Figura 88: Gráfico de barras pregunta 2.	109
Figura 89: Gráfico de barras pregunta 2.	109
Figura 90: Gráfica de pastel pregunta 3.	110
Figura 91: Respuesta a pregunta 4.	110
Figura 92: Respuesta a pregunta 4.	111
Figura 93: Respuesta a pregunta 4.	111

Figura 94: Respuesta a pregunta 5.	112
Figura 95: Respuesta a pregunta 5.	112
Figura 96: Respuesta a pregunta 5.	112
Figura 97: Carta de entrega de proyecto.	114

Índice de Tablas

Tabla 1: Capas modelo OSI	7
Tabla 2: Arquitectura de Microcontrolador	14
Tabla 3: Principales sensores	18
Tabla 4: Capa de soporte de servicios y aplicaciones IoT	22
Tabla 5: Capa de red IoT.....	23
Tabla 6: Capa de dispositivos IoT.....	25

Introducción

El Internet de las cosas es un tema que actualmente se encuentra en boga en varios campos como por ejemplo en lo tecnológico, social y económico. Actualmente varios productos electrónicos como sensores, cámaras de vigilancia, dispositivos de seguridad, entre otros, son capaces de conectarse al internet y en conjunto, pueden analizar diferentes tipos de datos que ayudarán a la evolución de cómo las personas actualmente viven o trabajan (Rose, Eldridge, & Chapin, 2015).

Es importante considerar que al implementar dispositivos IoT estos se pueden ver vulnerados ya que al mantener una conexión continua a internet pueden ser manipulados por personas exteriores al conjunto. Por tal motivo se aplicarán estándares de ciberseguridad, la cual permite resguardar información muy delicada de los dispositivos que se encuentran conectados al internet. Además de que la información se puede ver vulnerada por el rápido y acelerado crecimiento de las tecnologías de información ya que esto abre paso al delito cibernético sobre todo en la vulneración de la información personal (Gamón, 2017).

El conjunto se encuentra ubicado en el valle de los chillos en la parroquia de Alangasí en el sector conocido como “Playa Chica” a cinco minutos del centro comercial “San Luis”, el conjunto se llama “Campos de Arcadia” cuenta con un total de seis casas de las cuales cada una posee su un patio trasero y un jardín en la parte de al frente de las casas. Se trabajará en un inicio en la casa número cuatro para en un futuro expandir la tecnología a todas las casas del conjunto.

La tecnología que será aplicada en la casa número cuatro serán unos sensores de temperatura, humedad para realizar dispositivos de irrigación, de igual manera se realizará focos inteligentes, cámaras, sensores de movimiento y de puertas y ventanas que enviaran una alerta cuando se activen a la aplicación móvil. Al realizar los dispositivos se incluirá protocolos que cumplan con los estándares de ciberseguridad de igual manera para la aplicación la cual controlará los dispositivos.

Objetivos

Objetivo General

Desarrollar e implementar una aplicación para controlar dispositivos IoT y la ciberseguridad en el conjunto residencial.

Objetivos Específicos

- Analizar y comprender los dispositivos IoT y ciberseguridad enfocada a la seguridad informática.
- Desarrollar una aplicación Android para control de los dispositivos de IoT.
- Implementar la ciberseguridad enfocada a la seguridad informática.
- Implementar la aplicación móvil para el control de los dispositivos IoT.

1 Capítulo 1

1.1 Historia Conjunto residencial “Campos de Arcadia”

El conjunto residencial “Campos de Arcadia” está ubicado en Ecuador en la ciudad de Quito parroquia Alangasí sector playa chica, en las avenidas Río Pita y Río Zamora. Tuvo sus inicios en el año 2014. El proyecto estuvo a cargo del arquitecto Edgar Toscano en el cual se planteaba construir un total de 6 casas de las cuales 2 casas son esquineras con su respectivo diseño y 4 casas intermedias con un diseño en común. El proyecto fue concluido y entregado para el año 2016. Cada casa cuenta con acceso a 2 parqueaderos y a sus respectivas áreas verdes. En cuanto a las áreas verdes la primera casa esquinera cuenta con un patio trasero amplio en forma de L y un patio delantero, el resto de las casas, tomando en cuenta a la casa esquinera restante, cuentan con área verde delante de ellas y en cuanto al patio trasero es de forma rectangular. La distribución de las casas sin contar la segunda casa esquinera es de: 3 dormitorios, 3 baños, 1 sala, 1 cocina, 1 comedor, 1 baño social, 1 baño personal y 1 baño en el cuarto máster. La distribución de la segunda casa esquinera es diferente, ya que cuenta con 3 dormitorios y cada uno con su respectivo baño, un parqueadero cubierto, 1 sala, 1 comedor y 1 cocina.

El conjunto cuenta con ingreso por puerta peatonal e ingreso vehicular por un portón eléctrico, sin embargo, no cuenta con parqueadero de visitas, casa comunal y garita de guardianía.

En cuanto a la seguridad del conjunto, este cuenta con un cercado eléctrico alrededor del mismo y con un video portero eléctrico para abrir la puerta peatonal.

1.2 Internet

Internet es considerada la red más grande e importante del mundo, esta conecta a miles de millones de dispositivos electrónicos como computadores, teléfonos inteligentes, electrodomésticos, entre otros. Su aparición trajo consigo una transformación en el estilo de vida de las personas, todo orientado hacia la digitalización de la información y actualmente es considerado como el principal medio de transmisión de datos de todos los tiempos, gracias a este es que la comunicación ha alcanzado niveles antes inimaginables y es posible tener una conversación con cualquier persona alrededor del mundo en tiempo real. El destino de este es inimaginable ya que se encuentra en constante cambio y evolución.

Su origen se remonta a los años 60's en Estados Unidos de Norteamérica a manos de DARPA (Agencia de Proyectos de Investigación Avanzados de Defensa) con el fin de apoyar a científicos e investigadores en varias universidades a desarrollar nuevas tecnologías. No fue sino hasta el 5 de diciembre de 1969 que se desplego el proyecto ARPANET (Red de Agencias de Proyectos de Investigación Avanzada) gracias a Bob Taylor e Ivan Sutherland. Esta nueva red computacional estaba orientada a la comunicación y envío de datos militares a través del país, de esta manera, la información podía ser transmitida de costa a costa estadounidense teniendo así, una amplia ventaja estratégica en la carrera militar (Nicolás Jiménez, 2019).

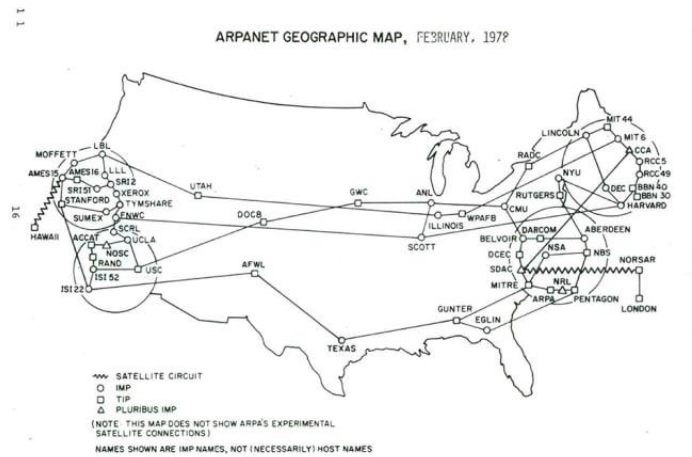


Figura 1: Red Informática ARPA. EE. UU.

Fuente: (Jornet, 2019)

Esta red fue la precursora del Internet que actualmente conocemos, la información era transmitida de un ordenador a otro en un flujo ininterrumpido y cualquier alteración o pérdida de datos podía resultar en la pérdida parcial o completa de la información. Por esta razón, se decidió utilizar el concepto de paquetes de datos, donde la información era particionada para poder ser transmitida. Es aquí donde aparece el modelo TCP/IP y surge con el fin de comunicar a los computadores a través de la red ya sea esta de manera local (LAN) (Local Area Network) o a través de redes amplias (WAN) (Wide Area Network).

1.3 Capas del Modelo OSI

El modelo OSI aparece en función a la necesidad de estandarización de arquitecturas de protocolos, esto debido a que existían muchas arquitecturas de carácter privado y abierto.

Debido a su complejidad y a que aparecieron modelos que simplificaban las conexiones, este modelo no fue tomado en cuenta por desarrolladores, sin embargo, es utilizado para comprender de mejor manera los conceptos básicos de redes.

El modelo OSI está conformado por siete capas o niveles, los cuales son:

N°	Nivel
1	Aplicación
2	Presentación
3	Sesión
4	Trasporte
5	Red
6	Enlace
7	Físico

Tabla 1: Capas modelo OSI

Fuente: Propia

Los niveles del modelo OSI pueden comunicarse de manera horizontal, siempre y cuando sea con los de su misma clase o nivel, y de manera vertical, cada nivel solo puede comunicarse con sus niveles adyacentes.

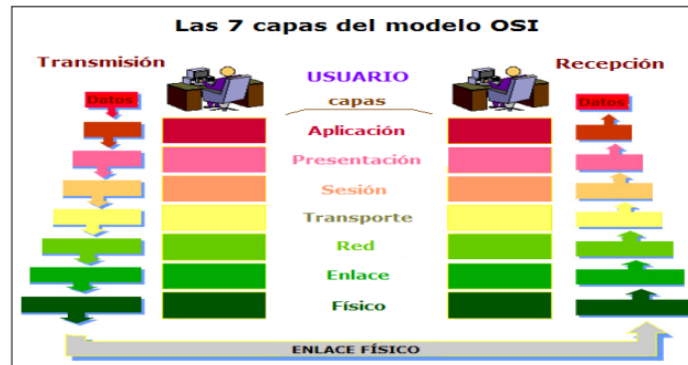


Figura 2: Las 7 capas del modelo OSI.

Fuente: (Chaupis Guardia, 2018)

1.4 Capas del Modelo TCP/IP

El modelo TCP/IP es un conjunto de protocolos los cuales respectivamente hacen referencia al nivel de transporte y al nivel de red. Una gran ventaja en comparación con otros modelos es que las aplicaciones que utilizan este modelo no discriminan el tipo de red ni sus características físicas y de esta manera las aplicaciones no se reconstruyen para los diferentes tipos de red. El conjunto de protocolos genera un modelo el cual es comúnmente llamado internet.

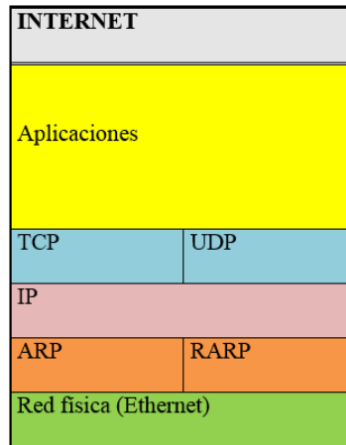


Figura 3: Capas Modelo Internet.

Fuente: (Chaupis Guardia, 2018)

1.5 Android Studio

Las aplicaciones móviles han ganado mucho terreno con el pasar de los años ya que ahora, después de los avances tecnológicos, es raro encontrar a una persona que no posea un dispositivo móvil, por lo que el desarrollo de aplicaciones móviles se ha convertido en un negocio muy amplio y generador de ganancias y de oportunidades. También, con la aparición de los dispositivos IoT los cuales ayudan mucho a las personas en las actividades diarias. Estos se conectan con los dispositivos móviles a través de las aplicaciones, por lo cual es necesario crear un entorno para la conexión y comunicación de estos dispositivos.

Para la creación de aplicaciones móviles existen muchos entornos dependiendo del sistema operativo que posea el celular, para el sistema operativo Android existe un IDE que es el oficial el cual se llama Android Studio, el cual es el entorno de desarrollo

integrado (IDE) oficial y este está basado en IntelliJ IDEA (Developers, 2021). Android Studio ofrece muchas funciones para aumentar la productividad cuando se desarrolla aplicaciones para el entorno Android, de las cuales podemos destacar las siguientes:

- Un emulador rápido y cargado de varias funciones
- Un entorno unificado donde se puede desarrollar para todos los dispositivos dentro del ambiente Android
- Un sistema de compilación flexible basado en Gradle
- Compatibilidad con C++ y NDK
- Compatibilidad con Google Cloud Plataform, que facilita la integración con Google Cloud Messaging y App Engine

(Developers, 2021)

Android Studio tiene muchas ventajas para el desarrollo de aplicaciones para el sistema operativo Android, además de que permite probar y ejecutar la aplicación con su propio emulador y en diferentes dispositivos permitiendo de esa manera observar los avances cuando el producto final esté terminado.

La estructura del proyecto que ofrece Android Studio incluye módulos con archivos de código fuente predefinidos y archivos con recursos para poder trabajar sin problemas. Los módulos que incluyen son Módulos de Apps para Android, Módulos de biblioteca y Módulos de Google App Engine. Dentro de los módulos están las carpetas de manifest en donde se encuentra el archivo AndroidManifest.xml, java el cual contiene el código fuente de Java, incluido con el código de prueba de JUnit, y la

carpeta res en donde se encuentran todos los recursos sin código, es decir, los que se manejan mediante la interfaz gráfica como son los diseños XML, strings de IU y las imágenes de mapa de bits (Developers, 2021).

La aplicación servirá para que el usuario tenga una experiencia más amigable con el sistema IoT a ser instalado en su hogar con el cual podrá controlar los dispositivos para encenderlos y apagarlos como son los focos inteligentes y los aspersores de agua, para recibir notificaciones en caso de que los sensores de movimiento se activen cuando el usuario no se encuentre en su hogar. El sistema será amigable con el usuario además de que el usuario debe poseer un dispositivo móvil con sistema operativo Android y esta aplicación cumplirá con estándares de ciberseguridad para proteger los datos de los usuarios además de que serán conexiones independientes con los demás usuarios de las demás casas.



Figura 4: Android Studio.

Fuente: (Developers, 2021)

1.6 Microcontroladores

En la actualidad los microcontroladores se han vuelto una herramienta indispensable dentro de la computación y podemos encontrarlos en la mayoría de los dispositivos electrónicos, como por ejemplo teléfonos inteligentes, computadores, refrigeradores y muchos otros. El papel principal que cumple un microprocesador es el de accionar determinada secuencia de patrones cuando sucede una determinada acción, esto gracias a su arquitectura que permite almacenar ordenes dentro de su memoria.

Los microcontroladores pueden llegar a asemejarse mucho a una computadora ya que poseen elementos particulares que les permiten almacenar y procesar información de manera eficiente, poseen memoria RAM, EEPROM, CPU, periféricos de entrada y salida. (Díaz Ronceros, 2020).

Una característica primordial de estos chips es que no funcionan a menos que sean programados con una determinada función, esto los hace muy importantes dentro de la ciencia de la computación porque pueden ser usados desde proyectos para la enseñanza de estudiantes o también para controlar procesos industriales complejos como por ejemplo el control de temperatura, humedad dentro de una granja de cultivo.



Figura 5: Microchip.

Fuente: (Díaz Ronceros, 2020)

1.6.1 Arquitectura de un microprocesador

Parte	Descripción
Procesador	Arquitectura Harvard con memoria de instrucciones y memoria de datos
Memoria ROM	Almacenamiento del programa de control
Memoria SRAM	Lectura y escritura de datos obtenidos para procesarlos
Líneas de entrada y salida	Fuente, conexión con periféricos, reset, conexión con cristal de oscilación para regular la frecuencia

Recursos Auxiliares	Dependiendo de función puede contener complementos que ayuden a cumplir su cometido
----------------------------	-------------------------------------------------------------------------------------

Tabla 2: Arquitectura de Microcontrolador

Fuente: (Díaz Ronceros, 2020)

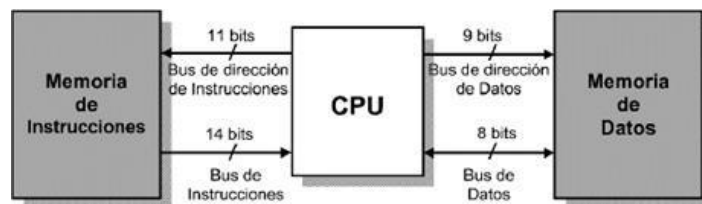


Figura 6: Arquitectura Harvard.

Fuente: (Díaz Ronceros, 2020)

1.6.2 Raspberry Pi

Raspberry Pi es una computadora pequeña, esto debido a que se encuentra fabricado dentro de una placa y cabe dentro de la categoría de computador de placa única o computador de placa simple, lo cual lo hace de fácil transporte y también de bajo costo. Fue principalmente desarrollado para que las personas, principalmente estudiantes, puedan aprender a desarrollar proyectos y entusiasmarse más con el área de la computación. Una característica que comparte igualmente con Arduino es que es de hardware libre (Solis Yagual, 2019)



Figura 7: Raspberry Pi.

Fuente: (Solis Yagual, 2019)

1.6.2.1 Componentes de Raspberry Pi

En la placa Raspberry Pi, podemos encontrar varios componentes que le permiten tener acceso al potencial necesario para ser considerada un computador, podemos tomar como ejemplo la placa actual de Raspberry Pi que es la cuarta versión modelo B, estos componentes son:

- GPU: VideoCore VI
- Memoria RAM: Podemos encontrarla en versiones de 1, 2 y 4 GB
- CPU: Procesador de cuatro núcleos, 1,5 GHz con brazo Cortex-A72
- Sistema en un Chip (SoC): Broadcom BCM2711
- Ranura para tarjeta SD
- Wi-Fi 802.11ac
- Bluetooth 5.0
- Puerto Ethernet
- Puertos de entrada y salida de audio y video

- Puertos USB
- Conector de alimentación USB tipo C
- 40 pines de conexión para posibles acoples o extensiones

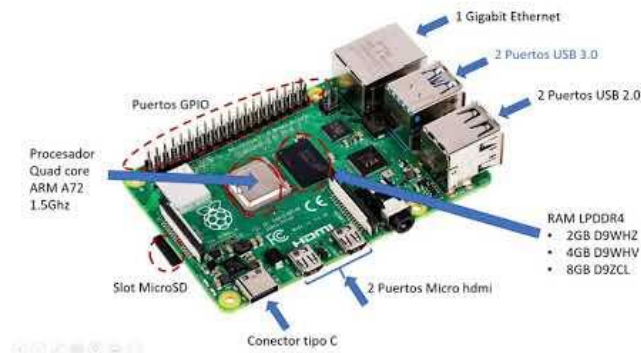





Figura 8: Partes de la tarjeta Raspberry Pi 4.

Fuente: (Creatividad Ahora, 2020)

1.7 Sensores

“Un sensor es un dispositivo encargado de tomar los datos directamente de la etapa denominada como campo y transmitirla hacia un dispositivo controlador” (Solis Yagual, 2019). Como lo menciona Carlos Solis, los sensores son dispositivos capaces de interactuar con el entorno y tomar datos con respecto a este. Son muy útiles en la actualidad y se han puesto de moda gracias a la aparición de los microcontroladores, ya que, gracias a estos, los sensores pueden tener una determinada función, como la de medir la humedad de un terreno y en función de esta, el microcontrolador pueda trabajar y realizar funciones específicas, como la de prender aspersores de agua que rieguen el terreno.

Principales Sensores

Sensor	Característica	Imagen
<p>Sensor de Humedad</p>	<p>Utilizado para riego en sembradíos y terrenos, mide la conductividad del suelo y en función de esta, determina la humedad.</p>	
<p>Sensor de Temperatura</p>	<p>Este sensor digital es capaz de medir la temperatura del medio en el que se encuentre.</p>	
<p>Sensor de Movimiento</p>	<p>Mediante señales infrarrojas, este sensor es capaz de detectar el movimiento de objetos, cumple muy bien su función que es</p>	


	utilizado dentro de proyectos de domótica.	
Sensor Magnético	El sensor magnético actúa como un circuito cerrado cuando las dos piezas que lo confirman están unidas, cuando se separan se hace un corto y envía la señal correspondiente.	

Tabla 3: Principales sensores

Fuente: Propia

2 Capítulo 2

2.1 ¿Qué es IoT?

En la actualidad los avances tecnológicos han sido muchos, sobre todo en cómo se maneja la información dentro y fuera del internet, pero ahora toda esta arquitectura de la información basada en el internet permite el intercambio de bienes y servicios entre cada uno de los elementos, equipos y objetos que se encuentran conectados a la red. Por lo tanto, IoT es la interconexión en red de todos aquellos dispositivos que se encuentran equipados con algún tipo de inteligencia y que tienen un impacto muy fundamental en la seguridad y la privacidad de todos aquellos actores que se encuentren involucrados.

El concepto de que varias máquinas se encuentren intercomunicadas entre ellas y con varios sistemas es conocido como M2M (comunicaciones de máquina a máquina). Los dispositivos electrónicos son varios actualmente, en el año 2020 los dispositivos electrónicos eran de aproximadamente 50 mil millones y para el año 2021 el doble ya que los dispositivos son aquellos que usamos diariamente como puede ser un foco, una cámara, un sensor de movimiento, un aspersor para jardín entre otros (Salazar & Silvestre, 2016), IoT ha provocado un cambio radical en el estilo de vida de las personas ya que ofrece muchas oportunidades de manejo y acceso de datos, servicios más específicos para las necesidades de las personas, cambios en la educación, seguridad y control del hogar, cambios en la salud, entre otros. Las empresas aumentan su productividad gracias a los avances tecnológicos, pero principalmente gracias a los dispositivos IoT que ayudan a crear un mundo mejor un

mundo conectado a internet un mundo conocido como “Smart Planet” (Pérez F & Guerra J, 2017).

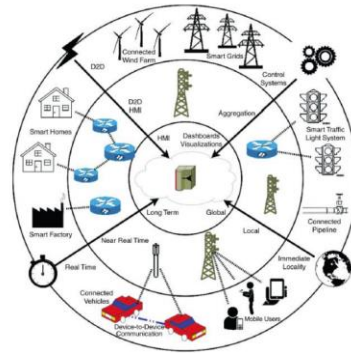


Figura 9: IoT conectadas.

Fuente: (Pérez F & Guerra J, 2017)

2.2 Arquitectura del Internet de las Cosas

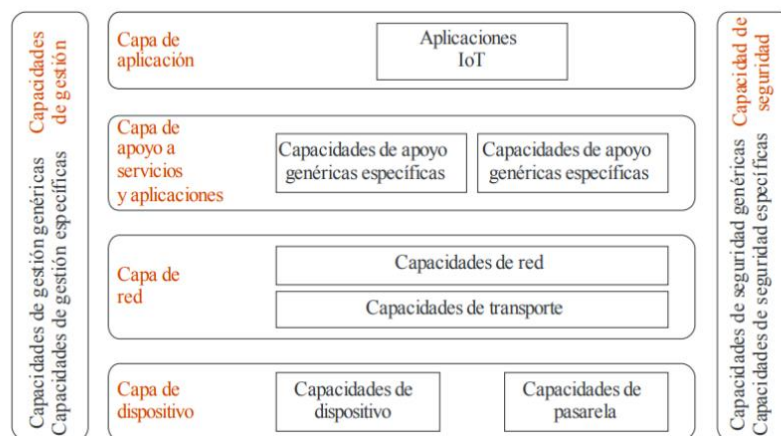


Figura 10: Modelo de referencia IoT.

Fuente: (Chitiva Bernal, 2020)

La arquitectura del internet de las cosas como se puede observar en la **Figura 10** está basada en una arquitectura multicapa en donde los dispositivos se conectarán entre sí para que al usarlos puedan prestar varios servicios por medio de cada una de sus cuatro capas principales. IoT llega a ser una estructura que ofrece varios servicios de aplicación que son habilitados por varias tecnologías. Estos servicios son muy importantes ya que dotan de mucha tecnología a ciudades, transporte, hogar, educación y salud los cuales son proporcionados por varios sensores colocados dentro de los dispositivos, además de servicios en la nube, localización y almacenamiento. Algo muy importante de los sistemas y aplicaciones de IoT es que su diseño esta dado para ofrecer muchos beneficios como son escalabilidad, seguridad, privacidad, confianza, fiabilidad, entre otros.



Figura 11: Aplicaciones IoT.

Fuente: Propia

2.2.1 Capa de aplicación IoT

En la capa de aplicación se encuentran las aplicaciones para IoT y también el control de estas, y esto se da por la computación en la nube y otras tecnologías de computación inteligente que se usan para el manejo de grandes cantidades de información y de control inteligente.

2.2.2 Capa de soporte de servicios y aplicaciones IoT

Clasificación de Descripción capacidades	
Capacidad de soporte genérico	Al utilizar dispositivos IoT, es necesario obtener, procesar y almacenar determinada información con el fin de precautelar el correcto funcionamiento de estos dispositivos.
Capacidad de soporte específico	Son funciones particulares y específicas de los dispositivos IoT, que a su vez pueden ser agrupadas para ofrecer diferentes resultados, por ejemplo, se agrupan sensores de lectura de humedad y temperatura para que se active una función específica.

Tabla 4: Capa de soporte de servicios y aplicaciones IoT

Fuente: (Chitiva Bernal, 2020)

2.2.3 Capa de red IoT

Clasificación de Descripción de capacidades	
Capacidad de Red	Ofrece el control y monitorización de la conexión de las redes, ya que permite obtener el acceso a recursos de transporte, autenticación, autorización y contabilidad.
Capacidad de Transporte	Ofrece conectividad al sistema de IoT para el correcto envío de información, datos de control, gestión y administración.

Tabla 5: Capa de red IoT

Fuente: (Chitiva Bernal, 2020)

2.2.4 Capa de dispositivos IoT

Clasificación de capacidades	Subclasificación	Descripción
Capacidad de Dispositivo	Interacción directa con Red de Comunicaciones	Los dispositivos IoT pueden recibir, enviar y obtener información directamente de la red

	Interacción indirecta con Red de Comunicaciones	Los dispositivos IoT pueden recibir, enviar y obtener información indirectamente de la red mediante capacidades de pasarela
	Redes ad – hoc	Los dispositivos IoT pueden crear redes de este tipo para poder escalar tanto en evolución como velocidad
	Modo reposo – activo	Los dispositivos IoT deben contar con los modos de reposo y activo con el fin de salvaguardar energía eléctrica.
Capacidad de Pasarela	Soporte de Interfaz Múltiple	Dentro de la capa de dispositivo y la capa de red, la capacidad de pasarela permite la conexión de dispositivos de manera alámbrica e inalámbrica, como, por ejemplo: Bluetooth, Wi-Fi, redes móviles 3G, 4G o 5G, ethernet, etc.
	Conversión de Protocolo	Cuando en la comunicación en la capa de dispositivos utilizan protocolos diferentes es necesaria la capacidad de pasarela, con el

		<p>fin de que no existan problemas de comunicación.</p> <p>Igualmente, en la comunicación de capa de red con capa de dispositivo, cuando existen protocolos diferentes es necesaria la conversión de protocolos.</p>
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 6: Capa de dispositivos IoT

Fuente: (Chitiva Bernal, 2020)

2.3 Dispositivos Sonoff

Con los avances tecnológicos lo que siempre se busca es hacer la vida del usuario un poco más sencilla y eficaz ya que en la actualidad el tiempo es muy corto por cuestiones de trabajo de movilidad y de otros factores que se involucran directamente con la vida de las personas, es por lo cual los dispositivos buscan que la vida del usuario sea un poco más sencilla y llevadera y eso es lo que buscan los dispositivos IoT, y las grandes empresas se han encargado de sacar al mercado muchos dispositivos de este índole el único problema es que estos dispositivos poseen un precio muy elevado y no son accesibles para todos a pesar de sus beneficios no abordan a cubrir todo el mercado y varias personas se quedan sin la oportunidad de obtener al menos un dispositivo. Pero los dispositivos se pueden realizar mediante la utilización de Arduino o de dispositivos como los Sonoff que funcionan de muy buena manera y se los puede modificar a gusto propio o gusto del usuario.

Sonoff es el producto ideal de equipos que pueden proveer a los usuarios un control inteligente de su hogar y de su entorno, son de bajo coste y son muy fáciles de modificar además de que se pueden programar con Arduino lo que da muchas facilidades ya que se puede conectar con una aplicación móvil y manejar todo el hogar desde el celular y al permitir conexiones Wi-Fi se puede conectar con varios electrodomésticos (Gálvez Marchán & Mite Vera, 2020). Se puede conectar sensores para medir temperatura, los cuales son muy útiles para colocar aspersores y regar el jardín así los usuarios ya no deben preocuparse de regar su césped ya que el dispositivo lo hará por sí solo, como se puede observar el dispositivos Sonoff tiene varios usos ya que permite modificar su placa para poderla programar y hacerlo útil para varios proyectos de IoT conectándolo a varios electrodomésticos, su bajo coste permite el acceso a todos los usuarios que quieren un control inteligente de su hogar lo cual facilitara mucho su vida haciendo que reduzca gastos como consumo de luz y agua y haciendo que optimice su tiempo realizando otras actividades sin preocupaciones.



Figura 12: Dispositivo Sonoff.

Fuente: Propia

2.3.1 Módulo Sonoff

Un módulo Sonoff este compuesto por cuatro bloques los cuales son la fuente de alimentación, relé, serial/TTL IC y el ESP8266. Sonoff es de código abierto lo que quiere decir que está disponible para todo el mundo para su manipulación y modificación como se puede observar en la **Figura 13** donde se encuentra especificada las partes del Sonoff Basic y en la **Figura 14** se encuentra la vista posterior con una modificación ya realizada en donde se encuentra soldado un cable a un pin para los dispositivos IoT a realizar.

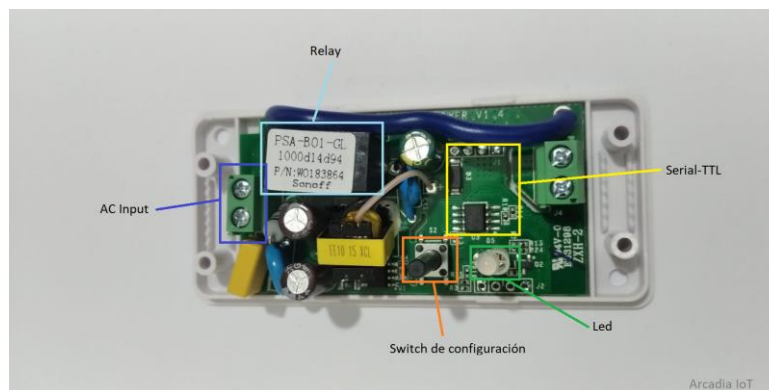


Figura 13: Módulos Sonoff Basic.

Fuente: Propia

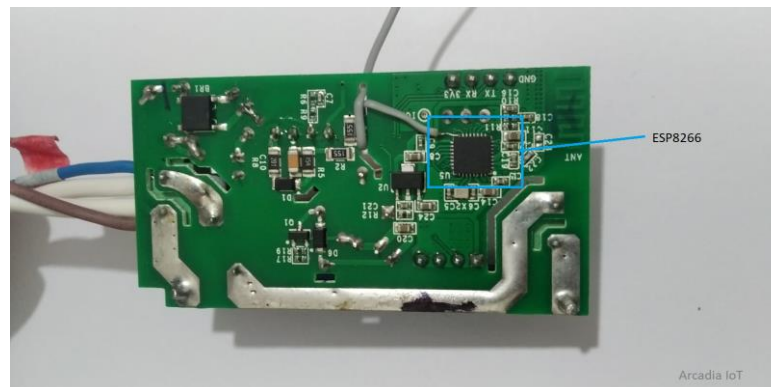


Figura 14: Módulo Sonoff Basic vista posterior.

Fuente: Propia

2.4 ¿Qué es ciberseguridad?

La definición de ciberseguridad según la Asociación de Auditoría y Control sobre los Sistemas de Información es la “protección de activos de información, a través del tratamiento de amenazas que ponen en riesgo la información que es procesada, almacenada y transportada por los sistemas de información que se encuentran interconectados” (Almeida & Herrera, 2019). También se conoce como ciberseguridad al conjunto de herramientas, políticas, conceptos de seguridad, acciones, prácticas idóneas y tecnologías que pueden ser utilizadas para proteger a los usuarios del ciber entorno (Almeida & Herrera, 2019), los usuarios son los más vulnerables a sufrir ataques cibernéticos ya que la mayoría que posee acceso a internet no conoce sobre los riesgos e implicaciones que conlleva tener ese acceso y el hecho también de tener un dispositivos electrónico como puede ser un celular, una laptop, dispositivos IoT, cámaras de seguridad, entre otros. El incremento de dispositivos al ciberespacio incrementó los ataques cibernéticos y ya nadie está exento de sufrir un ataque ya que usamos el internet sin la responsabilidad que se necesita, entramos en links

sospechosos, compartimos contraseñas con otras personas y estos son varios casos que se dan. Con ciberseguridad se busca garantizar que se almacenen y mantengan las propiedades de seguridad de los datos de los usuarios contra los riesgos de seguridad correspondientes en el ciberentorno. La revolución de la tecnología de información y comunicación (TIC) provocó que las grandes empresas comiencen a fabricar de manera masiva dispositivos electrónicos y eso aumento la cantidad de usuarios que los utilizan, según un estudio realizado en el Ecuador en el año 2017, la cantidad de habitantes era de 16 millones y las personas con acceso a internet era de 9 millones y la penetración de la red era del 57%, Ecuador aún no cuenta con una estrategia de seguridad cibernética aunque si ha logrado realizar avances significativos para la mejora de las capacidades cibernéticas y en el enfrentamiento de amenazas (Inter-American Development Bank, 2020). Para los ataques cibernéticos es muy importante conocer las leyes con el fin de saber qué hacer si un ataque de esta índole llegara a suceder. Dentro del Ecuador la ley N ° 2002-67 sobre comercio electrónico, firma electrónica y mensajes de datos describe acerca de las normas que rigen sobre delito cibernético y además señala sobre las reformas pertinentes del código penal (Inter-American Development Bank, 2020). La protección de datos y privacidad cuentan con protección constitucional, en donde la Constitución estipula que los ciudadanos tienen el derecho a la protección de sus datos personales, el problema es que existen leyes y reglamentos pero no hay una ley que cubra con la protección de los datos personales, sin embargo, existe un proyecto sobre la ley de protección de la privacidad de los datos personales y aunque este entro en vigencia antes de la pandemia aún no existe una ley que proteja a los usuarios en una filtración de datos.

En el siguiente cuadro comparativo se pueden observar los cambios y mejoras que ha tenido el Ecuador respecto a ciberseguridad, el estudio fue realizado por Inter-American Development Bank (IDB) con los datos recopilados en el año 2016 y los datos del año 2020.

Indicadores Ciberseguridad Ecuador			2016	2020
Cultura Cibernética y Sociedad	Mentalidad de Seguridad Cibernética		● ● ○ ○ ○	● ● ● ○ ○
	Confianza y Seguridad en Internet		● ● ○ ○ ○	● ● ○ ○ ○
	Comprensión del Usuario de la Protección de la Información en Línea		○ ○ ○ ○ ○	● ○ ○ ○ ○
	Mecanismos de Denuncia		○ ○ ○ ○ ○	● ○ ○ ○ ○
	Medios y Redes Sociales		○ ○ ○ ○ ○	● ○ ○ ○ ○
Formación, Capacitación y Habilidades de Seguridad Cibernética	Sensibilización		● ● ○ ○ ○	● ● ○ ○ ○
	Marco para la Formación		● ● ○ ○ ○	● ● ○ ○ ○
	Marco para la Capacitación Profesional		● ● ○ ○ ○	● ● ○ ○ ○
Marcos Legales y Regulatorios	Marcos Legales		● ○ ○ ○ ○	● ● ● ○ ○
	Sistema de Justicia Penal		● ● ○ ○ ○	● ● ○ ○ ○

Figura 15: Comparación ciberseguridad Ecuador 2016 vs 2020.

Fuente: Propia

Las mejoras en ciberseguridad en el Ecuador no son muchas ni muy significativas, pero al menos son un punto de partida para comenzar a tomar muy en cuenta la seguridad cibernética, en cuanto al área de cultura cibernética y sociedad se ha mejorado un poco en cuanto a la mentalidad de la seguridad cibernética, sobre todo el gobierno dándose cuenta de que debe proteger los datos de los ciudadanos que son muy sensibles y que pueden traer problemas si son expuestos, en cuanto a la confianza y seguridad del internet los números se han mantenido, lo que respecta a comprensión del usuario, mecanismos de denuncia, medios y redes sociales existe una mejora mínima, sobre todo en el hecho de mecanismos de denuncia después del proyecto de la ley de protección de datos. En cuanto al área de formación, capacitación y

habilidades de la seguridad cibernética no existe una mejora como tal, se mantienen los mismos parámetros desde el año 2016, en lo que respecta a programas de sensibilización no hay un aumento, más bien se lo mantiene y esto porque la mayoría de los programas son enfocados hacia las empresas y no hacia los usuarios que es donde existe la brecha y las vulnerabilidades de la seguridad cibernética. Para los marcos legales y regulatorios ya existen varias leyes y varios proyectos de ley para combatir la ciberdelincuencia, sobre todo desde los ataques en donde se expusieron millones de datos de los ecuatorianos de la base de datos de propiedad del Banco de Pichincha en donde hubo filtración de domicilios, de correos, de nombres completos y de números de cédula, dejando muy preocupado al pueblo ecuatoriano, es por lo que, el estado toma cartas en el asunto y comienza el proyecto de ley sobre la protección de datos con lo que se espera poder ya regular los ataques cibernéticos y tener un mejor conocimiento de las redes dentro del Ecuador y sobre todo de la seguridad de los datos de los ecuatorianos.

3 Capítulo 3

3.1 Arquitectura de Solución

Para la construcción de la infraestructura de IoT y ciberseguridad dentro del conjunto “Campos de Arcadia” se propone la siguiente solución observada en la **Figura 16.**

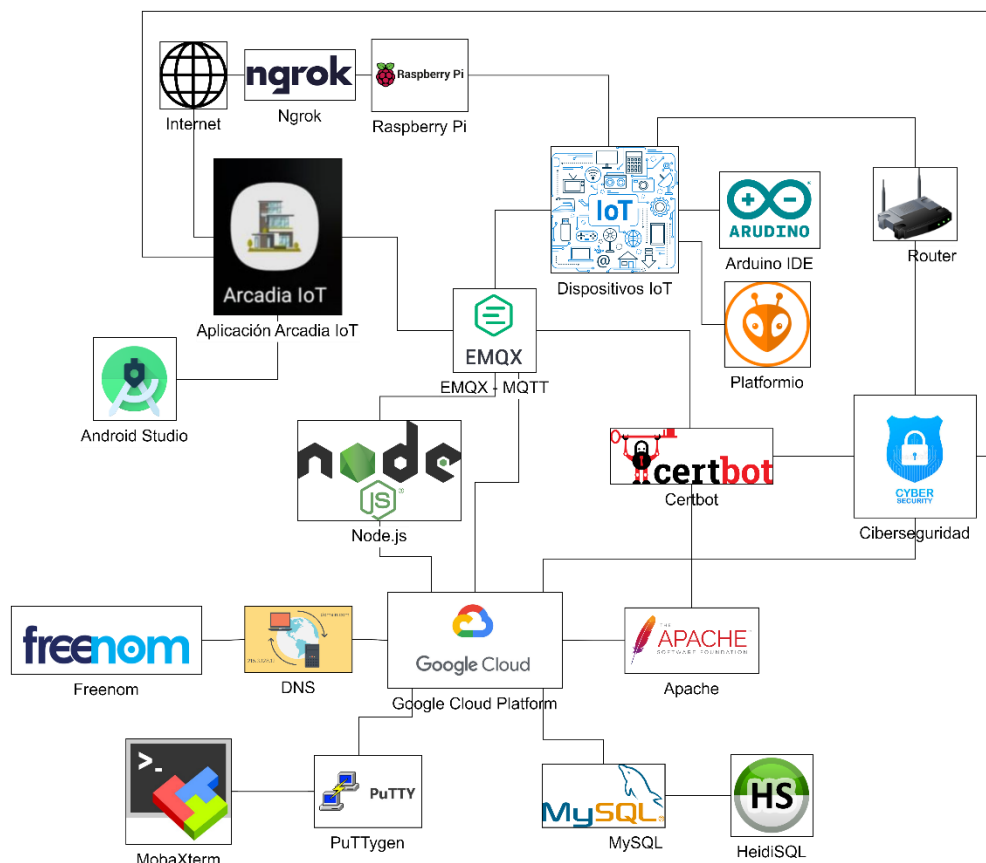


Figura 16: Diagrama Arquitectura de solución.

Fuente: Propia

Lo primordial para nuestro proyecto de titulación fue la creación de una máquina virtual dentro de Google Cloud Platform, se configura su DNS dándole un dominio a su IP Publica el cual se obtuvo a través de Freenom, en donde instalamos

los servicios de MySQL para la gestión de datos el cual se conectó con HeidiSQL para trabajar de manera gráfica, Apache con la finalidad de instalar los certificados de seguridad al proyecto a través de Certbot, Node.js para la conexión y administración entre los dispositivos y las bases de datos, EMQX para poder realizar pruebas y trabajar con el bróker MQTT el cual sirve de nodo de comunicación entre los dispositivos, la base de datos y la aplicación Arcadia IoT, todas estas instalaciones se realizaron a través MobaXterm que nos da la posibilidad de trabajar remotamente en la máquina virtual. A través de Android Studio se creó la aplicación Arcadia IoT la cual se conecta al bróker y de esa manera se comunica con los dispositivos IoT para su control. Para la generación y compilación de código de los dispositivos IoT se utilizó el IDE de Arduino y PlatformIO. Estos dispositivos IoT a excepción de las cámaras se conectan al bróker, ya que las cámaras al no poseer una conexión directa al internet, pero si una IP local Fija, mediante el uso de una raspberry Pi se publica la IP a través de un dominio generado por Ngrok, de esta manera se puede observar dentro de la aplicación la imagen en vivo. Se implemento ciberseguridad a nivel del router configurando los parámetros necesarios para evitar ataques de tipo DNS y ARP, en la aplicación con una pantalla de inicio de sesión con un usuario y una contraseña fuerte para evitar ataques por fuerza bruta, dentro de la máquina virtual con la instalación de certificados de seguridad y configuración de Firewalls.

3.2 Software

3.2.1 Kanban

Kanban es considerada una metodología ágil, puesto que se enfoca en el desarrollo de productos, servicios o software, ajustándose a los cambios que se presentan y entregando resultados rápida y eficientemente. Su origen se remonta a los años 50's en la empresa de Toyota, donde buscaban constantemente mejorar los tiempos de entrega, optimizando recursos, tiempo y llevando consigo un buen control de su inventario. Su éxito fue tal que poco a poco fue adaptándose a las nuevas tecnologías y requerimientos, por esta razón, fue introducida como una metodología para el desarrollo de software en el año 2004 y su objetivo es determinar y mostrar los procesos que deben realizarse en la construcción de un proyecto (Yépez Llerena & Armijos Guillen, 2020).

Esta metodología emplea una pizarra dividida en tres segmentos principales. Los nombres de estos segmentos en inglés son: To Do, Doing y Done, lo que en español quiere decir: por hacer, haciendo y hecho respectivamente. Se colocan en la pizarra todas las tareas que deben realizarse y los integrantes del desarrollo tomarán estas actividades y las realizarán. Una gran ventaja de usar esta metodología es que la pizarra está disponible para todos y pueden observar si una actividad está demorándose o si provoca un cuello de botella.

El desarrollo de software utilizando la metodología de Kanban cuenta con tres reglas importantes. La primera regla es la de visualizar el flujo de trabajo y esto se

logra colocando las tareas a realizarse en el tablero o pizarra, de esta manera todos los integrantes podrán observar las tareas que deben realizarse y tener muy claro el objetivo al que se desea llegar. La segunda regla es determinar el límite del trabajo en cada proceso, abarcar la mayor cantidad de actividades no es sinónimo de mayor productividad ya que esto puede desembocar en cuellos de botella, por esta razón se establece un límite y de esta manera los miembros involucrados en el proceso pueden también avanzar con sus tareas de sin pausas prolongadas. La tercera regla es la de controlar el tiempo que se toma en realizar una tarea, el principal objetivo de esta regla es la de poder optimizar el tiempo y medir el rendimiento que se tiene para completar un ítem (Yépez Llerena & Armijos Guillen, 2020).

Para la realización del proyecto de IoT dentro del conjunto residencial, todo lo referente a software fue dividido en procesos específicos y colocados en una pizarra, gracias a esto, se optimizó el tiempo y los recursos utilizados en el desarrollo de este.

Como se observa en la **Figura 17** y **Figura 19** a continuación, no todas las tareas podían ser realizadas al mismo tiempo y esto es debido a que se llevaba un proceso ordenado y sistemático, en el cual determinadas tareas no podían ser realizadas sin antes haber terminado otras y a su vez, existía un límite de tareas que podían realizarse al mismo tiempo, esto con el fin de evitar sobrecargas o cuellos de botella.

El proyecto y todo lo relacionado con Software, fue dividido en dos etapas. La primera etapa tuvo que ver con todo lo relacionado al servidor y la segunda etapa con lo relacionado al desarrollo de la aplicación y los dispositivos IoT.

3.2.1.1 Primera Etapa



Figura 17: Tablero Kanban Inicio Primera Etapa.

Fuente: Propia



Figura 18: Tablero Kanban-To Do Primera Etapa.

Fuente: Propia

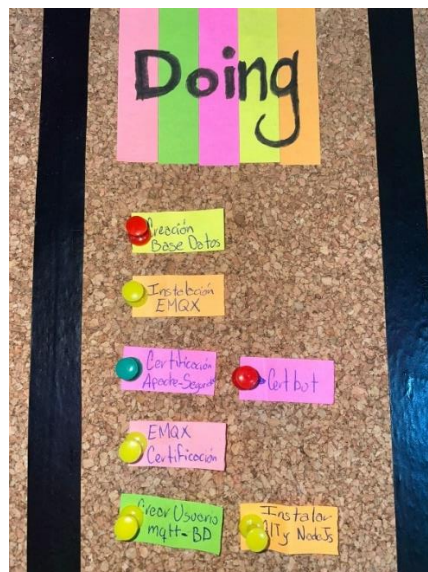


Figura 19: Tablero Kanban-Doing Primera Etapa.

Fuente: Propia

La lista de tareas por realizar es la siguiente:

- Creación Dominio Web
- Creación clave pública y privada
- Creación VM en Google Cloud Platform
- Configuración DNS
- Configuración Firewall VM
- Configuración Dominio Web
- Conexión VM con MobaXterm
- Instalación MySQL
- Instalación Apache
- Instalación y despliegue de base de datos en HeidiSQL
- Configuración MySQL
- Creación Base de Datos
- Instalación EMQX
- Certificación Apache-Seguridad con Certbot
- Certificación EMQX
- Crear usuario MQTT en la base de datos
- Instalar Node.js
- Configuración EMQX en MySQL
- Configuración de seguridad en EMQX
- Crear tabla Estado Casas
- Crear proyecto Node.js

- Ejecutar proyecto Node.js en segundo plano

3.2.1.2 Segunda Etapa



Figura 22: Tablero Kanban Inicio Segunda Etapa.

Fuente: Propia



Figura 23: Tablero Kanban-To Do Segunda Etapa

Fuente: Propia



Figura 24: Tablero Kanban-Doing Segunda Etapa.

Fuente: Propia

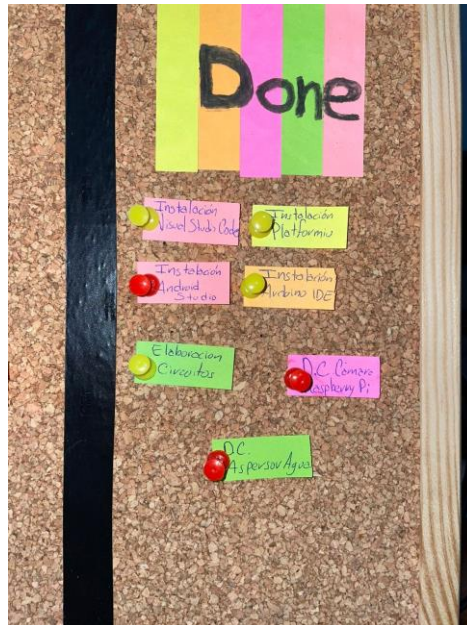


Figura 25: Tablero Kanban-Done Segunda Etapa.

Fuente: Propia

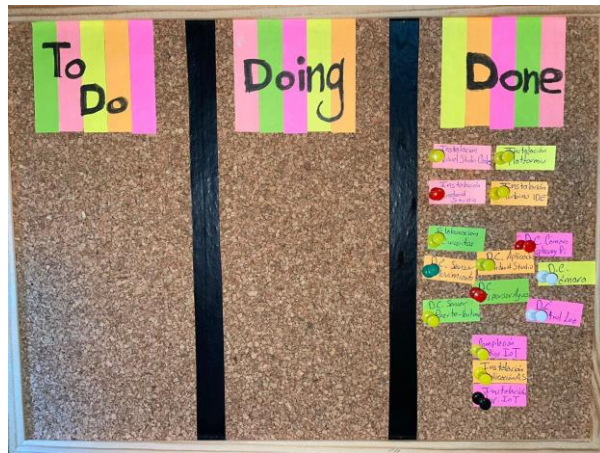


Figura 26: Tablero Kanban Final Segunda Etapa.

Fuente: Propia

La lista de tareas por realizar es la siguiente:

- Instalación Visual Studio Code
- Instalación Platformio
- Instalación Android Studio
- Instalación Arduino IDE
- Elaboración de circuitos
- Desarrollo de código de aplicación en Android Studio
- Desarrollo de código para dispositivo IoT – Control Luz
- Desarrollo de código para dispositivo IoT – Cámara
- Desarrollo de código para RaspberryPi – Servidor Cámaras
- Desarrollo de código para dispositivo IoT – Aspersor de Agua
- Desarrollo de código para dispositivo IoT – Sensor de Movimiento
- Desarrollo de código para dispositivo IoT – Sensor Puerta – Ventana
- Compilación de código en dispositivos IoT

- Instalación Aplicación Android
- Instalación de dispositivos IoT
- Finalización proyecto

3.2.2 Google Cloud Platform

Google cloud platform es la suite de infraestructuras y servicios que Google utilizaba a nivel interno pero que ahora está disponible para su uso externo para cualquier persona o empresa que desee utilizar los servicios de Google. Dentro de Google cloud platform se encuentran disponibles todas las herramientas que Google posee en la nube y que antes se usaban por separado, el entorno nos ofrece todas las herramientas imprescindibles para poder diseñar, hacer testings y hasta lanzar aplicaciones desde gcloud y además de que ofrece una gran escalabilidad y seguridad en comparación a las demás (Ordorica, 2020).

Google cloud platform ofrece una prueba gratuita por el periodo de 90 días o también ofrece una cantidad de 300 dólares para poder utilizarlos con los diferentes servicios que ofrece lo cual es muy útil sobre todo para proyectos en los que se necesita hacer pruebas para el producto final. Para el caso de nuestro proyecto de titulación fue más que necesario el uso de la prueba gratuito ya que los servicios utilizados incluyen una instancia VM con sistema operativo Ubuntu basado en Linux con 2 CPU virtuales y 1 GB de memoria que tiene un costo de \$7,11 y lo utilizamos para la parte de pruebas por lo que no fue necesario de una instancia VM con mejores características. La instancia VM nos sirve para generar el nodo de comunicación para la aplicación y los dispositivos IoT y este es un nodo en común que se encuentra en el internet, en cuando

al uso de la máquina virtual utilizamos MobaXterm en donde nos permite trabajar mediante comandos y en donde nos conectamos a través de la IP externa que la máquina virtual que nos proporciona Google cloud platform, para cumplir con seguridad utilizamos el protocolo de administración remota SSH. SSH o Secure Shell es un protocolo de administración remota que nos permite controlar y modificar nuestros servicios remotos mediante el internet a través de un mecanismo de autenticación por lo tanto se generó dos claves una clave publica, una clave privada y un usuario mediante la aplicación PuTTY para poder ingresar a la máquina virtual por medio de MobaXterm en donde colocamos la clave privada y el usuario la cual previamente fue colocada en la máquina virtual para que se genere la autenticación.

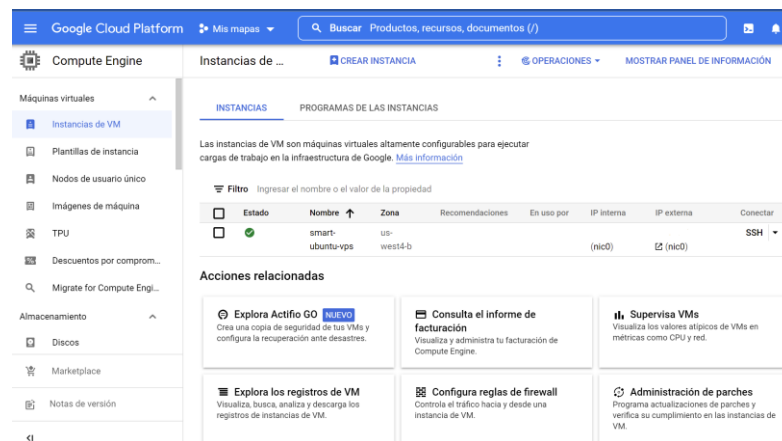


Figura 27: Google Cloud Platform.

Fuente: Propia

3.2.3 MobaXterm

Es uno de los programas más potentes para realizar tareas de administración de ordenadores y servidores, ofrece una gran cantidad de funcionalidades como por

ejemplo incorpora un cliente SSH muy completo, el cual fue utilizado para nuestro proyecto de titulación ya que de esa manera podemos entrar a la máquina virtual creada en Google cloud platform y es más fácil trabajar a nivel de comandos y además de que nos proporciona una interfaz gráfica para el manejo de archivos, dentro de MobaXterm realizamos todas las descargas necesarias para el proyecto y la comunicación de los dispositivos IoT. Partiendo de las directrices que proporciona la metodología de desarrollo Kanban se dividió las tareas, para la descarga de los elementos a utilizar para el proyecto dentro de MobaXterm es importante recalcar que todos deben hacerse en la carpeta raíz de la máquina virtual todo esto a través de comandos.

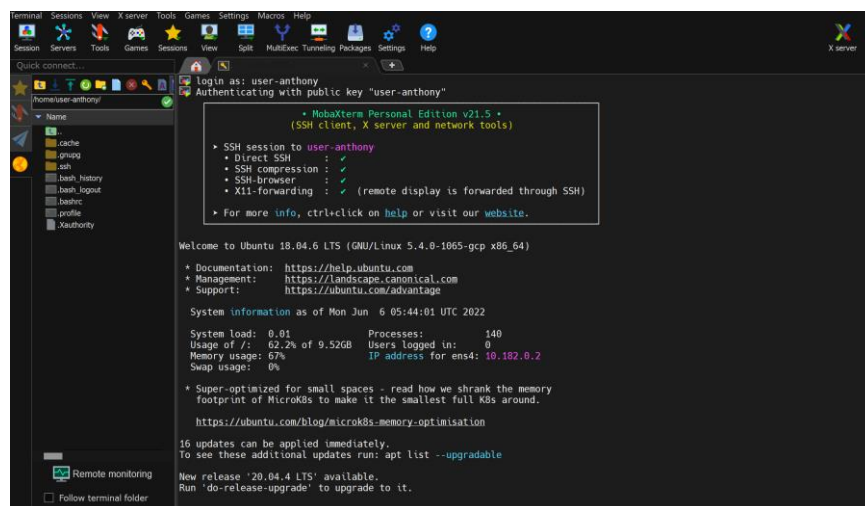


Figura 28: MobaXterm.

Fuente: Propia

3.2.4 DNS

Dentro de la configuración de la máquina virtual se habilitó el tráfico para http y https para poder ingresar vía web. Por lo que se creó un DNS el cual es un sistema

de nomenclatura jerárquico descentralizado para aquellos dispositivos que se encuentran conectados a redes IP como son el internet o una red privada, se utilizó el sistema freenom el cual permite crear dominios de manera gratuita, para el proyecto se creó el dominio de `www.arcadiaiot2022.tk` que es prácticamente la IP externa de la máquina virtual y es a donde todos los dispositivos apuntan.

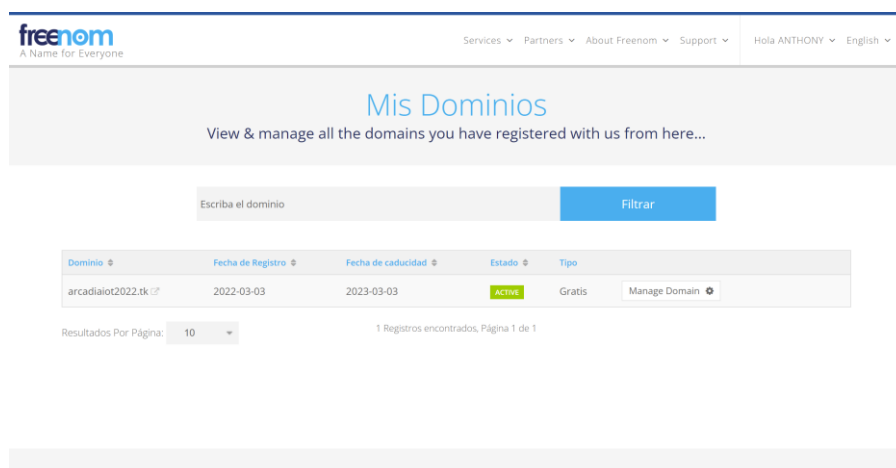


Figura 29: Dominio Freenom.

Fuente: Propia

3.2.5 Apache

Para manejar los entornos de la página web para el apartado de pruebas se utilizó Apache 2, Apache es un software de servidor web que nos permite servir contenido en la web es decir que cuando alguien quiere visitar un sitio web ingresa un nombre de dominio y luego el servidor web en este caso Apache se encarga de enviar los archivos solicitados es decir como un repartidor virtual (Márquez Díaz, Sampedro, & Vargas, 2002). Se utilizó dentro de Apache el permiso Apache full el cual habilita dos puertos que son el puerto 80 que permite un tráfico web sin encriptar y el puerto 443 que ya es un tráfico web encriptado, se editó la página web `/var/www/html` para

tener seguridad y en la página web se dejó solo un enunciado indicando que es una página de prueba y de esa manera nadie ajeno tenga acceso a la información de los dispositivos al estar conectados. Se necesita Apache para instalar los certificados para un entorno seguro, para EMQX es decir los certificados de Apache se apuntan a EMQX solo es necesario indicar la dirección en la que se encuentran.



Figura 30: Apache.

Fuente: (Redacción, 2022)

3.2.6 MySQL

Para el manejo de los estados y comunicación de los dispositivos IoT y para el manejo de ingresos de usuarios y contraseñas es necesario contar con una base de datos en donde se pueda almacenar estos datos y verificar el correcto funcionamiento de la aplicación y de los dispositivos IoT. Para la instalación de MySQL dentro de Ubuntu es importante realizarlo dentro de la carpeta raíz, se realizó la configuración para que tenga un nivel de seguridad fuerte, se habilito el acceso remoto a la base de datos, pero se proporcionó una contraseña para el ingreso al entorno de la base de datos. El manejo de la base de datos se realiza mediante un super usuario con su propia clave de ingreso el cual tiene permisos para manejar las tablas dentro de las bases de datos y también

permisos para manejar los comandos en un entorno gráfico el cual se conoce como HeidiSQL el cual es un gestor de base de datos en este caso de MySQL. La base está escuchando en el localhost lo cual es por defecto por seguridad, pero para poder manejar en un entorno gráfico y que sea más cómodo es necesario cambiar esa configuración para poder trabajar de manera remota, eso se puede volver a configurar cuando ya no sea necesario hacer cambios en la base de datos para que solo escuche en el localhost y cumpla con la seguridad correspondiente. La conexión tiene nombre de “arcadia” por el nombre del conjunto y dentro de la conexión se creó una base con nombre de “EstadoCasas” dentro de la base se encuentran las tablas “casa_1”, “casa_2”, “casa_3” en las tablas se puede visualizar el estado de todos los dispositivos con un 0 y un 1 en caso de encendido y apagado y también se puede ver un estado 2 el cual indica que se activó la alarma de los sensores. El funcionamiento es de la siguiente manera a través de la base de datos nos conectamos con el VPS, es decir al encender la carga el mensaje llega al servidor ese lo direcciona hacia el dispositivo y cuando se ejecuta la respuesta esta regresa al bróker y en ese momento se guarda en la base de datos y la respuesta pasa a los dispositivos, una ventaja es que si existe un corte de luz o un corte de internet los dispositivos guardan el ultimo estado en el que se quedaron por lo que cuando vuelva la luz o el internet estos estarán tal y como quedaron o encendidos o apagados. Se encuentra también la tabla “ClaveCasas” en donde están los datos para el ingreso a la aplicación el usuario y la clave los cuales son dados por nosotros para que el usuario no olvide la clave ni la pierda además de que se espera a futuro realizar un login en el que el usuario pueda ingresar sus datos con una contraseña fuerte para que ninguna persona ajena ingresa a la aplicación y pueda controlar los dispositivos.

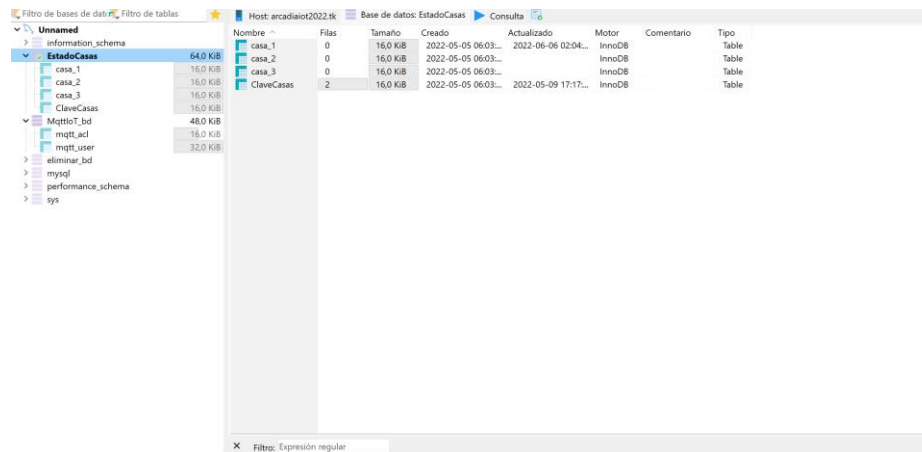


Figura 31: Heidi SQL.

Fuente: Propia

3.2.7 EMQX

Es un bróker MQTT de código abierto altamente escalable el cual nos permite la creación de un clúster con múltiples nodos lo cual nos facilita el manejo de miles de conexiones, es el protocolo de comunicación, es decir, el servidor de paso entre la parte de los dispositivos, el servidor y la aplicación (Vásconez, 2019). Existen varios servidores MQTT, uno de los más conocidos es mosquito, pero el que nos otorga más seguridad es EMQX. La versión instalada para nuestro proyecto de titulación es la versión 4.2.14 para Ubuntu, iniciamos el servicio bróker después de la instalación y se inicia en el puerto 18083 y se encuentra en modo inseguro. Para modo seguro es necesario habilitar reglas de firewall dentro de Google cloud platform en la sección de red de VPC para lo cual se habilitaron varios puertos entre ellos el 18084 que es el puerto seguro para que la información este encriptada. Al ingresar al entorno de EMQX en un inicio se encuentra en modo inseguro, además de que se puede ingresar con la clave predeterminada por lo cual es importante cambiar dicha clave para que

nadie más tenga acceso a la información, el entorno lo usamos para pruebas iniciales. Para poder tener al sitio en modo seguro es decir que la información este encriptada utilizamos certbot para poder crear los certificados, dentro de certbot es necesario indicar que es para software Apache y para el sistema Ubuntu, es por lo cual fue necesario instalar Apache para poder obtener las certificaciones, para su instalación ingresamos todos los comandos como indica en su sitio web e ingresamos nuestro dominio el cual es “www.arcadiaiot2022.tk” y certbot se encarga de instalar los certificados para el dominio y tenemos la ventaja que actualiza los certificados automáticamente. EMQX es el que apunta a los certificados por lo cual se realizó la edición de los archivos de configuración dentro de EMQX, se realizó el cambio al puerto seguro y se apuntó a la dirección de los certificados y el puerto inseguro se lo dejó comentado por lo que no se tiene acceso al puerto, todo se lo realizó a modo de comandos es la razón por la cual se descargó Apache para el proyecto de titulación. De esa manera se llegó a la conexión segura para el dominio por lo que ya se observó como cambio de http a https “https://www.arcadiaiot2022.tk/”. EMQX requiere de un usuario para trabajar con MySQL y es necesario crear el usuario por lo cual se crea la base “MqttIoT_bd”, se crea la tabla “mqtt_user” y dentro de la tabla se encuentra el usuario el cual puede conectarse tanto con la aplicación como los dispositivos para que se conecten con el servidor EMQX es por lo cual se asigna una contraseña al usuario, y los usuarios son por casas es decir cada casa tendrá su propio usuario y su propia contraseña para la conexión de los dispositivos con MQTT y con EMQX y todos los cambios se realizan desde la base en caso de querer una contraseña más compleja o más sencilla.

Es necesario ahora integrar el servicio de la base de datos dentro de EMQX por lo cual ingresamos en el archivo de configuración de MySQL, colocamos el nombre de la base de datos el usuario y la contraseña la cual creamos previamente y con eso establecemos la comunicación de EMQX con la base de datos MySQL, en el panel de pruebas de EMQX dentro de la sección de plugins habilitamos el servicio de la base de datos. Al crear la configuración para la comunicación entre EMQX y MySQL cualquier persona se puede conectar lo cual no cumpliría con estándares de ciberseguridad por lo que es necesario realizar un cambio en la configuración en la cual deshabilitamos la opción de permitir la conexión de usuarios anónimos es decir que solo se podrán conectar los usuarios que poseen las contraseñas para el ingreso. Para poder realizar las pruebas iniciales de los dispositivos dentro del ambiente de EMQX es preciso realizar cambios en el archivo de configuración en donde indicamos el dominio, la clave pública y la clave privada, además de indicar el usuario y la contraseña para seguir con los protocolos de seguridad y solo lo activamos para la parte de pruebas, ya finalizada la parte de pruebas lo desactivamos, de igual manera, utilizamos el software MQTTX para tener un entorno gráfico y sea más sencillo visualizar la comunicación de los dispositivos y que las pruebas sean satisfactorias.

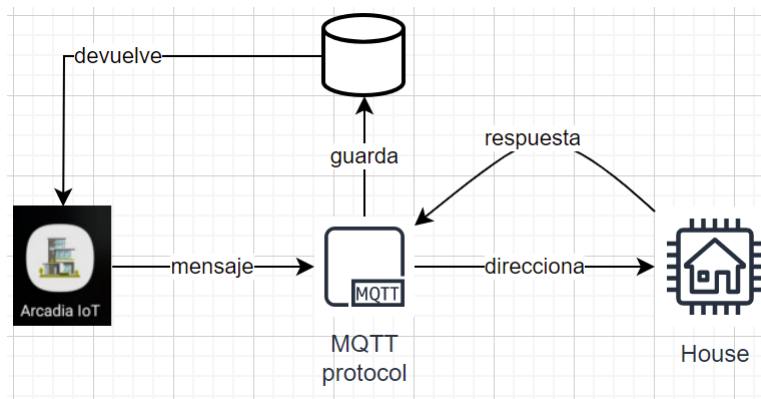


Figura 32: Funcionamiento de los dispositivos IoT.

Fuente: Propia

En la **Figura 32** se puede observar la comunicación y el funcionamiento de los dispositivos IoT para lo cual, el usuario al hacer uso de la aplicación envía el mensaje de la acción y este pasa por el bróker que es el protocolo de comunicación MQTT este direcciona la acción hacia los dispositivos IoT y dependiendo el mensaje que envió la aplicación realiza la acción el dispositivo devuelve la respuesta y esta pasa por el bróker el cual se encarga de guardar la información del mensaje en la base de datos para luego enviar el mensaje de la acción a la aplicación para que informe al usuario la acción que realizo de encender o apagar los dispositivos.

3.2.8 Node.js

Node.js es una plataforma que permite la ejecución de código JavaScript, es orientado a eventos asíncronos y nos permite elaborar aplicaciones de red escalables. Su principal ventaja radica en que solo se pondrán en marcha los métodos dentro del código, cuando exista una llamada a los mismos, caso contrario estos no permanecerán en ejecución. Cabe destacar a su vez que node.js brinda un gran soporte para trabajar

con bases de datos SQL, permitiendo agregar, eliminar o modificar registros dentro de la base de datos (Haro, Guarda, Zambrano Peñaherrera, & Ninahualpa Quiña, 2019).

Es por estas características que se utilizó Node.js en la construcción del proyecto de backend. Como principal actividad, se instala Node.js en el servidor y luego se procede a crear un proyecto dentro de la carpeta raíz. Se instalan los módulos requeridos y los paquetes necesarios para la comunicación con SQL y con el servicio MQTT, como se puede observar en la **Figura 33** y **Figura 34**.

```
Press ^C at any time to quit.
package name: (proyectocasas) control_casas_esp
version: (1.0.0) 1.0.0
description: proyecto casas
entry point: (index.js) index.js
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/ProyectoCasas/package.json:

{
  "name": "control_casas_esp",
  "version": "1.0.0",
  "description": "proyecto casas",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes) yes
npm notice
npm notice New minor version of npm available! 8.5.5 -> 8.7.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.7.0
npm notice Run npm install -g npm@8.7.0 to update!
npm notice
root@smart-ubuntu-vps:/home/ProyectoCasas#
```

Figura 33: Instalación Node.js.

Fuente: Propia

```
root@vps-iot-ubuntu-bionic:/home/DataInNodeJs# ls -l
index.js
package.json
root@vps-iot-ubuntu-bionic:/home/DataInNodeJs# npm install mysql
added 11 packages, and audited 12 packages in 1s

found 0 vulnerabilities
root@vps-iot-ubuntu-bionic:/home/DataInNodeJs# npm install mqtt
added 44 packages, and audited 56 packages in 2s

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
root@vps-iot-ubuntu-bionic:/home/DataInNodeJs#
```

Figura 34: Instalación de MYSQL y MQTT dentro de Node.js.

Fuente: Propia

Dentro del proyecto se localiza el archivo `index.js`, el cual interactuará con los datos que sean ingresados y guardará la información en la base de datos respectiva. Dentro del código de `index.js` nos encontramos con las respectivas conexiones a las bases de datos, verificaciones, estados de los dispositivos IoT y tramas necesarias para el ingreso y comunicación de la aplicación con el servidor y los dispositivos IoT. Código de Node.js (ver el Anexo F).

3.2.9 PlatformIO

PlatformIO es un ecosistema para el desarrollo de aplicaciones IoT el cual nos facilita la gestión del proyecto, de dependencias y con las librerías necesarias para el desarrollo de código, es compatible con más de 400 placas de desarrollo por lo cual nos fue de mucha utilidad para el desarrollo de nuestro proyecto de titulación. Dentro de nuestro proyecto utilizamos varios microcontroladores como son el ESP-8266, ESP8285 y ESP32-WROOM por lo que utilizar el IDE de Arduino no era suficiente además que dentro de las ventajas de PlatformIO es que el código se ejecuta más

rápido, además de que tiene detección de puerto automático, permite el desarrollo remoto, posee un depurador, gestor de librerías privadas, asistente inteligente de escritura, todo esto dentro de Visual Studio Code ya que al tener Visual Studio Code ya solo es necesario instalar la extensión y se podrá comenzar a trabajar de manera inmediata.

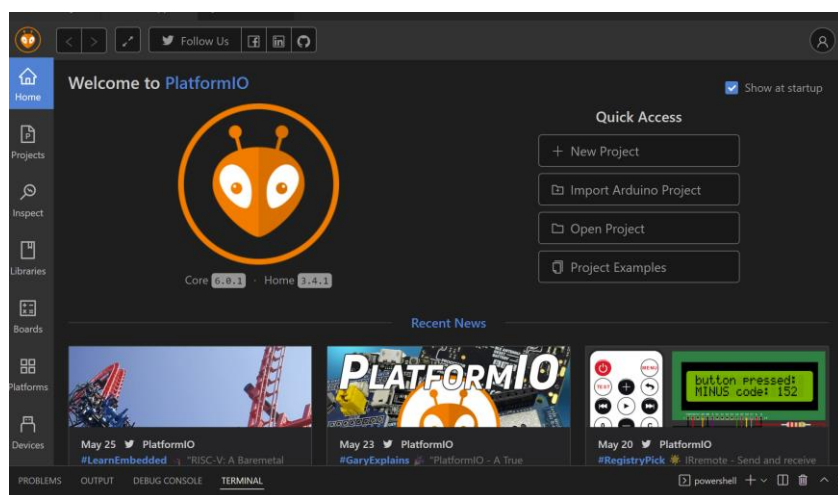


Figura 35: Interfaz PlatformIO.

Fuente: Propia

Como se puede observar en la **Figura 35** dentro de Visual Studio Code solo fue necesario agregar la extensión y ya se puede comenzar a trabajar con PlatformIO, tenemos la sección para crear un nuevo proyecto o importar un proyecto desde Arduino o abrir un proyecto ya realizado además de que también está la opción de ver proyectos de ejemplos que son de mucha utilidad para el desarrollo de nuevos proyectos.

Para el control de nuestros dispositivos creamos varios proyectos uno para cada microcontrolador utilizado ya que cada uno es diferente y necesitaba de compiladores

diferentes. Para el control de luces de sensores de movimiento y sensores de ventanas tenemos las placas ESP-8266 y ESP-8285.

3.2.10 Control Luces

Para los controles de luces el código fue el mismo, lo único diferente fue el compilador, la generación de código para el control de luces comienza con sus respectivas librerías y declaración de variables, se encuentra la parte importante para la conexión a internet en donde es necesario ingresar el nombre de la red y la contraseña para que los dispositivos se conecten a la red de cada hogar, también otro aspecto importante es la configuración MQTT para la comunicación con el bróker y el servidor con su respectivo usuario y contraseña el cual fue creado dentro de la base de datos dentro de la tabla MQTT si este valor se lo cambia dentro de la base de igual manera es necesario cambiarlo en el código para todos los dispositivos ya que todos funcionan con el mismo principio, otra parte a resaltar son los topic donde cada dispositivo apunta a cada casa para que no exista problemas en la comunicación y que cada dispositivo funcione de manera independiente para cada casa, por lo cual al momento de subir el código a los dispositivos se selecciona a que dispositivo pertenece y a que casa pertenece. En el código se encuentra una función de reconexión en caso de que el internet falle o si el dispositivo no encuentra la red en la que después de cada 5 segundos va a tratar de volverse a conectar a la red. En cuanto a seguridad dentro del código esta la parte del del ingreso del nombre de la red y de la contraseña de la red por lo que el dispositivo no se va a conectar a ninguna otra red ya que para hacerlo es necesario ingresar al código y realizar ese cambio también con la conexión MQTT en la que igual se necesita la contraseña y el usuario además del ingreso a la base de datos

3.2.12 Sensores de movimiento

El código para los sensores de movimiento posee el mismo principio que el de control de luces y que el de sensores de puertas y ventanas es decir es necesario del nombre de la red y de la contraseña de la red para que el dispositivo se conecte al internet además de que necesita la conexión MQTT para la comunicación con la aplicación en donde el usuario y la contraseña es el mismo en todos los dispositivos ya que esos datos se encuentran en la base de datos y si se lo cambia en los dispositivos la comunicación no va a funcionar, es por lo cual si se realiza un cambio en la base de datos hay que replicarlo dentro del código para los dispositivos, en cuanto al funcionamiento del sensor de igual manera posee el topic para distinguir los sensores de movimiento y también para distinguir de la casa en la que fueron colocados, los mensajes que recibe de entrada son los de encendido y apagado a través de la aplicación y el mensaje de salida es igual que al sensor de puertas y ventanas da una alarma en la aplicación móvil cuando se activan es decir cuando detectan movimiento lo cual es de mucha ayuda para la seguridad del hogar de los propietarios ya que reciben la alerta de manera inmediata en sus celulares, y en cuando a la reconexión es igual cada 5 segundos busca la señal de internet para poderse reconectar. En lo que respecta a la seguridad del dispositivo sigue el mismo principio que los demás cumpliendo con contraseñas de ingreso y con la conexión a internet única a la red de los propietarios. Código sensores de movimiento (ver el Anexo D).

3.2.14 ESP32-CAM

El módulo de cámara utilizado para el proyecto es el ESP32-CAM, este módulo es muy versátil ya que es capaz de transmitir video, posee conexión Wi-Fi, pines que permiten la configuración de este y su correspondiente alimentación de energía. Para el correspondiente desarrollo del código, se utilizó la aplicación de Arduino, dentro de esta, abrimos la pestaña herramientas y seleccionamos la placa ESP32 Wrover Module, ya que este es el modelo de la placa de la cámara y nos permitirá hacer el streaming de video.

Dentro del código, debemos configurar la red Wi-Fi y la contraseña a la que se va a conectar el dispositivo ESP, también debemos configurar y asignar una dirección IP a este módulo para que pueda conectarse Red Wi-Fi con esta IP estática. Cabe también recalcar que se deben configurar los parámetros de Gateway y subnet de la red. Código de cámaras (ver el Anexo G).

Una vez terminada la configuración del dispositivo ESP32-Cam, se conecta el módulo al computador utilizando un conversor USB serial TTL, conectando los pines RX con TX y TX con RX para que el dispositivo se encuentre en modo de grabación serial, además de esto, se deben conectar todos los pines de GND con el pin GPIO0 y la correspondiente fuente de energía. Se compila el código y se lo sube al módulo, de esta manera estará listo para funcionar.

Hasta el momento, el streaming de video del dispositivo sólo podrá ser visto desde un navegador que se encuentre conectado a la misma red, por lo que es necesario

utilizar una placa Raspberry Pi, la cual funcionará como servidor y podrá enviar el streaming de video hacia el internet. Para poder lograr esto, se necesita definir dominios web para cada una de las cámaras como se puede observar en la **Figura 40**. Es por lo que, utilizamos Ngrok, la cual es una plataforma que provee direcciones web para publicar proyectos en internet.

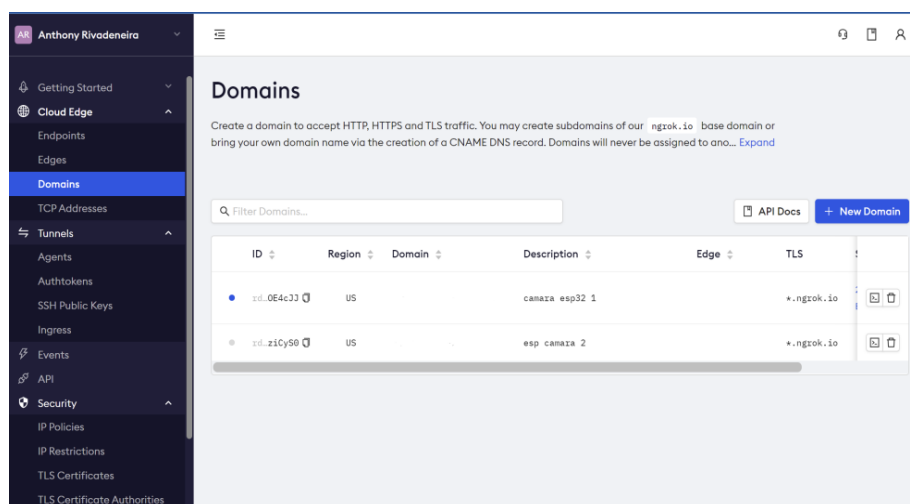


Figura 40: Configuración dominios Ngrok.

Fuente: Propia

Se configura la Raspberry Pi conectándola a la misma red a la que se encuentran conectadas las cámaras, además de esto se crean dos proyectos de Python, uno para cada cámara respectivamente, en el cual se detalla la dirección web a la que va a ser transmitida el streaming de cada cámara con su respectiva IP. El código Python que se ejecuta en la Raspberry Pi se encuentra en el apartado de Anexos.

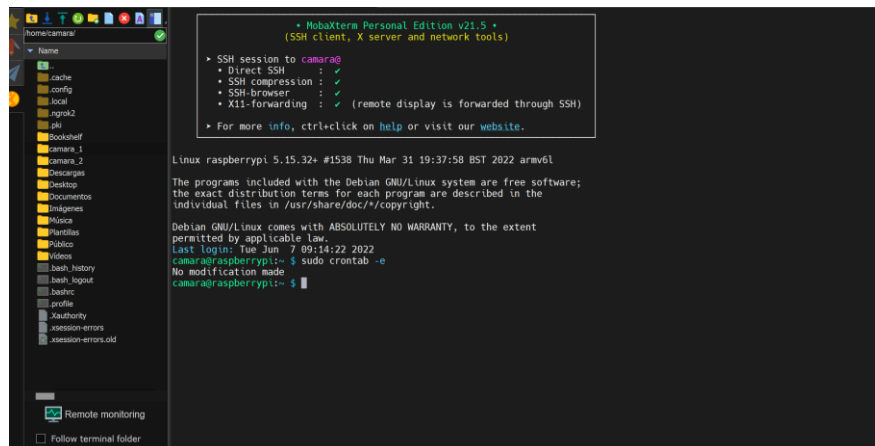


Figura 41: Terminal Raspberry Pi desde MobaXterm.

Fuente: Propia

Dentro de esto surge un problema y es que estos códigos de Python deben ejecutarse siempre, por lo que se configura la Raspberry Pi para que cuando ésta se encienda ejecute automáticamente estos archivos y así se pueda ver el streaming de video por medio de la aplicación desde cualquier parte del mundo. El comando utilizado se lo puede observar en la **Figura 42**, y dentro del comando se especifica los archivos que se va a ejecutar cuando se encienda lo cual se puede observar en la **Figura 43**.

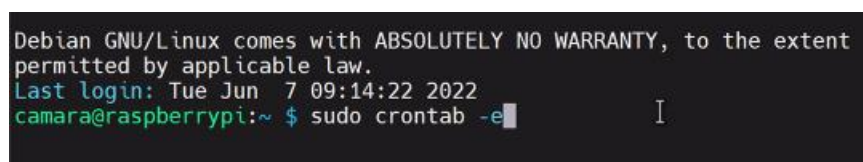


Figura 42: Configuración arranque Raspberry Pi.

Fuente: Propia

```
GNU nano 5.4 /tmp/crontab.QppCtG/crc
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
@reboot /home/camara/camara_1/camara_1.sh
@reboot /home/camara/camara_2/camara_2.sh
```

Figura 43: Crontab configuración.

Fuente: Propia

3.2.15 Ngrok

Ngrok Es una herramienta muy útil cuando se trata de publicar proyectos al internet, transforma el servidor local en un servidor web y de esta manera puede ser accedido por las personas que poseen el dominio web que a su vez es proporcionado de manera automática por Ngrok. (Cordoba Aguilar & Vergara Rodríguez, 2021)

Esta herramienta es de uso gratuito, y ofrece al usuario un dominio generado de manera aleatoria, cada vez que se publica un proyecto. Esto por una parte no es factible para el desarrollo de nuestra aplicación, ya que se necesita de un dominio estático para que pueda apuntar a la dirección IP de los módulos ESP32-CAM.

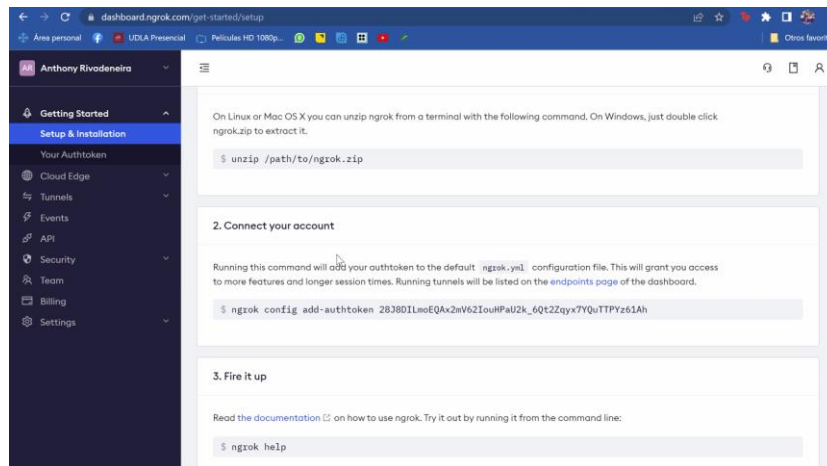


Figura 44: Entorno Ngrok.

Fuente: Propia

Ngrok, ofrece a sus usuarios una versión Pro y una versión Enterprise, sin embargo, estas opciones tienen una suscripción mensual de 25\$ y 65\$ respectivamente, por lo que, para poder tener los dominios estáticos, se adquirió la versión Pro (ngrok, 2022). De esta manera, en el código de Python de la Raspberry Pi, se coloca el dominio estático creado para que apunte a la dirección IP de la cámara y así poder observar el servicio de streaming de las cámaras. Cabe destacar que cualquier persona que posea el dominio web creado puede acceder a las cámaras, pero este, al ser un proyecto piloto, queda en consideración a ser mejorado adquiriendo la versión Enterprise, la cual posee la opción de restringir las direcciones IP que pueden acceder a los dominios.

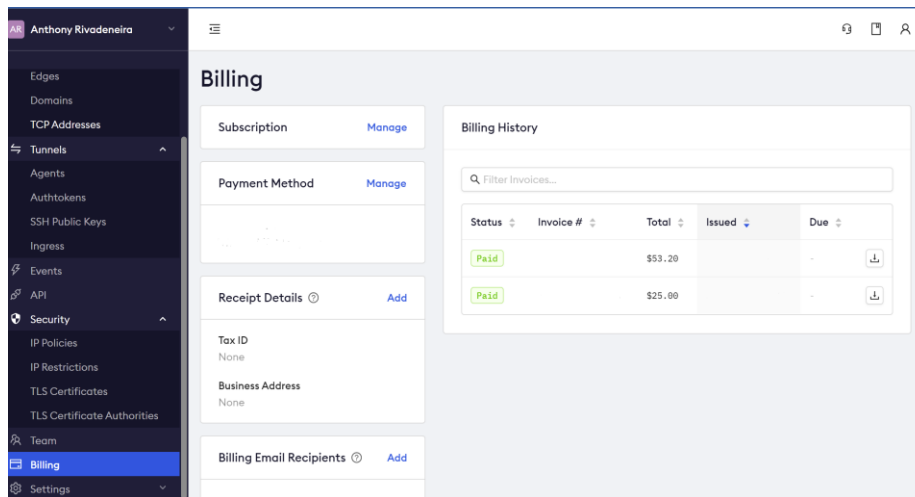


Figura 45: Sección de pago Nginx.

Fuente: Propia

The image shows a pricing card for the Nginx Pro version. The card has a light blue background and contains the following text: 'Pro' at the top, '\$25' in large font, 'per license per month' below it, and 'All FREE features plus...' in bold. Below this, a list of features is shown: Custom Domains, Reserved TCP Addresses, HTTPS & TCP Edges, Configurable Modules per URL Route, OAuth, Webhook Verification, Load Balancing, and No pre-set limits on usage.

Figura 46: Versión Pro Nginx.

Fuente: Propia

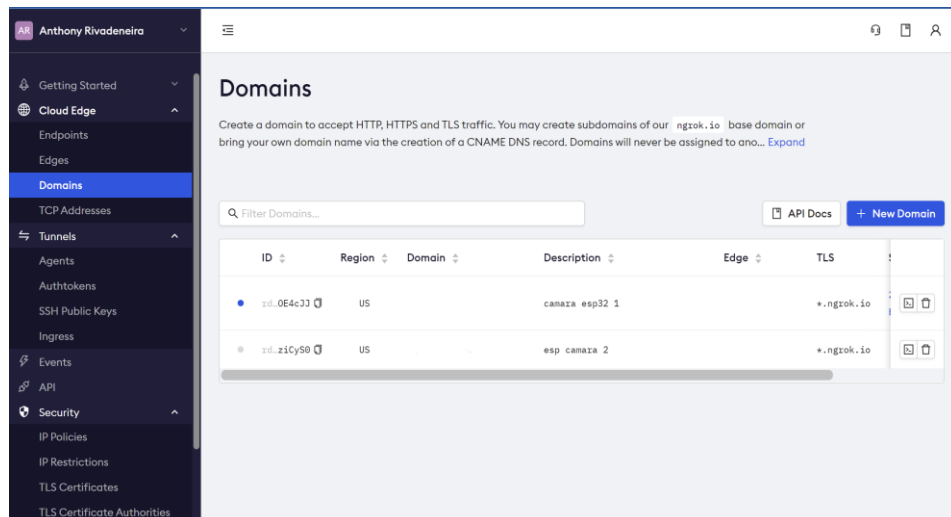


Figura 47: Dominios creados.

Fuente: Propia

3.2.16 Aplicación Android

La aplicación móvil que controlará a los dispositivos IoT, será desarrollada para dispositivos con el sistema operativo Android, por lo que se utilizará la aplicación de escritorio llamada Android Studio. Como fue descrito en el capítulo uno, Android Studio nos ofrece un entorno amigable para el desarrollo de aplicaciones móviles, nos brinda emuladores en los cuales se puede instalar la aplicación que se está desarrollando para poder realizar diferentes pruebas y sus respectivas correcciones.

Empezamos creando un nuevo proyecto en Android Studio, colocando los diferentes parámetros que nos pide para la creación, teniendo en cuenta que se va a empezar desde un proyecto vacío.

Por defecto la aplicación genera un activity con su respectivo Layout, éste lo usaremos para desarrollar la funcionalidad principal de la aplicación, que es la de

controlar los dispositivos IoT. La segunda funcionalidad que la aplicación deberá poseer es la de login, por lo que añadimos un nuevo activity vacío que será el de LoginActivity, es aquí en donde los usuarios deberán colocar las credenciales correspondientes de su hogar para poder ingresar hacia la funcionalidad principal de la aplicación, de esta manera podemos cerciorarnos que ninguna otra persona ajena al conjunto o más específicamente a la casa, pueda controlar los dispositivos, incluso si llega a obtener la aplicación, ya que solamente se proveerá del usuario y contraseña respectivo a los dueños de las casas. Debemos recordar que los usuarios y las contraseñas están almacenadas en una base de datos en la nube. Como podemos observar en la **Figura 48**, al crear el proyecto, se generan los elementos necesarios y la arquitectura de la aplicación, para el correcto funcionamiento de esta. Se ve a su vez la ubicación de los activities y los layouts.

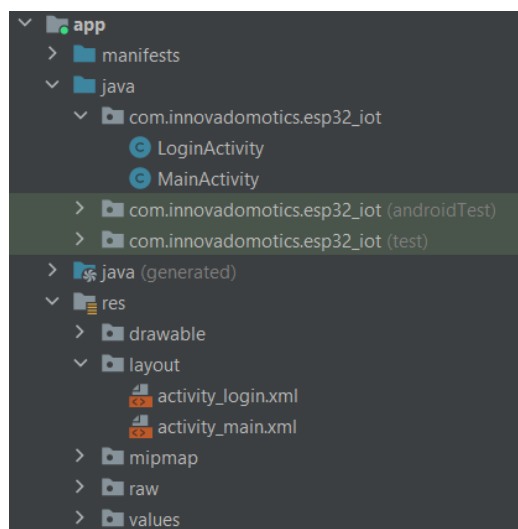


Figura 48: Estructura proyecto Android.

Fuente: Propia

En la carpeta de Layout seleccionamos el archivo activity_login.xml, colocaremos cuadros de texto en donde el usuario podrá ingresar las credenciales

proporcionadas por los administradores de la base de datos para poder ingresar a la aplicación y controlar los dispositivos. Los parámetros son: nombre de usuario y contraseña, existirá además un botón de ingresar para que se pueda verificar la información, esto se lo puede observar en la **Figura 49** y **Figura 50**. Dentro del código de LoginActivity, se importan las librerías y paquetes necesarios. Se declaran los botones y el cuadro de texto utilizados en el Layout, luego se realiza la conexión MQTT por medio del ingreso de las credenciales en el código, de esta manera se verifica si el usuario existe dentro de la base de datos, en el caso de no existir este usuario, se presentará un mensaje que dice error de conexión, caso contrario se presentará el mensaje de conexión exitosa y se desplegará el Layout de activity_main.xml que es en donde se encuentran los controles de los dispositivos IoT. Código de Aplicación Android Studio (ver el Anexo H).

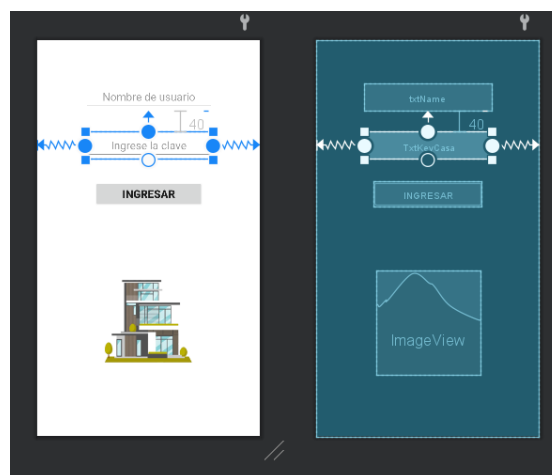


Figura 49: Layout activity_login.

Fuente: Propia



Figura 50: Ingreso Aplicación.

Fuente: Propia

Continuamos con la construcción de la aplicación, para esto abrimos la carpeta de Layout y seleccionamos el archivo `activity_main.xml`, se colocan de manera ordenada los componentes, tales como: botones, switches, cuadros de texto, etc., esto se lo puede observar en la **Figura 51** y **Figura 52**.

Una vez terminada la ubicación y correcta distribución de los componentes dentro del Layout, se procede con el nombramiento individual de cada componente, esto con el fin de reconocerlos al momento de trabajar en el código.

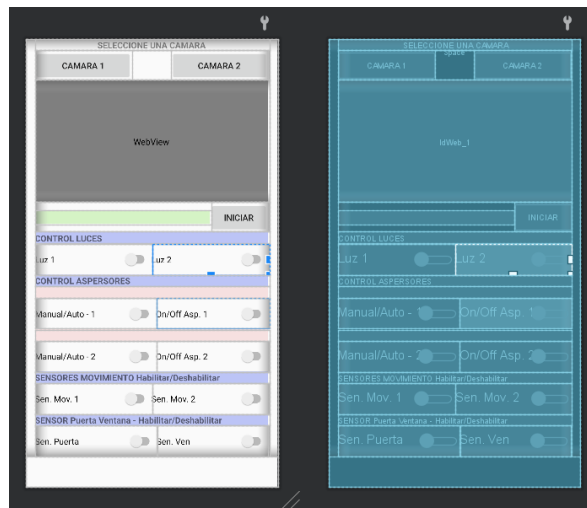


Figura 51: Layout activity_main.

Fuente: Propia

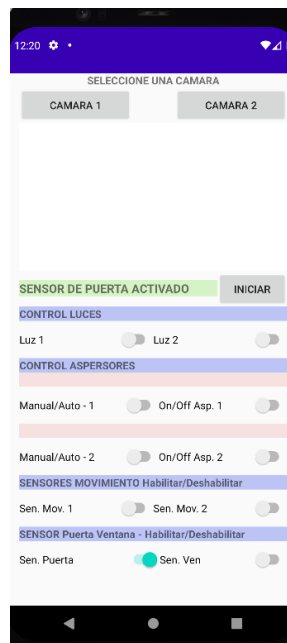


Figura 52: Pantalla principal.

Fuente: Propia

Dentro del MainActivity, empezaremos a desarrollar el código funcional que responderá a las acciones que se realicen del Layout. Se empieza con la importación de librerías y paquetes necesarios para poder desarrollar el proyecto. Al haber

nombrado los componentes dentro del Layout de una manera específica, se nos hará más fácil dentro del MainActivity el declararlos, recordando que cada uno tiene una propiedad específica como por ejemplo botones, switches, cuadros de texto, etc. Código de Aplicación Android Studio (ver el Anexo H).

La aplicación debe contar con parámetros de conexión MQTT para poder acceder a las bases de datos de donde se obtendrán los valores de las tablas, estados y credenciales. Código de Aplicación Android Studio (ver el Anexo H).

Una vez ya autenticado el usuario, se procede con el reconocimiento del hogar al que se está accediendo, y se suscribe a los Topics del hogar que le corresponda, así se obtendrán los estados actuales de los dispositivos IoT, ya sea que estén encendidos o apagados, en otras palabras, los estados se quedan guardados en la base de datos y cuando se suscriba el usuario a estos Topics, se le presentará el estado en el que se quedó. Existe también un cuadro de texto que nos presentará un mensaje después de realizar una acción, por ejemplo, puede ser luz encendida o luz apagada, sensor habilitado o sensor deshabilitado, de esta manera se sabrá si los dispositivos están trabajando correctamente y si se encuentran habilitados para funcionar.

Una vez terminado el proyecto de Android Studio, la aplicación podrá acceder a los registros de las bases de datos, ya sea de usuarios o de estados de los dispositivos y a la vez poder controlar los dispositivos IoT. El usuario podrá tener el control de luces, control de aspersores, habilitar o deshabilitar sensores de movimiento, habilitar o deshabilitar sensores de puertas o ventanas y también tener acceso al control de cámaras para observar el video en vivo que estas transmiten.

3.3 Hardware

3.3.1 Placas Dispositivos

Las placas de los dispositivos del control de luces y de los sensores tienen como base la placa de los dispositivos Sonoff, pero únicamente se lo uso como base del circuito ya que el microcontrolador solo controla por defecto la parte del relé, del led y del pulsador y para poder controlar las demás partes de la placa para los dispositivos se soldó un pin en el ESP para declararlo como entrada y nos sirve también para las conexiones extras para los sensores de movimiento y los sensores de puertas y ventanas. Para poder subir el código a los dispositivos fue necesario soldar unos pines para la grabación los cuales son VCC, GND, RX y TX y se colocan unos postes.



Figura 53: Placa sin los postes soldados a los pines.

Fuente: Propia



Figura 54: Placa con los postes soldados a los pines.

Fuente: Propia

Como se puede observar en la **Figura 53** no tiene aún los pines de carga lo que es como viene la placa del dispositivo, pero para el uso del dispositivo se soldó los pines como se aprecia en la **Figura 54** y de esa manera ya se puede subir el código a los dispositivos para su uso dependiendo al que corresponda.

3.3.1.1 Control luces

Al tener la base de la placa soldamos los postes para los pines de grabación y poder subir el código para el funcionamiento del control de luces. La forma para subir el código al dispositivo es de la siguiente manera al ya tener soldado los postes a los pines se realiza una conexión mediante un convertor USB a serial TTL donde es necesario conectar el convertor con los pines del dispositivo y la conexión es GND se lo conecta con GND luego TX se conecta con RX y RX se conecta con TX esta parte se la realiza cruzada como se observa en la **Figura 56**. El cable de color blanco en la **Figura 56** es el que va conectado a GND ese va directo el cable de color café en la placa está en TX por lo que al conectarse en al convertor USB está conectado en RX

y el cable de color negro en la placa está en RX por lo tanto en el conversor USB está conectado en TX.



Figura 55: Conversor USB a serial TTL.

Fuente: Propia

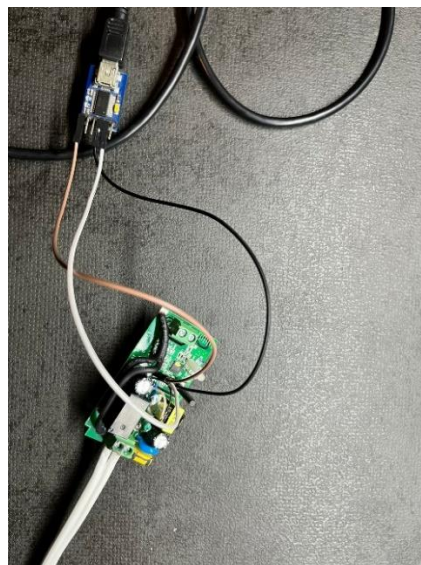


Figura 56: Conexiones entre la placa y el conversor serial.

Fuente: Propia

Luego de realizar las conexiones y conectar el conversor en el puerto USB se procede a compilar el código dependiendo del ESP con el que se está trabajando se selecciona el compilador, se ejecuta y se sube el código en el proceso es necesario tener aplastado el pulsador y no soltarlo hasta terminar el proceso y cuando el código comience a subirse al dispositivo es necesario energizarlo y con ello se termina el proceso y el dispositivo ya se puede conectar a la red y funcionar correctamente.

En cuanto a la entrada del dispositivo el input ingresa la parte que da el voltaje al dispositivo y por la parte de output sale la conexión hacia el foco es decir la salida de energía como se puede observar en la **Figura 57**.

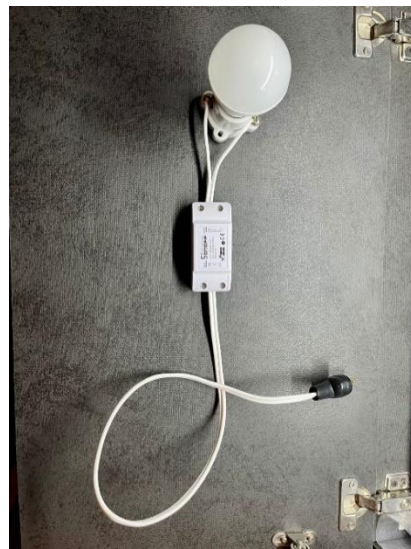


Figura 57: Entrada y salida del control de luces.

Fuente: Propia

El funcionamiento del dispositivo es sencillo como esta explicado en el apartado software en el capítulo 3 al ya estar subido el código en el dispositivo este ya se encuentra conectado a la red por lo que ya se lo puede controlar mediante la aplicación, se lo coloca al control de luces en puntos estratégicos de poca luz en los

hogares y al realizar la acción dentro de la aplicación para encender el foco este recibe el mensaje y el foco se enciende todo por el funcionamiento dentro de la placa y con el proceso explicado anteriormente.

3.3.1.2 Sensor puertas y ventanas

Para los sensores de puertas y ventanas la placa fue la misma, pero fue necesario realizar un cambio muy importante el cual es necesario para poder controlar el sensor, por lo tanto, al microcontrolador ESP se le soldó un pin para conectar el sensor y otro pin se soldó a GND para el voltaje del sensor como se puede observar en la **Figura 58**, de igual manera se soldaron los pines de grabación para poder subir el código.



Figura 58: Conexión a GND del sensor de ventana.

Fuente: Propia

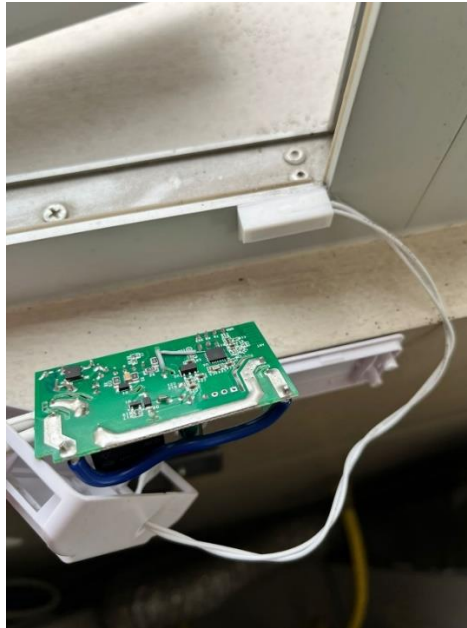


Figura 59: Conexión a ESP del sensor de ventana.

Fuente: Propia

En la forma de subir el código no cambia en nada con respecto al control de luces, es decir al tener ya los postes soldados a los pines se realiza la conexión con el conversor USB a serial TTL de la misma forma GND del conversor se conecta con GND de la placa, RX del conversor se conecta con TX de la placa y TX del conversor se conecta con TX como se observa en la **Figura 60** ya teniendo las conexiones listas el procedimiento es el mismo se compila el código y luego se ejecuta y se sube el código, y es necesario en el proceso tener presionado el pulsador y en el momento que ya busca la señal es necesario energizarlo es decir conectarlo a la electricidad de esta manera ya el dispositivo se conecta a la red y está listo.

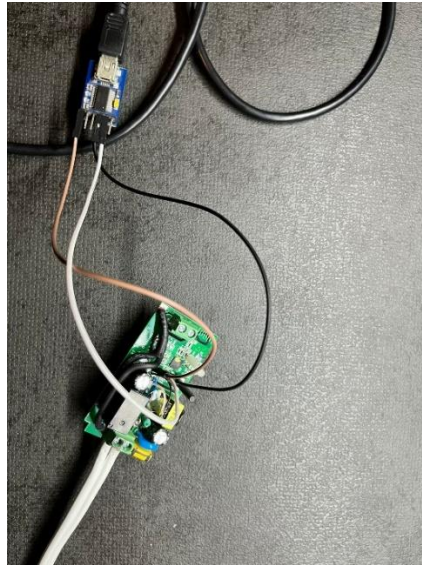


Figura 60: Conexiones entre la placa y el conversor serial.

Fuente: Propia

Para el input ingresa los cables para la energía del dispositivo, pero la diferencia de este con el control de luces es que no tiene un output conectado de momento como se puede observar en la **Figura 61**, pero se lo puede realizar para poder conectar una alarma adicional a la que ya posee la aplicación una que este conectada al dispositivo y que cuando se active esta suene y sea un indicador más para los propietarios, se lo espera poder realizar a futuro.



Figura 61: Input y Output del sensor de puertas y ventanas.

Fuente: Propia

El funcionamiento del dispositivo de igual manera se encuentra en el apartado de software en el capítulo 3, el dispositivo ya conectado a la aplicación se lo coloca en una ventana y una puerta o dos ventanas o dos puertas en los lugares estratégicos en el hogar para que los propietarios puedan tener cierta seguridad adicional en su hogar y tranquilidad ya que si el dispositivo es activado una alarma sonara en el dispositivo e indicara un mensaje del dispositivo que fue activado sea una puerta o una ventana.

3.3.1.3 Sensor Movimiento

En los sensores de movimiento en principio la placa es la misma que las anteriores pero esta posee un cambio para poder conectar el sensor de movimiento, ya que el sensor de movimiento funciona a 5 V necesitamos de un regulador ya que el circuito como tal solo nos da un máximo de 3.3 V por lo que el regulador es de 110 V a 5 V por lo tanto los 5 V alimentan a la parte del sensor de movimiento y la salida del sensor es de 3.3 V de tal manera que la salida sigue estando dentro de los niveles del circuito. La conexión es de la siguiente manera la salida del sensor de movimiento está conectado al pin del ESP para su control, y el GND del regulador está conectado al GND de la placa para así cerrar el circuito como se puede observar en la **Figura 62** en donde se realizaban las pruebas del circuito y ya el circuito completo se puede observar en la **Figura 63** en donde ya se encuentra incluso con la caja en donde se lo coloco.

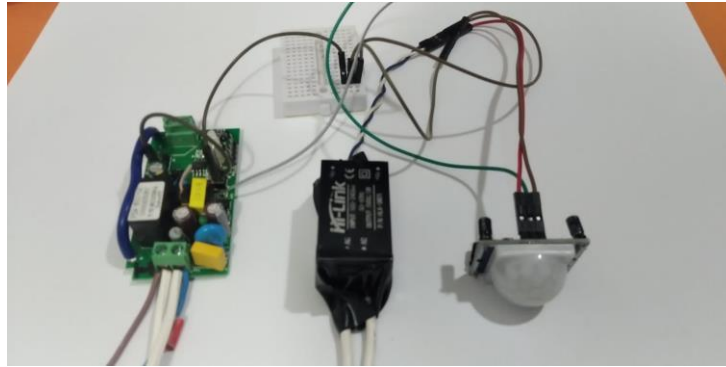


Figura 62: Conexión sensor de movimiento con la placa y el regulador de voltaje.

Fuente: Propia



Figura 63: Circuito completo sensor de movimiento.

Fuente: Propia

De igual manera fue necesario soldar los pines para la grabación de código en el dispositivo, los postes soldados a los pines funcionan de la misma manera que los dispositivos de control de luces y los sensores de puertas y ventanas por lo que es el mismo procedimiento, en la **Figura 64** se puede observar a los cables conectados al dispositivo para subir el código.



Figura 64: Conexiones entre la placa y el conversor serial.

Fuente: Propia

En lo que respecta al circuito en el input igual ingresa los cables para la energía del dispositivo y el output esta sin ocupar por el momento como se observa en la **Figura 65** ya que al igual que en los sensores de puertas y ventanas se puede conectar un dispositivo extra como puede ser una luz para que cuando se active el sensor de movimiento también se active una luz para indicar el lugar y también la alerta que llega a la aplicación móvil por lo que se espera a futuro poder incorporar esta funcionalidad. De igual manera el funcionamiento del dispositivo ya se encuentra en el apartado de software en el capítulo 3 en donde al activarse el sensor de movimiento llega el mensaje y además suena la alarma en el celular para indicar que el sensor ha sido activado.



Figura 65: Input y Output del sensor de movimiento.

Fuente: Propia

3.3.1.4 Aspersores

En el dispositivo IoT de aspersores fue necesaria la creación de un circuito para poder unificar los sensores de temperatura y humedad con el módulo ESP32-WROVER. Dentro de este circuito se suelda el sensor de humedad, el sensor de temperatura con protección de agua, un relé que activará o desactivará la electroválvula, un convertidor DC-DC Step-Down el cual permite reducir el voltaje de 12 V a 5 V, esto debido a que, si bien la fuente provee 12 V sólo es funcional para la electroválvula, mas no para el microcontrolador, ya que este funciona a 5 V.

Por la complejidad de este dispositivo IoT y poseer dos sensores diferentes, es imposible utilizar el dispositivo Sonoff Basic, ya que este solamente provee un pin de control para un solo dispositivo o sensor, y como fue mencionado el dispositivo IoT de aspersores, posee dos sensores.



Figura 66: Dispositivo IoT Aspersor

Fuente: Propia

El funcionamiento de este dispositivo IoT puede ser automático o manual, si se activa la opción de automático, por el código desarrollado y compilado en el módulo ESP32-WROVER se analizará constantemente los parámetros de temperatura y de humedad del suelo, dependiendo de estos factores se activará o se desactivará la electroválvula para poder así regar el patio. Si se activa la opción manual, debemos tener en consideración que el agua seguirá fluyendo a menos que deshabitemos el dispositivo.

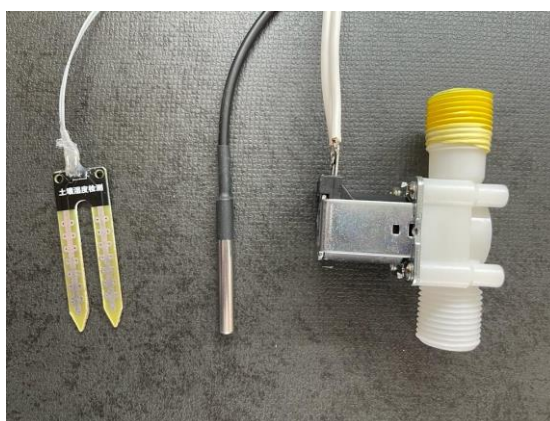


Figura 67: Sensores de temperatura y humedad, electroválvula

Fuente: Propia

3.3.1.5 Cámara

El dispositivo IoT de cámara funciona con el módulo ESP32-CAM, este módulo posee pines de salida lo cual facilita su programación y conexión a una fuente para su funcionamiento. Se realizó un circuito para adaptar y conectar estos pines de una manera más eficiente y también para poder tener acceso a los pines de programación

Para subir el código al módulo, se utilizó el IDE de Arduino, el cual nos permite escribir código y subirlo a los diferentes módulos que soporte la aplicación, en este caso se lo subió en el módulo ESP32-CAM, seleccionando el tipo de módulo y compilándolo.



Figura 68: Módulo ESP32-CAM

Fuente: Propia

Debido a que la fuente de energía entrega un voltaje de 5,5 V, se colocó un diodo para que el voltaje que llegue al módulo sea de 5 V como se puede evidenciar en la **Figura 69**, que es el adecuado y recomendado para su funcionamiento.

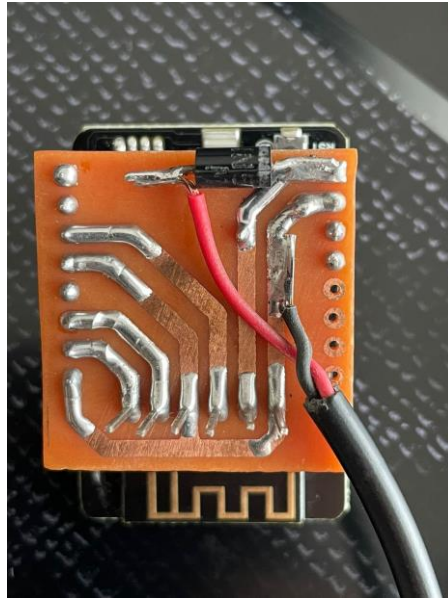


Figura 69: Circuito para módulo de cámara con diodo

Fuente: Propia

3.3.2 Eagle

Eagle es una aplicación desarrollada por Autodesk la cual nos ayuda en la automatización de diseño electrónico, su principal característica es que permite a sus usuarios poder diseñar PCB (Placa Circuitos Impresos) para posteriormente imprimirlas, quemarlas y colocar los dispositivos electrónicos en el circuito (Autodesk, 2022).

Dentro del proyecto utilizamos Eagle ya que se necesitó conectar un módulo ESP32-WROOM a diferentes componentes electrónicos entre los cuales tenemos un relé, un sensor de temperatura, una resistencia para poder medir la humedad del suelo y una electroválvula. Cabe mencionar que estos dispositivos electrónicos funcionan con diferentes voltajes por lo que fue necesario quemar en la circuitería pistas adicionales para la colocación de transformadores de voltaje, resistencias, transistor y

diodos que permitan la transformación del voltaje y que puedan funcionar correctamente los componentes eléctricos.

El diseño de la placa comienza con la colocación de los dispositivos eléctricos y electrónicos que van a ser utilizados en esta, después la misma herramienta nos permite trazar las pistas que conectan a todos estos dispositivos. Se realizaron dos estilos de placas, una para el dispositivo IoT de aspersor y otra para los módulos ESP32-CAM.

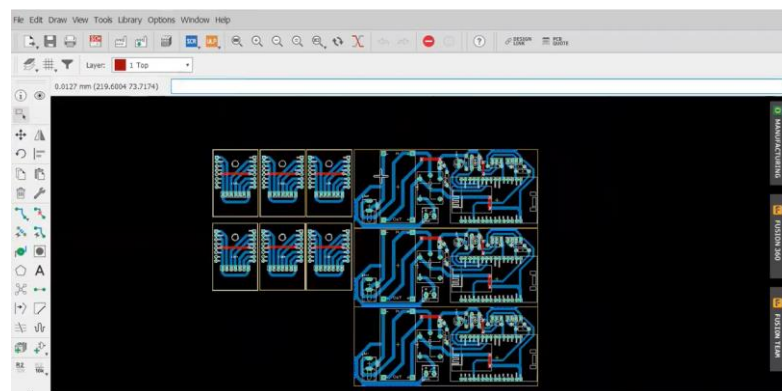


Figura 70: Interfaz Eagle.

Fuente: Propia

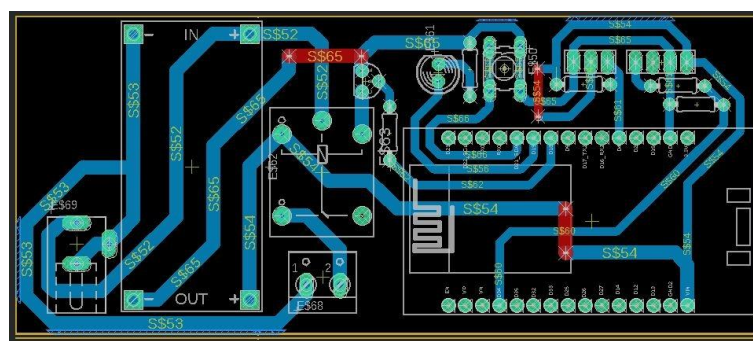


Figura 71: Placa aspersores.

Fuente: Propia



Figura 73: Planchado.

Fuente: Propia

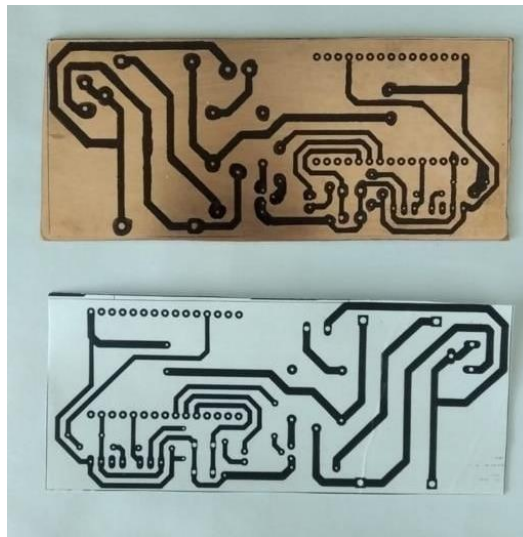


Figura 74: Placa impresa.

Fuente: Propia



Figura 75: Ácido.

Fuente: Propia

Una vez tenemos la placa con las pistas impresas, procedemos a colocarla en ácido para que destruya el cobre excedente y queden solo las pistas. Se lo deja reposar un tiempo, y cuando se lo saca, la placa estará lista con las pistas necesarias. Después de esto, se realizan los agujeros por donde entrarán los pines de los dispositivos como se evidencia en la **Figura 76** y se procede a soldarlos en la placa.

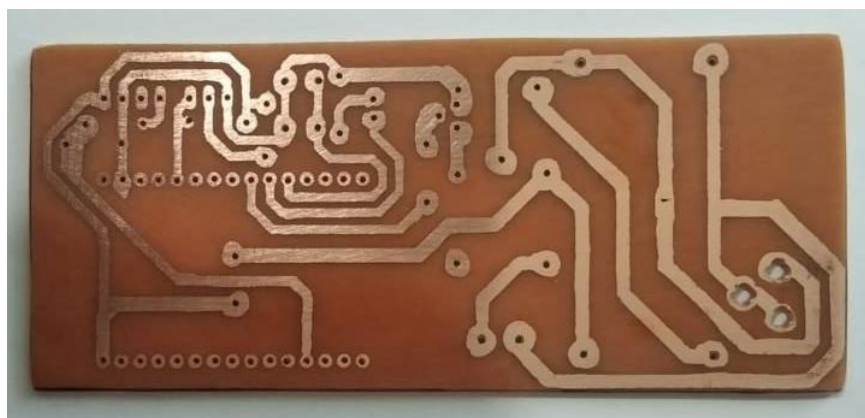


Figura 76: Circuito aspersor.

Fuente: Propia

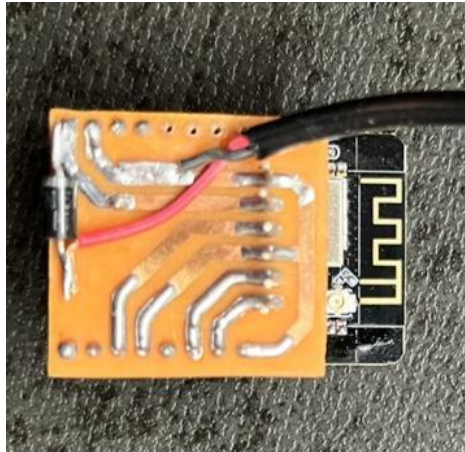


Figura 77: Circuito cámara.

Fuente: Propia

Una vez terminado el proceso de quemado con el ácido, tendremos el circuito listo para colocar y soldar los componentes electrónicos y de esta manera construir los dispositivos IoT.

4 Capítulo 4

4.1 Ataque DNS

El DNS (Sistema de nombres de dominio) es un sistema muy importante de internet, ya que es el encargado de traducir los dominios web a direcciones IP con la finalidad de observar y acceder a las páginas de internet.

Se debe tener un especial cuidado cuando se trata de los servidores DNS, ya que estos son los que dirigen el tráfico web hacia las direcciones IP correspondientes de cada nombre de dominio que poseen las páginas o servicios web. No es de extrañarse que los atacantes quieran adueñarse o controlar estos servidores DNS, ya que pueden llegar a tener acceso y control a información privada, tal como: contraseñas, cuentas, nombres de usuario, entre otras (Pozuelo Saiz de Bustamante, 2020).

4.1.1 Ataque DDOS (Distributed Denial of Service)

Los ataques más comunes a los que están expuestos los DNS son los ataques DDOS los cuales se centran en la denegación de un servicio mediante la sobrecarga de estos, en este caso los servidores, para que así no se les permita procesar las peticiones DNS. Esto es logrado a partir de la amplificación de peticiones, los atacantes envían consultas pequeñas pero que estas a su vez llegan a ser complejas y expandirse a gran escala, de tal manera se logrará colapsar el servicio, como por ejemplo, el de pedir información adicional pero suplantando la dirección IP de origen con la misma

dirección IP del servicio que está siendo atacado, por lo que le llegará a si mismo la informacion extensa, esto lo hacen en reiteradas ocasiones para que llegue a colapsar el servidor (Pozuelo Saiz de Bustamante, 2020).

4.1.2 Ataques Random Subdomain

Este tipo de ataques están enfocados en detener por completo el servicio, para esto, los atacantes envían muchas peticiones DNS a dominios válidos y existentes pero que a su vez incluyen peticiones a subdominios que no existen, generando de esta manera errores los cuales si son abundantes terminarán por saturar el servicio y colapsarlo (Pozuelo Saiz de Bustamante, 2020).

4.2 ARP Spoofing

El ataque ARP Spoofing o también conocido como ataque de suplantación de identidad usando como medio el protocolo ARP, el protocolo ARP tiene como principal objetivo identificar la dirección MAC el cual pertenece a una dirección IP y para lograrlo es necesario enviar un paquete la cual es conocida como la solicitud ARP (Maldonado Simbaña & Pilatuña Collaguazo, 2021). Tomando en cuenta que para navegar por la red para poder enviar paquetes de datos a través de TCP/IP un servidor necesita conocer ciertos datos claves los cuales son la máscara de la subred, la dirección IP y la dirección MAC. Todos los dispositivos reciben la máscara de subred y la dirección IP esto se produce de manera automática al establecerse la conexión, luego se vincula la dirección MAC o dirección hardware con una dirección IP por lo que conocemos como ARP, es un protocolo fundamental para nuestras conexiones.

Pero existe un problema con el protocolo ARP y es que en ocasiones presenta vulnerabilidades y pueden ocurrir ataques como es la suplantación de ARP o conocido como ARP Spoofing.

El ataque ARP Spoofing es muy utilizado por los piratas informáticos ya que es uno de los ataques confiables y acertados ya que prácticamente se basa en el envenenamiento de las tablas ARP, de esta manera logran entrar en una red para el robo de los paquetes de datos que pasan por la red local por lo tanto pueden controlar el tráfico e incluso podrían detenerlo (Jiménez, 2021). Para lograr el ataque los piratas informáticos podrían enviar mensajes falsificados ARP a una LAN, con lo cual conseguiría vincular la dirección MAC con la dirección IP de un servidor lo cual es necesario tal y como esta explicado en el significado de protocolo ARP. Con lo cual consigue recibir cualquier información que ingrese a través de la dirección IP y de esa manera tomar el control del tráfico por completo. Lo que buscan los piratas informáticos es obtener la información del tipo de datos que se transmiten los cuales en su mayoría son datos sensibles como son nombre de usuario, contraseñas, conversaciones, cookies, entre otros. Este ataque puede ser tanto a nivel empresarial como a nivel de un usuario particular como puede ser el propietario de un hogar es por lo cual importante enfocarse en este tipo de ataques por la cantidad de información que pasa por los dispositivos IoT aun que esta información no sea de riesgo para el usuario puede ser una manera de infiltrarse.

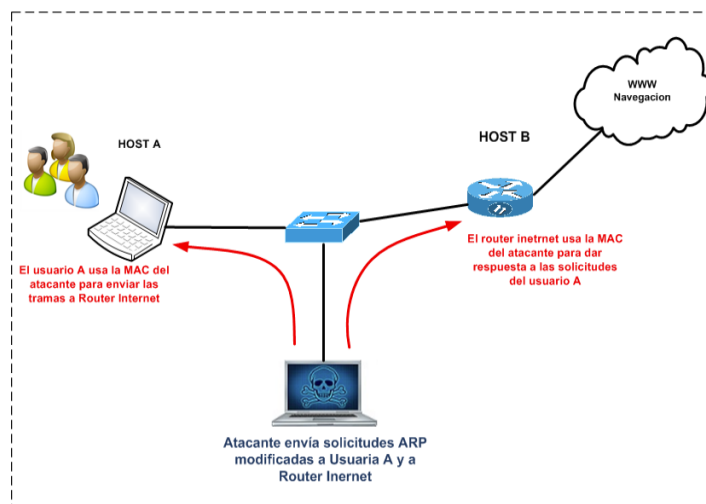


Figura 78: Ataque ARP Spoofing.

Fuente: (Darktrace, 2020)

En la **Figura 78** se puede observar cómo ocurren los ataques por ARP Spoofing por lo que se puede identificar el tráfico originado por el Host A hacia internet el cual es interceptado por el pirata informático lo que genera un impacto muy fuerte ya que la información en su mayoría son datos sensibles de los usuarios que pueden ser interceptados y/o modificados por el ciberdelincuente.

Tomando en cuenta el peligro que involucra este tipo de ataque y además de la cantidad de dispositivos conectados y que proporcionan bastante información, aunque esta no sea sensible es importante de igual manera tomar medidas de seguridad, una de ellas es mediante la aplicación gratuita Arpwatch el cual permite tener un control sobre la actividad del tráfico Ethernet, ya que construye una base de datos en base de las direcciones MAC con su respectiva IP de manera que si en un futuro la aplicación detecta un cambio en estas entradas genera una alarma para su respectiva gestión un problema con esta aplicación es que lleva un tiempo en el aprendizaje de la asignación de la dirección y la detección del ataque y es evidente en grandes volúmenes de

entradas pero la ventaja es como es a nivel de hogar no se manejan grandes volúmenes de datos por lo que es una solución viable para la tranquilidad de los propietarios y usuarios además de que están protegidos para sus actividades diarias en el uso del internet. Otra solución es la seguridad en la red para lo cual como recomendaciones es proporcionar seguridad WPA 2 con una contraseña que cumpla con estándares de letra mayúscula, números y caracteres especiales es decir fuertes y complejas además de tener todos los dispositivos actualizados para evitar los posibles ataques (Jiménez, 2021), en caso de que ocurra un ataque ARP Spoofing la solución es establecer la MAC del Gateway como estática de esta manera se mitiga el ataque ya que los ciberdelincuentes no podrán intervenir en el canal de comunicación (Maldonado Simbaña & Pilatuña Collaguazo, 2021). De esta manera se puede proteger a la red de estos ataques ya que como la ciberseguridad está enfocada al ámbito hogar lo mejor en estos casos es proteger la red para evitar dichos ataques ya que los dispositivos IoT están conectados directamente a la red y al evitar estos ataques se evitan que se puede capturar la trama que da los dispositivos y que los ciberdelincuentes controles los dispositivos.

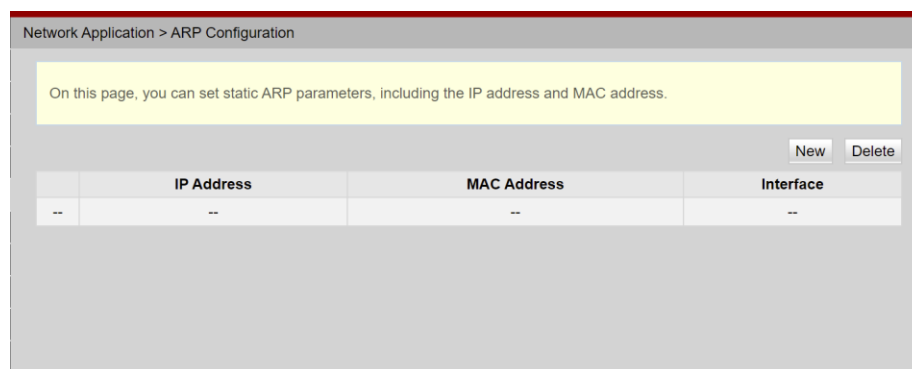


Figura 79: Configuración ARP.

Fuente: Propia

4.3 Recomendaciones de seguridad a nivel de router

A la hora de desarrollar el proyecto, tomamos en consideración varias medidas de seguridad, como por ejemplo la creación de usuarios específicos, contraseñas fuertes para ingreso a bases de datos, certificaciones y una buena práctica de desarrollo de código. Sin embargo, se analizó que puede existir un problema de seguridad a nivel de router ya que se pueden acceder a las tramas e interceptar paquetes enviados, por esta razón, se modificaron los parámetros de seguridad de este. Configuramos el tipo de contraseña de acceso al router, colocándola en modo WPA/WPA2 PreSharedKey. Este tipo de contraseña es muy segura y es la recomendada para el uso de dispositivos IoT dentro de un hogar, existe una única clave con la que todos podrán tener acceso a la red. Existe otro tipo de contraseña segura y esta es la WPA/WPA2 Enterprise, sin embargo, este tipo de contraseña es normalmente utilizado en empresas ya que su característica principal es que se crean credenciales y contraseñas para cada dispositivo que vaya a acceder a la red, este tipo de contraseña no es recomendada para el uso en un hogar.

En la propia configuración del router nos encontramos con la pestaña de seguridad y esta nos da la posibilidad de activar o desactivar determinadas prevenciones ante ataques DoS (Denegación de Servicio). No todos los parámetros a prevenir están activados por defecto por lo que procedimos a activarlos como se observa en la **Figura 80**. De esta manera, el router se encontrará más seguro ante posibles ataques a la red.

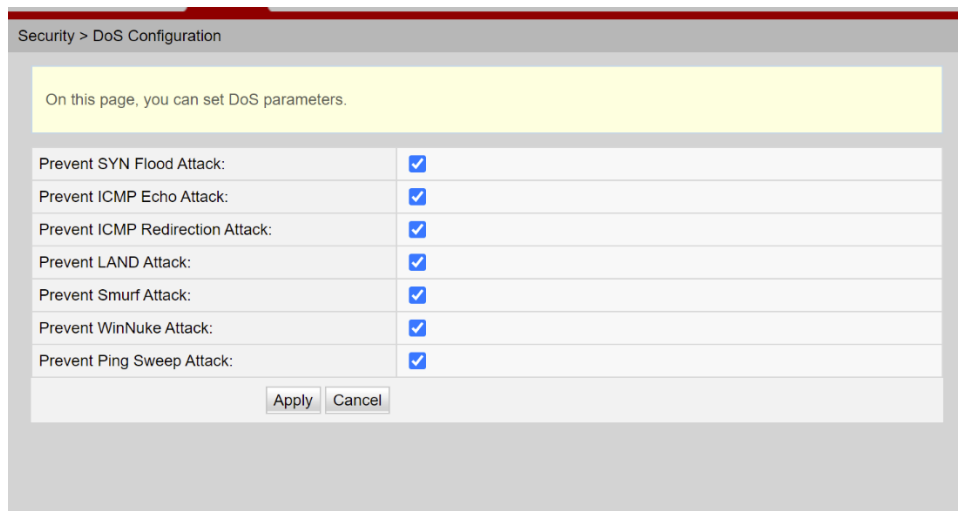


Figura 80: Seguridad en Router.

Fuente: Propia

A continuación, se detallan los tipos de ataque DoS que puede prevenir el router:

- **SYN Flood Attack:** En el ataque SYN (Synchronize) Flood, los atacantes envían una inundación de paquetes TCP / SYN con una dirección IP falsificada, lo que genera que el servidor trate de responder estos mensajes y deja abierto los puertos en espera de respuesta, sin embargo, esto provoca la saturación del servidor y no responderá de manera correcta (Haro Huerta, 2022).
- **ICMP Echo Attack:** Este tipo de ataque se focaliza en ralentizar o detener el tráfico de red. Esto es logrado por los atacantes enviando una gran cantidad de solicitudes ping. Esta solicitud ping sirve para saber si un dispositivo se encuentra conectado y a la escucha para recibir o enviar paquetes. El problema radica en que, al momento de inundar el nodo con estas solicitudes, este nodo solo trata de responder estas solicitudes, y no puede responder de manera correcta o simplemente no puede recibir datos ni información (Silva, 2020).

- **ICMP Redirection Attack:** Este tipo de ataque se centra en poder enviar paquetes de ICMP falsos para que se redirija el tráfico hacia la máquina del atacante, pudiendo así observar los paquetes que se envían y se reciben. El atacante se convierte en man in the middle y puede espiar o modificar las tramas de las víctimas (Martínez Canata & Rodríguez Gómez, 2016).
- **LAND Attack:** Este tipo de ataque ya no es común observarlo debido a que se lo realiza dentro de una red de área local. Los atacantes envían paquetes que contienen la misma dirección tanto en el origen como en el destino, lo que provoca es que el sistema colapse y solo pueda detenerse cuando es detectado o es reiniciado el sistema (Silva, 2020).
- **Smurf Attack:** El ataque smurf tiene como principal objetivo el saturar un servidor inundándolo con paquetes ICMP (Protocolo de control de mensajes de internet) o solicitud ping (eco) a servidores de difusión, lo que amplifica el tráfico y volviéndose un bucle infinito (Haro Huerta, 2022).
- **WinNuke Attack:** Este tipo de ataque logra que se bloquee el ordenador y se ponga la pantalla azul, afecta principalmente al sistema operativo Windows 95 y consiste en enviar un string al puerto 139 TCP (Transfer Control Protocol) (Academic, 2010).
- **Ping Sweep Attack:** Este ataque envía solicitudes ping a manera de difusión amplia hacia los anfitriones de una determinada red. De este modo, el atacante puede saber si se encuentra en línea y puede perpetrar un ataque, probablemente una inundación de solicitudes ping (Zanabria Ticona & Cayo Mamami, 2018).

4.4 Ciberseguridad aplicada

Los dispositivos IoT al estar conectados a la red tienen un cierto riesgo a que sufran de ataques cibernéticos sobre todo que alguien externo podría entrar y controlar los dispositivos o capturar la actividad de estos, también podrían entrar a ver información de las conexiones de los dispositivos claves y datos sensibles otorgados por los propietarios. Por lo cual para el desarrollo de los dispositivos y todo lo que corresponde nos fijamos en la seguridad cumpliendo con ciertos parámetros para poder otorgar más seguridad a los propietarios de los posibles ataques cibernéticos, además cabe recalcar que las seguridades están enfocadas al entorno de hogar.

En cuanto respecta a los dispositivos y su comunicación esta se da a través del servidor MQTT el cual es un protocolo de comunicación para poder comunicarse con la aplicación esta comunicación es cifrada ya que se encuentra dentro del puerto seguro y posee una contraseña y un usuario el cual es el mismo para todos los dispositivos y se extrae desde la base de datos. Para la conexión de los dispositivos con la red de igual manera es necesario en el código poner el nombre de la red y la contraseña para al momento de subirlo al dispositivo este solo se pueda conectar únicamente a dicha red, para poder ver la información del bróker y las conexiones con la base de datos y demás se necesita de una contraseña para poder ingresar la cual cumple con varios parámetros para que no sea descifrada. La página que nos da por defecto Apache que en un inicio da información de la conexión fue sustituida por una página de prueba para que nadie en el exterior tenga acceso a esa información. En cuanto a certificados de seguridad gracias Apache incluimos a certbot que nos da los certificados de seguridad para poder cifrar los datos que se extraen en el servidor además de que esta

conexión se da por el puerto seguro. Dentro de la base de datos se encuentra información sensible de los dispositivos como el usuario y contraseña de la conexión MQTT como también se encuentra los estados de los dispositivos el nombre de ellos y también el usuario y la contraseña de ingreso a la aplicación por lo que tenemos un super usuario con su contraseña para el manejo de la base de datos también posee la contraseña para la conexión de la base de datos por lo que será muy difícil que alguien del exterior pueda tener acceso a los datos. Para la aplicación de igual manera tiene la conexión MQTT la cual posee su contraseña que es la misma de los dispositivos pero tiene algo muy importante que es para poder ingresar a la aplicación es necesario de un usuario y una contraseña el cual será entregado a cada propietario para que pueda realizar uso de la aplicación la contraseña cumple parámetros de seguridad como mayúsculas números y carácter especial para que en caso de que se extravié el celular o este sea robado nadie ajeno al dueño pueda tener acceso a la aplicación. En seguridad para posibles ataques de suplantación de ARP se ingresó a la configuración del router y en el apartado de configuración ARP se puede agregar la IP del Wi-Fi para protegerse de posibles ataques de ARP de igual manera se lo puede realizar a través de comandos en el CMD lo recomendable es realizarlo si se tiene sospecha de un ataque de esta índole.

Cómo parte también de la aplicación de ciberseguridad en el proyecto se procedió con la Configuración DNS por parte del servidor, esto se lo hizo en Google Cloud Platform. Anterior a esto se creó y reservó un dominio web, este es colocado dentro de la configuración DNS y nos ayuda a apuntar los dispositivos IoT a este dominio. De tal manera, se crea una mayor seguridad al no revelar la dirección IP

publica ya que como conocemos, existen varios tipos de ataques DoS que se pueden perpetrar hacia un servidor web.

5 Capítulo 5

5.1 Instalación de los dispositivos

Los dispositivos instalados fueron acorde al pedido de cada propietario para su hogar, lo único diferente en los demás hogares fue la no instalación de cámaras de seguridad, puesto que ya tenían contratado un servicio aparte para ello. En cuanto a los demás dispositivos se cumplió con lo ofrecido a los propietarios con lo que son sensores de movimiento, sensores de ventanas y puertas, controles de luces y el aspersor para el patio trasero.



Figura 81: Dispositivos IoT - Sensor de movimiento y cámara

Fuente: Propia



Figura 82: Dispositivos IoT - Cámara sobre adorno, Sensor de movimiento sobre mesa y Luz debajo de mesa.

Fuente: Propia



Figura 83: Dispositivos IoT - Sensor de Ventana

Fuente: Propia



Figura 84: Dispositivos IoT - Sensor de Ventana

Fuente: Propia

5.2 Encuesta de satisfacción

Para revisar la satisfacción de los usuarios se realizó una encuesta la cual se les hizo llenar después de la instalación de los dispositivos con lo que podemos ver si es que en un futuro los propietarios quisieran adquirir más dispositivos para su hogar. En la **Figura 85** se encuentra la plantilla de la encuesta.

Encuesta de satisfacción

Propietarios Conjunto Campos de Arcadia

Objetivo: Analizar la satisfacción del usuario al tener instalado el sistema de IoT (Internet de las cosas) en su hogar y en la aplicación móvil.

Instructivo: Sea lo más objetivo posible y siga las instrucciones.

1. ¿En una escala del 1 al 5 siendo 1 muy malo y 5 muy bueno como considera que fue la instalación de los dispositivos IoT?

1	2	3	4	5
---	---	---	---	---

2. ¿Consideraría incorporar más dispositivos IoT en un futuro si es así cuáles serían?

SI	NO
----	----

CÁMARA	SENSORES DE MOVIMIENTO	SENSORES DE VELOCIDAD	ASPERISORES	CONTROL DE LUCES
--------	------------------------	-----------------------	-------------	------------------

3. ¿En una escala del 1 al 5 siendo 1 muy malo y 5 muy bueno como considera el funcionamiento de la aplicación? y ¿Por qué?

1	2	3	4	5
---	---	---	---	---

4. ¿Qué mejoras le gustaría en los dispositivos IoT?

5. ¿Qué mejoras le gustaría que tuviera la aplicación móvil?

Figura 85: Plantilla encuesta de satisfacción.

Fuente: Propia

Encuesta de satisfacción

Propietarios Conjunto Campos de Arcadia

Objetivo: Analizar la satisfacción del usuario al tener instalado el sistema de IoT (Internet de las cosas) en su hogar y en la aplicación móvil.

Instructivo: Sea lo más objetivo posible y siga las instrucciones.

1. ¿En una escala del 1 al 5 siendo 1 muy malo y 5 muy bueno como considera que fue la instalación de los dispositivos IoT?

1	2	3	4	5
---	---	---	---	---

2. ¿Consideraría incorporar más dispositivos IoT en un futuro si es así cuáles serían?

SI	NO
----	----

CÁMARA	SENSORES DE MOVIMIENTO	SENSORES DE VELOCIDAD	ASPERISORES	CONTROL DE LUCES
--------	------------------------	-----------------------	-------------	------------------

3. ¿En una escala del 1 al 5 siendo 1 muy malo y 5 muy bueno como considera el funcionamiento de la aplicación? y ¿Por qué?

1	2	3	4	5
---	---	---	---	---

Es fácil de entender y ayuda mucho a la seguridad

4. ¿Qué mejoras le gustaría en los dispositivos IoT?

Que los dispositivos puedan tener la opción de tener diferentes colores

5. ¿Qué mejoras le gustaría que tuviera la aplicación móvil?

No solo Negro
Que sea más llamativa y acorde a aplicaciones como Facebook, es decir, más bonita.

Figura 86: Encuesta llena por propietario.

Fuente: Propia

5.2.1 Análisis de respuestas

Pregunta 1

¿En una escala del 1 al 5 siendo 1 muy malo y 5 muy bueno como considera que fue la instalación de los dispositivos IoT?

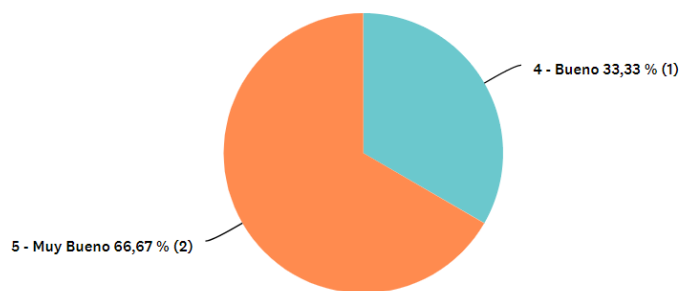


Figura 87: Gráfica de pastel pregunta 1.

Fuente: Propia

Análisis

En base a las respuestas de los residentes, se puede concluir que en la instalación de los dispositivos IoT no existieron inconvenientes o molestias ya que tuvo un alto porcentaje de aceptación. Teniendo un 66,67% en Muy Bueno y 33,33% en Bueno.

Pregunta 2

¿Consideraría incorporar más dispositivos IoT en un futuro si es así cuáles serían?

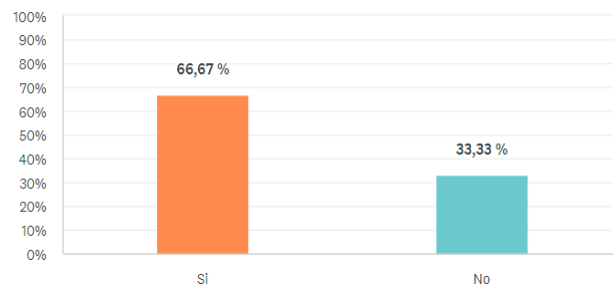


Figura 88: Gráfico de barras pregunta 2.

Fuente: Propia

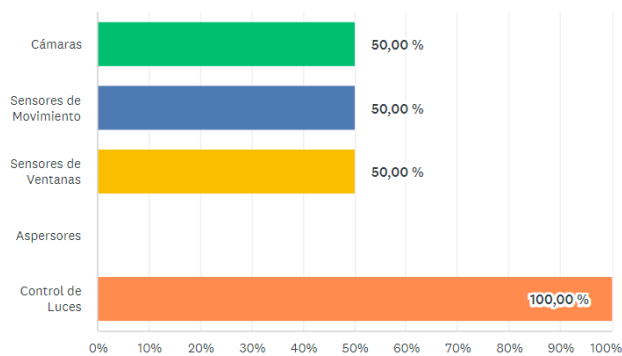


Figura 89: Gráfico de barras pregunta 2.

Fuente: Propia

Análisis

En base a las respuestas de los residentes, se puede concluir que la mayoría de los residentes si desean expandir el número de dispositivos IoT en los que prevalecen control de luces, con un 100% y los sensores de movimiento y de puertas de ventanas con las cámaras con un 50%.

Pregunta 3

¿En una escala del 1 al 5 siendo 1 muy malo y 5 muy bueno como considera el funcionamiento de la aplicación? y ¿Por qué?

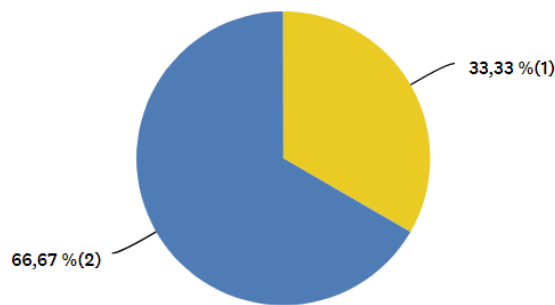


Figura 90: Gráfica de pastel pregunta 3.

Fuente: Propia

Análisis

En base a las respuestas de los residentes, se puede concluir que la aplicación móvil tuvo una aceptación en cuanto a su funcionamiento, teniendo un 66,67% en Muy Bueno y un 33,33% en Bueno.

Pregunta 4

¿Qué mejoras le gustaría en los dispositivos IoT?

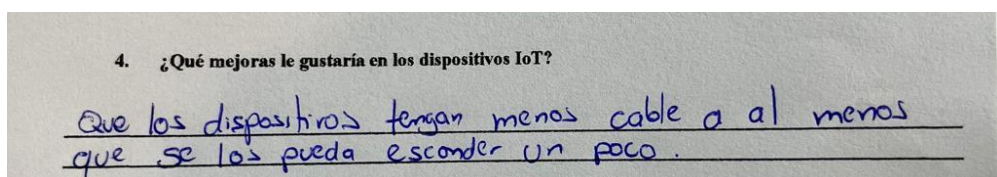


Figura 91: Respuesta a pregunta 4.

Fuente: Propia

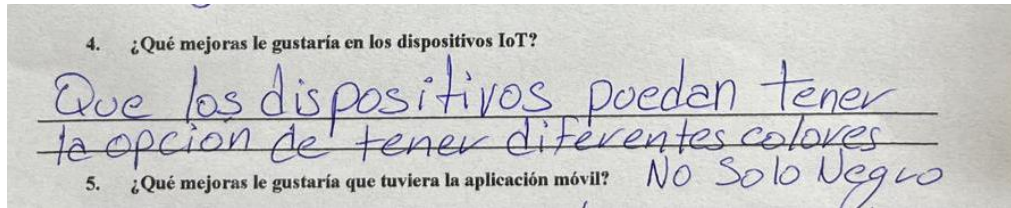


Figura 92: Respuesta a pregunta 4.

Fuente: Propia

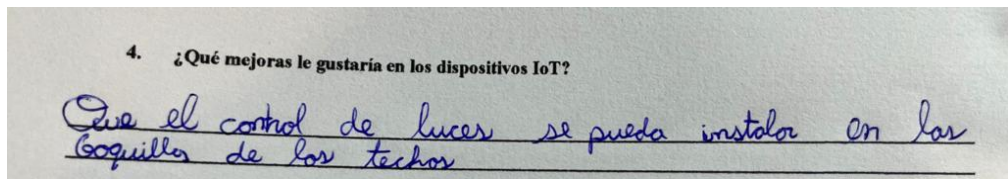


Figura 93: Respuesta a pregunta 4.

Fuente: Propia

Análisis

En base a las respuestas de los residentes, se puede concluir que como posibles mejoras para los dispositivos IoT se tiene que estos tengan más colores disponibles, que en el caso del control de luces estos se puedan instalar en las boquillas del techo y que el uso de cable de los dispositivos no sea tan extenso o que este se lo pueda esconder.

Pregunta 5

¿Qué mejoras le gustaría que tuviera la aplicación móvil?

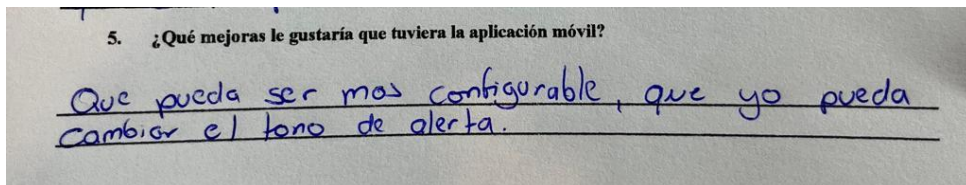


Figura 94: Respuesta a pregunta 5.

Fuente: Propia

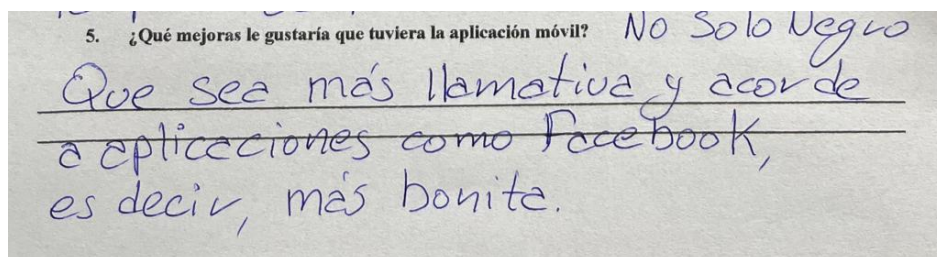


Figura 95: Respuesta a pregunta 5.

Fuente: Propia

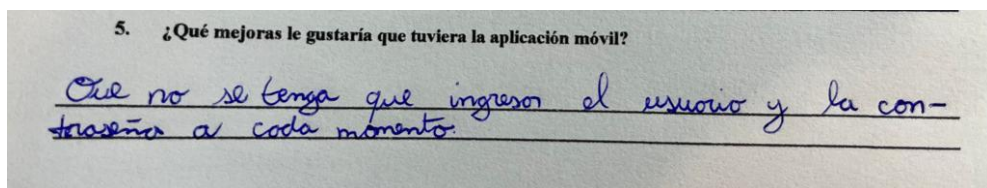


Figura 96: Respuesta a pregunta 5.

Fuente: Propia

Análisis

En base a las respuestas de los residentes, se puede concluir que las mejoras requeridas son tanto a nivel de back-end como de front-end, en su interfaz más

llamativa, en la selección de diferentes tipos de alarmas y eliminando el constante requerimiento de login.

5.2.2 Conclusión general de encuestas de satisfacción

En base a las respuestas recopiladas podemos concluir que tanto la aplicación como los dispositivos IoT tuvieron una muy buena aceptación por parte de los residentes del conjunto y la mayoría tienen planteado adquirir más dispositivos. Se tomará muy en cuenta las recomendaciones de mejora en la aplicación y en los dispositivos para mejorar la satisfacción de los residentes con el fin de expandir el proyecto a un futuro.

5.3 Entrega del proyecto

En las 3 casas se logró instalar todos los dispositivos aceptados por los propietarios para lo cual se presentó una carta de finalización del proyecto al presidente del conjunto para que lo revise y lo firme en donde se indica que se instaló todos los dispositivos y que el mismo nos sirve de respaldo para el trabajo de titulación, en la **Figura 97** se puede observar la carta firmada del proyecto entregado.



Figura 97: Carta de entrega de proyecto.

Fuente: Propia

6 Capítulo 6

6.1 Conclusiones

- Finalizado el presente proyecto de titulación se puede concluir que los dispositivos IoT satisfacen las necesidades de los habitantes del “conjunto Campos de Arcadia” donde su mayor preocupación era la inseguridad que ha estado acechando a nuestra sociedad en los últimos años.
- Con la inseguridad con la que constantemente se vive al navegar en internet además de que cada vez hay más dispositivos conectados a la red se puede evidenciar que se implementó parámetros de seguridad tanto a los dispositivos como a la aplicación y también a la red del hogar para que estos no sean vulnerados y los propietarios puedan estar con la tranquilidad que no van a sufrir ningún ataque y que sus datos personales no se verán afectados.
- El software de platformio es de mucha utilidad al momento de desarrollar código y compilarlo en los microcontroladores además en cuestión a tiempo, procesa el código de una manera rápida y efectiva a comparación del IDE de Arduino, en donde compilar el código y subirlo, toma un largo tiempo.
- En base a la arquitectura de solución propuesta se concluye que, se obtuvieron los resultados esperados al modificar los inconvenientes encontrados durante el desarrollo del proyecto, dejando así, una arquitectura de solución viable para la implementación de IoT y ciberseguridad.

6.2 Recomendaciones

- Por lo observado durante el desarrollo del presente proyecto de titulación, se recomienda a tener en consideración el desarrollo de la aplicación que controle los dispositivos IoT para el sistema operativo IOS, ya que no todos los usuarios poseen un dispositivo móvil con el sistema operativo Android.
- Aumentar el catálogo de dispositivos IoT que se pueden instalar en los hogares, esto debido a que se solicitaron dispositivos con sensores diferente a los que se planteó dentro del alcance del proyecto de titulación, como por ejemplo sensores de humo y de gases.
- El proyecto tiene un gran potencial dentro del campo de servicios de IoT y ciberseguridad y además una gran escalabilidad, por lo que se podría apuntar al desarrollo no solo de hogares si no a la de la automatización de procesos a escala industrial.
- El servicio de cámaras es muy importante ahora para poder identificar las amenazas que suceden todos los días, es por lo cual dentro del proyecto de titulación se planteó un servicio de videovigilancia por streaming utilizando el servicio de ngrok con una conexión a una Raspberry Pi, sin embargo, este servicio tiene un costo de suscripción mensual con un costo relativamente alto para el tipo de proyecto por lo que se recomienda buscar diferentes alternativas para disminuir costos y mejorar el servicio de videovigilancia para que no sea necesaria una conexión extra a un rapsberry Pi.
- Tomando en cuenta las recomendaciones de los propietarios, podemos evidenciar las mejoras a futuro que pueden tener los dispositivos IoT y la aplicación, para que no sea solo un enfoque de hogar sino también para un

edificio donde existen varios departamentos o también un control general para un conjunto.

7 Anexos

7.1 Anexo A Código Python

```
import os

import time

time.sleep(10)

comando_inicial = "./ngrok authtoken -----"

os.system(comando_inicial)

iniciar_camara_1 = './ngrok http --region=us --hostname=-----.ngrok.io
http://255.255.255.255'

os.system(iniciar_camara_1)
```

7.2 Anexo B Código Control Luces

```
#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <EEPROM.h>

#define eeprom_dir_1 10

#define eeprom_dir_2 12

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

#define D0 16 //GPIO16 - WAKE UP

#define D1 5 //GPIO5

#define D2 4 //GPIO4

#define D3 0 //GPIO0 ok ok

#define D4 2 //GPIO2 - TXD1
```

```

#define D5 14 //GPIO14 - HSCLK

#define D6 12 //GPIO12 - HMISO ok ok

#define D7 13 //GPIO13 - HMOSI - RXD2 ok ok

#define D8 15 //GPIO15 - HCS - TXD2

#define RX 3 //GPIO3 - RXD0

#define TX 1 //GPIO1 - TXD0

uint8 eeprom_on_off, eeprom_en_dis, eeprom_main, eeprom_mqtt =0;

const int BotonConfig = D3;

const int ledPin1 = D7;

const int Rele_1 = D6;

int EstBotConfig = 0;

boolean toggle_led = 0;

boolean toggle_rele = 0;

boolean est_OnOff = 0;

const char* ssid = "Nombre de la red";

const char* password = "Contraseña de la red";

//*****

//*** CONFIGURACION MQTT ***

//*****

const char *mqtt_server = "arcadiaiot2022.tk";

//const char *mqtt_server = "111.222.111.222";

const int mqtt_port = puerto en el que escucha;

const char *mqtt_user = "usuario mqtt";

const char *mqtt_pass = "contraseña de usuario";

```

```

//*****

#define EnableCasa_1_1 1

#define EnableCasa_1_2 0

#define EnableCasa_2_1 0

#define EnableCasa_2_2 0

#define EnableCasa_3_1 0

#define EnableCasa_3_2 0

#if EnableCasa_1_1 == 1

const String Topic_Init = "Casa_1";

const String DataIn = "/Luz_1";

const String DataOut = "/EstLuz_1";

#endif

#if EnableCasa_1_2 == 1

const String Topic_Init = "Casa_1";

const String DataIn = "/Luz_2";

const String DataOut = "/EstLuz_2";

#endif

#if EnableCasa_2_1 == 1

const String Topic_Init = "Casa_2";

const String DataIn = "/Luz_1";

const String DataOut = "/EstLuz_1";

#endif

#if EnableCasa_2_2 == 1

const String Topic_Init = "Casa_2";

```

```

const String DataIn = "/Luz_2";

const String DataOut = "/EstLuz_2";

#endif

#if EnableCasa_3_1 == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/Luz_1";

const String DataOut = "/EstLuz_1";

#endif

#if EnableCasa_3_2 == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/Luz_2";

const String DataOut = "/EstLuz_2";

#endif

//-----

// TOPIC OUT => Casa_1/Luz_x  MENSAJES QUE RECIBE  on  off

//*****

WiFiClient espClient;

PubSubClient client(espClient);

void EnviarMsgTopic(String StrMsg, String TopicSend);

//*****

/** DECLARACION FUNCIONES **

//*****

void setup_wifi(void);

void callback(char* topic, byte* payload, unsigned int length);

```

```

void reconnect(void);

void setup()
{
    Serial.begin(115200);

    EEPROM.begin(50);

    pinMode(ledPin1, OUTPUT);

    pinMode(Rele_1, OUTPUT);

    pinMode(BotonConfig, INPUT_PULLUP);

    Serial.println("Iniciando ConfESP .....");

    digitalWrite(Rele_1, 0);

    setup_wifi();

    client.setServer(mqtt_server, mqtt_port);

    client.setCallback(callback);

    digitalWrite(ledPin1, HIGH);

    toggle_rele = 0;

    eeprom_main = 1;

    eeprom_mqtt = 0;
}

void loop() {

    if((eeprom_main == 1)&&(eeprom_mqtt == 1))

    {

        eeprom_main, eeprom_mqtt = 0;

        eeprom_on_off = EEPROM.read(eeprom_dir_1);

        if (eeprom_on_off == 1)

```

```

{
  EnviarMsgTopic("on",DataOut);

  est_OnOff = 1;

  toggle_rele = 1;

  digitalWrite(ledPin1, LOW);

  digitalWrite(Rele_1, toggle_rele);
}

else

{
  EnviarMsgTopic("off",DataOut);

  est_OnOff = 0;

  toggle_rele = 0;

  digitalWrite(ledPin1, HIGH);

  digitalWrite(Rele_1, toggle_rele);
}
}

EstBotConfig = digitalRead(BotonConfig);

if (EstBotConfig == LOW)

{
  delay(50);

  EstBotConfig = digitalRead(BotonConfig);

  if (EstBotConfig == LOW)

  {
    Serial.println("Control Manual");
  }
}

```

```

String StrMsg;

toggle_rele =! toggle_rele;

if (toggle_rele == 1){

    StrMsg = "on";

    est_OnOff = 1;

    toggle_rele = 1;

    digitalWrite(ledPin1, LOW);

    digitalWrite(Rele_1, toggle_rele);

    EEPROM.write(eeprom_dir_1,1);EEPROM.commit();

}

if (toggle_rele == 0){

    StrMsg = "off";

    est_OnOff = 0;

    toggle_rele = 0;

    digitalWrite(ledPin1, HIGH);

    digitalWrite(Rele_1, toggle_rele);

    EEPROM.write(eeprom_dir_1,0);EEPROM.commit();

}

EnviarMsgTopic(StrMsg,DataOut);

delay(400);

}

}

if (!client.connected()) {

    reconnect();

```

```

    }

    client.loop();
}

//*****

//***  CONEXION WIFI  ***

//*****

void setup_wifi(void)
{
    delay(10);

    // Nos conectamos a nuestra red Wifi

    Serial.println();

    Serial.print("Conectando a ");

    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        digitalWrite(ledPin1, toggle_led);

        Serial.print(".");

        delay(500);

        toggle_led =! toggle_led;

    }

    Serial.println("");

    Serial.println("Conectado a red WiFi!");

    Serial.println("Dirección IP: ");

    Serial.println(WiFi.localIP());
}

```

```

}

void callback(char* topic, byte* payload, unsigned int length)
{
    String incoming = "";

    Serial.print("Mensaje recibido desde -> ");Serial.print(topic);

    Serial.println("");

    for (int i = 0; i < length; i++)
    {
        incoming += (char)payload[i];
    }

    incoming.trim();

    Serial.println("Mensaje -> -> -> ->" + incoming);

    if (incoming == "on"){

        Serial.println("SE PROCEDE A ENCENDER LA LUZ");

        toggle_rele = 1;

        digitalWrite(ledPin1, LOW);

        digitalWrite(Rele_1, toggle_rele);

        EnviarMsgTopic("on",DataOut);

        EEPROM.write(eeprom_dir_1,1);EEPROM.commit();

        est_OnOff = 1;

    }

    if (incoming == "off"){

        Serial.println("SE PROCEDE A APAGAR LA LUZ");

        toggle_rele = 0;

```

```

digitalWrite(ledPin1, HIGH);

digitalWrite(Rele_1, toggle_rele);

EnviarMsgTopic("off",DataOut);

EEPROM.write(eeprom_dir_1,0);EEPROM.commit();

est_OnOff = 0;

}

}

void reconnect(void)
{
while (!client.connected())
{
Serial.print("Intentando conexión Mqtt...");

// Creamos un cliente ID
String clientId = "EspLuz_";

clientId += String(random(0xffff), HEX);

// Intentamos conectar
if (client.connect(clientId.c_str(),mqtt_user,mqtt_pass))
{
Serial.println("Conectado!");

String StrTopic = Topic_Init + DataIn;

Serial.print("Nos suscribimos a: ");Serial.println(StrTopic);

int str_len = StrTopic.length() + 1;

char char_Topic[str_len];

StrTopic.toCharArray(char_Topic, str_len);

```

```

        client.subscribe(char_Topic);

        eeprom_mqtt = 1;

    } else {

        Serial.print("falló :( con error -> ");

        Serial.print(client.state());

        Serial.println(" Intentamos de nuevo en 5 segundos");

        delay(5000);

    }

}

}

void EnviarMsgTopic(String StrMsg, String TopicSend)

{

    int str_len = StrMsg.length() + 1;

    char char_Msg[str_len];

    StrMsg.toCharArray(char_Msg, str_len);

    Serial.print("TramaOut: ");Serial.println(StrMsg);

    String StrTopic = Topic_Init + TopicSend;

    str_len = StrTopic.length() + 1;

    char char_Topic[str_len];

    StrTopic.toCharArray(char_Topic, str_len);

    Serial.print("StrTopic: ");Serial.println(StrTopic);

    client.publish(char_Topic, char_Msg);

}

```

7.3 Anexo C Código de Sensores de Puertas y Ventanas

```
#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <EEPROM.h>

#define eeprom_dir_1 10

#define eeprom_dir_2 12

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

#define D0 16 //GPIO16 - WAKE UP

#define D1 5 //GPIO5

#define D2 4 //GPIO4

#define D3 0 //GPIO0 ok ok

#define D4 2 //GPIO2 - TXD1

#define D5 14 //GPIO14 - HSCLK

#define D6 12 //GPIO12 - HMISO ok ok

#define D7 13 //GPIO13 - HMOSI - RXD2 ok ok

#define D8 15 //GPIO15 - HCS - TXD2

#define RX 3 //GPIO3 - RXD0

#define TX 1 //GPIO1 - TXD0

uint8 eeprom_on_off, eeprom_en_dis, eeprom_main, eeprom_mqtt =0;

const int BotonConfig = D3;

const int SenPuerVen = D2;

const int ledPin1 = D7;

const int Rele_1 = D6;
```

```

int EstSensPuerVen = 0;

boolean toggle_led = 0;

boolean enable_sensor = 0;

uint16 ContMensajes = 0;

const char* ssid    = "Nombre de la red";

const char* password = "contraseña de la red";

//*****

//*** CONFIGURACION MQTT ***

//*****

const char *mqtt_server = "arcadiaiot2022.tk";

//const char *mqtt_server = "111.222.111.222";

const int mqtt_port = puerto en el que escucha;

const char *mqtt_user = "usuario mqtt";

const char *mqtt_pass = "contraseña de usuario";

//*****

// SENSOR DE PUERTA VENTANA

//*****

#define EnableCasa_1_p 1

#define EnableCasa_1_v 0

#define EnableCasa_2_p 0

#define EnableCasa_2_v 0

#define EnableCasa_3_p 0

#define EnableCasa_3_v 0

#if EnableCasa_1_p == 1

```

```
const String Topic_Init = "Casa_1";

const String DataIn = "/SenPuerta";

const String DataOut = "/EstPuerta";

#endif

#if EnableCasa_1_v == 1

const String Topic_Init = "Casa_1";

const String DataIn = "/SenVentana";

const String DataOut = "/EstVentana";

#endif

#if EnableCasa_2_p == 1

const String Topic_Init = "Casa_2";

const String DataIn = "/SenPuerta";

const String DataOut = "/EstPuerta";

#endif

#if EnableCasa_2_v == 1

const String Topic_Init = "Casa_2";

const String DataIn = "/SenVentana";

const String DataOut = "/EstVentana";

#endif

#if EnableCasa_3_p == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/SenPuerta";

const String DataOut = "/EstPuerta";

#endif
```

```

#if EnableCasa_3_v == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/SenVentana";

const String DataOut = "/EstVentana";

#endif

//*****

// SENSOR DE PUERTA

//*****

// TOPIC IN => Casa_1/SenPuerta  MENSAJES QUE RECIBE  on off

// TOPIC OUT => Casa_1/EstPuerta  MENSAJE QUE ENVIA  peligro on off

//*****

// SENSOR DE VENTANA

//*****

// TOPIC IN => Casa_1/SenVentana  MENSAJES QUE RECIBE  on off

// TOPIC OUT => Casa_1/EstVentana  MENSAJE QUE ENVIA  peligro on off

//*****

WiFiClient espClient;

PubSubClient client(espClient);

//*****

//*** DECLARACION FUNCIONES ***

//*****

void setup_wifi(void);

void callback(char* topic, byte* payload, unsigned int length);

void reconnect(void);

```

```

void EnviarMsgTopic(String StrMsg, String TopicSend);

void setup()
{
    Serial.begin(115200);

    EEPROM.begin(50);

    pinMode(ledPin1, OUTPUT);
    pinMode(Rele_1, OUTPUT);

    pinMode(BotonConfig, INPUT_PULLUP);
    pinMode(SenPuerVen, INPUT_PULLUP);

    Serial.println("Iniciando ConfESP .....");

    digitalWrite(Rele_1, 0);

    setup_wifi();

    client.setServer(mqtt_server, mqtt_port);

    client.setCallback(callback);

    digitalWrite(ledPin1, HIGH);

    enable_sensor = 0;

    ContMensajes = 100;

    eeprom_main = 1;

    eeprom_mqtt = 0;
}

void loop() {

    if((eeprom_main == 1)&&(eeprom_mqtt == 1))

    {

        eeprom_main, eeprom_mqtt = 0;
    }
}

```

```

eeprom_on_off = EEPROM.read(eeprom_dir_1);

if (eeprom_on_off == 1)
{
    enable_sensor = 1;

    ContMensajes =0;

    EnviarMsgTopic("on",DataOut);
}
else
{
    enable_sensor = 0;

    ContMensajes = 100;

    EnviarMsgTopic("off",DataOut);
}

digitalWrite(ledPin1, HIGH);

digitalWrite(Rele_1, 0);
}

if (enable_sensor == 1)
{
    EstSensPuerVen = digitalRead(SenPuerVen);

    if (EstSensPuerVen == HIGH)
    {
        delay(10);

        EstSensPuerVen = digitalRead(SenPuerVen);

        delay(10);
    }
}

```

```

    if (EstSensPuerVen == HIGH)
    {
        if (ContMensajes < 1)
        {
            Serial.println("Puerta Ventana - Abierta");
            EnviarMsgTopic("peligro",DataOut);
            digitalWrite(ledPin1, LOW);
            digitalWrite(Rele_1, 1);
            delay(500);
            ContMensajes ++;
        }
    }
}
else{
    digitalWrite(ledPin1, HIGH);
    digitalWrite(Rele_1, 0);
}
}
/*
EstSensPuerVen = digitalRead(BotonConfig);
if (EstSensPuerVen == LOW)
{
    delay(10);
    if (EstSensPuerVen == LOW)
    {

```

```

        Serial.println("ModoConfiguracion");

        delay(400);

    }

}

*/

if (!client.connected()) {

    reconnect();

}

client.loop();

}

//*****

//***  CONEXION WIFI  ***

//*****

void setup_wifi(void)

{

    delay(10);

    // Nos conectamos a nuestra red Wifi

    Serial.println();

    Serial.print("Conectando a ");

    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        digitalWrite(ledPin1, toggle_led);

        Serial.print(".");

```

```

    delay(500);

    toggle_led =! toggle_led;

}

Serial.println("");

Serial.println("Conectado a red WiFi!");

Serial.println("Dirección IP: ");

Serial.println(WiFi.localIP());

}

void callback(char* topic, byte* payload, unsigned int length)

{

    String incoming = "";

    Serial.print("Mensaje recibido desde -> ");Serial.print(topic);

    Serial.println("");

    for (int i = 0; i < length; i++)

    {

        incoming += (char)payload[i];

    }

    incoming.trim();

    Serial.println("Mensaje -> -> -> ->" + incoming);

    if (incoming == "on"){

        Serial.println("HABILITA SENSOR");

        enable_sensor = 1;

        ContMensajes =0;

        EnviarMsgTopic("on",DataOut);

```

```

        EEPROM.write(eeprom_dir_1,1);EEPROM.commit();
    }

    if (incoming == "off"){

        Serial.println("DESHABILITA SENSOR");

        enable_sensor = 0;

        ContMensajes = 100;

        EnviarMsgTopic("off",DataOut);

        EEPROM.write(eeprom_dir_1,0);EEPROM.commit();

    }

    digitalWrite(ledPin1, HIGH);

    digitalWrite(Rele_1, 0);

}

void reconnect(void)

{

    while (!client.connected())

    {

        Serial.print("Intentando conexión Mqtt...");

        // Creamos un cliente ID

        String clientId = "esp32_";

        clientId += String(random(0xffff), HEX);

        // Intentamos conectar

        if (client.connect(clientId.c_str(),mqtt_user,mqtt_pass))

        {

            Serial.println("Conectado!");

```

```

String StrTopic = Topic_Init + DataIn;

Serial.print("Nos suscribimos a: ");Serial.println(StrTopic);

int str_len = StrTopic.length() + 1;

char char_Topic[str_len];

StrTopic.toCharArray(char_Topic, str_len);

client.subscribe(char_Topic);

eeprom_mqtt = 1;

} else {

Serial.print("falló :( con error -> ");

Serial.print(client.state());

Serial.println(" Intentamos de nuevo en 5 segundos");

delay(5000);

}

}

}

void EnviarMsgTopic(String StrMsg, String TopicSend)

{

int str_len = StrMsg.length() + 1;

char char_Msg[str_len];

StrMsg.toCharArray(char_Msg, str_len);

Serial.print("TramaOut: ");Serial.println(StrMsg);

String StrTopic = Topic_Init + TopicSend;

str_len = StrTopic.length() + 1;

char char_Topic[str_len];

```

```

    StrTopic.toCharArray(char_Topic, str_len);

    Serial.print("StrTopic: ");Serial.println(StrTopic);

    client.publish(char_Topic, char_Msg);

}

```

7.4 Anexo D Código de Sensores de Movimiento

```

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <EEPROM.h>

#define eeprom_dir_1 10

#define eeprom_dir_2 12

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

#define D0 16 //GPIO16 - WAKE UP

#define D1 5 //GPIO5

#define D2 4 //GPIO4

#define D3 0 //GPIO0

#define D4 2 //GPIO2 - TXD1

#define D5 14 //GPIO14 - HSCLK

#define D6 12 //GPIO12 - HMISO

#define D7 13 //GPIO13 - HMOSI - RXD2

#define D8 15 //GPIO15 - HCS - TXD2

#define RX 3 //GPIO3 - RXD0

#define TX 1 //GPIO1 - TXD0

uint8 eeprom_on_off, eeprom_en_dis,eeprom_main, eeprom_mqtt =0;

```

```

const int BotonConfig = D3;

const int SenPuerVen = D2;

const int ledPin1 = D7;

const int Rele_1 = D6;

int EstSensPuerVen = 0;

boolean toggle_led = 0;

boolean enable_sensor = 0;

boolean SendMensajes = 0;

boolean est_OnOff = 0;

const char* ssid = "Nombre de la red";

const char* password = "contraseña de la red";

//*****

//*** CONFIGURACION MQTT ***

//*****

const char *mqtt_server = "arcadiaiot2022.tk";

//const char *mqtt_server = "111.222.111.222";

const int mqtt_port = puerto que escucha;

const char *mqtt_user = "usuario mqtt";

const char *mqtt_pass = "contraseña de usuario";

//*****

// SENSOR DE MOVIMIENTO

//*****

#define EnableCasa_1_1 1

#define EnableCasa_1_2 0

```

```
#define EnableCasa_2_1 0

#define EnableCasa_2_2 0

#define EnableCasa_3_1 0

#define EnableCasa_3_2 0

#if EnableCasa_1_1 == 1

const String Topic_Init = "Casa_1";

const String DataIn = "/SenMov_1";

const String DataOut = "/EstMov_1";

#endif

#if EnableCasa_1_2 == 1

const String Topic_Init = "Casa_1";

const String DataIn = "/SenMov_2";

const String DataOut = "/EstMov_2";

#endif

#if EnableCasa_2_1 == 1

const String Topic_Init = "Casa_2";

const String DataIn = "/SenMov_1";

const String DataOut = "/EstMov_1";

#endif

#if EnableCasa_2_2 == 1

const String Topic_Init = "Casa_2";

const String DataIn = "/SenMov_2";

const String DataOut = "/EstMov_2";

#endif
```

```

#if EnableCasa_3_1 == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/SenMov_1";

const String DataOut = "/EstMov_1";

#endif

#if EnableCasa_3_2 == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/SenMov_2";

const String DataOut = "/EstMov_2";

#endif

//*****

// TOPIC DE ENTRADA Casa_1/SenMov_1 MENSAJE on off

// TOPIC DE ENVIO Casa_1/EstMov_1 MENSAJE peligro on off

//*****

WiFiClient espClient;

PubSubClient client(espClient);

//*****

/** DECLARACION FUNCIONES **/

//*****

void setup_wifi(void);

void callback(char* topic, byte* payload, unsigned int length);

void reconnect(void);

void EnviarMsgTopic(String StrMsg, String TopicSend);

void setup()

```

```

{
  Serial.begin(115200);

  EEPROM.begin(50);

  pinMode(ledPin1, OUTPUT);

  pinMode(Rele_1, OUTPUT);

  pinMode(BotonConfig, INPUT_PULLUP);

  pinMode(SenPuerVen, INPUT_PULLUP);

  Serial.println("Iniciando ConfESP .....");

  digitalWrite(Rele_1, 0);

  setup_wifi();

  client.setServer(mqtt_server, mqtt_port);

  client.setCallback(callback);

  digitalWrite(ledPin1, HIGH);

  enable_sensor = 0;

  SendMensajes = 0;

  eeprom_main = 1;

  eeprom_mqtt = 0;
}

void loop() {

  if((eeprom_main == 1)&&(eeprom_mqtt == 1))

  {

    eeprom_main, eeprom_mqtt = 0;

    eeprom_on_off = EEPROM.read(eeprom_dir_1);
  }
}

```

```

if (eeprom_on_off == 1)
{
    enable_sensor = 1;

    SendMensajes =0;

    EnviarMsgTopic("on",DataOut);

    est_OnOff = 1;
}
else
{
    enable_sensor = 0;

    SendMensajes = 0;

    EnviarMsgTopic("off",DataOut);

    est_OnOff = 0;
}

digitalWrite(ledPin1, HIGH);

digitalWrite(Rele_1, 0);
}

if (enable_sensor == 1)
{
    EstSensPuerVen = digitalRead(SenPuerVen);

    if (EstSensPuerVen == LOW)
    {
        SendMensajes = 1;
    }
}

```

```

    if ((EstSensPuerVen == HIGH) &&(SendMensajes == 1))
    {
        SendMensajes = 0;
        Serial.println("Movimiento Detectado");
        EnviarMsgTopic("peligro",DataOut);
        digitalWrite(ledPin1, LOW);
        digitalWrite(Rele_1, 1);
    }
}

if (!client.connected()) {
    reconnect();
}

client.loop();
}

//*****

//***  CONEXION WIFI  ***

//*****

void setup_wifi(void)
{
    delay(10);

    // Nos conectamos a nuestra red Wifi

    Serial.println();

    Serial.print("Conectando a ");

    Serial.println(ssid);
}

```

```

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

    digitalWrite(ledPin1, toggle_led);

    Serial.print(".");

    delay(500);

    toggle_led =! toggle_led;

}

Serial.println("");

Serial.println("Conectado a red WiFi!");

Serial.println("Dirección IP: ");

Serial.println(WiFi.localIP());

}

void callback(char* topic, byte* payload, unsigned int length)

{

    String incoming = "";

    Serial.print("Mensaje recibido desde -> ");Serial.print(topic);

    Serial.println("");

    for (int i = 0; i < length; i++)

    {

        incoming += (char)payload[i];

    }

    incoming.trim();

    Serial.println("Mensaje -> -> -> ->" + incoming);

    if (incoming == "on"){

```

```

Serial.println("HABILITA SENSOR");

enable_sensor = 1;

SendMensajes =0;

EnviarMsgTopic("on",DataOut);

EEPROM.write(eeprom_dir_1,1);EEPROM.commit();

est_OnOff = 1;
}

if (incoming == "off"){

Serial.println("DESHABILITA SENSOR");

enable_sensor = 0;

SendMensajes = 0;

EnviarMsgTopic("off",DataOut);

EEPROM.write(eeprom_dir_1,0);EEPROM.commit();

est_OnOff = 0;

}

digitalWrite(ledPin1, HIGH);

digitalWrite(Rele_1, 0);

}

void EnviarMsgTopic(String StrMsg, String TopicSend)

{

int str_len = StrMsg.length() + 1;

char char_Msg[str_len];

StrMsg.toCharArray(char_Msg, str_len);

Serial.print("TramaOut: ");Serial.println(StrMsg);

```

```

String StrTopic = Topic_Init + TopicSend;

str_len = StrTopic.length() + 1;

char char_Topic[str_len];

StrTopic.toCharArray(char_Topic, str_len);

Serial.print("StrTopic: ");Serial.println(StrTopic);

client.publish(char_Topic, char_Msg);

}

void reconnect(void)

{

while (!client.connected())

{

Serial.print("Intentando conexión Mqtt...");

// Creamos un cliente ID

String clientId = "esp32_";

clientId += String(random(0xffff), HEX);

// Intentamos conectar

if (client.connect(clientId.c_str(),mqtt_user,mqtt_pass))

{

Serial.println("Conectado!");

String StrTopic = Topic_Init + DataIn;

Serial.print("Nos suscribimos a: ");Serial.println(StrTopic);

int str_len = StrTopic.length() + 1;

char char_Topic[str_len];

StrTopic.toCharArray(char_Topic, str_len);

```

```

    client.subscribe(char_Topic);

    eeprom_mqtt = 1;

} else {

    Serial.print("falló :( con error -> ");

    Serial.print(client.state());

    Serial.println(" Intentamos de nuevo en 5 segundos");

    delay(5000);

}

}

}

```

7.5 Anexo E Código Aspersores de Agua

```

#include <Arduino.h>

#include <WiFi.h>

#include <PubSubClient.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include <Ticker.h>

#include <EEPROM.h>

Ticker tiempo_1;

#define EEPROM_SIZE 50

#define eeprom_dir_1 10

#define eeprom_dir_2 12

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

const char* ssid = "Nombre de la res";

```

```

const char* password = "contraseña de la red";

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

#define LimHumedad 80

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

byte eeprom_on_off, eeprom_en_dis, eeprom_main, eeprom_mqtt =0;

const int BotonConfig = 21;

const int ledPin1 = 19;

const int PinAspersor = 18;

const int PinSenHumd = 34;

int ValAnlgRead, PorcHumedad = 0;

float temperatureC = 0;

boolean ReadSensores, Ctr_Manual_Auto = false;

//*****

/** CONFIGURACION MQTT **

//*****

const char *mqtt_server = "arcadiaiot2022.tk";

//const char *mqtt_server = "111.222.111.222";

const int mqtt_port = Puerto en el que escucha;

const char *mqtt_user = "Usuario mqtt";

const char *mqtt_pass = "contraseña";

//*****

#define EnableCasa_1_1 0

#define EnableCasa_1_2 1

#define EnableCasa_2_1 0

```

```

#define EnableCasa_2_2 0

#define EnableCasa_3_1 0

#define EnableCasa_3_2 0

#if EnableCasa_1_1 == 1

const String Topic_Init = "Casa_1";

const String DataIn = "/Esparzor_1";

const String DataOut_Est = "/EstEspazor_1"; // Indica el estado del esparzor

const String DataOut_Ctr = "/EstControl_1";

const String Data_HT = "/HumTemp_1";

#endif

#if EnableCasa_1_2 == 1

const String Topic_Init = "Casa_1";

const String DataIn = "/Esparzor_2";

const String DataOut_Est = "/EstEspazor_2"; // Indica el estado del aspensor

const String DataOut_Ctr = "/EstControl_2";

const String Data_HT = "/HumTemp_2";

#endif

#if EnableCasa_2_1 == 1

const String Topic_Init = "Casa_2";

const String DataIn = "/Esparzor_1";

const String DataOut_Est = "/EstEspazor_1"; // Indica el estado del aspensor

const String DataOut_Ctr = "/EstControl_1";

const String Data_HT = "/HumTemp_1";

#endif

```

```

#if EnableCasa_2_2 == 1

const String Topic_Init = "Casa_2";

const String DataIn = "/Esparzor_2";

const String DataOut_Est = "/EstEspazor_2"; // Indica el estado del aspersor

const String DataOut_Ctr = "/EstControl_2";

const String Data_HT = "/HumTemp_2";

#endif

#if EnableCasa_3_1 == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/Esparzor_1";

const String DataOut_Est = "/EstEspazor_1"; // Indica el estado del aspersor

const String DataOut_Ctr = "/EstControl_1";

const String Data_HT = "/HumTemp_1";

#endif

#if EnableCasa_3_2 == 1

const String Topic_Init = "Casa_3";

const String DataIn = "/Esparzor_2";

const String DataOut_Est = "/EstEspazor_2"; // Indica el estado del aspersor

const String DataOut_Ctr = "/EstControl_2";

const String Data_HT = "/HumTemp_2";

#endif

// DATOS QUE ENVIA EL SERVIDOR

// Casa_1/Esparzor_1   on    enciende el aspersor

// Casa_1/Esparzor_1   off   enciende el aspersor

```

```

// Casa_1/Espazor_1   ctr_on   activa el modo manual

// Casa_1/Espazor_1   ctr_off   desactiva el modo manual - por defecto esta en
manual

//*****

//DATOS QUE ENVIA EL ESP

// EstEspazor_1   on   of

// EstControl_1   ctr_on   ctr_off

//*****

WiFiClient espClient;

PubSubClient client(espClient);

boolean toggle,OnOffAutomatico = 0;

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

const int oneWireBus = 4; // GPIO where the DS18B20 is connected to

OneWire oneWire(oneWireBus); // Setup a oneWire instance to communicate with
any OneWire devices

DallasTemperature sensors(&oneWire); // Pass our oneWire reference to Dallas
Temperature sensor

//*****

//*** DECLARACION FUNCIONES ***

//*****

void setup_wifi(void);

void callback(char* topic, byte* payload, unsigned int length);

void reconnect(void);

void funcion_1(void);

```

```

void EnviarMsgTopic(String StrMsg, String TopicSend);

void setup()
{
  Serial.begin(115200);

  EEPROM.begin(EEPROM_SIZE);

  pinMode(ledPin1, OUTPUT);

  pinMode(PinAspersor, OUTPUT);

  pinMode(BotonConfig, INPUT_PULLUP);

  //XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

  Serial.println("Iniciando ConfESP32 .....");

  digitalWrite(PinAspersor, LOW);

  //XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

  setup_wifi();

  client.setServer(mqtt_server, mqtt_port);

  client.setCallback(callback);

  sensors.begin();

  digitalWrite(ledPin1, LOW);

  ReadSensores = false;

  OnOffAutomatico = false;

  Ctr_Manual_Auto = false;

  tiempo_1.attach(15, funcion_1);

  eeprom_main = 1;

  eeprom_mqtt = 0;
}

```

```

void loop()
{
  if((eeprom_main == 1)&&(eeprom_mqtt == 1))
  {
    eeprom_main, eeprom_mqtt = 0;

    eeprom_on_off = EEPROM.read(eeprom_dir_1);
    eeprom_en_dis = EEPROM.read(eeprom_dir_2);
    if (eeprom_en_dis == 1)
    {
      Serial.println("Control Manual on");
      EnviarMsgTopic("ctr_on",DataOut_Ctr);
      Ctr_Manual_Auto = true;
      if(PorcHumedad > LimHumedad) OnOffAutomatico = true;
      if(PorcHumedad < LimHumedad) OnOffAutomatico = false;
    }
    else
    {
      Serial.println("Control Manual off");
      EnviarMsgTopic("ctr_off",DataOut_Ctr);
      Ctr_Manual_Auto = false;
    }
    if (Ctr_Manual_Auto == false)
    {
      if (eeprom_on_off == 1)

```

```

    {
        Serial.println("SE ACTIVA EL ASPERSOR");
        digitalWrite(ledPin1, HIGH);
        digitalWrite(PinAspersor, HIGH);
        EnviarMsgTopic("on",DataOut_Est);
    }
else
    {
        Serial.println("SE APAGA EL ASPERSOR");
        digitalWrite(ledPin1, LOW);
        digitalWrite(PinAspersor, LOW);
        EnviarMsgTopic("off",DataOut_Est);
    }
}
}

if(ReadSensores == true)
{
    ReadSensores = false;
    // Leyendo la humedad
    //totalmente seco el valor es maximo
    //mas cercano al 0 es mas humedo
    ValAnlgRead = analogRead(PinSenHumd);
    Serial.println("*****");
    Serial.println(ValAnlgRead);
}

```

```

//aspersor 1 valor - 336 totalmente seco - 282 humedo

//aspersor 2 valor - 342 totalmente seco - 263 humedo

//ASPERSOR 1

/*

if (ValAnlgRead > 336){ValAnlgRead = 336;}

PorcHumedad = map(ValAnlgRead, 0, 336, 100, 0); // 100 -> Muy Seco - 0 ->
Muy Humedo

*/

//ASPERSOR 2

if (ValAnlgRead > 342){ValAnlgRead = 342;}

PorcHumedad = map(ValAnlgRead, 0, 342, 100, 0); // 100 -> Muy Seco - 0 ->
Muy Humedo

Serial.println(PorcHumedad);

sensors.requestTemperatures();

temperatureC = sensors.getTempCByIndex(0);

EnviarMsgTopic((String(PorcHumedad)+"@"+String(temperatureC,2)),Data_HT);

Serial.println(PorcHumedad);

Serial.println(temperatureC);

if (Ctr_Manual_Auto == true)

{

if((PorcHumedad > LimHumedad)&&(OnOffAutomatico == true))

{

Serial.println("SE ACTIVA EL ASPERSOR");

```

```

    digitalWrite(ledPin1, HIGH);

    digitalWrite(PinAspersor, HIGH);

    EnviarMsgTopic("on",DataOut_Est);

    OnOffAutomatico = false;

}

if((PorcHumedad < LimHumedad) &&(OnOffAutomatico == false))

{

    Serial.println("SE APAGA EL ASPERSOR");

    digitalWrite(ledPin1, LOW);

    digitalWrite(PinAspersor, LOW);

    EnviarMsgTopic("off",DataOut_Est);

    OnOffAutomatico = true;

}

}

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

if((digitalRead(BotonConfig)) == LOW)

{

    Serial.println("PULSADOR PRESIONADO");

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

if (!client.connected()) {

    reconnect();

}

```

```

    client.loop();
}

void EnviarMsgTopic(String StrMsg, String TopicSend)
{
    int str_len = StrMsg.length() + 1;

    char char_Msg[str_len];

    StrMsg.toCharArray(char_Msg, str_len);

    Serial.print("TramaOut: ");Serial.println(StrMsg);

    String StrTopic = Topic_Init + TopicSend;

    str_len = StrTopic.length() + 1;

    char char_Topic[str_len];

    StrTopic.toCharArray(char_Topic, str_len);

    Serial.print("StrTopic: ");Serial.println(StrTopic);

    client.publish(char_Topic, char_Msg);
}

void funcion_1(void)
{
    ReadSensores = true;
}

//*****

//***  CONEXION WIFI  ***

//*****

void setup_wifi(void)
{

```

```

delay(10);

// Nos conectamos a nuestra red Wifi

Serial.println();

Serial.print("Conectando a ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

    digitalWrite(ledPin1, toggle);

    Serial.print(".");

    delay(500);

    toggle =! toggle;

}

Serial.println("");

Serial.println("Conectado a red WiFi!");

Serial.println("Dirección IP: ");

Serial.println(WiFi.localIP());

}

void callback(char* topic, byte* payload, unsigned int length)

{

    String incoming = "";

    Serial.print("Mensaje recibido desde -> ");Serial.print(topic);

    Serial.println("");

    for (int i = 0; i < length; i++)

    {

```

```

    incoming += (char)payload[i];
}

incoming.trim();

Serial.println("Mensaje -> -> -> ->" + incoming);

String str_topic = String(Topic_Init) + String(DataIn);

if (String(topic) == str_topic)
{
    //XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

    if (incoming == "ctr_on"){

        Serial.println("Control Manual on");

        EnviarMsgTopic("ctr_on",DataOut_Ctr);

        Ctr_Manual_Auto = true;

        if(PorcHumedad > LimHumedad) OnOffAutomatico = true;

        if(PorcHumedad < LimHumedad) OnOffAutomatico = false;

        EEPROM.write(eeprom_dir_2, 1);EEPROM.commit();delay(5);

    }

    if (incoming == "ctr_off"){

        Serial.println("Control Manual off");

        EnviarMsgTopic("ctr_off",DataOut_Ctr);

        Ctr_Manual_Auto = false;

        EEPROM.write(eeprom_dir_2, 0);EEPROM.commit();delay(5);

    }

    if (Ctr_Manual_Auto == false)

    {

```

```

    if (incoming == "on"){

        Serial.println("SE ACTIVA EL ASPERSOR");

        digitalWrite(ledPin1, HIGH);

        digitalWrite(PinAspersor, HIGH);

        EnviarMsgTopic("on",DataOut_Est);

        EEPROM.write(eeprom_dir_1, 1);EEPROM.commit();delay(5);

    }

    if (incoming == "off"){

        Serial.println("SE APAGA EL ASPERSOR");

        digitalWrite(ledPin1, LOW);

        digitalWrite(PinAspersor, LOW);

        EnviarMsgTopic("off",DataOut_Est);

        EEPROM.write(eeprom_dir_1, 0);EEPROM.commit();delay(5);

    }

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

}

}

void reconnect(void)

{

    while (!client.connected())

    {

        Serial.print("Intentando conexión Mqtt...");

        // Creamos un cliente ID

```

```

String clientId = "esp32_";

clientId += String(random(0xffff), HEX);

// Intentamos conectar

if (client.connect(clientId.c_str(),mqtt_user,mqtt_pass))

{

    Serial.println("Conectado!");

    String StrTopic = Topic_Init + DataIn;

    Serial.print("Nos suscribimos a: ");Serial.println(StrTopic);

    int str_len = StrTopic.length() + 1;

    char char_Topic[str_len];

    StrTopic.toCharArray(char_Topic, str_len);

    client.subscribe(char_Topic);

    eeprom_mqtt = 1;

} else {

    Serial.print("falló :( con error -> ");

    Serial.print(client.state());

    Serial.println(" Intentamos de nuevo en 5 segundos");

    delay(5000);

}

}
}

```

7.6 Anexo F Código Node.js

```

var mysql = require('mysql');

var mqtt = require('mqtt');

```

```

//CREDENCIALES MYSQL

var con = mysql.createConnection({

    host: "arcadiaiot2022.tk",

    user: "Usuario",

    password: "Contraseña",

    database: "EstadoCasas"

});

//CREDENCIALES MQTT

var options = {

    port: numero de puerto,

    host: 'arcadiaiot2022.tk',

    clientId: 'esp_arc' + Math.round(Math.random() * (0 - 10000) * -1),

//acces_control_iot_ - clientes distintos pero con el mismo inicio

    username: 'Usuario',

    password: 'Contraseña',

    keepalive: 60, //envio un ping para que no se caiga la conexion

    reconnectPeriod: 1000,

    protocolId: 'MQIsdp',

    protocolVersion: 3,

    clean: true,

    encoding: 'utf8' //data set

};

var client = mqtt.connect("mqtt://arcadiaiot2022.tk", options);

//SE REALIZA LA CONEXION

```

```

client.on('connect', function() {

    console.log("Conexion MQTT Exitosa!");

    //client.subscribe('+/#', function(err) {          // se suscribe a todos los topicos

    //client.subscribe('MyCasa/#', function(err) {

    client.subscribe('Casa_1/#', function(err) { console.log("Subscripcion exitosa!");});

    client.subscribe('Casa_2/#', function(err) { console.log("Subscripcion exitosa!");});

    client.subscribe('Casa_3/#', function(err) { console.log("Subscripcion exitosa!");});

    })

function ActualizarTabla(StrCasa, StrColumna,StrValor) {

    var    query    =    "UPDATE    `EstadoCasas`.`"+StrCasa+"`    SET

`"+StrColumna+"`="+StrValor+" WHERE `id`=1;";

    // var query = "UPDATE `EstadoCasas`.`casa_1` SET `luz_1`=1' WHERE

`id`=1;";

    //var    query    =    "INSERT    INTO    `EspIoT_bd`.`esp_iot`

(`data_volt`,`data_amp`,`data_pow`) VALUES (" + vvolt + "," + vamp + "," + vpow +

");";

    con.query(query, function(err, result, fields) {

        if (err) throw err;

        console.log("Fila insertada correctamente");

    });

    /*

    var StrColumna = "luz_1";

    var StrValor = "0";

```

```

        var query = "UPDATE `EstadoCasas`.`casa_1` SET
`"+StrColumna+"`='"+StrValor+"' WHERE `id`=1;";

        //var query = "UPDATE `EstadoCasas`.`casa_1` SET `luz_1`=0' WHERE
`id`=1;";

        //var query = "INSERT INTO `EspIoT_bd`.`esp_iot`
(`data_volt`,`data_amp`,`data_pow`) VALUES (" + vvolt + "," + vamp + "," + vpow +
");";

        con.query(query, function(err, result, fields) {

            if (err) throw err;

            console.log("Fila insertada correctamente");

        });

    */

}

function LeerTabla(StrCasa) {

    var query = "SELECT `luz_1`, `luz_2`, `est_sen_p`, `est_sen_v`,
`est_sen_m1`, `est_sen_m2`, `est_espar_1`, `est_ctr_1`, `est_espar_2`, `est_ctr_2`
FROM `EstadoCasas`.`"+StrCasa+"` WHERE `id`=1;";

    con.query(query, function(err, result, fields) {

        DataTotal = result[0].luz_1.toString() + "," +
result[0].luz_2.toString() + "," +
result[0].est_sen_p.toString() + "," +
result[0].est_sen_v.toString() + "," +
result[0].est_sen_m1.toString() + "," +
result[0].est_sen_m2.toString() + "," +

```

```

result[0].est_espar_1.toString() + "," +
result[0].est_ctr_1.toString() + "," +
result[0].est_espar_2.toString() + "," +
result[0].est_ctr_2.toString();

console.log(DataTotal);

if (StrCasa == "casa_1"){client.publish("Casa_1/TablaMysql", DataTotal);}
if (StrCasa == "casa_2"){client.publish("Casa_2/TablaMysql", DataTotal);}
if (StrCasa == "casa_3"){client.publish("Casa_3/TablaMysql", DataTotal);}

//console.log(result);

//console.log(result[0].luz_1);

//console.log(result[0].luz_2);

//console.log(result[0].est_sen_p);

if (err) throw err;

    console.log("Fila leida correctamente");

});

/*

    RowDataPacket {

        luz_1: 0,

        luz_2: 0,

        est_sen_p: 0,

        est_sen_v: 0,

        est_sen_m1: 0,

        est_sen_m2: 0,

        est_espar_1: 1,

```

```

        est_ctr_1: 0,
        est_espar_2: 1,
        est_ctr_2: 0
    }
    */
}

function ValidarIngreso(NumCasa,ClaveEscrita) {
    console.log(NumCasa);
    console.log(ClaveEscrita);
    var query = "SELECT `user`, `key` FROM `EstadoCasas`.`ClaveCasas`
WHERE `id`="+NumCasa+"";

    con.query(query, function(err, result, fields) {
        DataUsuario = result[0].user.toString();
        DataKey = result[0].key.toString();
        console.log(DataUsuario);
        console.log(DataKey);
        DataTotal = DataUsuario +"@"+DataKey;
        if(NumCasa == 1)
        {
            if(DataTotal == ClaveEscrita)
            {
                //console.log("exito - key");
                client.publish("Casa_1/key_valida", "exito");
            }
        }
    });
}

```

```

else
{
    //console.log("error - key");
    client.publish("Casa_1/key_valida", "error");
}
}
if(NumCasa == 2)
{
    if(DataTotal == ClaveEscrita)
    {
        //console.log("exito - key");
        client.publish("Casa_2/key_valida", "exito");
    }
    else
    {
        //console.log("error - key");
        client.publish("Casa_2/key_valida", "error");
    }
}
if(NumCasa == 3)
{
    if(DataTotal == ClaveEscrita)
    {
        //console.log("exito - key");

```

```

        client.publish("Casa_3/key_valida", "exito");
    }
    else
    {
        //console.log("error - key");
        client.publish("Casa_3/key_valida", "error");
    }
}
if (err) throw err;
    console.log("Fila leida correctamente");
});
}

//CUANDO SE RECIBE MENSAJE
client.on('message', function(topic, message) {
    console.log("Mensaje recibido desde -> " + topic + " Mensaje -> " +
message.toString());

    var StrTopic = topic.toString();
    var StrMsg = message.toString();

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

if (StrTopic == "Casa_1/EstTotalDisp")
{
    if (StrMsg == "data")
    {
        console.log("PETICION DE TODOS LOS SENSORES");
        LeerTabla("casa_1");
    }
}

if (StrTopic == "Casa_1/validar_key")
{
    //console.log(StrMsg);
    ValidarIngreso(1,StrMsg)
}

if (StrTopic == "Casa_1/EstLuz_1")
{
    if (StrMsg == "on"){console.log("Luz 1 on");ActualizarTabla("casa_1",
"luz_1","1");}
    if (StrMsg == "off"){console.log("Luz 1 off");ActualizarTabla("casa_1",
"luz_1","0");}
}

if (StrTopic == "Casa_1/EstLuz_2")
{
    if (StrMsg == "on"){console.log("Luz 2 on");ActualizarTabla("casa_1",
"luz_2","1");}
}

```

```

        if (StrMsg == "off"){console.log("Luz 2 off");ActualizarTabla("casa_1",
"luz_2","0");}

    }

    if (StrTopic == "Casa_1/EstPuerta")

    {

        if (StrMsg == "on"){console.log("EstPuerta on");ActualizarTabla("casa_1",
"est_sen_p","1");}

        if (StrMsg == "off"){console.log("EstPuerta off");ActualizarTabla("casa_1",
"est_sen_p","0");}

        if (StrMsg == "peligro")

        {

            client.publish("Casa_1/key_valida", "peligro");

            console.log("EstPuerta peligro");ActualizarTabla("casa_1", "est_sen_p","2");

        }

    }

    if (StrTopic == "Casa_1/EstVentana")

    {

        if (StrMsg == "on"){console.log("EstVentana on");ActualizarTabla("casa_1",
"est_sen_v","1");}

        if (StrMsg == "off"){console.log("EstVentana off");ActualizarTabla("casa_1",
"est_sen_v","0");}

        if (StrMsg == "peligro")

        {

            client.publish("Casa_1/key_valida", "peligro");

```

```

        console.log("EstVentana          peligro");ActualizarTabla("casa_1",
"est_sen_v","2");
    }
}
if (StrTopic == "Casa_1/EstMov_1")
{
    if (StrMsg == "on"){console.log("EstMov_1 on");ActualizarTabla("casa_1",
"est_sen_m1","1");}
    if (StrMsg == "off"){console.log("EstMov_1 off");ActualizarTabla("casa_1",
"est_sen_m1","0");}
    if (StrMsg == "peligro")
    {
        client.publish("Casa_1/key_valida", "peligro");
        console.log("EstMov_1          peligro");ActualizarTabla("casa_1",
"est_sen_m1","2");
    }
}
if (StrTopic == "Casa_1/EstMov_2")
{
    if (StrMsg == "on"){console.log("EstMov_2 on");ActualizarTabla("casa_1",
"est_sen_m2","1");}
    if (StrMsg == "off"){console.log("EstMov_2 off");ActualizarTabla("casa_1",
"est_sen_m2","0");}
    if (StrMsg == "peligro")

```

```

    {
        client.publish("Casa_1/key_valida", "peligro");
        console.log("EstMov_2          peligro");ActualizarTabla("casa_1",
"est_sen_m2","2");
    }
}
if (StrTopic == "Casa_1/EstEspazor_1")
{
    if (StrMsg == "on"){console.log("EstEspazor_1 on");ActualizarTabla("casa_1",
"est_espar_1","1");}
    if (StrMsg == "off"){console.log("EstEspazor_1 off");ActualizarTabla("casa_1",
"est_espar_1","0");}
}
if (StrTopic == "Casa_1/EstControl_1")
{
    if (StrMsg == "ctr_on"){console.log("EstControl_1
on");ActualizarTabla("casa_1", "est_ctr_1","1");}
    if (StrMsg == "ctr_off"){console.log("EstControl_1
off");ActualizarTabla("casa_1", "est_ctr_1","0");}
}
if (StrTopic == "Casa_1/EstEspazor_2")
{
    if (StrMsg == "on"){console.log("EstEspazor_2 on");ActualizarTabla("casa_1",
"est_espar_2","1");}

```

```

        if (StrMsg == "off"){console.log("EstEspazor_2 off");ActualizarTabla("casa_1",
"est_espar_2","0");}

    }

    if (StrTopic == "Casa_1/EstControl_2")

    {

        if      (StrMsg      ==      "ctr_on"){console.log("EstControl_2
on");ActualizarTabla("casa_1", "est_ctr_2","1");}

        if      (StrMsg      ==      "ctr_off"){console.log("EstControl_2
off");ActualizarTabla("casa_1", "est_ctr_2","0");}

    }

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

    if (StrTopic == "Casa_2/EstTotalDisp")

    {

        if (StrMsg == "data")

        {

            console.log("PETICION DE TODOS LOS SENSORES");

            LeerTabla("casa_2");

        }

    }

    if (StrTopic == "Casa_2/validar_key")

    {

```

```

        //console.log(StrMsg);

        ValidarIngreso(2,StrMsg)

    }

    if (StrTopic == "Casa_2/EstLuz_1")

    {

        if (StrMsg == "on"){console.log("Luz 1 on");ActualizarTabla("casa_2",
"luz_1","1");}

        if (StrMsg == "off"){console.log("Luz 1 off");ActualizarTabla("casa_2",
"luz_1","0");}

    }

    if (StrTopic == "Casa_2/EstLuz_2")

    {

        if (StrMsg == "on"){console.log("Luz 2 on");ActualizarTabla("casa_2",
"luz_2","1");}

        if (StrMsg == "off"){console.log("Luz 2 off");ActualizarTabla("casa_2",
"luz_2","0");}

    }

    if (StrTopic == "Casa_2/EstPuerta")

    {

        if (StrMsg == "on"){console.log("EstPuerta on");ActualizarTabla("casa_2",
"est_sen_p","1");}

        if (StrMsg == "off"){console.log("EstPuerta off");ActualizarTabla("casa_2",
"est_sen_p","0");}

        if (StrMsg == "peligro"){

```

```

        client.publish("Casa_2/key_valida", "peligro");

        console.log("EstPuerta peligro");ActualizarTabla("casa_2", "est_sen_p","2");}

    }

    if (StrTopic == "Casa_2/EstVentana")

    {

        if (StrMsg == "on"){console.log("EstVentana on");ActualizarTabla("casa_2",
"est_sen_v","1");}

        if (StrMsg == "off"){console.log("EstVentana off");ActualizarTabla("casa_2",
"est_sen_v","0");}

        if (StrMsg == "peligro"){

            client.publish("Casa_2/key_valida", "peligro");

            console.log("EstVentana peligro");ActualizarTabla("casa_2", "est_sen_v", "2");}

        }

        if (StrTopic == "Casa_2/EstMov_1")

        {

            if (StrMsg == "on"){console.log("EstMov_1 on");ActualizarTabla("casa_2",
"est_sen_m1","1");}

            if (StrMsg == "off"){console.log("EstMov_1 off");ActualizarTabla("casa_2",
"est_sen_m1","0");}

            if (StrMsg == "peligro"){

                client.publish("Casa_2/key_valida", "peligro");

                console.log("EstMov_1                peligro");ActualizarTabla("casa_2",
"est_sen_m1","2");}

            }

        }

```

```

if (StrTopic == "Casa_2/EstMov_2")
{
    if (StrMsg == "on"){console.log("EstMov_2 on");ActualizarTabla("casa_2",
"est_sen_m2","1");}

    if (StrMsg == "off"){console.log("EstMov_2 off");ActualizarTabla("casa_2",
"est_sen_m2","0");}

    if (StrMsg == "peligro"){
        client.publish("Casa_2/key_valida", "peligro");
        console.log("EstMov_2                peligro");ActualizarTabla("casa_2",
"est_sen_m2","2");}
    }

if (StrTopic == "Casa_2/EstEspazor_1")
{
    if (StrMsg == "on"){console.log("EstEspazor_1 on");ActualizarTabla("casa_2",
"est_espar_1","1");}

    if (StrMsg == "off"){console.log("EstEspazor_1 off");ActualizarTabla("casa_2",
"est_espar_1","0");}

    }

if (StrTopic == "Casa_2/EstControl_1")
{
    if (StrMsg == "ctr_on"){console.log("EstControl_1
on");ActualizarTabla("casa_2", "est_ctr_1","1");}

    if (StrMsg == "ctr_off"){console.log("EstControl_1
off");ActualizarTabla("casa_2", "est_ctr_1","0");}
}

```

```

}

if (StrTopic == "Casa_2/EstEspazor_2")

{

    if (StrMsg == "on"){console.log("EstEspazor_2 on");ActualizarTabla("casa_2",
"est_espar_2","1");}

    if (StrMsg == "off"){console.log("EstEspazor_2 off");ActualizarTabla("casa_2",
"est_espar_2","0");}

}

if (StrTopic == "Casa_2/EstControl_2")

{

    if (StrMsg == "ctr_on"){console.log("EstControl_2
on");ActualizarTabla("casa_2", "est_ctr_2","1");}

    if (StrMsg == "ctr_off"){console.log("EstControl_2
off");ActualizarTabla("casa_2", "est_ctr_2","0");}

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX

if (StrTopic == "Casa_3/EstTotalDisp")

{

    if (StrMsg == "data")

```

```

    {
        console.log("PETICION DE TODOS LOS SENSORES");
        LeerTabla("casa_3");
    }
}

if (StrTopic == "Casa_3/validar_key")
{
    //console.log(StrMsg);
    ValidarIngreso(3,StrMsg)
}

if (StrTopic == "Casa_3/EstLuz_1")
{
    if (StrMsg == "on"){console.log("Luz 1 on");ActualizarTabla("casa_3",
"luz_1","1");}

    if (StrMsg == "off"){console.log("Luz 1 off");ActualizarTabla("casa_3",
"luz_1","0");}

}

if (StrTopic == "Casa_3/EstLuz_2")
{
    if (StrMsg == "on"){console.log("Luz 2 on");ActualizarTabla("casa_3",
"luz_2","1");}

    if (StrMsg == "off"){console.log("Luz 2 off");ActualizarTabla("casa_3",
"luz_2","0");}

}

```

```

if (StrTopic == "Casa_3/EstPuerta")
{
    if (StrMsg == "on"){console.log("EstPuerta on");ActualizarTabla("casa_3",
"est_sen_p","1");}

    if (StrMsg == "off"){console.log("EstPuerta off");ActualizarTabla("casa_3",
"est_sen_p","0");}

    if (StrMsg == "peligro"){
        client.publish("Casa_3/key_valida", "peligro");
        console.log("EstPuerta peligro");ActualizarTabla("casa_3", "est_sen_p","2");}
    }
if (StrTopic == "Casa_3/EstVentana")
{
    if (StrMsg == "on"){console.log("EstVentana on");ActualizarTabla("casa_3",
"est_sen_v","1");}

    if (StrMsg == "off"){console.log("EstVentana off");ActualizarTabla("casa_3",
"est_sen_v","0");}

    if (StrMsg == "peligro"){
        client.publish("Casa_3/key_valida", "peligro");
        console.log("EstVentana peligro");ActualizarTabla("casa_3", "est_sen_v","2");}
    }
if (StrTopic == "Casa_3/EstMov_1")
{
    if (StrMsg == "on"){console.log("EstMov_1 on");ActualizarTabla("casa_3",
"est_sen_m1","1");}

```

```

        if (StrMsg == "off"){console.log("EstMov_1 off");ActualizarTabla("casa_3",
"est_sen_m1","0");}

        if (StrMsg == "peligro"){
            client.publish("Casa_3/key_valida", "peligro");
            console.log("EstMov_1                peligro");ActualizarTabla("casa_3",
"est_sen_m1","2");}
    }

    if (StrTopic == "Casa_3/EstMov_2")
    {
        if (StrMsg == "on"){console.log("EstMov_2 on");ActualizarTabla("casa_3",
"est_sen_m2","1");}

        if (StrMsg == "off"){console.log("EstMov_2 off");ActualizarTabla("casa_3",
"est_sen_m2","0");}

        if (StrMsg == "peligro"){
            client.publish("Casa_3/key_valida", "peligro");
            console.log("EstMov_2                peligro");ActualizarTabla("casa_3",
"est_sen_m2","2");}
    }

    if (StrTopic == "Casa_3/EstEspazor_1")
    {
        if (StrMsg == "on"){console.log("EstEspazor_1 on");ActualizarTabla("casa_3",
"est_espar_1","1");}

        if (StrMsg == "off"){console.log("EstEspazor_1 off");ActualizarTabla("casa_3",
"est_espar_1","0");}
    }

```

```

}

if (StrTopic == "Casa_3/EstControl_1")

{

    if (StrMsg == "ctr_on"){console.log("EstControl_1
on");ActualizarTabla("casa_3", "est_ctr_1","1");}

    if (StrMsg == "ctr_off"){console.log("EstControl_1
off");ActualizarTabla("casa_3", "est_ctr_1","0");}

}

if (StrTopic == "Casa_3/EstEspazor_2")

{

    if (StrMsg == "on"){console.log("EstEspazor_2 on");ActualizarTabla("casa_3",
"est_espar_2","1");}

    if (StrMsg == "off"){console.log("EstEspazor_2 off");ActualizarTabla("casa_3",
"est_espar_2","0");}

}

if (StrTopic == "Casa_3/EstControl_2")

{

    if (StrMsg == "ctr_on"){console.log("EstControl_2
on");ActualizarTabla("casa_3", "est_ctr_2","1");}

    if (StrMsg == "ctr_off"){console.log("EstControl_2
off");ActualizarTabla("casa_3", "est_ctr_2","0");}

}

//client.publish("MyCasa/DataIn", "<<<envio_data: " + message.toString() +
">>>");

```

```

    //console.log("Respuesta Enviada ...");
});

//nos conectamos con las tablas de datos de -----
//para leer la informacion de estas tablas
con.connect(function(err) {
    if (err) throw err;
    //una vez conectados, podemos hacer consultas.
    console.log("Conexion a MYSQL exitosa!!!")
});
//para mantener la sesion con mysql abierta
//hace una consulta y con eso mysql no desconecta
//al cliente
setInterval(function() {
    var query = 'SELECT 1 + 1 as result';
    con.query(query, function(err, result, fields) {
        if (err) throw err;
    });
}, 5000);

```

7.7 Anexo G Código ESP32-CAM

```

#include "src/OV2640.h"
#include <WiFi.h>
#include <WebServer.h>
#include <WiFiClient.h>

```

```

// Select camera model

#define CAMERA_MODEL_WROVER_KIT

#define CAMERA_MODEL_ESP_EYE

#define CAMERA_MODEL_M5STACK_PSRAM

#define CAMERA_MODEL_M5STACK_WIDE

#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

#include "home_wifi_multi.h"

#define SSID1 "Nombre red WiFi"

#define PWD1 "Contraseña WiFi"

OV2640 cam;

WebServer server(80);

const char HEADER[] = "HTTP/1.1 200 OK\r\n" \
    "Access-Control-Allow-Origin: *\r\n" \
    "Content-Type:                                multipart/x-mixed-replace;
boundary=12345678900000000000000987654321\r\n";

const char BOUNDARY[] = "\r\n--12345678900000000000000987654321\r\n";

const char CTNTTYPE[] = "Content-Type: image/jpeg\r\nContent-Length: ";

const int hdrLen = strlen(HEADER);

const int bdrLen = strlen(BOUNDARY);

const int cntLen = strlen(CTNTTYPE);

IPAddress local_IP(255, 255, 255, 255);

IPAddress gateway(255, 255, 255, 255);

IPAddress subnet(255, 255, 255, 255);

```

```

void handle_jpg_stream(void)
{
    char buf[32];

    int s;

    WiFiClient client = server.client();

    client.write(HEADER, hdrLen);

    client.write(BOUNDARY, bdrLen);

    while (true)
    {
        if (!client.connected()) break;

        cam.run();

        s = cam.getSize();

        client.write(CTNTTYPE, cntLen);

        sprintf( buf, "%d\r\n\r\n", s );

        client.write(buf, strlen(buf));

        client.write((char *)cam.getfb(), s);

        client.write(BOUNDARY, bdrLen);

    }
}

const char JHEADER[] = "HTTP/1.1 200 OK\r\n" \
    "Content-disposition: inline; filename=capture.jpg\r\n" \
    "Content-type: image/jpeg\r\n\r\n";

const int jhdLen = strlen(JHEADER);

void handle_jpg(void)

```

```

{
    WiFiClient client = server.client();

    cam.run();

    if (!client.connected()) return;

    client.write(JHEADER, jhdLen);

    client.write((char *)cam.getfb(), cam.getSize());
}

void handleNotFound()
{
    String message = "Server is running!\n\n";

    message += "URI: ";

    message += server.uri();

    message += "\nMethod: ";

    message += (server.method() == HTTP_GET) ? "GET" : "POST";

    message += "\nArguments: ";

    message += server.args();

    message += "\n";

    server.send(200, "text / plain", message);
}

void setup()
{
    Serial.begin(115200);

    //while (!Serial);      //wait for serial connection.

    camera_config_t config;

```

```
config.ledc_channel = LEDC_CHANNEL_0;

config.ledc_timer = LEDC_TIMER_0;

config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;

config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

// Frame parameters

// config.frame_size = FRAMESIZE_UXGA;
config.frame_size = FRAMESIZE_QVGA;

config.jpeg_quality = 12;
```

```

    config.fb_count = 2;

#if defined(CAMERA_MODEL_ESP_EYE)

    pinMode(13, INPUT_PULLUP);
    pinMode(14, INPUT_PULLUP);

#endif

    cam.init(config);

    IPAddress ip;

    WiFi.mode(WIFI_STA);

    if (!WiFi.config(local_IP, gateway, subnet)) {

        Serial.println("STA Failed to configure");

    }

    WiFi.begin(SSID1, PWD1);

    while (WiFi.status() != WL_CONNECTED)

    {

        delay(500);

        Serial.print(F("."));

    }

    ip = WiFi.localIP();

    Serial.println(F("WiFi connected"));

    Serial.println("");

    Serial.println(ip);

    Serial.print("Stream Link: http://");

    Serial.print(ip);

    Serial.println("/mjpeg/1");

```

```

server.on("/mjpeg/1", HTTP_GET, handle_jpg_stream);

server.on("/jpg", HTTP_GET, handle_jpg);

server.onNotFound(handleNotFound);

server.begin();

}

void loop()

{

server.handleClient();

}

```

7.8 Anexo H Código Aplicación Android Studio

LoginActivity

```

package com.innovadomotics.esp32_iot;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.media.MediaPlayer;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import org.eclipse.paho.android.service.MqttAndroidClient;

import org.eclipse.paho.client.mqttv3.IMqttActionListener;

import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;

```

```

import org.eclipse.paho.client.mqttv3.IMqttToken;

import org.eclipse.paho.client.mqttv3.MqttCallback;

import org.eclipse.paho.client.mqttv3.MqttClient;

import org.eclipse.paho.client.mqttv3.MqttConnectOptions;

import org.eclipse.paho.client.mqttv3.MqttException;

import org.eclipse.paho.client.mqttv3.MqttMessage;

public class LoginActivity extends AppCompatActivity {

    Button IdBtnLogin;

    EditText txtName, TxtKeyCasa;

    //XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

    //CONEXION MQTT

    MqttAndroidClient client;

    final private String username = "Usuario mqtt";

    final private String password = "Contraseña";

    String Topic_Gen = "Casa_1";

    String MensajeInEmqx;

    int EnableAllSend = 0;

    //XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_login);

        IdBtnLogin = (Button) findViewById(R.id.IdBtnLogin);

        txtName = (EditText) findViewById(R.id.txtName);

```

```

TxtKeyCasa = (EditText) findViewById(R.id.TxtKeyCasa);

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//-----

String clientId = MqttClient.generateClientId();

client      =      new      MqttAndroidClient(this.getApplicationContext(),
"tcp://arcadiaiot2022.tk:1883", clientId);

//CONEXION CON EL SERVIDOR MQTT

try {

    MqttConnectOptions options = new MqttConnectOptions();

    options.setMqttVersion(MqttConnectOptions.MQTT_VERSION_3_1);

    //String Topic_Gen2 = "Casa_1";

    byte[] payload = "some payload".getBytes();

    options.setWill(Topic_Gen, payload ,0,false);

    options.setUserName(username);

    options.setPassword(password.toCharArray());

    IMqttToken token = client.connect(options);

    token.setActionCallback(new IMqttActionListener() {

        @Override

        public void onSuccess(IMqttToken asyncActionToken) {

            //String Topic_Gen3 = "Casa_1";

            //CUANDO LA CONEXION ES EXITOSA SE PROCEDE CON LA

SUSCRIPCION DE LOS TOPICOS

            SuscripcionTopic(Topic_Gen + "/key_valida",0);

```

```

        Toast.makeText(LoginActivity.this, "CONEXION EXITOSA",
Toast.LENGTH_LONG).show();

        SendTopicMsg(Topic_Gen+ "/EstTotalDisp","data");
    }

    @Override

    public void onFailure(IMqttToken asyncActionToken, Throwable
exception) {

        // Something went wrong e.g. connection timeout or firewall problems

        //Log.d(TAG, "onFailure");

        Toast.makeText(LoginActivity.this, "ERROR DE CONEXION",
Toast.LENGTH_LONG).show();

    }

});

} catch (MqttException e) {

    e.printStackTrace();

}

client.setCallback(new MqttCallback() {

    @Override

    public void connectionLost(Throwable cause) {

    }

    //CUANDO LLEGA UN DATO EN ESTE SITIO SE VERIFICA EL MISMO

    //QUE ES TIPO STRING PERO SE CONVIERTE A LOS QUE SE

REQUIERE

    @Override

```

```

        public void messageArrived(String topic, MqttMessage message) throws
Exception {

        //AQUI SE VISUALIZA CUANDO UN MENSAJE LLEGA
        if(topic.matches(Topic_Gen + "/TablaMysql"))
        {

            MensajeInEmqx = new String(message.getPayload());

            String[] parts = MensajeInEmqx.split(",");

            if (parts[2].equals("2")) {

                MediaPlayer    mp    =    MediaPlayer.create(LoginActivity.this,
R.raw.alerta);

                mp.start();

            }

            if (parts[3].equals("2")) {

                MediaPlayer    mp    =    MediaPlayer.create(LoginActivity.this,
R.raw.alerta);

                mp.start();

            }

            if (parts[4].equals("2")) {

                MediaPlayer    mp    =    MediaPlayer.create(LoginActivity.this,
R.raw.alerta);

                mp.start();

            }

            if (parts[5].equals("2")) {

```

```

        MediaPlayer mp = MediaPlayer.create(LoginActivity.this,
R.raw.alerta);

        mp.start();
    }
}

if(topic.matches(Topic_Gen + "/key_valida"))
{
    MensajeInEmqx = new String(message.getPayload());
    if(MensajeInEmqx.equals("exito"))
    {
        finishAffinity();

        Intent intend = new Intent(LoginActivity.this , MainActivity.class);
        startActivity(intend);
    }

    if(MensajeInEmqx.equals("error"))
    {
        Toast.makeText(LoginActivity.this, "CLAVE O USUARIO
INCORRECTOS", Toast.LENGTH_LONG).show();
    }

    if(MensajeInEmqx.equals("peligro"))
    {
        MediaPlayer mp = MediaPlayer.create(LoginActivity.this,
R.raw.alerta);

        mp.start();
    }
}
}

```

```

    }

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX

//PERMITE VISUALIZAR LOS DATOS QUE INGRESAN

//String MensajeInEmqx = "topic: " + topic + ", msg: " + new
String(message.getPayload());

//String MensajeInEmqx = "T: " + topic + ",M: " + new
String(message.getPayload());

//IdTvwTest.setText(MensajeInEmqx);

//Toast.makeText(LoginActivity.this, MensajeInEmqx,
Toast.LENGTH_LONG).show();

}

```



```

try {
    MqttMessage message = new MqttMessage((mensaje).getBytes());
    client.publish(topic,message);
} catch (MqttException e)
{
    e.printStackTrace();
}
}

public void SuscripcionTopic(String topic,int qos)
{
    try {
        IMqttToken subToken = client.subscribe(topic, qos);
        subToken.setActionCallback(new IMqttActionListener() {
            @Override
            public void onSuccess(IMqttToken asyncActionToken) {
                // The message was published
                // Toast.makeText(MainActivity.this, "The message was published",
Toast.LENGTH_LONG).show();
            }
            @Override
            public void onFailure(IMqttToken asyncActionToken,
                Throwable exception) {
                // The subscription could not be performed, maybe the user was not
                // authorized to subscribe on the specified topic e.g. using wildcards

```

```

        Toast.makeText(LoginActivity.this, "NO SE REALIZO LA
SUSCRIPCION", Toast.LENGTH_LONG).show();

    }

});

} catch (MqttException e) {

    e.printStackTrace();

}

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
}

```

MainActivity

```

package com.innovadomotics.esp32_iot;

import androidx.appcompat.app.AppCompatActivity;

import android.content.SharedPreferences;

import android.media.MediaPlayer;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.view.View;

import android.webkit.WebView;

import android.widget.Button;

import android.widget.CompoundButton;

import android.widget.Switch;

import android.widget.TextView;

```

```

import android.widget.Toast;

import org.eclipse.paho.android.service.MqttAndroidClient;

import org.eclipse.paho.client.mqttv3.IMqttActionListener;

import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;

import org.eclipse.paho.client.mqttv3.IMqttToken;

import org.eclipse.paho.client.mqttv3.MqttCallback;

import org.eclipse.paho.client.mqttv3.MqttClient;

import org.eclipse.paho.client.mqttv3.MqttConnectOptions;

import org.eclipse.paho.client.mqttv3.MqttException;

import org.eclipse.paho.client.mqttv3.MqttMessage;

import com.google.gson.Gson;

import com.google.gson.reflect.TypeToken;

import java.lang.reflect.Type;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    //-----

    ArrayList<String> ArrayDataPut, ArrayDataGet;

    ArrayList<String>[] ArrayMatrizPut ,ArrayMatrizGet;

    SharedPreferences shared;

    //RECORDAR QUE EN ANDROID MANIFES HAY PERMISOS PARA
CONEXION A INTERNET

    // EN BLUID.GRADLE ESTA LA CONFIGURACION

    //EL QUE PUBLICA EN ARDUINO - SE SUSCRIBE DESDE ANDROID Y
VICEVERSA

```

```

Button IdBtnCam_1, IdBtnCam_2,IdBtnLimpiar;

WebView IdWeb_1;

Switch

IdSwLuz_1,IdSwLuz_2,IdSwAsp_1,IdSwAsp_2,IdSwSm_1,IdSwSm_2,IdSwSPtr,Id
SwSVen;

Switch IdSwEnbAsp_1, IdSwEnbAsp_2;

TextView TxtDataAsp_1,TxtDataAsp_2,IdTwwTest;

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

//CONEXION MQTT

MqttAndroidClient client;

final private String username = "Usuario Mqtt";

final private String password = "Contraseña";

String Topic_Gen = "Casa_1";

String MensajeInEmqx;

int EnableAllSend = 0;

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    //-----

    shared = PreferenceManager.getDefaultSharedPreferences(MainActivity.this);

    ArrayDataPut = new ArrayList();

    ArrayDataGet = new ArrayList();

```

```

ArrayMatrizPut = new ArrayList[1]; //Crea la array
ArrayMatrizGet = new ArrayList[1]; //Crea la array
//inicializa los componentes
ArrayMatrizPut [0] = new ArrayList<String>();
ArrayMatrizGet[0] = new ArrayList<String>();
IdBtnCam_1 = (Button) findViewById(R.id.IdBtnCam_1);
IdBtnCam_2 = (Button) findViewById(R.id.IdBtnCam_2);
IdBtnLimpiar = (Button) findViewById(R.id.IdBtnLimpiar);
IdWeb_1 = (WebView) findViewById(R.id.IdWeb_1);
IdSwLuz_1 = (Switch) findViewById(R.id.IdSwLuz_1);
IdSwLuz_2 = (Switch) findViewById(R.id.IdSwLuz_2);
IdSwAsp_1 = (Switch) findViewById(R.id.IdSwAsp_1);
IdSwAsp_2 = (Switch) findViewById(R.id.IdSwAsp_2);
IdSwSm_1 = (Switch) findViewById(R.id.IdSwSm_1);
IdSwSm_2 = (Switch) findViewById(R.id.IdSwSm_2);
IdSwSPtr = (Switch) findViewById(R.id.IdSwSPtr);
IdSwSVen = (Switch) findViewById(R.id.IdSwSVen);
IdSwEnbAsp_1 = (Switch) findViewById(R.id.IdSwEnbAsp_1);
IdSwEnbAsp_2 = (Switch) findViewById(R.id.IdSwEnbAsp_2);
TxtDataAsp_1 = (TextView) findViewById(R.id.TxtDataAsp_1);
TxtDataAsp_2 = (TextView) findViewById(R.id.TxtDataAsp_2);
IdTwvTest = (TextView) findViewById(R.id.IdTwvTest);
IdTwvTest.setText(" ");
EnableAllSend = 0;

```

```

IdSwLuz_1.setEnabled(false);

IdSwLuz_2.setEnabled(false);

IdSwAsp_1.setEnabled(false);

IdSwAsp_2.setEnabled(false);

IdSwSm_1.setEnabled(false);

IdSwSm_2.setEnabled(false);

IdSwSPtr.setEnabled(false);

IdSwSVen.setEnabled(false);

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

IdTwvTest.setText(" ");

try{

    ArrayDataGet.clear();

    ArrayMatrizGet[0].clear();

    ArrayMatrizGet[0] = getArrayList("MyArray_01");

    //Toast.makeText(MainActivity.this,

ArrayMatrizGet[0].toString(),Toast.LENGTH_SHORT).show();

    //Toast.makeText(MainActivity.this,

ArrayMatrizGet[0].get(0),Toast.LENGTH_SHORT).show();

    IdTwvTest.setText(ArrayMatrizGet[0].get(0));

}catch(RuntimeException e){

}finally{

    System.out.println("finally");

}

```

```

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//-----

String clientId = MqttClient.generateClientId();

client = new MqttAndroidClient(this.getApplicationContext(),
"tcp://arcadiaiot2022.tk:1883", clientId);

//CONEXION CON EL SERVIDOR MQTT

try {

    MqttConnectOptions options = new MqttConnectOptions();

    options.setMqttVersion(MqttConnectOptions.MQTT_VERSION_3_1);

    //String Topic_Gen2 = "Casa_1";

    byte[] payload = "some payload".getBytes();

    options.setWill(Topic_Gen, payload ,0,false);

    options.setUsername(username);

    options.setPassword(password.toCharArray());

    IMqttToken token = client.connect(options);

    token.setActionCallback(new IMqttActionListener() {

        @Override

        public void onSuccess(IMqttToken asyncActionToken) {

            //String Topic_Gen3 = "Casa_1";

            //CUANDO LA CONEXION ES EXITOSA SE PROCEDE CON LA
SUSCRIPCION DE LOS TOPICOS

            SuscripcionTopic(Topic_Gen + "/EstLuz_1",0);

            SuscripcionTopic(Topic_Gen + "/EstLuz_2",0);

```

```

        SuscripcionTopic(Topic_Gen + "/EstPuerta",2);

        SuscripcionTopic(Topic_Gen + "/EstVentana",2);

        SuscripcionTopic(Topic_Gen + "/EstMov_1",2);

        SuscripcionTopic(Topic_Gen + "/EstMov_2",2);

        //ASPERSORES

        SuscripcionTopic(Topic_Gen + "/HumTemp_1",0);

        SuscripcionTopic(Topic_Gen + "/HumTemp_2",0);

        SuscripcionTopic(Topic_Gen + "/EstEspazor_1",0);

        SuscripcionTopic(Topic_Gen + "/EstEspazor_2",0);

        SuscripcionTopic(Topic_Gen + "/EstControl_1",0);

        SuscripcionTopic(Topic_Gen + "/EstControl_2",0);

        SuscripcionTopic(Topic_Gen + "/TablaMysql",0);

        Toast.makeText(MainActivity.this, "CONEXION EXITOSA",
Toast.LENGTH_LONG).show();

        //REALIZA LA PETICION DE LOS ESTADOS

        SendTopicMsg(Topic_Gen+ "/EstTotalDisp","data");
    }

    @Override

    public void onFailure(IMqttToken asyncActionToken, Throwable
exception) {

        // Something went wrong e.g. connection timeout or firewall problems

        //Log.d(TAG, "onFailure");

        Toast.makeText(MainActivity.this, "ERROR DE CONEXION",
Toast.LENGTH_LONG).show();

```

```

    }

});

} catch (MqttException e) {

    e.printStackTrace();

}

client.setCallback(new MqttCallback() {

    @Override

    public void connectionLost(Throwable cause) {

    }

    //CUANDO LLEGA UN DATO EN ESTE SITIO SE VERIFICA EL MISMO

    //QUE ES TIPO STRING PERO SE CONVIERTE A LOS QUE SE

REQUIERE

    @Override

    public void messageArrived(String topic, MqttMessage message) throws

Exception {

        //AQUI SE VISUALIZA CUANDO UN MENSAJE LLEGA

        if(topic.matches(Topic_Gen + "/TablaMysql"))

        {

            MensajeInEmqx = new String(message.getPayload());

            String[] parts = MensajeInEmqx.split(",");

            if (parts[0].equals("0")) { IdSwLuz_1.setChecked(false);}

            if (parts[0].equals("1")) { IdSwLuz_1.setChecked(true);}

            if (parts[1].equals("0")) { IdSwLuz_2.setChecked(false);}

            if (parts[1].equals("1")) { IdSwLuz_2.setChecked(true);}

```

```

if (parts[2].equals("0")) { IdSwSPtr.setChecked(false);}

if (parts[2].equals("1")) { IdSwSPtr.setChecked(true);}

if (parts[2].equals("2")) {

    IdTwvTest.setText("SENSOR DE PUERTA ACTIVADO");

    IdSwSPtr.setChecked(true);

    MediaPlayer mp = MediaPlayer.create(MainActivity.this,
R.raw.alerta);

    mp.start();

}

if (parts[3].equals("0")) { IdSwSVen.setChecked(false);}

if (parts[3].equals("1")) { IdSwSVen.setChecked(true);}

if (parts[3].equals("2")) {

    IdTwvTest.setText("SENSOR DE VENTANA ACTIVADO");

    IdSwSVen.setChecked(true);

    MediaPlayer mp = MediaPlayer.create(MainActivity.this,
R.raw.alerta);

    mp.start();

}

if (parts[4].equals("0")) { IdSwSm_1.setChecked(false);}

if (parts[4].equals("1")) { IdSwSm_1.setChecked(true);}

if (parts[4].equals("2")) {

    ArrayMatrizPut [0].add("SENSOR DE MOVIMIENTO 1
ACTIVADO");

    IdSwSm_1.setChecked(true);

```

```

        MediaPlayer mp = MediaPlayer.create(MainActivity.this,
R.raw.alerta);

        mp.start();
    }

    if (parts[5].equals("0")) { IdSwSm_2.setChecked(false);}
    if (parts[5].equals("1")) { IdSwSm_2.setChecked(true);}
    if (parts[5].equals("2")) {
        ArrayMatrizPut [0].add("SENSOR DE MOVIMIENTO 2
ACTIVADO");

        IdSwSm_2.setChecked(true);

        MediaPlayer mp = MediaPlayer.create(MainActivity.this,
R.raw.alerta);

        mp.start();
    }

    if (parts[6].equals("0")) { IdSwAsp_1.setChecked(false);}
    if (parts[6].equals("1")) { IdSwAsp_1.setChecked(true);}
    if (parts[7].equals("0")) { IdSwEnbAsp_1.setChecked(false);}
    if (parts[7].equals("1")) { IdSwEnbAsp_1.setChecked(true);}
    if (parts[8].equals("0")) { IdSwAsp_2.setChecked(false);}
    if (parts[8].equals("1")) { IdSwAsp_2.setChecked(true);}
    if (parts[9].equals("0")) { IdSwEnbAsp_2.setChecked(false);}
    if (parts[9].equals("1")) { IdSwEnbAsp_2.setChecked(true);}

    //IdTwvTest.setText(MensajeInEmqx);

    /*

```

```

*
* result[0].luz_1.toString() + "," +
  result[0].luz_2.toString() + "," +
  result[0].est_sen_p.toString() + "," +
  result[0].est_sen_v.toString() + "," +
  result[0].est_sen_m1.toString() + "," +
  result[0].est_sen_m2.toString() + "," +
  result[0].est_espar_1.toString() + "," +
  result[0].est_ctr_1.toString() + "," +
  result[0].est_espar_2.toString() + "," +
  result[0].est_ctr_2.toString();
* */
}
if(topic.matches(Topic_Gen + "/HumTemp_1"))
{
  MensajeInEmqx = new String(message.getPayload());
  String[] parts = MensajeInEmqx.split("@");
  TxtDataAsp_1.setText("1) Humedad: "+parts[0]+ " - Temperatura: "+
parts[1]);
}
if(topic.matches(Topic_Gen + "/HumTemp_2"))
{
  MensajeInEmqx = new String(message.getPayload());
  String[] parts = MensajeInEmqx.split("@");

```



```

        if(MensajeInEmqx.matches("ctr_on"))    {    IdTwvTest.setText("
");IdTwvTest.setText("Esparzor Habilidadado"); }

        if(MensajeInEmqx.matches("ctr_off"))   {    IdTwvTest.setText("
");IdTwvTest.setText("Esparzor DesHabilidadado"); }

    }

    if(topic.matches(Topic_Gen + "/EstControl_2"))
    {

        MensajeInEmqx = new String(message.getPayload());

        if(MensajeInEmqx.matches("ctr_on"))    {    IdTwvTest.setText("
");IdTwvTest.setText("Esparzor Habilidadado"); }

        if(MensajeInEmqx.matches("ctr_off"))   {    IdTwvTest.setText("
");IdTwvTest.setText("Esparzor DesHabilidadado"); }

    }

//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

    if(topic.matches(Topic_Gen+ "/EstLuz_1"))
    {

        MensajeInEmqx = new String(message.getPayload());

        if(MensajeInEmqx.matches("on"))       {    IdTwvTest.setText("
");IdTwvTest.setText("Luz Encendida"); }

        if(MensajeInEmqx.matches("off"))      {    IdTwvTest.setText("
");IdTwvTest.setText("Luz Apagada"); }

    }

    if(topic.matches(Topic_Gen+ "/EstLuz_2"))

```

```

        {
            MensajeInEmqx = new String(message.getPayload());

            if(MensajeInEmqx.matches("on"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Luz Encendida"); }

            if(MensajeInEmqx.matches("off"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Luz Apagada"); }

        }

        if(topic.matches(Topic_Gen+ "/EstPuerta"))

        {

            MensajeInEmqx = new String(message.getPayload());

            if(MensajeInEmqx.matches("on"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Sensor Habilitado"); }

            if(MensajeInEmqx.matches("off"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Sensor DesHabilitado"); }

            if(MensajeInEmqx.matches("peligro"))

            {

                //Toast.makeText(getApplicationContext(),"Peligro    Sensor    Puerta",
Toast.LENGTH_SHORT).show();

                ArrayMatrizPut [0].clear();

                ArrayMatrizPut [0].add("SENSOR DE PUERTA ACTIVADO");

                saveArrayList(ArrayMatrizPut [0], "MyArray_01");

                IdTwvTest.setText("SENSOR DE PUERTA ACTIVADO");

                MediaPlayer    mp    =    MediaPlayer.create(MainActivity.this,
R.raw.alerta);

```

```

        mp.start();
    }
}
if(topic.matches(Topic_Gen+ "/EstVentana"))
{
    MensajeInEmqx = new String(message.getPayload());
    if(MensajeInEmqx.matches("on"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Sensor Habilitado"); }
    if(MensajeInEmqx.matches("off"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Sensor DesHabilitado"); }
    if(MensajeInEmqx.matches("peligro"))
    {
        //Toast.makeText(getBaseContext(),"Peligro    Sensor    Puerta",
Toast.LENGTH_SHORT).show();
        ArrayMatrizPut [0].clear();
        ArrayMatrizPut [0].add("SENSOR DE VENTANA ACTIVADO");
        saveArrayList(ArrayMatrizPut [0], "MyArray_01");
        IdTwvTest.setText("SENSOR DE VENTANA ACTIVADO");
        MediaPlayer    mp    =    MediaPlayer.create(MainActivity.this,
R.raw.alerta);
        mp.start();
    }
}
if(topic.matches(Topic_Gen+ "/EstMov_1"))

```

```

    {
        MensajeInEmqx = new String(message.getPayload());

        if(MensajeInEmqx.matches("on"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Sensor Habilitado"); }

        if(MensajeInEmqx.matches("off"))        {        IdTwvTest.setText("
");IdTwvTest.setText("Sensor DesHabilitado"); }

        if(MensajeInEmqx.matches("peligro"))
        {
            //Toast.makeText(getBaseContext(),"Peligro    Sensor    Puerta",
Toast.LENGTH_SHORT).show();

            ArrayMatrizPut [0].clear();

            ArrayMatrizPut [0].add("SENSOR    DE    MOVIMIENTO    1
ACTIVADO");

            saveArrayList(ArrayMatrizPut [0], "MyArray_01");

            IdTwvTest.setText("SENSOR DE MOVIMIENTO 1 ACTIVADO");

            MediaPlayer    mp    =    MediaPlayer.create(MainActivity.this,
R.raw.alerta);

            mp.start();

        }
    }

    if(topic.matches(Topic_Gen+ "/EstMov_2"))
    {
        MensajeInEmqx = new String(message.getPayload());

```

```

        if(MensajeInEmqx.matches("on"))        {        IdTvwTest.setText("
");IdTvwTest.setText("Sensor Habilitado"); }

        if(MensajeInEmqx.matches("off"))        {        IdTvwTest.setText("
");IdTvwTest.setText("Sensor DesHabilitado"); }

        if(MensajeInEmqx.matches("peligro"))
        {
            //Toast.makeText(getBaseContext(),"Peligro    Sensor    Puerta",
Toast.LENGTH_SHORT).show();

            ArrayMatrizPut [0].clear();

            ArrayMatrizPut    [0].add("SENSOR    DE    MOVIMIENTO    2
ACTIVADO");

            saveArrayList(ArrayMatrizPut [0], "MyArray_01");

            IdTvwTest.setText("SENSOR DE MOVIMIENTO 2 ACTIVADO");

            MediaPlayer    mp    =    MediaPlayer.create(MainActivity.this,
R.raw.alerta);

            mp.start();

        }
    }

    //PERMITE VISUALIZAR LOS DATOS QUE INGRESAN

    //String MensajeInEmqx = "topic: " + topic + ", msg: " + new
String(message.getPayload());

    //MensajeInEmqx    =    "T:    "    +    topic    +    ",M:    "    +    new
String(message.getPayload());

    //IdTvwTest.setText(MensajeInEmqx);

```

```

        //Toast.makeText(MainActivity.this,
        MensajeInEmqx,
Toast.LENGTH_LONG).show();

    }

    @Override

    public void deliveryComplete(IMqttDeliveryToken token) {

    }

});

//-----

IdBtnCam_1.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        //Toast.makeText(getApplicationContext(),"Se a presionado el boton 1",
Toast.LENGTH_SHORT).show();

        IdWeb_1.loadUrl("-----");

    }

});

IdBtnCam_2.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        //Toast.makeText(getApplicationContext(),"Se a presionado el boton 1",
Toast.LENGTH_SHORT).show();

        IdWeb_1.loadUrl("-----");

    }

});

```

```

IdBtnLimpiar.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        ArrayMatrizPut [0].clear();

        ArrayMatrizPut [0].add(" ");

        saveArrayList(ArrayMatrizPut [0], "MyArray_01");

        IdTvwTest.setText(" ");

        EnableAllSend = 1;

        IdSwLuz_1.setEnabled(true);

        IdSwLuz_2.setEnabled(true);

        IdSwAsp_1.setEnabled(true);

        IdSwAsp_2.setEnabled(true);

        IdSwSm_1.setEnabled(true);

        IdSwSm_2.setEnabled(true);

        IdSwSPtr.setEnabled(true);

        IdSwSVen.setEnabled(true);

    }

});

IdSwLuz_1.setOnCheckedChangeListener(new

CompoundButton.OnCheckedChangeListener() {

    @Override

    public void onCheckedChanged(CompoundButton buttonView, boolean

isChecked) {

        if (EnableAllSend == 1) {

```

```

        if (IdSwLuz_1.isChecked()) {
            SendTopicMsg(Topic_Gen+ "/Luz_1","on");
            //Toast.makeText(getBaseContext(),"LUZ_1      ON      111",
Toast.LENGTH_SHORT).show();
        } else {
            SendTopicMsg(Topic_Gen+ "/Luz_1","off");
            //Toast.makeText(getBaseContext(),"LUZ_1      OFF      000",
Toast.LENGTH_SHORT).show();
        }
    }
}
});

IdSwLuz_2.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void  onCheckedChanged(CompoundButton  buttonView, boolean
isChecked) {
        if (EnableAllSend == 1) {
            if (IdSwLuz_2.isChecked()) {
                SendTopicMsg(Topic_Gen+ "/Luz_2","on");
                //Toast.makeText(getBaseContext(),"LUZ_2      ON      111",
Toast.LENGTH_SHORT).show();
            } else {
                SendTopicMsg(Topic_Gen+ "/Luz_2","off");

```

```

        //Toast.makeText(getBaseContext(),"LUZ_2      OFF      000",
Toast.LENGTH_SHORT).show();

    }

}

});

    IdSwAsp_1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

    @Override

    public void  onCheckedChanged(CompoundButton  buttonView, boolean
isChecked) {

        if (EnableAllSend == 1) {

            if (IdSwAsp_1.isChecked()) {

                //Toast.makeText(getBaseContext(),"ASP_1      ON      ok      111",
Toast.LENGTH_SHORT).show();

                SendTopicMsg(Topic_Gen+ "/Esparzor_1","on");

            } else {

                //Toast.makeText(getBaseContext(),"ASP_1      OFF      ok      000",
Toast.LENGTH_SHORT).show();

                SendTopicMsg(Topic_Gen+ "/Esparzor_1","off");

            }

        }

    }

});

```

```

        IdSwEnbAsp_1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {

                if (EnableAllSend == 1) {

                        if (IdSwEnbAsp_1.isChecked()) {

                                SendTopicMsg(Topic_Gen+ "/Esparzor_1","ctr_on");

                                //Toast.makeText(getBaseContext(),"Ventana      ON      111",
Toast.LENGTH_SHORT).show();

                                } else {

                                        SendTopicMsg(Topic_Gen+ "/Esparzor_1","ctr_off");

                                        //Toast.makeText(getBaseContext(),"Ventana      OFF      000",
Toast.LENGTH_SHORT).show();

                                }

                        }

                });

        IdSwAsp_2.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {

                if (EnableAllSend == 1) {

```

```

        if (IdSwAsp_2.isChecked()) {
            SendTopicMsg(Topic_Gen+ "/Esparzor_2","on");
            //Toast.makeText(getBaseContext(),"ASP_2      ON      111",
Toast.LENGTH_SHORT).show();
        } else {
            SendTopicMsg(Topic_Gen+ "/Esparzor_2","off");
            //Toast.makeText(getBaseContext(),"ASP_2      OFF      000",
Toast.LENGTH_SHORT).show();
        }
    }
});

IdSwEnbAsp_2.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
        if (EnableAllSend == 1) {
            if (IdSwEnbAsp_2.isChecked()) {
                SendTopicMsg(Topic_Gen+ "/Esparzor_2","ctr_on");
                //Toast.makeText(getBaseContext(),"Ventana      ON      111",
Toast.LENGTH_SHORT).show();
            } else {
                SendTopicMsg(Topic_Gen+ "/Esparzor_2","ctr_off");

```

```

        //Toast.makeText(getBaseContext(),"Ventana      OFF      000",
Toast.LENGTH_SHORT).show();

    }

}

});

    IdSwSm_1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

    @Override

    public void  onCheckedChanged(CompoundButton  buttonView, boolean
isChecked) {

        if (EnableAllSend == 1) {

            if (IdSwSm_1.isChecked()) {

                SendTopicMsg(Topic_Gen+ "/SenMov_1","on");

                //Toast.makeText(getBaseContext(),"SM_1      ON      111",
Toast.LENGTH_SHORT).show();

            } else {

                SendTopicMsg(Topic_Gen+ "/SenMov_1","off");

                //Toast.makeText(getBaseContext(),"SM_1      OFF      000",
Toast.LENGTH_SHORT).show();

            }

        }

    }

});

```

```

        IdSwSm_2.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {

                if (EnableAllSend == 1) {

                        if (IdSwSm_2.isChecked()) {

                                SendTopicMsg(Topic_Gen+ "/SenMov_2","on");

                                //Toast.makeText(getBaseContext(),"SM_2      ON      111",
Toast.LENGTH_SHORT).show();

                                } else {

                                        SendTopicMsg(Topic_Gen+ "/SenMov_2","off");

                                        //Toast.makeText(getBaseContext(),"SM_2      OFF      000",
Toast.LENGTH_SHORT).show();

                                }

                        }

                });

        IdSwSPtr.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {

                if (EnableAllSend == 1) {

```

```

        if (IdSwSPtr.isChecked()) {
            SendTopicMsg(Topic_Gen+ "/SenPuerta","on");
            //Toast.makeText(getBaseContext(),"Puerta      ON      111",
Toast.LENGTH_SHORT).show();
        } else {
            SendTopicMsg(Topic_Gen+ "/SenPuerta","off");
            //Toast.makeText(getBaseContext(),"Puerta      OFF      000",
Toast.LENGTH_SHORT).show();
        }
    }
});

IdSwSVen.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
        if (EnableAllSend == 1) {
            if (IdSwSVen.isChecked()) {
                SendTopicMsg(Topic_Gen+ "/SenVentana","on");
                //Toast.makeText(getBaseContext(),"Ventana      ON      111",
Toast.LENGTH_SHORT).show();
            } else {
                SendTopicMsg(Topic_Gen+ "/SenVentana","off");

```

```

        //Toast.makeText(getApplicationContext(),"Ventana OFF 000",
Toast.LENGTH_SHORT).show();

    }

}

});

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

public void saveArrayList(ArrayList<String> list, String key){

    SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(MainActivity.this);

    SharedPreferences.Editor editor = prefs.edit();

    Gson gson = new Gson();

    String json = gson.toJson(list);

    editor.putString(key, json);

    editor.apply();

}

public ArrayList<String> getArrayList(String key){

    SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(MainActivity.this);

    Gson gson = new Gson();

```

```

String json = prefs.getString(key, null);

Type type = new TypeToken<ArrayList<String>>() {}.getType();

return gson.fromJson(json, type);
}

@Override

public void onPause()

{

    finishAffinity();

    super.onPause();

    //IdWeb_1 = (WebView) findViewById(R.id.IdWeb_1);

    //IdWeb_1.destroy();

}

@Override

public void onStop()

{

    finishAffinity();

    super.onStop();

    //IdWeb_1 = (WebView) findViewById(R.id.IdWeb_1);

    //IdWeb_1.destroy();

}

@Override

public void onDestroy()

{

    finishAffinity();

```



```

        // Toast.makeText(MainActivity.this, "The message was published",
Toast.LENGTH_LONG).show();

    }

    @Override

    public void onFailure(IMqttToken asyncActionToken,

        Throwable exception) {

        // The subscription could not be performed, maybe the user was not

        // authorized to subscribe on the specified topic e.g. using wildcards

        Toast.makeText(MainActivity.this, "NO SE REALIZO LA

SUSCRIPCION", Toast.LENGTH_LONG).show();

    }

});

} catch (MqttException e) {

    e.printStackTrace();

}

}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

}

```

Bibliografía

- Academic. (2010). *WinNuke*. Retrieved from <https://es-academic.com/dic.nsf/eswiki/1227221>
- Agudelo, C. A. (2020). Arquitectura para automatizar respuesta a incidentes de seguridad de la información relacionados con ataques internos mediante la ejecución de técnicas spoofing. (*Tesis de Maestría*). Institución Universitaria ITM, Medellín.
- Almeida, C. A., & Herrera, L. R. (2019). La ciberseguridad en el Ecuador, una propuesta de organización. *Revista de Ciencias de Seguridad y Defensa*, 156-169.
- Autodesk. (2022). *Eagle PCB design made easy for every engineer*. Retrieved from https://www.autodesk.com/products/eagle/overview?us_oa=dotcom-us&us_si=4a0a9fe1-dd10-4c9e-98a3-04bcf2123ad0&us_st=EAGLE&term=1-YEAR&tab=subscription
- Canales, M. P. (2021, Enero 22). *Derechos Digitales*. Retrieved from <https://www.derechosdigitales.org/15138/proteccion-de-datos-personales-en-ecuador-el-momento-es-ahora/#:~:text=El%20objetivo%20de%20la%20ley,de%20agentes%20p%C3%BAblicos%20y%20privados>.
- Chaupis Guardia, L. D. (2018). *Redes Físicas*. (*Tesis de grado*). Universidad Nacional de Educación, Lima.
- Chitiva Bernal, Y. E. (2020). Internet of Things (IoT) DISEÑO DE UNA RED DE IoT PARA EL HOGAR. (*Tesis de Grado*). Universidad Cooperativa de Colombia, Bogotá.
- Cordoba Aguilar, R. A., & Vergara Rodríguez, D. A. (2021). ASESOR CHATBOT DE CRÉDITO PARA LA FINANCIACIÓN DE MATRÍCULA. (*Tesis de Grado*). Universidad Distrital Francisco José de Caldas, Bogotá.
- Creatividad Ahora. (2020, Julio 20). Descripción resumida de las partes de la tarjeta Raspberry Pi 4 [video]. Youtube. Retrieved from https://www.youtube.com/watch?v=bRyvAj_xbv8
- Darktrace. (2020). *Cyber AI Platform Anrigena*. Retrieved from <https://www.darktrace.com/en/technology/>
- Developers, A. (2021, Mayo 17). *Android Developers*. Retrieved from <https://developer.android.com/studio/intro?hl=es-419>

- Díaz Ronceros, E. (2020). Relevancia de la ejecución experimental de proyectos con microcontroladores en el aprendizaje de la ingeniería electrónica. *Educación*, 48-72.
- Díaz, J. M. (2019, Octubre). Riesgos y vulnerabilidades de la denegación de servicio distribuidos en internet de las cosas. *Revista de Bioética y Derecho*(46). Retrieved from http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1886-58872019000200006&lang=es
- Gálvez Marchán, C. A., & Mite Vera, F. J. (2020). Diseño e implementación de un sistema de domótica mediante el uso de módulos sonoff y de la tecnología de google assistant, para la congregación de la iglesia del Nazareno en el sector sur de Guayaquil. (*Tesis de Grado*). Universidad de Guayaquil., Guayaquil.
- Gamón, V. P. (2017, Abril 26). Internet, la nueva era del delito: ciberdelito, ciberterrorismo, legislación, y ciberseguridad. *Urvio*, 80-93. Retrieved from <http://200.41.82.22/bitstream/10469/12243/1/RFLACSO-06-Pons.pdf>
- Haro Huerta, R. M. (2022). ANÁLISIS DE VULNERABILIDADES EN LA RED DEL ISP: "CAFANET" PARROQUIA ISLA DE BEJUCAL, AÑO 2022. (*Tesis de Grado*). Universidad Técnica de Babahoyo, Babahoyo.
- Haro, E., Guarda, T., Zambrano Peñaherrera, A. O., & Ninahualpa Quiña, G. (2019, Enero). Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot. *Revista Ibérica de Sistemas e Tecnologías de Informação*, pp. 309-321. Retrieved from <https://www.proquest.com/openview/a78cfaa62708fd24f38ac8d1025050eb/1?pq-origsite=gscholar&cbl=1006393>
- Hirare, C. S. (2017, Junio). Ciberseguridad presentación del Dossier. *Urvio*(20), 8-15. Retrieved from <http://200.41.82.22/bitstream/10469/12197/1/RFLACSO-01-Sancho.pdf>
- Inter-American Development Bank. (2020). *IDB*. Retrieved from <https://publications.iadb.org/publications/spanish/document/Reporte-Ciberseguridad-2020-riesgos-avances-y-el-camino-a-seguir-en-America-Latina-y-el-Caribe.pdf>
- Jiménez, J. (2021, Abril 29). *Redes Zone*. Retrieved from <https://www.redeszone.net/tutoriales/redes-cable/ataques-arp-spoofing-evitar/>
- Jornet, R. (2019, Octubre 29). *Así era ARPANET, la "abuela de Internet"*. Retrieved from <https://www.domestika.org/es/blog/2380-asi-era-arpamet-la-abuela-de-internet>

- Maldonado Simbaña, N. F., & Pilatuña Collaguazo, M. D. (2021). Desarrollo de guías prácticas para la ejecución de ataques y aseguramiento de la información. (*Tesis de Grado*). Escuela Politécnica Nacional, Quito.
- Márquez Díaz, J., Sampedro, L., & Vargas, F. (2002). Instalación y configuración de Apache un servidor Web gratis. *Redalyc*, 10-23.
- Martínez Canata, J. M., & Rodríguez Gómez, R. A. (2016). IMPLEMENTACIÓN Y PRUEBA DE CONCEPTO DE UN LABORATORIO DE SEGURIDAD PARA EL S.O. ANDROID UTILIZANDO LIVE-USB. Retrieved from https://www.researchgate.net/profile/Rafael-Rodriguez-Gomez/publication/308971646_Implementacion_y_prueba_de_concepto_de_un_laboratorio_de_seguridad_para_el_SO_Android_utilizando_Live-USB/links/57fbcddf08aea0db5a3f5ece/Implementacion-y-prueba-de-concepto-
- ngrok. (2022). *ngrok pricing*. Retrieved from <https://ngrok.com/pricing>
- Nicolás Jiménez, I. (2019). DESARROLLO DE UN MÓDULO DE SEGURIDAD PARA NODOS DE AGREGACIÓN EMPLEADOS EN INTERNET DE LAS COSAS. (*Tesis de Grado*). Universidad Autónoma de Madrid, Madrid.
- Ordorica, I. (2020, Agosto 19). *Incentro*. Retrieved from <https://www.incentro.com/es-ES/blog/que-es-google-cloud-platform>
- Peláez, J. P. (2019). *Internet de las cosas (IoT) con Arduino. Manual práctico* (Vol. I). Madrid, España: Ediciones Paraninfo. Retrieved from https://books.google.es/books?hl=es&lr=&id=73yJDwAAQBAJ&oi=fnd&pg=PP1&dq=arduino&ots=tC2n6yTlln&sig=E-_gt5OedZCYu9d7DbymOCoyz0k#v=onepage&q=arduino&f=false
- Pérez F, A. F., & Guerra J, L. G. (2017). Internet de las Cosas. *Perspectiv@s*, 45-49.
- Pozuelo Saiz de Bustamante, J. I. (2020). ANÁLISIS Y DISEÑO DEL PLAN DE MIGRACIÓN PARA LA MEJORA DEL RENDIMIENTO Y SEGURIDAD DE UN SERVIDOR CACHÉ DNS. (*Tesis de Grado*). Universidad Politécnica de Madrid, Madrid.
- Quiña Haro, E. D., & Reza Zurita, C. P. (2022). Análisis de vulnerabilidades en la comunicación inalámbrica utilizando el ESP32 para mitigar intrusiones. (*Tesis de Grado*). Universidad Politécnica Salesiana Sede Quito, Quito.
- Redacción. (2022, Junio 8). *Concepto Definición*. Retrieved from <https://conceptodefinicion.de/apache/>
- Rose, K., Eldridge, S., & Chapin, L. (2015, Octubre 15). *Internet Society*. Retrieved from <https://www.internetsociety.org/es/resources/doc/2015/iot-overview/>
- Salazar , J., & Silvestre, S. (2016). *Internet de las cosas* . Praga: TechPedia.

- Silva, A. (2020). *DISTRIBUTED DENIAL OF SERVICE ATTACK IN NETWORKS*. Universidad del Algarve, Faro. Retrieved from <http://eprints.rclis.org/40181/1/new-report-as.pdf>
- Sobrevilla, G., Hernández, J., Velasco Elizondo, P., & Soriano, S. (2017). Aplicando Scrum y Prácticas de Ingeniería de Software para la Mejora Continua del Desarrollo de un Sistema Ciber-Físico. *ReCIBE*, 1-15.
- Solis Yagual, C. R. (2019). DISEÑO DEL PROTOTIPO DE UN GUANTE CONTROLADO CON ARDUINO QUE PERMITE LA DETECCIÓN DE OBSTÁCULOS POR MEDIO DE SENSORES PARA MEJORAR EL DESPLAZAMIENTO DENTRO DEL HOGAR A PERSONAS CON BAJOS RECURSOS QUE CUENTEN CON DISCAPACIDAD VISUAL. (*Tesis de Grado*). Universidad de Guayaquil, Guayaquil.
- Vásconez, G. J. (2019). Diseño e implementación de un sistema de adquisición de señales ECG en tiempo real utilizando la tecnología Long Term Evolution (LTE). (*Tesis de Grado*). Universidad Nacional de Chimborazo, Riobamba.
- Yépez Llerena, E. D., & Armijos Guillen, K. F. (2020). APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE CONTROL ACADÉMICO UNACH. (*Tesis de Grado*). Universidad Nacional de Chimborazo, Riobamba.
- Zanabria Ticona, E. D., & Cayo Mamami, E. (2018). SEGURIDAD INFORMÁTICA EN DISPOSITIVOS MÓVILES CON SISTEMAS OPERATIVOS ANDROID MEDIANTE PENTESTING. (*Tesis de Grado*). Universidad Nacional del Altiplano, Puno.