



ÁNALISIS DE UNA ARQUITECTURA IoT EN APLICACIONES MÉDICAS PARA 5G

TESIS

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN ELECTRÓNICA

PRESENTA

Francisco Antonio Palos Angulo

CIUDAD OBREGÓN, SONORA

JULIO DE 2020

AGRADECIMIENTOS

El amor recibido y la paciencia con la que cada día se preocupaba mi madre por mis avances y desarrollo de mis proyectos, es simplemente único y se refleja en la vida de un hijo.

Gracias a mi madre Fidelia Angulo Lara por ser la principal promotora de mis sueños, gracias a ella por confiar y creer en mí y en mis expectativas, por estar dispuesta a acompañarme en cada uno de ellos que siempre fueron largas y agotadoras noches de estudio, noches en las que sin su apoyo y en ocasiones compañía no lo hubiera logrado, la llegada de palabras sinceras y llenas de amor para mí. Gracias a mi hermano Francisco Alberto Palos Angulo por siempre desear y anhelar siempre lo mejor para mi vida, gracias por cada consejo y por cada una de sus palabras que me guiaron durante mi vida.

Gracias a mis compañeros, cada uno en su estilo me ayudaron a crecer en todo este tiempo. Gracias, porque siendo tan diferentes ahuyentaron la monotonía, volviéndose como una familia la cual me apoyó, me cuidó, y respetó, por haberme aportado tanto aprendizaje y para la culminación de este momento.

A mis maestros, por la paciencia, la entrega y la experiencia que me aportaron en cada una de mis clases y proyectos, el hacer de ellas siempre un reto, obligarme a seguir esforzándome cada día y sacar lo mejor de mí en cada uno de ellos. Gracias a todos ellos dejan un legado que me acompañará en cada uno de mis proyectos futuros.

En especial a mi asesor de tesis Dr. Armando García por la oportunidad y la confianza en la realización de este proyecto, que confió siempre en mis decisiones, convicciones y conocimientos para la culminación de este, que sin su confianza y conocimientos no lo hubiera logrado.

Gracias a la vida por este nuevo triunfo, gracias a todas las personas que me apoyaron incondicionalmente y creyeron en la realización de esta tesis.

ÍNDICE GENERAL

Página

Lista de Figuras	v
-------------------------------	----------

Lista de Tablas	viii
------------------------------	-------------

Resumen	ix
----------------------	-----------

CAPÍTULO I. INTRODUCCIÓN.....	1
--------------------------------------	----------

1.1	Antecedentes	1
1.2	Planteamiento del problema.....	2
1.3	Objetivo	3
1.4	Justificación	3
1.5	Delimitaciones	4
1.6	Limitaciones	4

CAPÍTULO II. MARCO TEÓRICO.....	5
--	----------

2.1	Internet de las cosas	5
2.1.1	IoT protocolos de comunicación.....	5
2.1.2	Arquitectura general de IoT.....	6
2.1.2.1	Arquitecturas existentes de IoT y sus aplicaciones.	10
2.2	“M. health”	14
2.3	5G (fifth generation wireless technology).....	14
2.3.1	Arquitectura 5G	18
2.4	Bluetooth	20
2.5	Protocolo MQTT	22
2.6	VPS (Un servidor virtual privado).....	26
2.7	SSH	26
2.8	Certificado SSL	26
2.9	Protocolo FTP	26
2.10	Presentación de propuestas.....	27

CAPÍTULO III. MÉTODO	29
-----------------------------------	-----------

3.1	Sujetos o población de la investigación	29
3.2	Procedimiento.	30

3.3	Investigación	30
3.4	Análisis	30
3.5	Propuesta teórica	31
3.6	Propuesta práctica de desarrollo	31
3.7	Validación de resultados	31
3.8	Herramientas y Materiales	31
CAPÍTULO IV. DESARROLLO		32
4.1	Propuestas	32
4.2	Análisis de tecnologías.....	34
4.2.1	Bluetooth	34
4.2.2	MQTT	35
4.3	Topología de red y conexión.....	41
4.4	Funcionamiento de sistema	42
4.5	Configuración de desarrollador	43
4.6	Implementación y configuración del VPS.....	45
4.7	“HOSTING”	47
4.8	Registro de dominio.	48
4.9	Certificado de seguridad SSL.	50
4.10	Servidor FTP	51
4.11	Creación de página web.....	52
4.12.1	Página web principal	53
4.12.2	Sistema de registro de cuentas.....	54
4.12.3	“Dashboard”.....	56
4.13	Base de datos	58
4.14	MQTT EMQ	60
4.15	Integración de NODEJS	63
4.15.1	Instalación de NODEJS en VPS	63
4.15.2	Funcionamiento del programa	63
4.16	Firmware del ESP32	65
CAPÍTULO V. RESULTADOS.....		67
5.1	Página web principal	67
5.1.1	Control de usuarios	68
5.1.1.1	Funcionamiento “Login.php”	69
5.1.1.2	Funcionamiento “Register.php”	70

5.2	Conexión entre MQTT y Pagina Web.	71
5.3	Publicación de mensajes y suscripción MQTT entre EMQ y pagina Web.....	71
5.3.1	Suscripción	71
5.3.2	Publicación	74
5.4	ESP32	76
5.5	Base de datos	77
5.5.1	Página web.....	77
5.5.2	NODEJS	79
5.6	Consumo de recursos en la máquina del VPS.	80
5.7	Prueba de seguridad	81
CAPÍTULO VI. CONCLUSIÓN		82
BIBLIOGRAFÍA		84
APÉNDICE		87

LISTA DE FIGURAS

	Página
Figura 1. Requerimientos clave para diseños de IoT.....	7
Figura 2. Arquitectura general IoT, (A: Modelo tres capas) (B: Modelo cinco capas).....	8
Figura 3. Arquitectura IoT para “healthcare” [6].	10
Figura 4. Primera propuesta de Arquitectura IoT para “M. health” Wifi [10].	11
Figura 5. Segunda propuesta Arquitectura IoT para “M. health” Wifi [11].	11
Figura 6. Modelo de arquitectura basada en IoT para casas inteligentes [12].	12
Figura 7. Arquitectura IoT para conexión entre vehículos [13].	13
Figura 8. Generaciones existentes de red de tecnología celular.....	14
Figura 9. Arquitectura general 5G.	18
Figura 10. Representación de la arquitectura 5G virtualizada.	19
Figura 11. Arquitectura MQTT	23
Figura 12. Diagrama de flujo de la metodología utilizada.	30
Figura 13. Diagrama a bloques de las dos arquitecturas propuestas.	32
Figura 14. Ejemplo de arquitectura básica MQTT	35
Figura 15. Ejemplo de jerarquía de “topic”	36
Figura 16. Consumo de energía de los tres diferentes niveles QoS de MQTT.	38
Figura 17. Comparación de eficiencia y rendimiento entre MQTT y HTTP en mensajes recibidos y enviados.....	39
Figura 18. Topología de desarrollo de red para arquitectura propuesta 2.....	41
Figura 19. Diagrama de funcionamiento básico de sistema.....	42
Figura 20. Interfaz de configuración internet infinitum modo control de acceso.	43
Figura 21. Interfaz de configuración internet infinitum DHCP.	44
Figura 22. Ventana de símbolo del sistema Windows.	45

Figura 23. Interfaz PUTY estableciendo conexión con VPS.	47
Figura 24. Ventana principal del panel de control de VESTACP.....	48
Figura 25. Estatus del dominio creado con freenom.....	49
Figura 26. Certificado SSL	50
Figura 27. Paquete de herramientas instaladas en Atom.	51
Figura 28. Panel de configuración de FTP en VestaCP.....	51
Figura 29. Diagrama de la estructura de la página web.....	52
Figura 30. Página por “default” de entrada al dominio sin diseño gráfico.....	53
Figura 31. Página web con modificación grafica.....	53
Figura 32. Interfaz de registro “Sign up”.	54
Figura 33. Interfaz de acceso “Login”.	55
Figura 34. Interfaz “Dashboard” del usuario	56
Figura 35. Agregación de dispositivos del usuario.....	57
Figura 36. Eliminación de dispositivos del usuario.....	57
Figura 37. Base de datos “PhpMyadmin”.	58
Figura 38. Configuración Heidi para trabajar con nuestra base de datos.....	58
Figura 39. Esquema de base de datos SQL.....	59
Figura 40. Dentro de la configuración de base de datos.....	60
Figura 41. “Log in” de administrador en EMQ.....	61
Figura 42. Panel de control EMQ de usuario	62
Figura 43. Consola de instalación de “nodejs”	63
Figura 44. Diagrama de funcionamiento del programa “nodejs”.	64
Figura 45. Herramienta VSCODE con PLATFORMIO integrado.	65
Figura 46. Esquema de funcionamiento “Firmware” ESP32.	66
Figura 47. Propagación y alcance del DNS.	68

Figura 48. Acceso denegado a usuario en “Log In” .	69
Figura 49. Errores al crear cuenta de usuario en “register.php”	70
Figura 50. Consola del navegador web Opera.	71
Figura 51. “Dashboard.php” modo suscripción editado en Atom.	72
Figura 52. Valores configurados de prueba en EMQ.	72
Figura 53. Recibimiento de valores en “dashboard” de página web.	73
Figura 54. Respuesta de consola ante los mensajes recibidos.	73
Figura 55. “Dashboard.php” modo publicación editado en Atom.	74
Figura 56. “Dashboard” accionando los dos interruptores.	74
Figura 57. Respuesta de consola ante los mensajes publicados.	75
Figura 58. EMQ “dashboard” en modo suscripción.	75
Figura 59. Consola de monitoreo serial hacia ESP32.	76
Figura 60. Implementación Esp32 con DHT11.	77
Figura 61. Datos generados de las cuentas registradas en “register.php”	78
Figura 62. Datos generados de “devices.php”	78
Figura 63. Datos generados por el ESP32.	79
Figura 64. Monitoreo de recursos del VPS.	80
Figura 65. Prueba de seguridad en Nessus.	81
Figura 66. Diagnóstico de vulnerabilidad en seguridad SSL.	81

LISTA DE TABLAS

	Página
Tabla I. Brokers MQTT y sus características y limitaciones.	25
Tabla II. Especificaciones transmisión de Bluetooth 5 vs bluetooth 4.2 [29].	34
Tabla III. MQTT VS HTTP	37
Tabla IV. Resultados experimentales obtenidos de los niveles de calidad MQTT.	38
Tabla V. Reglas de entrada AWS.....	46

RESUMEN

El paradigma de las nuevas tecnologías se aproxima y el mundo se está preparando para las nuevas revoluciones que estas traen de la mano. El Internet de las cosas es una tecnología prometedora que tiende a revolucionar y conectar el mundo global a través de dispositivos inteligentes heterogéneos a través de una conectividad perfecta. Sin embargo, la nueva tecnología móvil 5G está también llena de retos sin solución, la cual pretende innovar a gran escala las actuales condiciones con las que nos conectamos.

En este trabajo se propone una solución de adaptación entre estos dos conceptos para enfocarlo en aplicaciones de “Mobile Health” de la mejor manera posible. Se realiza una arquitectura funcional con los mejores recursos y herramientas existentes en la actualidad, la cual tiene como meta lograr los retos planteados al inicio de la investigación.

Para el desarrollo de esta arquitectura se estudiaron y analizaron las diferentes tecnologías que cuentan con potencial de aplicación en este tema, se debatieron y se compararon para la toma de decisiones, la cual fue la base de este proyecto. A partir de la arquitectura obtenida se propuso una manera adecuada de hacer internet de las cosas para poner a prueba la misma. Se desarrolló un entorno web pensando en la escalabilidad en beneficio del usuario, facilitándole el desarrollo del sistema a implementar. Se usaron servicios gratuitos y de código abierto brindado por diferentes compañías, tales como hosting, bróker MQTT, DNS y plantillas prefabricadas para la página web. La integración de estos servicios en conjunto hace lograr el alcance de este proyecto que es la implementación de la arquitectura.

Para evaluar el desempeño de la arquitectura desarrollada, se hicieron pruebas en tiempo real haciendo uso del bróker para la publicación y suscripción de mensajes usando diferentes simulando diferentes parámetros o variables a través de la página web. Se confirmaron diferentes métodos de seguridad para brindarle la estabilidad al sistema en general.

Los resultados demostraron una correcta coordinación del sistema en conjunto. La base de datos registró las acciones de la página web, al mismo tiempo que se visualizaban los parámetros a medir en la interfaz creada, así como también la publicación o envío de mensajes desde la interfaz hacia el panel de control del bróker corroborando la

comunicación bidireccional que debe existir en un esquema de internet de las cosas. Se demostró la rapidez de la comunicación del protocolo MQTT, la eficiencia en cuanto a consumo de recursos y la capacidad en temas de seguridad.

CAPÍTULO I. INTRODUCCIÓN

En este capítulo se presenta la propuesta de investigación. Inicialmente se describen los antecedentes de la investigación, puntualizando el contexto donde se presenta el problema; se define claramente el problema a afrontar, así como el objetivo propuesto. Posteriormente, se justifica el desarrollo del proyecto, indicando los beneficios del mismo. Por último, se establece el alcance y las limitantes de la investigación.

1.1 Antecedentes

A medida que los avances tecnológicos se presentan en todas las áreas de la ingeniería, se presentan nuevos retos y problemas al mismo tiempo. Gracias al internet se ha hecho posible la conexión de millones de personas que ya no se pueden hacer a la idea de un mundo sin él. El internet de las personas cambio al mundo, pero se emerge también otro internet que no es solo acerca de conectar personas sino cosas, de aquí surge IoT.

Dado que el concepto IoT crece en popularidad en poco tiempo, también crecen requisitos de tecnología para manejar grandes cantidades de datos transmitidos para tener una eficiencia en una amplia banda ancha en un futuro no lejano, en otras palabras, la próxima

generación de dispositivos conectados a IoT. Algunos retos prioritarios son; mayor capacidad, mejorar la velocidad de datos y disminución de la latencia. El desarrollo de la próxima generación de comunicación móvil indiscutible es la quinta generación (5G) que promete satisfacer las necesidades de arquitecturas IoT complejas [1] .

Internet de las cosas (IoT) es un elemento crítico de la Industria 4.0. Debido a las graduales exigencias de los clientes, la arquitectura presente de IoT no será confiable y receptiva para aplicaciones IoT de próxima generación y futuros servicios [2].

La nueva generación de redes de telecomunicaciones (quinta generación o 5G) ha empezado en el mercado a finales de 2018 y continuará expandiéndose globalmente, independientemente de las innovaciones en velocidad, se especula que la tecnología 5G desarrolle un masivo ecosistema de IoT donde las redes pueden cumplir con las necesidades de comunicación y conexión de cantidades masivas de dispositivos conectados, con las compensaciones correctas entre velocidad, latencia y costo.

El Internet de las cosas (IoT) se ha posicionado como el medio para la conectividad global entre cualquier dispositivo, con 20,400 millones de dispositivos conectados estimados en todo el mundo para 2020. El IoT depende en protocolos de Internet, como las comunicaciones máquina a máquina (M2M), para conseguir la interoperabilidad y atender las necesidades de dominios como la salud electrónica(“e-health”), las redes o ciudades inteligentes. En el caso particular de la salud electrónica, los pacientes pueden ser monitoreados sensores en red que reúnen datos personales y envían a centros médicos o de procesamiento para rastrear afecciones crónicas o por razones profilácticas. La continua monitorización constante del paciente para una respuesta rápida también es común en las unidades de cuidados intensivos [3].

1.2 Planteamiento del problema

Actualmente debido a que tanto el concepto de IoT y 5G son relativamente nuevos, no se tiene definido una arquitectura fija o no se ha acordado o creado como estándar en aplicaciones de salud, solo se tienen pocas propuestas tanto como en esta aplicación como en muchas otras.

Al entrar en juego la tecnología en comunicaciones 5G se tiene el reto de adaptar una arquitectura que interactúe óptimamente con IoT en este sector de aplicación.

Mediante esta problemática y al hacer mención de las evidencias, se tienen las siguientes incertidumbres; ¿cómo podemos lograr una arquitectura optima de IoT sin tener una referencia solida o una base estandarizada aplicada y enfocada a M. Health? ¿cómo podemos coordinar esta arquitectura para trabajar con la ya próxima tecnología 5G?

1.3 Objetivo

Proponer una arquitectura IoT que sea aplicable a 5G para aplicaciones de “M.Health”. Debido a la inexistencia, así como desarrollar parte de la arquitectura con herramientas y servicios existentes de manera profesional y que sea “Open source”. Con el fin de adaptarse a las demandas de las nuevas revoluciones y avances que se tienen en la tecnología con propósitos médicos.

1.4 Justificación

El crecimiento de la población lleva de la mano al crecimiento en número de dispositivos tecnológicos consumidos, por consecuencia, la saturación de las redes contemporáneas donde la tecnología 5G llega a proporcionar un medio mucho más amplio. Así como también al no tener un entorno adecuado para el manejo de esta información se daría un desaprovechamiento de datos que pueden ser usados para diversas cosas, como, por ejemplo, prevenir tragedias médicas y aumentar la calidad de vida de la moderna humanidad. Al tener estos datos de diferentes parámetros fisiológicos, se puede tener una inmediata respuesta mejorando la eficiencia en centros médicos u hospitales al facilitarle la información de trazabilidad del paciente ya sea que este haya contado con un dispositivo de monitoreo móvil. Este trabajo les sirve a empresas desarrolladoras de tecnología o que brindan servicios de la misma para dar pie a nuevos conceptos de aplicación.

De manera que se fundamentan las bases teóricas sobre el objetivo de este proyecto, puede retomarse para su continuidad y así lograr un producto o servicio comercial.

1.5 Delimitaciones

Debido a la falta de equipo para hacer pruebas físicas este proyecto de investigación se busca hacer una propuesta teórica y diversas simulaciones en software para probar resultados experimentales.

1.6 Limitaciones

La exigencia de recursos de la simulación a través de las distintas herramientas de software determinará la demanda de hardware que el equipo de cómputo podrá proporcionar. Se dependerán de estos recursos para llevar a cabo las simulaciones que satisfagan a los requerimientos del tamaño de simulación. El tiempo y la desinformación en algunos ámbitos también es una limitación importante puesto que los conceptos de interés abarcan muchos temas.

CAPÍTULO II. MARCO TEÓRICO

En esta sección se describen los conceptos teóricos que sustentan el presente trabajo; Primeramente, se hace una descripción de los conceptos acerca de IoT, haciendo hincapié en las arquitecturas y sus diferentes aplicaciones con el propósito de hacer comparación. Posteriormente se describe la tecnología 5G.

2.1 Internet de las cosas

El principal objetivo del internet de las cosas es conectar lo que no está conectado, refiriendo a que los objetos que no están conectados a una red de computadoras llamada internet, será conectada para que puedan comunicarse e interactuar con el usuario u otros objetos o dispositivos. Eso podrá permitir censar y controlar un mundo físico haciéndolos más inteligentes [4].

2.1.1 IoT protocolos de comunicación.

Estos protocolos son los más utilizados en la aplicación M. health.

Hyper Text Transfer Protocol (HTTP): es el protocolo de comunicación que permite la transferencia de información en la web. Este protocolo se encuentra dentro de la capa de aplicación.

Transmission Control Protocol (TCP) and the Internet Protocol (IP) (TCP/IP): es un apilamiento de protocolos donde se aprovecha lo mejor de los dos, el cual proporciona una transmisión fiable de paquetes de datos sobre la red [5].

Message Queue Telemetry Transport (MQTT): es un protocolo de conexión maquina a máquina (M2M) orientado a internet de las cosas, este protocolo va dirigido a dispositivos que tienen procesamiento y memoria muy limitados [6].

Protocolo de internet versión 6 (IPv6): La nueva versión de protocolo de internet (IP) se basado en IPv4. Se estableció para ampliar el rango de direcciones para otorgar a dispositivos [7].

2.1.2 Arquitectura general de IoT.

El objetivo de esta sección es sentar las bases en cuanto al ámbito internet de las cosas se orienta, para la estructuración de una arquitectura de IoT de nuestro interés.

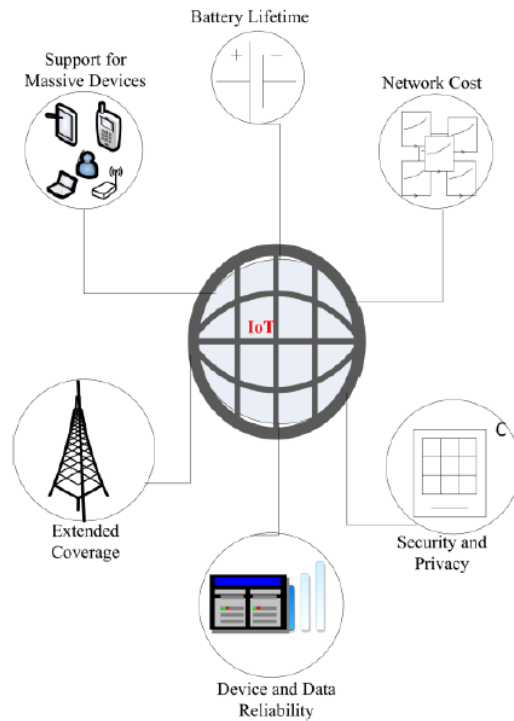


Figura 1. Requerimientos clave para diseños de IoT.

Como se aprecia en la Figura 1, son 6 factores que se tienen que tener a consideración en el desarrollo IoT como lo son: Seguridad y privacidad, el costo o presupuesto de la red, el tiempo de vida de las baterías en dispositivos, el soporte para masivas cantidades de dispositivos, una amplia cobertura, fiabilidad de datos y dispositivos.

Aunque todavía no existe una solución unificada para IoT en este momento, se han propuesto varias tecnologías de comunicación diferentes que están actualmente en funcionamiento y se han implementado en varios dispositivos en todo el mundo. Se utilizarán estándares de comunicación fijos y de corto alcance para la mayoría de las conexiones para lograr conectividad IoT masiva e IoT crítica a través de IoT celular tradicional o redes de área amplia de baja potencia (LPWAN) [8].

Las tecnologías LPWA son para aplicaciones IoT debido a sus características únicas que incluyen cobertura de área amplia, alta eficiencia energética, ancho de banda del canal, velocidad de datos y bajo consumo de energía.

No existe un consenso actual que determina una arquitectura IoT universalmente, las existentes han sido propuestas por diferentes investigadores. En la Figura 2 se ilustra las formas generales de una arquitectura mayormente usada la cual se integra por capas.

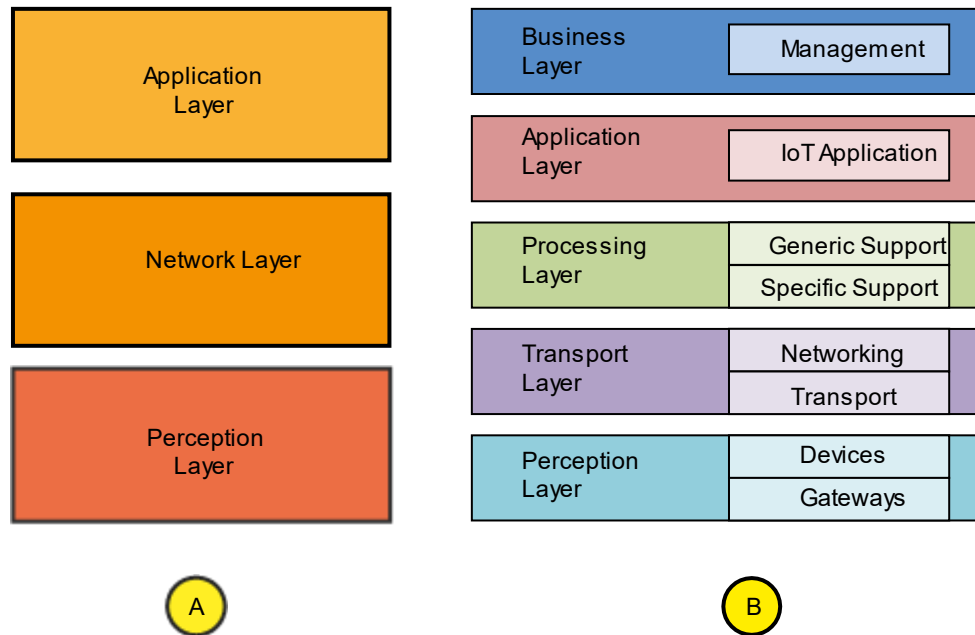


Figura 2. Arquitectura general IoT, (A: Modelo tres capas) (B: Modelo cinco capas)

Modelo de tres capas A). – Este modelo da la idea principal de Internet de las cosas, la conforman las capas de: percepción, red y aplicación.

- i) **Capa de percepción:** En esta capa representa las cosas físicas o dispositivos como sensores. La misma se encarga de recolectar y procesar información para la siguiente capa.
- ii) **Capa de red:** Es la responsable de conectar los dispositivos inteligentes, de red y servidores, así como también transmitir y procesar la información que proporcionan los sensores en la capa de percepción.
- iii) **Capa de aplicación:** Esta capa se encarga de brindar una interfaz mediante la cual los usuarios finales pueden comunicarse o relacionarse con un dispositivo. Define las aplicaciones que se le puede dar a IoT como, por ejemplo, casas inteligentes, salud inteligente, ciudades inteligentes, etc.

Modelo de cinco capas B). – Esta capa da una idea más detallada y profunda de internet de las cosas que está conformada por las capas de: percepción, transporte, procesado, aplicación y negocios.

- i) Capa de Transporte: Esta capa van incluidas las tecnologías como Bluetooth, Zigbee, Wifi, 3G etc. Las cuales son usadas para transferir información de manera segura provenientes de los sensores de la capa de percepción hacia la siguiente capa de procesado.
- ii) Capa de procesado: También conocida como Middleware, se encarga de procesar los datos recibidos de la capa de transporte, se toman decisiones y se entregan servicios a los usuarios que lo soliciten a través de los protocolos de red. Es responsable del servicio de administración y recibe la información para almacenarla en la base de datos.
- iii) Capa de negocios: Gestiona todas las actividades y servicios de sistemas de internet de las cosas. Se responsabiliza de construir modelos de negocios, gráficas y diagramas basados recibidos en la capa de aplicación, también tiene la tarea de analizar, diseñar, evaluar y monitorear las aplicaciones de internet de las cosas, en esta capa se pueden gestionar y monitorear las demás capas [9].

2.1.2.1 Arquitecturas existentes de IoT y sus aplicaciones.

El objetivo de esta sección de las diferentes arquitecturas en distintas aplicaciones es dar una referencia y una perspectiva amplia de lo que existe donde podemos a comparar a grandes rasgos hacia dónde va orientada la propuesta.

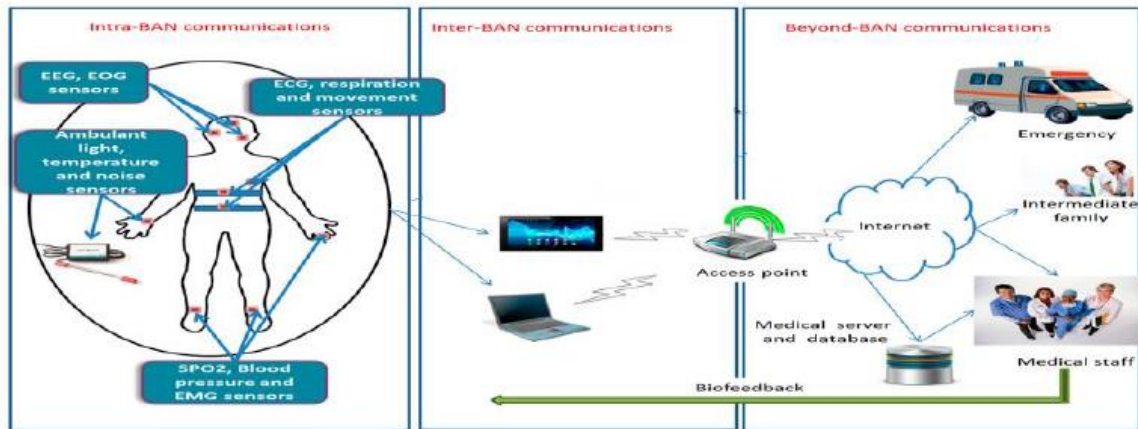


Figura 3. Arquitectura IoT para "healthcare" [6].

En la Figura 3 se muestra una arquitectura básica en 3 etapas la cual el paciente puede estar monitoreando diferentes parámetros médicos los cuales están conectados de manera alámbrica a un computadora la que procede a hacer la conexión a internet para así poder estar conectados con los que deben estar atentos a la información o a quienes estén cuidando del paciente para su eficiente intervención o cuidado.

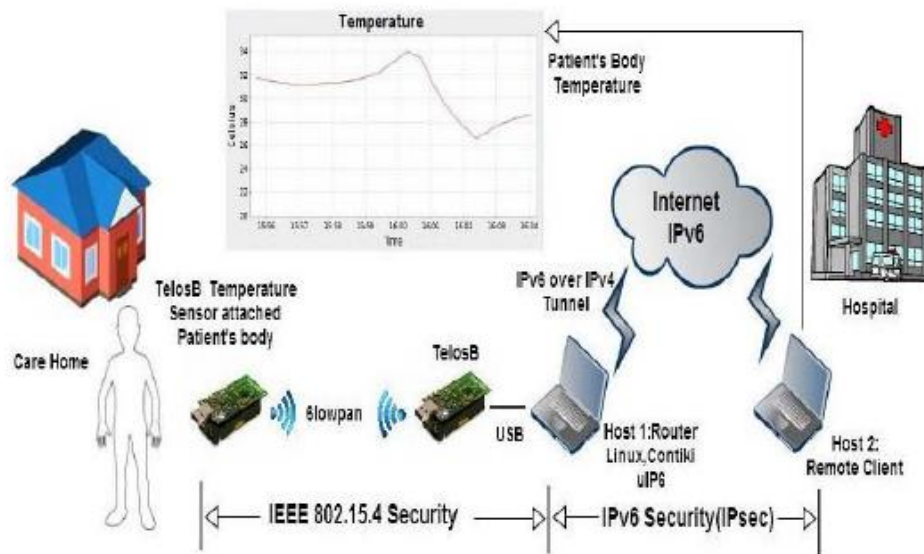


Figura 4. Primera propuesta de Arquitectura IoT para “M. health” Wifi [10].

En la Figura 4 podemos observar una arquitectura que va enfocada a personas que pueden estar siendo monitoreadas de manera “móvil” la cual implica que el sistema donde está conectado o conectados los sensores del mismo, está transmitiendo los datos de manera remota la cual viene siendo wifi en este caso.

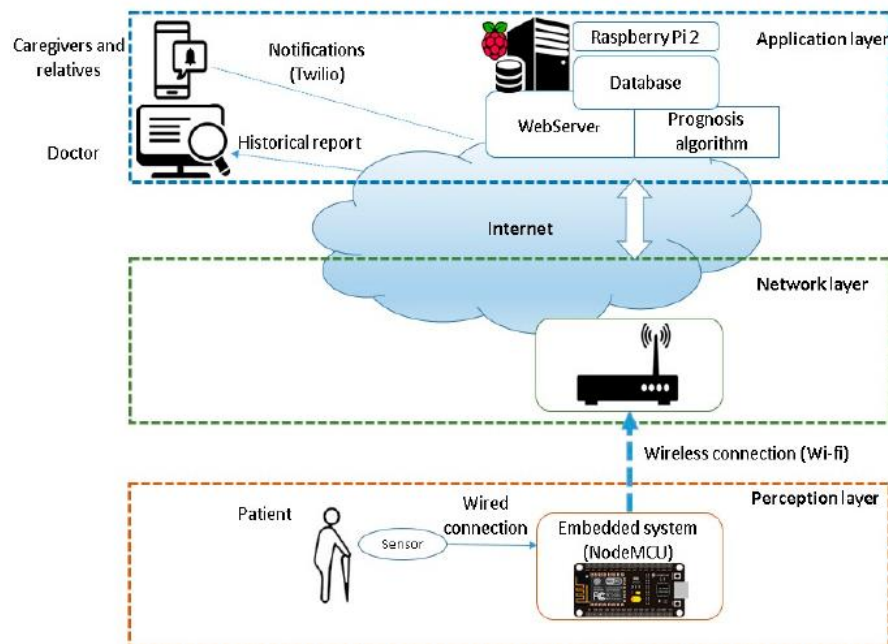


Figura 5. Segunda propuesta Arquitectura IoT para “M. health” Wifi [11].

La arquitectura de la Figura 5 es semejante a la Figura 4, la diferencia es que el sistema embebido que el paciente porta, establece comunicación directa con el router del proveedor de internet. También podemos observar que tiene su propio sistema de almacén de datos donde se involucra un servidor propio establecido con Raspberry Pi 2 donde se crea su propia base de datos y su propio algoritmo de pronóstico entre otras cosas.

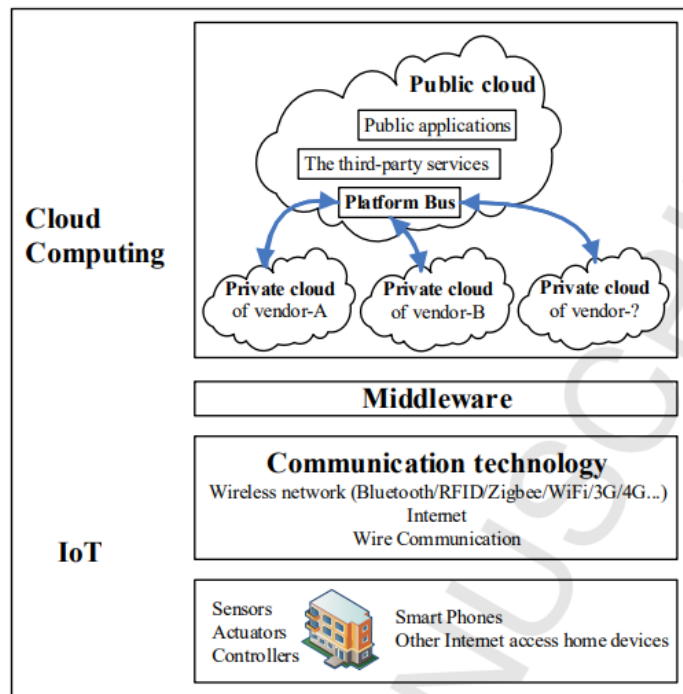


Figura 6. Modelo de arquitectura basada en IoT para casas inteligentes [12].

En la Figura 6 se aprecia una arquitectura para domótica o casas inteligentes donde podemos notar que se observa donde “Middleware” es intermediario entre la tecnología de comunicación y la nube publica donde operan directamente la información.

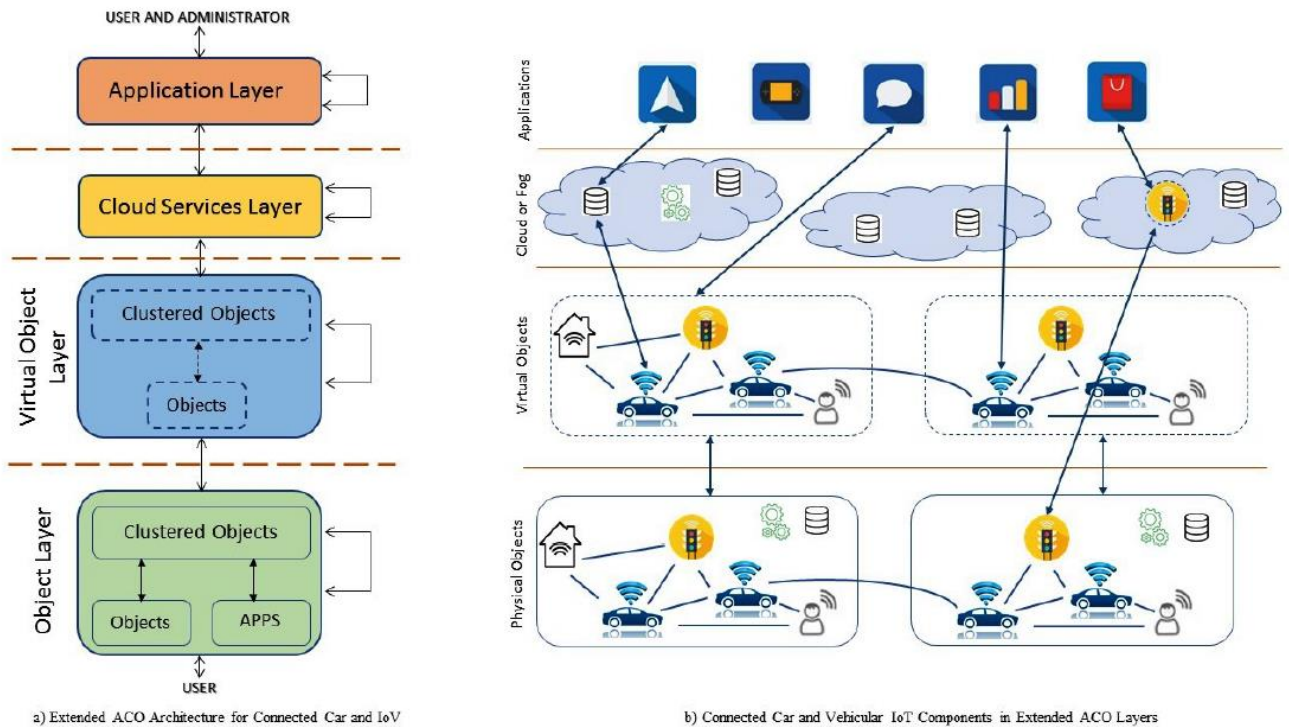


Figura 7. Arquitectura IoT para conexión entre vehículos [13].

En la Figura 7 se presenta una arquitectura IoT en un ecosistema de vehículos conectados o comunicados. Donde apreciamos cuatro capas donde se hace mención visual de cada etapa y sus características.

Se aclara que estas no son todas las aplicaciones existentes, cabe mencionar que se omitieron por ejemplo las aplicaciones donde WSN implica por la naturaleza del objetivo no comparten la misma estructura de la aplicación a analizar que es “M. health”, Sin embargo, se hizo referencia a arquitecturas distintas características a la misma donde se tiene un sistema embebido donde se hace conexión a la red por dispositivo independiente y no se cuenta con una red de sensores que se conectan a un solo dispositivo.

2.2 “M. health”

Se basa en el concepto de “movilidad y estar siempre conectado” se define como el computo móvil y todo lo que conlleva como censado y comunicación enfocado al cuidado de la salud [14].

2.3 5G (fifth generation wireless technology)

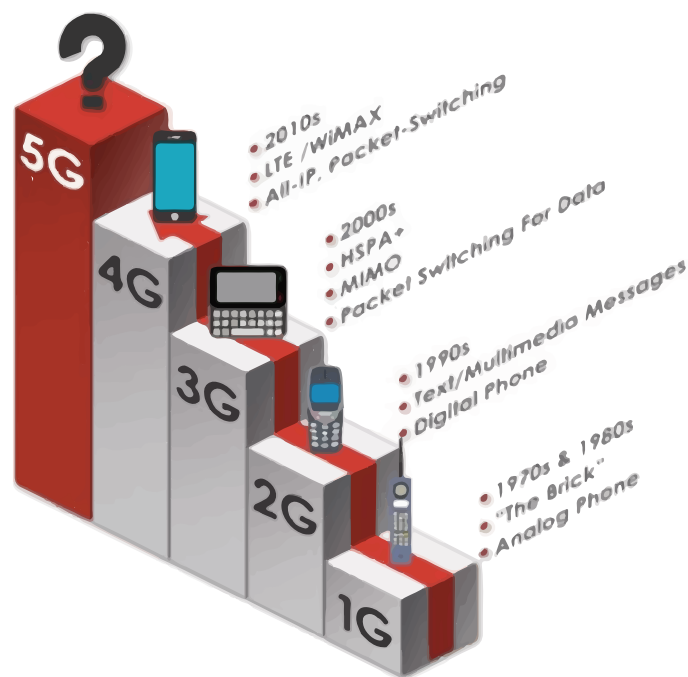


Figura 8. Generaciones existentes de red de tecnología celular.

La quinta generación de tecnología inalámbrica la cual viene a innovar a su antecesora 4G LTE, trae de la mano otros desarrollos de tecnología en cuanto a redes y telecomunicaciones. Este concepto será el futuro próximo de la siguiente etapa tecnológica y que abre la carta a y que les permite a otras tecnologías el desarrollo de servicios y tecnologías prometedoras a nivel mundial. La denominación de 5G se refiere a la quinta generación de redes móviles que conocemos. Atrás quedó la antigua red de 1G, la de aquellos primeros teléfonos móviles que solo permitían hablar. La tecnología 2G introdujo los SMS, y poco a poco nuestro 'smartphone' se convirtió en una herramienta de comunicación cada vez más amplia. Primero se incorporó la conexión a Internet (3G) y

después llegó la banda ancha (4G), lo que trajo consigo la reproducción de vídeos en tiempo real (streaming) o la realidad aumentada, algo a lo que ya estamos muy acostumbrados, pero que hace unos años eran completamente inviables [15].

La siguiente generación en redes móviles 5G está prevista para asegurar que los dispositivos en cantidades masivas y nuevos servicios como la banda ancha móvil mejorada (eMBB), masiva comunicación tipo-maquina (mMTC), comunicaciones críticas y operaciones de red sean eficientemente respaldadas. Se espera que los requisitos básicos tales como alto rendimiento, baja latencia en términos de entrega de datos, alta escalabilidad para permitir un número masivo de dispositivos, una técnica eficiente de consumo de energía y la provisión de una solución de conectividad ubicua para usuarios finales se apoyen eficientemente usando el 5G red móvil para el IoT. En consecuencia, considerando el mecanismo de seguridad de las redes celulares existentes que se basan en la protección de la conectividad básica y la privacidad de los usuarios finales, se espera que el sistema celular 5G garantice que se establezca un mecanismo de seguridad mejorado en toda la red para abordar problemas de autenticación, autorización, y contabilidad (AAA) para dispositivos IoT interconectados heterogéneos [8] [16].

Investigaciones ha demostrado que las futuras redes móviles 5G tienen que atender el despliegue masivo de IoT con miles de millones de objetos inteligentes y sensores conectados que serán una representación global del mundo real y para apoyar la provisión de casos de uso de IoT de misión crítica, que requieren respuestas en tiempo real y automatización de procesos dinámicos en diferentes campos de operaciones, incluyendo vehículo a infraestructura (V2I), movimiento de alta velocidad, vehículo a vehículo (V2V), así como un sistema de control de proceso [17].

Se espera que la nueva red de radio 5G que se está considerando actualmente se adapte a los casos de uso de IoT masivos y críticos, ya que la demanda de comunicaciones de la máquina continúa creciendo ampliamente para conectar una gran cantidad de dispositivos inteligentes con los beneficios del uso de redes celulares. A la luz de esto, actualmente se están introduciendo nuevas mejoras en los sistemas M2M y NB-IoT como se especifica en el 3GPP Release-14 actual para IoT celular, siendo la primera fase normativa para los estándares 5G. Actualmente, la estandarización 3GPP está trabajando para garantizar que se introduzcan mejoras adicionales de los KPI en las redes 4G existentes para garantizar

que la red móvil 5G esté diseñada desde cero para acomodar el creciente alcance de los casos de uso de IoT en el mercado y minimizar el costo de desarrollar nuevas redes.

Se espera que la nueva red de radio 5G que se está considerando actualmente se adapte a los casos de uso de IoT masivos y críticos, ya que la demanda de comunicaciones de la máquina continúa creciendo ampliamente para conectar una gran cantidad de dispositivos inteligentes con los beneficios del uso de redes celulares. A la luz de esto, actualmente se están introduciendo nuevas mejoras en los sistemas M2M y NB-IoT como se especifica en el 3GPP Release-14 actual para IoT celular, siendo la primera fase normativa para los estándares 5G. Actualmente, la estandarización 3GPP está trabajando para garantizar que se introduzcan mejoras adicionales de los KPI en las redes 4G existentes para garantizar que la red móvil 5G esté diseñada desde cero para acomodar el creciente alcance de los casos de uso de IoT en el mercado y minimizar el costo de desarrollar nuevas redes.

NGMN prevé una larga lista de tecnologías potenciales que puede cerrar la brecha a 5G. Algunos ejemplos son

- Acceso al espectro: uso flexible del espectro con licencia, espectro integrado sin licencia, uso de mayores bandas de frecuencia, modo dúplex;
- Enlace de radio: nuevas formas de onda, avanzado acceso a tecnologías múltiples, diseño de marcos de radio / numerología, masivo MIMO y esquemas mejorados de antenas múltiples, receptores avanzados, coordinación de interferencia, tecnologías para transmisión de paquetes pequeños, red centrada en UE;
- Capacidad de acceso por radio: densificación (células pequeñas / redes ultradensas), conectividad dual (capacidad / cobertura diseño de sistema dividido), coordinación mejorada de múltiples RAT, comunicaciones de dispositivo a dispositivo, backhauling inalámbrico (por ejemplo, auto-backhauling y retransmisión);
- Flexibilidad de red: redes definidas por software, red central móvil virtualizada, RAN virtualizada, división flexible de funciones entre los nodos de la red, declarada integrado nodo central, micro servidores;
- Uso eficiente / adaptable de los recursos de la red: optimización de tráfico, uso compartido de red mejorado para múltiples operadores, arquitectura de servicio escalable, big data, contextaware / red centrada en el usuario, optimización de

contenido y transmisión adaptativa, gestión inteligente heterogénea, medidas embebidas del desempeño de la red.

- Otros habilitadores: tecnologías para conectividad masiva, toda la red de transporte óptico, centrada en la información redes, redes de malla, fronthauling mejorado, privacidad y seguridad.

Si bien varias conferencias están ayudando a aumentar su visibilidad, se alienta a la industria a madurar tecnologías y abordar cualquier problema a tiempo para 5G. Un nuevo diseño del sistema se debe aprovechar al máximo los beneficios de estas tecnologías emergentes.

El sistema 5G consta de tres capas:

- Capa de recursos de infraestructura: los recursos físicos de una red convergente fijo-móvil, que comprende el acceso de nodos, nodos de nube (que pueden ser procesamiento o almacenamiento de recursos), dispositivos 5G, nodos de red y enlaces asociados. Los recursos están expuestos a las capas superiores y a la entidad de orquestación a través de los principios de virtualización.
- Capa de habilitación empresarial: biblioteca de todas las funciones requeridas dentro de una red convergente en forma de bloques de construcción de arquitectura modular, incluidas las funciones realizadas por módulos de software que se pueden recuperar del repositorio a la ubicación deseada, y un conjunto de parámetros de configuración para ciertas partes de la red, por ejemplo, acceso por radio. Esas funciones y capacidades son llamado a pedido de la entidad de orquestación, a través de API relevantes.
- Capa de aplicación empresarial: aplicaciones específicas y servicios del operador, empresa, verticales o terceras partes que utilizan la red 5G. Estas tres capas están articuladas por una entidad de orquestación, que juega un papel central en esta arquitectura. Tiene la capacidad para gestionar una red virtualizada de este tipo de extremo a extremo, además de la automatización tradicional de OSS y capacidades SON [18].

2.3.1 Arquitectura 5G

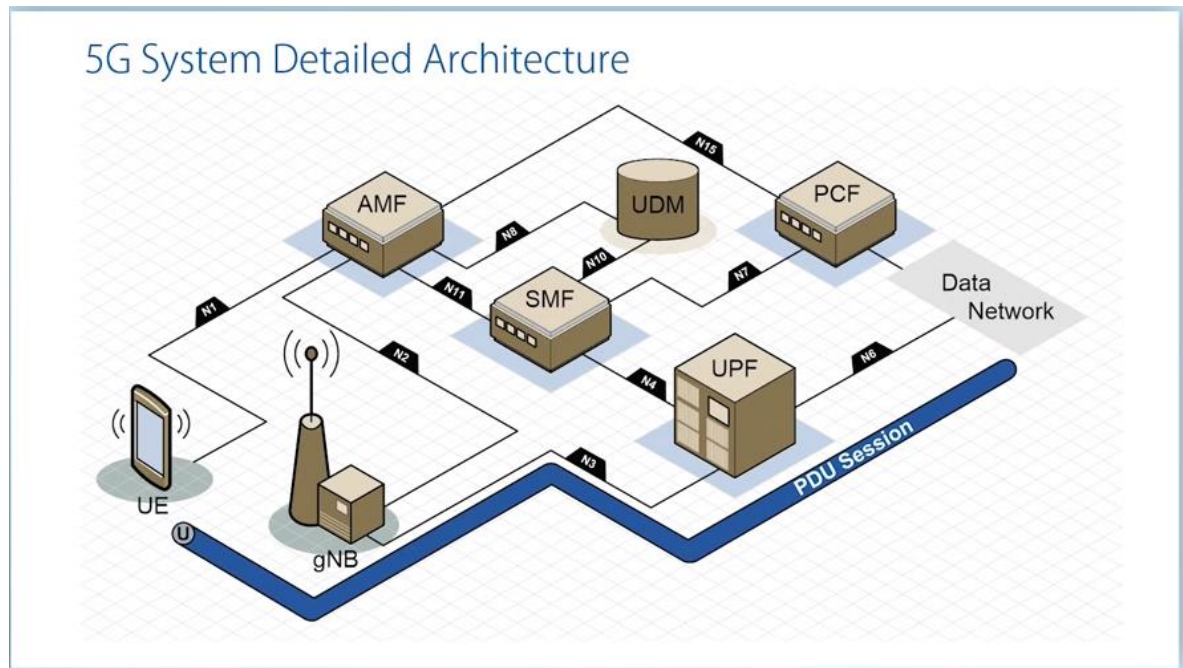


Figura 9. Arquitectura general 5G.

En la Figura 9 se muestra una arquitectura que compone 8 elementos.

PDU Session. Provee Qos Flow únicos para cada usuario otorgándoles un “ID”, es el que da las características específicas para cada usuario basado en latencia.

AMF (core access and mobility management function). Toma cuidado por la gestión de movilidad, así como también por la seguridad y registro. Trabaja con la base de datos de registro para ver si el usuario está permitido o no. Provee al dispositivo con una identidad temporal en cualquier señal de la red

SMF (session management function). Gestión de sesión. Se encarga del establecimiento y la modificación y arranque del “PDU session”. Como parte rutinariamente usa la política de control de función para determinar el sí o no de una sesión particular de datos es permitida. Se encarga también de seleccionar cual “UPF” por ende selecciona también el IP basado en asignar la dirección si es ipv4 o ipv6.

UPF (user plane function). Es un punto de anclaje para movilidad de NG-RAN, mientras me muevo alrededor de red de radio acceso, se está moviendo de una GnB a otra (antena), pero el UPF permanecerá en un punto de anclaje en la red para que el juego de la conectividad del usuario siempre estará corriendo de GnB a nuestro UPF en la red de automóviles. Se asegura que los datos correctos se envían al QoS correcto entre otras cosas.

UDM. (unified data management). Es un repositorio central de información de suscriptores directamente involucrada en la autorización de acceso porque cargara con llaves de seguridad, está involucrado en el registro y gestión de movilidad porque monitoreara donde está el suscriptor.

PCF (policy control function). Estas decisiones de política dinámica están basadas en condiciones las cuales serán consideradas por el bloque SMF y AMF.

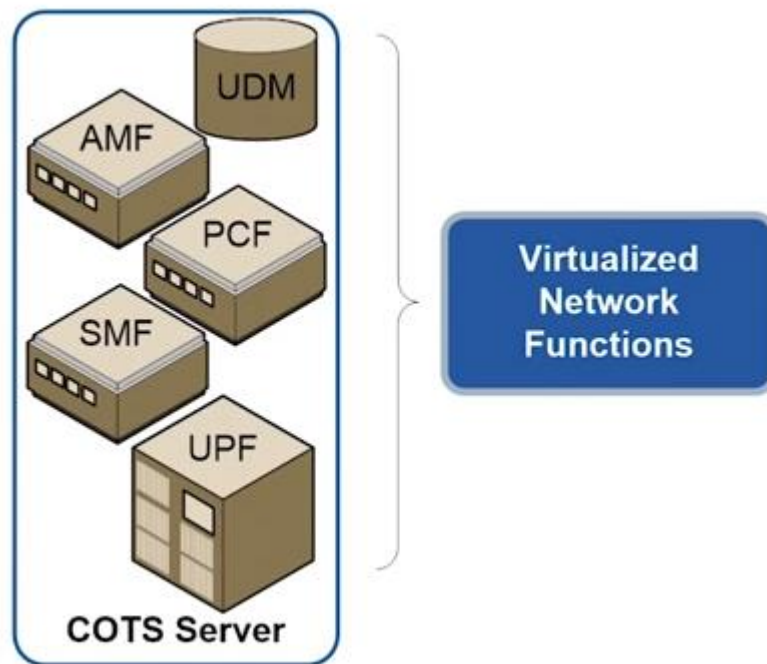


Figura 10. Representación de la arquitectura 5G virtualizada.

Estos procesos definidos previamente pueden ser virtualizados a manera que se pueden procesar en software lo cual resulta ser más óptimo en la utilización de recursos.

2.4 Bluetooth

Bluetooth es una tecnología inalámbrica de bajo consumo de energía la cual opera en la banda de radio ISM (porciones del espectro de radio con propósitos industriales, científicos y médicos) la cual se ha expandido a través del tiempo y ha proveído flexibilidad en rango, ancho de banda, los cuales son importantes para aplicaciones de IoT.

Bluetooth da la opción para operar en diferentes topologías de red como lo son punto a punto, estrella, malla y topologías de broadcast. Existen dos versiones en las que Bluetooth opera; LE y classic. La “LE” es conocida por su muy bajo consumo y se dispersa en celulares inteligentes, tabletas y computadoras entre otros dispositivos. Classic es la versión original de la radio Bluetooth y sigue siendo muy usada para aplicaciones de streaming de audio.

- Espectro de radio. - El espectro de radio en general se considera de 30 Hz hasta 300Ghz. Entre más corta la frecuencia mayor el rango. Sin embargo, Entre menor sea la frecuencia menor la tasa de datos que pueda soportar. Para seleccionar el espectro de radio se toma en cuenta la compensación entre rango y tasa de datos. Bluetooth usa 2.4 Ghz ISM banda de espectro (2400 a 2483.5 MHz). La cual habilita un buen balance entre rango y rendimiento. Para agregar, la banda 2.4 Ghz está disponible en todo el mundo haciéndolo un estándar de conectividad inalámbrica de bajo consumo.
- Capa física. – La PHY (capa física) de la tecnología inalámbrica define el esquema de modulación y otras técnicas que usan el envío de información a bandas de radio frecuencias (RF) específicas. Bluetooth ofrece múltiples opciones de PHY, cada una con diferentes características que determinan el rango efectivo y las velocidades de transmisión de datos
- Sensibilidad de recepción. – Es la medida mínima que puede ser percibida por el receptor o en la cual puede ser establecida una conexión. Se puede interpretar de igual manera de que tan bien se escucha el sonido más escaso. Bluetooth especifica que el dispositivo debe de ser capaz de lograr una mínima sensibilidad de recepción de -70dBm a -82dBm dependiendo del PHY usado.

- Poder de transmisión. – Entre mayor sea la potencia de transmisión, mayor la probabilidad de que se escuche a largas distancias, sin embargo, se compromete el consumo de energía. Bluetooth admite potencias de transmisión de -20 dBm (0.01 mW) a +20 dBm (100 mW).
- Ganancia de antena. – La antena cubre la transformación de energía en ondas electromagnéticas que viajan a través del aire. Se cuenta con variedad de opciones de antena. Los dispositivos que incorporan Bluetooth comúnmente logran una ganancia de antena en el rango de -10dBi a +10dBi.
- Atenuación. – La pérdida de poder de la señal transmitida que ocurre naturalmente al propagarse las ondas electromagnéticas por el aire. Puede darse por la intercepción o interferencias de algún material. Las cuales pueden ser humedad, paredes de vidrio, madera, metal o concreto, etc.

Bluetooth es administrado por Bluetooth “Special Interest Group” (SIG), que cuenta con más de 35,000 compañías miembros en las áreas de telecomunicaciones, computación, redes y electrónica de consumo. El IEEE estandarizó Bluetooth como IEEE 802.15.1.

Bluetooth 5 tiene como objetivo ofrecer un rendimiento significativo mejoras en comparación con las versiones anteriores de Bluetooth con respecto a la velocidad, el alcance y la capacidad de transmisión. En la feroz competencia para dominar el estándar de comunicación IoT, estas nuevas ventajas podrían ayudar a BLE a ser una de las mejores opciones para IoT. Bluetooth 5 atrajo rápidamente la atención de los inversores, especialmente las nuevas empresas y empresas de capital riesgo, que miran con interés al floreciente mercado de la IoT. Actualmente Es difícil predecir cuáles serán los estándares inalámbricos adoptados en IoT. De hecho, la dinámica y El mundo en evolución de las cosas inteligentes y conectadas es todavía en su infancia. Sin embargo, considerando las mejoras significativas en velocidad, consumo de energía, alcance y capacidad, parece que Bluetooth 5 es un candidato fuerte [19].

2.5 Protocolo MQTT

Desarrollado por la compañía de tecnología IBM y dado a conocer en 1999. MQTT (MQ Telemetry Transport o Message Queuing Telemetry Transport) es un protocolo estandarizado abiertamente por OASIS e ISO ESTÁNDAR (ISO/IEC 20922) de publicación/suscripción. Fue pensado para tener una comunicación eficiente y precisa de baja demora en la red con corto ancho de banda.

Este protocolo corre bajo TCP/IP o sobre otros protocolos de red con soporte bidireccional y sin pérdidas de datos, se define por tener dos entidades de red: un “bróker de mensaje” y sus clientes. El bróker se encarga de recibir los mensajes de los clientes y direcciona la ruta de los mensajes. Cada vez que un mensaje es publicado será recibido por el resto de dispositivos adheridos a un tópico del protocolo.

Sus principales características se basan en las tres calidades del servicio que brinda:

A lo más una vez (Qos 0). - Es básicamente el protocolo TCP, pero con pérdida de mensajes. Este nivel podría ser usado, por ejemplo, para aplicaciones de sensado, donde la pérdida de una lectura no es crítica ya que, tarde o temprano, nos llegará otra.

Al menos una vez (Qos 1). – Dónde la llegada de los mensajes está asegurada, pero considerando duplicidades.

Exactamente una vez (Qos 2). - Dónde la llegada de los mensajes está asegurada exactamente una vez. Este nivel puede ser usado, por ejemplo, con sistemas de pago donde la duplicidad o la pérdida de mensajes es más crítica y podría dar lugar a cargos incorrectos.

El protocolo MQTT es idóneo para aplicaciones de Internet de las Cosas en las cuales se envían cantidades pequeñas de información y por tanto no se necesita un gran ancho de banda. TST implementa MQTT de forma nativa en sus dispositivos, haciendo los datos disponibles en un MQTT broker, que no es más que un servidor de traducción y almacenamiento de datos.

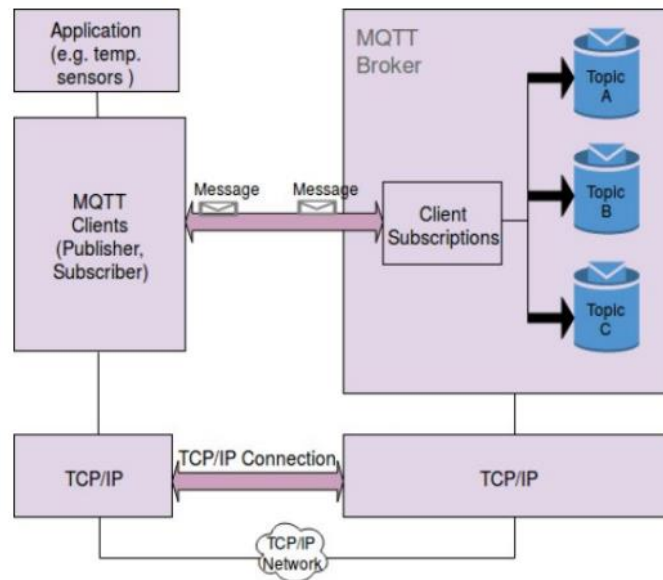


Figura 11. Arquitectura MQTT

En la Figura 11 se muestra la arquitectura general de MQTT se puede dividir en dos partes principales.

1) Cliente: El cliente puede ser un editor o suscriptor y siempre establece la conexión de red al servidor (Corredor). Puede hacer lo siguiente

- Publica mensajes para los usuarios interesados.
- Suscribe en un tema interesado para recibir mensajes.
- Cancela la suscripción para extraer de los tópicos suscritos.
- Desconectarse del broker.

2) Broker: El broker controla la distribución de información, es el principal responsable de recibir todos los mensajes de publicador, filtrándolos, decide quién está interesado en él y luego envía los mensajes a todos los clientes suscritos. Desempeña las siguientes labores:

- Aceptar solicitudes del cliente.
- Recibe mensajes publicados por los usuarios.
- Procesa diferentes solicitudes como Suscribirse y darse de baja de los usuarios.
- Después de recibir mensajes del editor, los envía a los usuarios interesados.

¿Dónde usar MQTT? El entorno restringido de los dispositivos embebidos como la limitada capacidad de procesamiento o dispositivos que tienen una conexión inestable a la red, mayormente son candidatos para el protocolo MQTT. Algunas de sus aplicaciones son: energía y servicios públicos, redes sociales y lo más importante de este proyecto es el impacto que puede tomar en el tema del cuidado de la salud. “**Healthcare**” que es un concepto directamente relacionado con “**M. Health**”, al usar MQTT, las asociaciones que se dedican al cuidado de la salud necesitan crear una solución de control flexible. Estas serían diferentes puntos que se esperan acordar para la atención de las víctimas.

- Mantener el monitoreo de víctimas aun estando fuera de la clínica.
- Mejorando y actualizando las pruebas subsecuentes.
- Lograr avances en la industria de la información y sus principios de colección de lo mismo.

La organización trabajó con IBM para dar una respuesta a cuál cliente MQTT es implementado para una máquina de observación en casa que reúne diagnósticos en cualquier punto que la víctima está cerca de un sistema base. Luego reenvía la información indicativa a través de la web al dominio principal, que es dado a una aplicación que analiza las medidas y da a conocer al equipo de atención médica si hay indicios de que la víctima tal vez este teniendo complicaciones. Ahorra dinero en efectivo para la asociación, así como también a sus víctimas, ya que existe un requisito limitado para que las víctimas vayan hospital para chequeos regulares si están bien.

Brokers de MQTT. El broker es el corazón de cada arreglo MQTT. Proporciona un enlace de conexión entre aplicaciones o dispositivos físicos y sistemas empresariales. Los brokers están a cargo de la suscripción, determinan las sesiones, los mensajes perdidos y la seguridad en general, incluyendo la autenticación y autorización.

Tabla I. Brokers MQTT y sus características y limitaciones.

Broker	Descripción	Características	Limitaciones
Mosquitto	Soporta la Versión MQTT 3.1 y es un fuente abierta.	<ul style="list-style-type: none"> - Todas las QoS - Autenticación - Puente - Tópicos dinámico - Sockets web 	<ul style="list-style-type: none"> - Agrupación - Menos configuración, No permite conexión simultánea con el uso de autenticación
RSMB (Really small message broker)	Es un pequeño broker que soporta V3 y V3.1.	<ul style="list-style-type: none"> - Todas las QoS - Puente - Tópicos dinámico 	<ul style="list-style-type: none"> -Seguridad - Agrupación -Sockets web
MQTT.js	Broker entre cliente/servidor producido por una API en JavaScript	<ul style="list-style-type: none"> - Todas las QoS - Puente - Tópicos dinámico - Sockets web -SSL 	<ul style="list-style-type: none"> -Puente -Autenticación -Agrupación
HiveMQ	La empodera una organización que adjunta todos los dispositivos y servicios con un esfuerzo nominal por persecución de fabrica	<ul style="list-style-type: none"> - Todas las QoS - Autenticación - Puente - Tópicos dinámico - Sockets web -TLS/SSL - Agrupación 	<ul style="list-style-type: none"> - Fuente abierta -Degradación de desempeño a causa de TLS
VerneMQ	VerneMQ podría ser superior, bróker de mensaje MQTT distribuido. Soporta MQTT versión 3.1 y 3.1.1	<ul style="list-style-type: none"> - Todas las QoS - Autenticación - Puente - Tópicos dinámico - Sockets web -Encriptación 	<ul style="list-style-type: none"> - Degradación de desempeño a causa de TLS

Se supone que IoT proporcionara conectividad avanzada de servicios, dispositivos y sistemas que van más allá de las asociaciones máquina-máquina (M2M) e incluye una gama de aplicaciones, protocolos y dominios. La simplicidad de MQTT y el código fuente abierto hace que este protocolo sea adecuado para restringidos entornos como IoT el cual tiene baja potencia, limitada capacidad de cálculo y memoria, y ancho de banda limitado [20] [21].

2.6 VPS (Un servidor virtual privado)

Es una partición virtual dentro de un servidor físico que le asigna recursos exclusivos a cada partición. Éstas otorgan acceso raíz que permite instalar un sistema operativo y trabajar con alto grado de libertad. El hosting VPS es uno de los servicios de alojamiento más populares que puedes elegir para tu sitio web. Utiliza tecnología de virtualización para proporcionarte recursos dedicados (privados) en un servidor con múltiples usuarios [22].

2.7 SSH

SSH o Secure Shell, es un protocolo de administración remota que le permite a los usuarios controlar y modificar sus servidores remotos a través de Internet a través de un mecanismo de autenticación. Proporciona un mecanismo para autenticar un usuario remoto, transferir entradas desde el cliente al host y retransmitir la salida de vuelta al cliente. El servicio se creó como un reemplazo seguro para el Telnet sin cifrar y utiliza técnicas criptográficas para garantizar que todas las comunicaciones hacia y desde el servidor remoto sucedan de manera encriptada [23].

2.8 Certificado SSL

Es cifrar las comunicaciones mediante el uso de criptografía en diversos servicios online, como el correo electrónico o la web. Constituye un estándar de Internet, elaborado, mantenido y reconocido por los organismos de dirección técnica de la red de redes, con la cual cosa es universal, independiente de fabricante y cuyo uso es facilitado a cualquier desarrollador de soluciones que trabaje creando software y servicios en Internet [24].

2.9 Protocolo FTP

El Protocolo de transferencia de archivos (en inglés File Transfer Protocol o FTP) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un

equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo [25].

2.10 Presentación de propuestas.

IoT es un cambio de paradigma para hacer frente a la movilidad y conectividad humana. Se tiene un reto en investigaciones y desafíos tecnológicos que deben cumplirse con soluciones de ingeniería viables. A continuación, destacamos algunos de los desafíos técnicos más importantes del IoT futuro:

- Escalabilidad: la gran cantidad de dispositivos y usuarios y las interacciones requeridas entre ellos hacen que la escalabilidad sea un gran desafío.
- Interoperabilidad: la heterogeneidad de habilitar dispositivos y plataformas hacen de la interoperabilidad otro desafío clave en el IoT.
- Eficiencia en las comunicaciones: tener un IoT viable, sensores de baja potencia, transceptores inalámbricos, comunicaciones y redes para M2M será crucial para comunicaciones eficientes.
- Seguridad y privacidad: enormes volúmenes de datos. emergiendo del mundo físico, M2M, y nuevas comunicaciones en el futuro IoT implica otro gran desafío, que implica extraer los datos, proporcionando seguridad acceso y preservación de la privacidad de los datos.
- Puntualidad y frescura de los datos: garantizar la puntualidad y la frescura de los datos es otro desafío importante que debe cumplirse.
- Movilidad, acceso y continuidad del servicio: La ubicuidad de IoT trae consigo el desafío de abordar la movilidad, el acceso ad hoc, y continuidad del servicio.
- Nomenclatura práctica, resolución y descubrimiento: El acceso global y el descubrimiento previstos en IoT también requiere soluciones prácticas de nombres, resolución y descubrimiento.

Anteriormente se mencionó las consideraciones que una arquitectura genérica debe de tener en la Figura 1. Requerimientos clave para diseños de IoT. Todas estas consideraciones tienen fundamento en lo que demanda la tecnología en desarrollo 5G [26].

Se plantean dos propuestas de arquitecturas IoT para la integración 5G con propósitos de aplicación "M. Health", con las consideraciones estudiadas en el la sección, se tiene dos conceptos con un inicio distinto, pero siguiendo el mismo camino. Presentada de manera general donde se demuestran las etapas que se cubren para lograr su objetivo.

CAPÍTULO III. MÉTODO

En esta sección se presenta la metodología aplicada para desarrollar una propuesta de arquitectura IoT para 5G. Se presentan los sujetos de estudio, se describe de manera general el procedimiento utilizado, así como los materiales y herramientas empleadas.

3.1 Sujetos o población de la investigación

El presente trabajo se llevó a cabo por el alumno Francisco Antonio Palos Angulo en las instalaciones internas del Instituto Tecnológico de Sonora, Campus Náinari. En el periodo enero de 2020 a mayo del 2020. Asesorada por el Dr. Armando García Berumen.

3.2 Procedimiento.

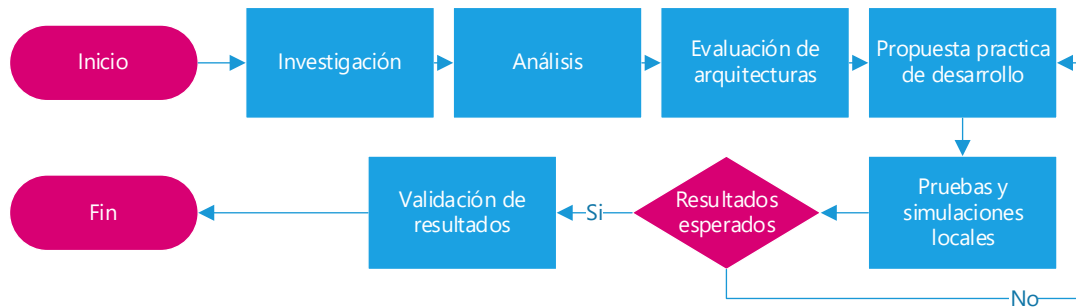


Figura 12. Diagrama de flujo de la metodología utilizada.

3.3 Investigación

Esta etapa consistió en la indagación, ya que los argumentos de este tema fueron bajo investigación, fue necesaria la recopilación información y/o datos, se hicieron de fuentes de confianza considerable, siendo libros u artículos oficiales sobre el tema a investigar [27] [28].

3.4 Análisis

La segunda etapa se basó en analizar las propuestas y evidencias de diferentes artículos sobre avances y rendimiento que las tecnologías desempeñan, a través de la comparación de las bases que brindan las fuentes. Ya que en la actualidad no se tiene una forma oficial de hacer las cosas o protocolos dependiendo de la aplicación, cada empresa maneja a su mejor postor, habiendo infinidad de aplicaciones, se pronostica que habrá diferentes propuestas sobre arquitecturas y métodos a segur. Habiendo muchas tecnologías propuestas presentadas, se retomarán las más adecuadas al tópico y se debatirá sobre cuál es la mejor como resultado de esa comparación presentadas con evidencia y estudios.

3.5 Propuesta teórica

Primeramente, se trató de analizar las tecnologías en hardware, software y protocolos que se viene con la tecnología 5G. Una vez que se realiza una comparación completa y obtenida una retroalimentación de las diferentes tecnologías y aplicaciones, se trató de dar la mejor propuesta fundamentada en estudios y resultados para lograr una arquitectura de IoT para 5G enfocado aplicaciones médicas.

3.6 Propuesta práctica de desarrollo

En base a lo propuesto teóricamente, se le dio prioridad para trabajar en el prototipo a las herramientas que se cuentan en la actualidad para desarrollar un sistema funcional sin perder el enfoque hacia la adaptación futura en cuanto a 5G. En base a esta propuesta se hizo el prototipo sobre el cual se trabajó el proyecto.

3.7 Validación de resultados

Teniendo una simulación exitosa se puede dar un seguimiento a la hipótesis o propuesta que se obtiene. Se trató de cubrir todo el proceso que demande la arquitectura en cuanto su propuesta en un tiempo considerable de operación para así poder dar un análisis que explote los puntos de este tema.

3.8 Herramientas y Materiales

Se utilizaron diferentes herramientas de edición y desarrollo de software de análisis y para adecuar una arquitectura estructurada en una computadora de escritorio con procesador Intel Core i5 (tercera generación), 8gb de RAM, disco duro SSD de 256 Gb con sistema operativo instalado Windows 10 64 bits. Se adquirió el dispositivo ESP32 junto con el sensor DHT11 los cuales se trabajaron en una placa de prototipado (protoboard).

CAPÍTULO IV. DESARROLLO

Para el desarrollo de la arquitectura IoT en aplicaciones médicas para 5G, se presenta la descripción de dos posibles arquitecturas propuestas óptimas para los requisitos de IoT modernos contemplando la integración de la tecnología 5G.

4.1 Propuestas

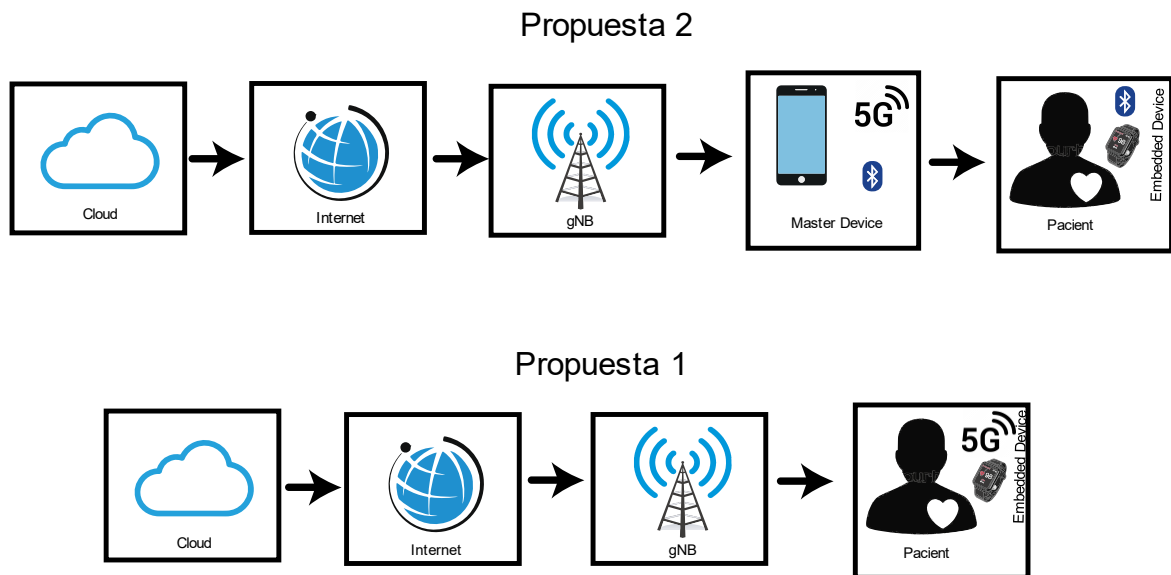


Figura 13. Diagrama a bloques de las dos arquitecturas propuestas.

Propuesta 1: En la Figura 13 se presenta el esquema. Se propone un dispositivo embebido que el paciente portara para la medición del parámetro de interés médico por lo que este hará el censado correspondiente, contara con la capacidad de estar comunicado con la red 5G, la cual debe brindarse por alguna compañía a través de una tarjeta SIM. Con esto se logra una conexión a internet la cual podrá facilitar el acceso a la publicación de datos a un servidor o nube.

Propuesta 2: En esta arquitectura se ofrece un dispositivo embebido el cual estará integrado con sensores de interés, conexión bluetooth 5.0; esta con el propósito de transmitir datos a un “dispositivo maestro” que tiene el rol de intermediario entre el dispositivo de monitoreo y la red, que convenientemente será un smartphone el cual se encargará de hacer el proceso de conexión a internet tanto por wifi o por su integración 5G, la cual permitirá pasar los datos a la nube donde se tiene el mismo propósito que en la propuesta 1.

En ambas propuestas se propone usar el protocolo de mensajería MQTT el cual tiene como base TCP/IP la cual abarca cinco capas del modelo OSI. Se pretende usar el protocolo IPv6 por su mayor número de direcciones IP.

En cuanto al control de datos, donde una vez los datos lleguen a la nube, desde este punto podemos controlar estos datos obtenidos donde se puede aplicar “Backend”, para posteriormente hacer “Frontend” para su posterior visualización o monitoreo y manipulación de datos.

Este proyecto se orientó más hacia la propuesta dos, ya que es más practica en corto plazo para su desarrollo, así como también las ventajas que esta trae, sin embargo, las dos son posibles de realizar.

4.2 Análisis de tecnologías

4.2.1 Bluetooth

Tabla II. Especificaciones transmisión de Bluetooth 5 vs bluetooth 4.2 [29].

Especificaciones	Bluetooth 5	Bluetooth 4.2
Velocidad	Dos veces más alta comparado con bluetooth 4.2 soporta alrededor de 2 Mbps	Menor, acerca de 1 Mbps
Rango	Mayor, cuatro veces más comparado a la versión 4.2, soporta 200 metros en línea de vista (LOS, line of sight) camino en entornos exteriores, soporta 40 metros en entornos interiores	Menor, soporta 50 metros en exteriores y 10 metros en interiores
Requerimiento de potencia	Alto	Bajo
Capacidad de mensaje	Largo, alrededor de 255 bytes	Corto, alrededor de 31 bytes de los cuales proporciona de 17 a 20 bytes para la carga de datos reales
Robustez para operar en ambientes congestionados	Mayor	Menor
Vida de batería	Mayor	Menor
Rendimiento de datos teóricos	2 Mbps, da acerca de 1.6 con sobrecarga	1 Mbps
Fiabilidad	Alta	Baja
Vida digital	Alta	Baja
Control de seguridad	Mejor	Menos segura comparada a 5.0
Soporte para dispositivos IoT	Si	No

En esta tabla notamos especificaciones de interés tales como la velocidad que maneja esta tecnología, la cual es 2 Mbps que viene estando sobrada para la aplicación, pero lo que realmente es el punto de interés es: el rendimiento de potencia y el rango en el que puede estar transmitiendo fielmente, los cuales indican tener especificaciones adecuadas para esta aplicación, ya que el usuario cuando no porta su celular o dispositivo maestro para hacer la comunicación, puede dejarlo a una distancia relativamente considerable, lo

suficiente como para estar en una casa, en un gimnasio, etc. En cuanto al rendimiento; ya que se pretende que sea un dispositivo inalámbrico que utilizara batería, es de importancia que use los mínimos recursos suficiente para la publicación de datos a la red IoT y recibir notificaciones de la misma.

Bluetooth es una tecnología que tiene cerca de 20 años en el mercado por lo que se ha hecho un estándar en telecomunicaciones de corto alcance entre dispositivos por lo que la mayoría de los celulares por no decir que todos cuentan con esta tecnología. Por ende, no será problema de expansión en un futuro ya que este tipo de tecnologías están adaptadas para trabajar con versiones anteriores. Esta es la principal razón por la que se elige bluetooth, por su practicidad y su abundancia en el mercado actual entre muchas otras virtudes que tiene.

4.2.2 MQTT

Para IoT es de suma importancia comunicar o mandar información lo más rápido posible, MQTT fue pensado para comunicación maquina a máquina por lo cual lo hace más simple, ligero, abierto, fácil de implementar y más directo. Está diseñado para tener una mínima sobrecarga sobre todo cuando aparecen sistemas embebidos o limitados. Este protocolo permite hacer una topología de red estrella conformada como se muestra en la Figura 14.

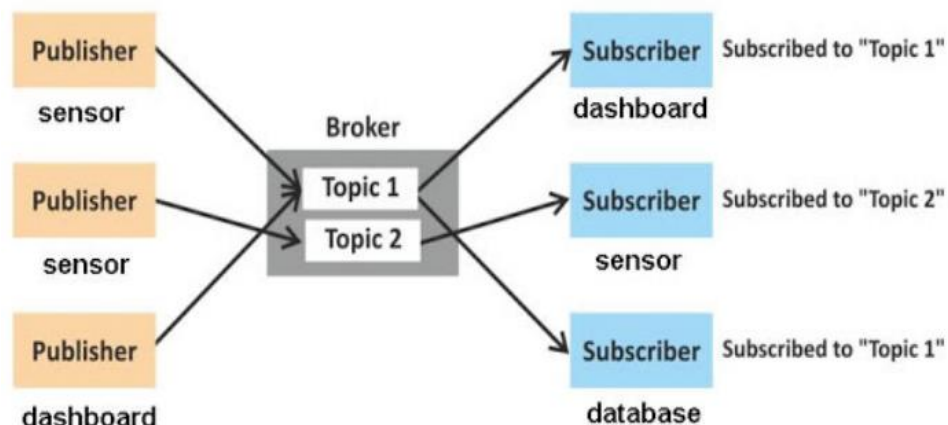


Figura 14. Ejemplo de arquitectura básica MQTT

Como elemento principal tenemos el bróker donde los diferentes clientes que hacen función como editores o suscriptores, los puede haber en el mismo dispositivo, en la parte izquierda podemos ver como esos editores publican en diferentes “topics” o temas, donde los suscriptores que aparecen de lado derecho de la figura, se suscriben a esos “topics” o temas para que el bróker le envíe los datos correspondientes al tema al que se han suscrito. A grandes rasgos ese es el concepto básico que tiene la arquitectura MQTT.

Conceptos de MQTT. - *Topic*. - Se refiere a una cadena de texto que el bróker utiliza para filtrar los mensajes para cada cliente conectado. Puede contener varios niveles y cada nivel de tema está separado por una barra inclinada.



Figura 15. Ejemplo de jerarquía de “topic”.

¿Es MQTT un protocolo fiable?: Si, por que implementa un servicio de calidad (QoS, Quality of service). Este servicio determina la fiabilidad con la que el mensaje se entregará a los receptores.

El QoS.- se especifica en cada mensaje que se envía y puede haber tres niveles de calidad:

- QoS 0: el mensaje se envía como máximo una vez. Esto implica que no garantiza que se entregue.
- QoS 1: el mensaje se envía al menos una vez. Se garantiza la entrega en el receptor, pero puede que el emisor envíe duplicados.
- QoS 2: se garantiza que el mensaje llegara al receptor y que solo se enviara una vez por el emisor.

Utilizar un nivel de calidad u otro dependerá de la fiabilidad que queramos tener en nuestro sistema. Cuanto mayor sea el nivel de calidad mayor cantidad de recursos se utilizan y menor será el rendimiento [28].

En la actualidad la gran mayoría de dispositivos usan el protocolo HTTP, esto por ser uno de los más populares y usados, sin embargo, MQTT creció rápidamente por los beneficios que brinda. Por ende, se puede hacer referencia al comparando los dos protocolos.

Tabla III. MQTT VS HTTP

	MQTT	HTTP
Propósito	Mensajería	Documentos
Eficiencia de protocolo	Alto	Promedio
Eficiencia de energía	Alto	Promedio
Calidad de servicio	Si	No
Lenguajes pares el cliente	Muchos	Muchos

En la Tabla III se observan las principales características que los dos protocolos brindan y es que podemos notar que ambos tienen diferentes propósitos por lo que los parámetros como eficiencia en cuanto a protocolo en sí, eficiencia en energía, la calidad de servicio, se inclinan más de un lado que del otro.

Rendimiento

Se tomó de base un experimento con características similares a la aplicación de interés con la cual usaron un ESP-WROOM-02 conectado a un router de internet wifi para generar paquetes de datos como editor, simultáneamente está recibiendo paquetes de datos como suscriptor donde una Raspberry pi juega el rol de bróker, la cual recibe los paquetes de datos del editor y reenvía los paquetes de datos recibidos al suscriptor. Y se obtuvieron los resultados con las tres diferentes calidades de servicios, como se muestra en la Figura 16.

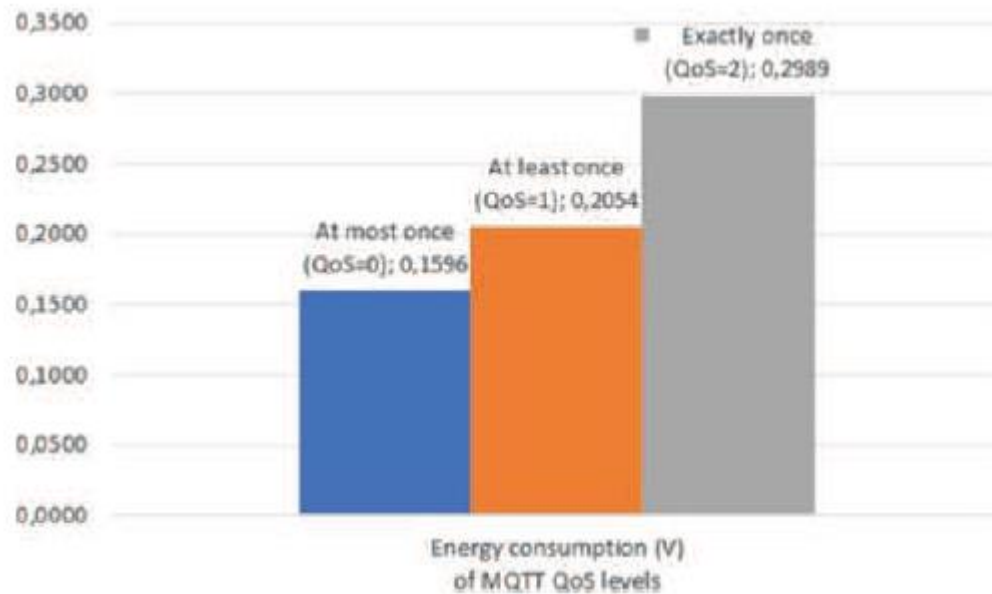


Figura 16. Consumo de energía de los tres diferentes niveles QoS de MQTT.

Se evaluó la diferencia en consumo de energía entre los diferentes niveles de QoS.

Tabla IV. Resultados experimentales obtenidos de los niveles de calidad MQTT.

Nivel MQTT QoS	Energía consumida de la batería(V)
QoS = 0	0.1596
QoS= 1	0.2054
QoS = 2	0.2989

Con estos resultados, se concluye existe una variación del consumo de energía dependiendo de la calidad de servicio que apliquemos en la práctica. El nivel 1 consume 29% más que el nivel 0, el nivel 2 consume 87% más energía que el nivel 0 y 45% más que el nivel 1. En base a esta referencia es crucial decidir por el un nivel de calidad. Por lo tanto, en este proyecto, ya que los parámetros a medir son de suma importancia para la salud y atención del usuario, se tiene como prioridad la fiabilidad de los datos en comunicación con su entorno, por lo que un nivel 3 de calidad es el mejor postor, aunque esto signifique sacrificar un poco más de consumo de batería [30].

En otro estudio realizado por Stephen Nicholas, se comparó MQTT vs. HTTPS para 1024 1 byte mensajes, usando una simple aplicación de mensajería usando “Comet long polling” se obtuvo de resultado la imagen de la Figura 17 de las tablas experimentales.

Receiving				
	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.43%	16.13%	3.45%	4.23%
Messages / Hour	1708	160278	3628	263314
% Battery / Message	0.01709	0.00010	0.00095	0.00002
Messages Received	240 / 1024	1024 / 1024	524 / 1024	1024 / 1024

Sending				
	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.79%	17.80%	5.44%	3.66%
Messages / Hour	1926	21685	5229	23184
% Battery / Message	0.00975	0.00082	0.00104	0.00016

Figura 17. Comparación de eficiencia y rendimiento entre MQTT y HTTP en mensajes recibidos y enviados.

En la misma Figura 17 apreciamos la comparación entre ambos protocolos usando wifi y 3G, se realizó la prueba en mensajes enviados y recibidos, se evaluó la cantidad de batería por hora, mensajes por hora y la cantidad de mensajes por porcentaje de batería. Observando el comportamiento de ambos protocolos se puede dar cuenta que MQTT casi siempre fue mejor y más eficiente en todos los casos, en algunos insignificamente mejor y en otros notablemente superior [31].

Resumen

Ya que se prevé que el número de dispositivos en el futuro crecerá de manera exponencial, lo primero que buscamos es eficiencia, esto es fundamental para la incorporación de la tecnología 5G. En nuestra aplicación tenemos “M. health” que es el enfoque donde parten las características de la arquitectura, ya que, al monitorear una persona, no requiere tanto proceso de datos, buscamos la seguridad en que el sistema embebido que se esté usando, vistiendo o llevando consigo, sea lo más autónomo posible. Ya que los datos que transmite el usuario, son de crucial importancia para la ejecución de acciones que se pueden tomar en consecuencia.

¿Por qué usar MQTT entre otros protocolos?

Porque el tipo de información a enviar. Para pequeños mensajes es mejor MQTT, mientras que para bloques de datos conviene HTTP. MQTT es recomendable en nodos alimentados a baterías ya que mejora el uso de la energía. Para comunicación entre dispositivos siempre es conveniente utilizar MQTT, de hecho, para eso se pensó en principio el protocolo. MQTT puede adaptarse a redes poco confiables mediante la configuración de la calidad de servicio QoS. Es un protocolo bastante liviano y fácil de implementar en dispositivos pequeños que no tienen mucha capacidad de procesamiento.

¿Por qué usar Bluetooth 5?0?

Esta tecnología es muy popular y se ha vuelto un estándar en la industria de los dispositivos móviles, de ahí surge su practicidad que es un punto fuerte. No obstante, gracias a los avances de alcance y consumo de energía lo hacen aún más adecuado. Ya que la idea principal es el monitoreo de usuarios o pacientes mediante un dispositivo que pueda ser “llevable” o “vestible” y que el mismo tenga comunicación directa con un smartphone, todas estas virtudes que bluetooth otorga, se adaptan muy bien a la aplicación.

Tomando como punto de partida la arquitectura propuesta número dos, en cuanto la tecnología 5G, al no contar con equipo de pruebas y experimentación con, así como tampoco se cuenta con una infraestructura funcional en zona local. Al hablar también de bluetooth no existen tarjetas de desarrollo de prototipado de Bluetooth 5 en la fecha actual de elaboración de este proyecto. Serán omitidas en este desarrollo, sin embargo, se puede proceder a diseñar una arquitectura donde se sustituyan estas tecnologías para tener resultados de igual manera, en caso de 5G al hacer una conexión con internet, se piensa para probarse con dispositivos. sin embargo, no se pierde la esencia del objetivo ya que toda la arquitectura fue pensada para ser compatible con el propósito principal y dar una posible solución a la problemática.

4.3 Topología de red y conexión.

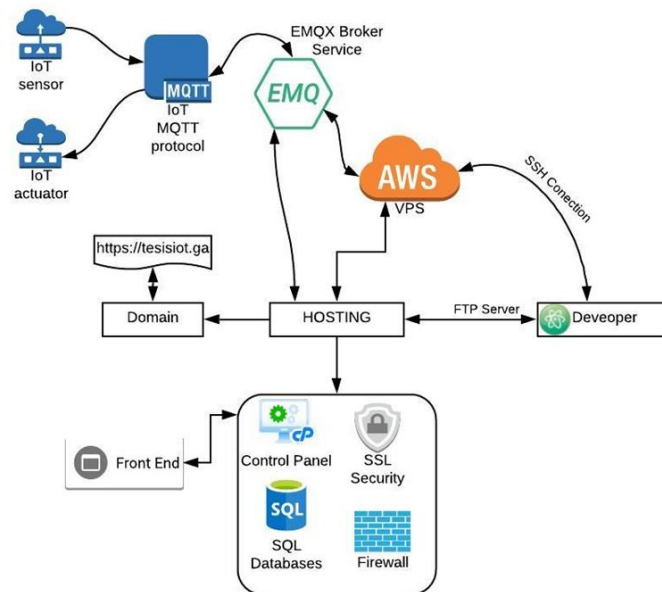


Figura 18. Topología de desarrollo de red para arquitectura propuesta 2.

Se estructuró la tipología general ilustrado en Figura 18. Topología de desarrollo de red para arquitectura propuesta 2. La cual se basa en tres partes intercomunicadas la cual permitió un sistema de conexión de diferentes tipos. Estas partes corresponden a servicios que ofrece la industria en el ámbito de internet de las cosas y más. Existen diversas compañías que dan competencia en estos servicios, sin embargo, se optó por las más convenientes.

4.4 Funcionamiento de sistema

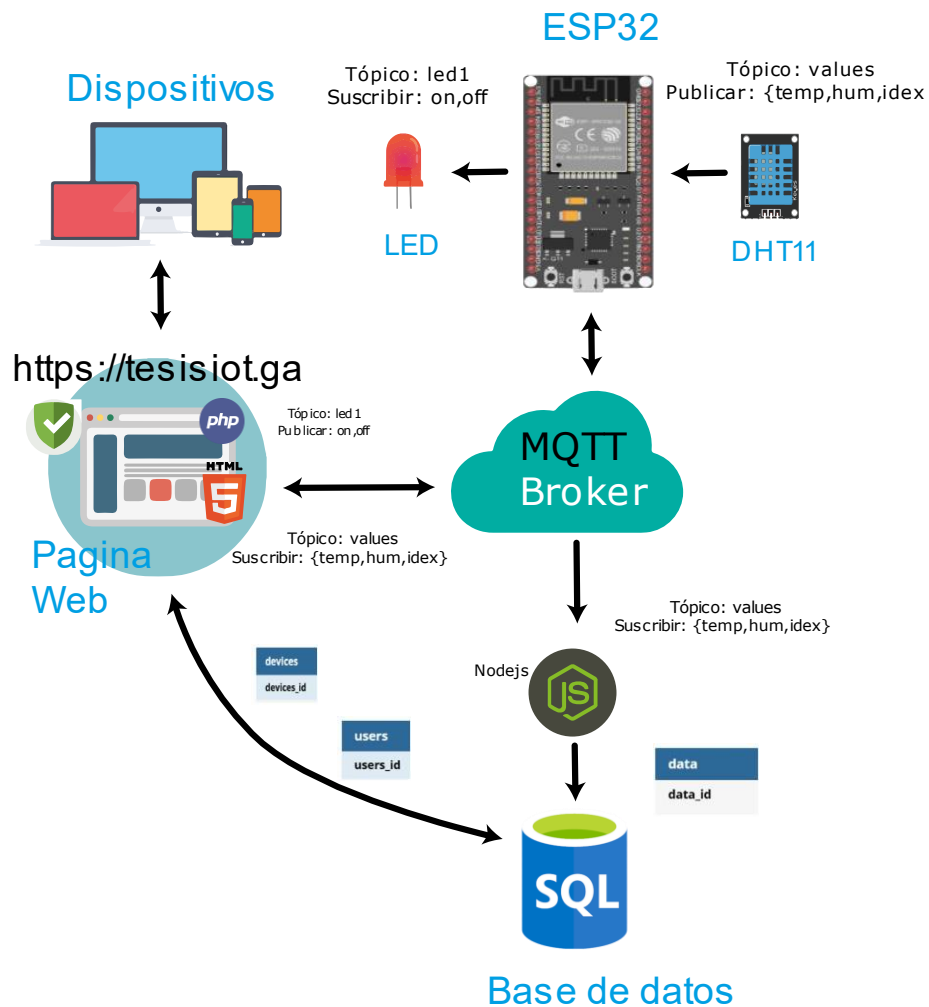


Figura 19. Diagrama de funcionamiento básico de sistema.

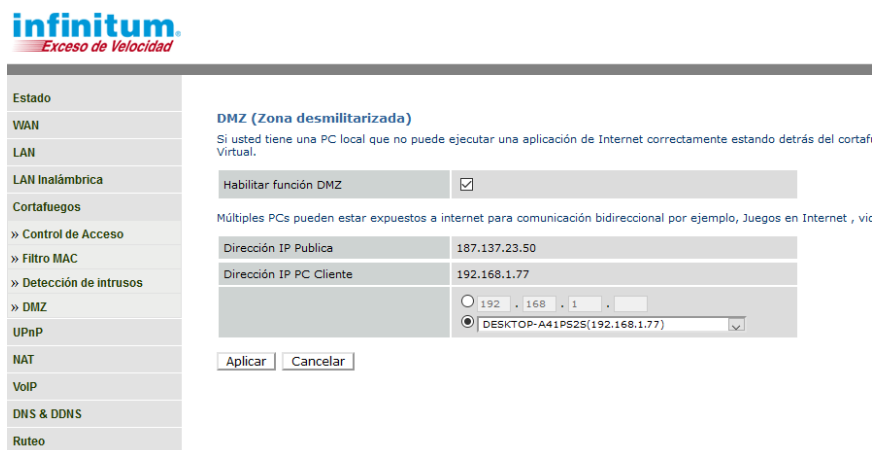
Se trabajó con el dispositivo ESP32 el cual se encargó en parte de mandar los datos censados a través del protocolo MQTT publicando hacia la base de datos con ayuda de "nodejs". Este dispositivo también está bajo suscripción del tópico que le permite accionar cuando la página web le ordene. La página web se suscribe al tópico que le permite visualizar los datos que el ESP32 manda a MQTT, al estar conectado mediante el mismo "websocket", también manda la publicación con los botones correspondientes.

Los datos que son ingresados en la página web están directamente relacionados con la base de datos, tales como información de usuarios y dispositivos.

4.5 Configuración de desarrollador

Al trabajar con servicios web y servidores, es necesario tener establecidos algunos parámetros. Ya que se utilizaron herramientas en línea, es necesario determinar quién es el administrador y quien tiene los permisos para hacer modificaciones en caso de anomalías o fallos en el funcionamiento del sistema.

Configuración del “router”. El proveedor de servicio internet domestico estándar, trae predeterminada su configuración de facto en el hardware del “router”. Aunque trae algunos puertos de interés liberados, es necesario asegurarnos de no tener problemas en un futuro. Para eso se realizó el procedimiento de la Figura 20.



The screenshot shows the 'infinitem' web interface with the slogan 'Exceso de Velocidad'. On the left is a navigation menu with options: Estado, WAN, LAN, LAN Inalámbrica, Cortafuegos, » Control de Acceso, » Filtro MAC, » Detección de intrusos, » DMZ, UPnP, NAT, VoIP, DNS & DDNS, and Ruteo. The 'DMZ' option is selected. The main content area is titled 'DMZ (Zona desmilitarizada)' and includes a description: 'Si usted tiene una PC local que no puede ejecutar una aplicación de Internet correctamente estando detrás del cortafi Virtual.' Below this, there is a checkbox 'Habilitar función DMZ' which is checked. A note states: 'Múltiples PCs pueden estar expuestos a internet para comunicación bidireccional por ejemplo, Juegos en Internet , vic'. There are two input fields: 'Dirección IP Publica' with the value '187.137.23.50' and 'Dirección IP PC Cliente' with the value '192.168.1.77'. Below these is a radio button interface for selecting a device, with 'DESKTOP-A41PS2S(192.168.1.77)' selected. At the bottom are 'Aplicar' and 'Cancelar' buttons.

Figura 20. Interfaz de configuración internet infinitem modo control de acceso.

Ingresado en nuestro navegador web, la siguiente dirección <http://192.168.1.254> y al haber puesto nuestros datos de usuario. En la sección control de acceso, se modificó la parte de DMZ. Zona desmilitarizada es el nombre de esta sección, a la cual se le estableció el IP local e IP publica, dichas direcciones que fueron apuntadas a una dirección MAC.

El anterior procedimiento nos provee de mucha facilidad de manipulación de puertos de comunicación. Sin embargo, el dispositivo que tiene estos privilegios queda con una gran vulnerabilidad en seguridad informática, pero con las debidas precauciones se vuelve una óptima configuración.

Lo primordial aquí, es que esta configuración nos permite determinar el dispositivo que va a ser capaz de tener permisos de administrador, en este caso es la computadora DESKTOP-A41PS2S ilustrada en la imagen Figura 20. Interfaz de configuración internet infinitum modo control de acceso. El “router” del proveedor de internet una vez realizada esta configuración le permite, mas no garantiza, una dirección IP publica fija. Para garantizar que esto funcione. Es necesario establecer la IP local de nuestro dispositivo de manera estática.

Ya que, al reiniciar nuestro dispositivo, o al reiniciar nuestro “router”, los demás dispositivos que estaban conectados al mismo “router”, se distribuyen de manera indeterminada las direcciones IP brindadas por dicho “router”, como se muestra en la Figura 21.

DHCP Estatico

ID	IP	Dirección MAC
1	192 . 168 . 1 . 77	00 : 28 : F8 : 63 : A1 : C8
2	0 . 0 . 0 . 0	00 : 00 : 00 : 00 : 00 : 00
3	0 . 0 . 0 . 0	00 : 00 : 00 : 00 : 00 : 00
4	0 . 0 . 0 . 0	00 : 00 : 00 : 00 : 00 : 00
5	0 . 0 . 0 . 0	00 : 00 : 00 : 00 : 00 : 00

Figura 21. Interfaz de configuración internet infinitum DHCP.

En la sección LAN, apartado de DHCP, el proveedor nos brinda el anterior menú para configurar nuestra ip fija. En este menú proporcionamos la dirección MAC de nuestro dispositivo y una ip local. Así le ordenamos a nuestro router, la dirección MAC establecida tiene privilegios con dicha ip local. Para declarar la ip local, preferentemente nos fijamos en cual se tiene en el momento, ingresando al símbolo de sistema de Windows, con el comando “ipconfig -all”.

```

Seleccionar C:\WINDOWS\system32\cmd.exe
Configuración automática habilitada . . . : sí

Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . :
Descripción . . . . . : Intel(R) Dual Band Wireless-AC 8265
Dirección física. . . . . : 00-28-F8-63-A1-C8
DHCP habilitado . . . . . : sí
Configuración automática habilitada . . . : sí
Dirección IPv6 . . . . . : 2806:1016:a:25d:c18c:e86d:c129:9d64(Preferido)
Dirección IPv6 temporal. . . . . : 2806:1016:a:25d:a86f:c80:a2f1:a12a(Preferido)
Vínculo: dirección IPv6 local. . . . . : fe80::c18c:e86d:c129:9d64%8(Preferido)
Dirección IPv4. . . . . : 192.168.1.77(Preferido)
Máscara de subred . . . . . : 255.255.255.0
Concesión obtenida. . . . . : jueves, 16 de julio de 2020 10:24:32 a. m.
La concesión expira . . . . . : viernes, 17 de julio de 2020 11:14:15 a. m.

Puerta de enlace predeterminada . . . . : fe80::1%8
                                           192.168.1.254
Servidor DHCP . . . . . : 192.168.1.254
IAID DHCPv6 . . . . . : 67119352
DUID de cliente DHCPv6. . . . . : 00-01-00-01-25-00-7D-C0-00-22-4D-9A-92-CD
Servidores DNS. . . . . : fe80::1%8
                                           192.168.1.254
NetBIOS sobre TCP/IP. . . . . : habilitado

```

Figura 22. Ventana de símbolo del sistema Windows.

En la Figura 22, se visualiza la información que el hardware nos proporciona, es importante tener identificado el adaptador de red correspondiente. Aquí se pudo corroborar la IP local, dirección IP, nombre de adaptador y otras configuraciones.

4.6 Implementación y configuración del VPS

En la parte central se cuenta con un servidor VPS (Servidores virtuales privados) brindado por “Amazon Web Services”, esta empresa es la que brindó los elementos de hardware suficientes para llevar a cabo todos los procesos computacionales. Se configuró con un sistema operativo Ubuntu versión 18 por su flexibilidad, eficiencia y seguridad para correr un servidor con estos propósitos, en un disco duro de estado sólido de 8Gb de almacenamiento. Este sistema se situó en la zona más cercana de la localidad la cual viene a ser el este de Ohio, Estados Unidos.

En cuanto a configuración de red se abrieron los puertos de protocolo TCP que se enlistan en la Tabla V. Estos se encargan de establecer las conexiones requeridas necesarias para IoT y que funcionen los elementos planteados en la topología:

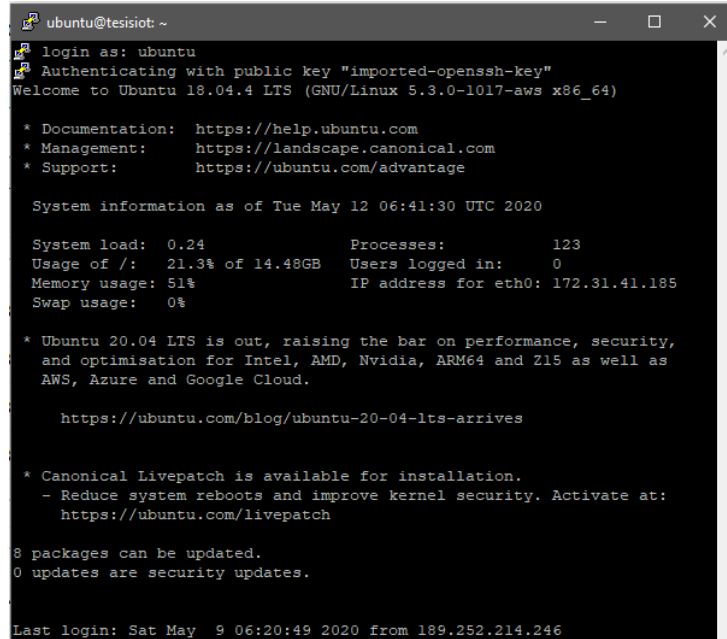
Tabla V. Reglas de entrada AWS

Tipo	Protocolo	Puerto	Fuente	Descripción
SSH	TCP	22	Ip Propia	SSH
HTTP	TCP	80	Cualquiera	HTTP
Custom TCP	TCP	8080	Cualquiera	VESTA
Custom TCP	TCP	8083	Cualquiera	-
Custom TCP	TCP	18083	Cualquiera	-
Custom TCP	TCP	8883	Cualquiera	-
Custom TCP	TCP	8090	Cualquiera	-
Custom TCP	TCP	1883	Cualquiera	EMQX
HTTP	TCP	443	Cualquiera	HTTPS
Custom TCP	TCP	8093	Cualquiera	EMQX
Custom TCP	TCP	12000-12100	189.252.214.246/32	FTP PASIVO
Custom TCP	TCP	21	Cualquiera	FTP
Custom TCP	TCP	8094	Cualquiera	EMQX
MYSQL/AURORA	TCP	3306	Cualquiera	MYSQL

Se tienen estas reglas de entrada descrita donde usamos diversos puertos para hacer uso de diversos servicios, en cuanto a las reglas de salida se tiene una configuración abierta a cualquier destinatario.

Dado que el CPU donde apunta nuestro servidor requiere de apagarse y/o volver a encender. Se reservó una IP fija para evitar que esta cambie al detectar que nuestro CPU tiene una ip local y publica diferente entregada por el router de servicio de internet que se cuenta en este caso con TELMEX router Huawei modelo HG8245H que actualmente también se configuró para establecerle una IP local fija al MAC adress correspondiente. Al final se obtuvieron los siguientes datos estables: Ip local: 192.168.1.219, IP Publica de ordenador local: 189.252.153.26, Ip pública del VPS: 3.15.255.39.

Para tener comunicación con el ordenador virtual del VPS que en este caso se tiene con un sistema operativo Ubuntu, al ser Windows un sistema operativo con sus diferencias en sus interfaces, se requiere una conexión por SSH. Mediante Windows la conexión se tiene que hacer mediante el cliente PUTTY una vez ya configurado también con PUTTYgen.



```
ubuntu@tesisiot: ~  
login as: ubuntu  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1017-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Tue May 12 06:41:30 UTC 2020  
  
System load:  0.24          Processes:      123  
Usage of /:   21.3% of 14.48GB  Users logged in:  0  
Memory usage: 51%          IP address for eth0: 172.31.41.185  
Swap usage:   0%  
  
* Ubuntu 20.04 LTS is out, raising the bar on performance, security,  
and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as  
AWS, Azure and Google Cloud.  
  
https://ubuntu.com/blog/ubuntu-20-04-lts-arrives  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
https://ubuntu.com/livepatch  
  
8 packages can be updated.  
0 updates are security updates.  
  
Last login: Sat May  9 06:20:49 2020 from 189.252.214.246
```

Figura 23. Interfaz PUTY estableciendo conexión con VPS.

Se ingresa como usuario Ubuntu donde se muestra un breve historial de introducción con información del set up de hardware y software, fechas de ingreso, números de procesos, entre otras cosas.

4.7 “HOSTING”

VestaCP este panel de gestión gratuito con código abierto para servicios de VPS es una herramienta muy potente y muy liviano, con las características que son otorgadas son más que suficiente para darle soporte a nuestra arquitectura, tiene compatibilidad con Ubuntu y de las herramientas que proporciona, que se hicieron uso son la base de datos MySQL, Firewall, seguridad de conexión SSL, motor de FTP. Se establecieron ajustes de configuración con el nombre del host tesisiot.ga. Se procedió a instalar mediante la consola PUTTY con dirección al sistema operativo del servidor, con los comandos predeterminados por Vesta. Estableces tu usuario y contraseña para tener privilegios de administrador en el panel de control y al finalizar te brinda una dirección ip misma del servidor VPS.

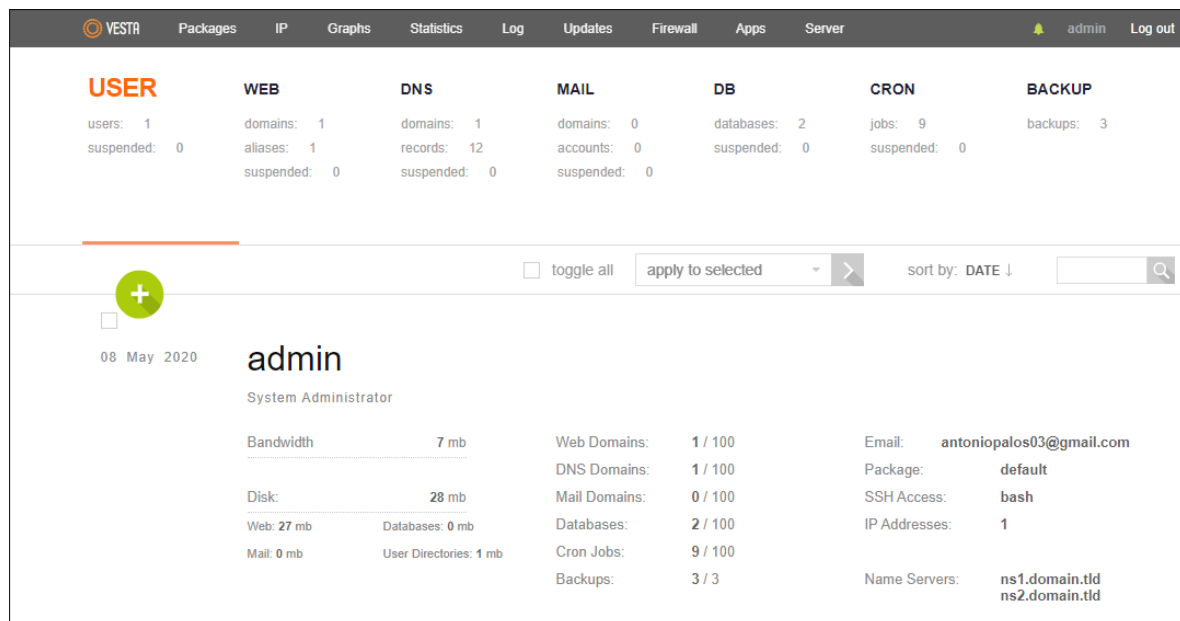


Figura 24. Ventana principal del panel de control de VESTACP.

Al finalizar la instalación se confirmó el acceso al panel de control al poner el IP de nuestro VPS y seleccionar el puerto 8083, una vez entrado al “login”, vemos nuestros datos y las diferentes herramientas que podemos usar, lo anterior se muestra en la Figura 24.

4.8 Registro de dominio.

Para pasar la representación numérica única que proporciona el servidor a un DNS se recurrió a un dominio gratuito, dado que es más difícil recordar un número, este se transforma a caracteres. Se tiene a la empresa líder en dominios gratuitos llamada Freenom, Figura 25. Sin embargo, no por ser un dominio gratuito significa que tendrá un desempeño des privilegiado.

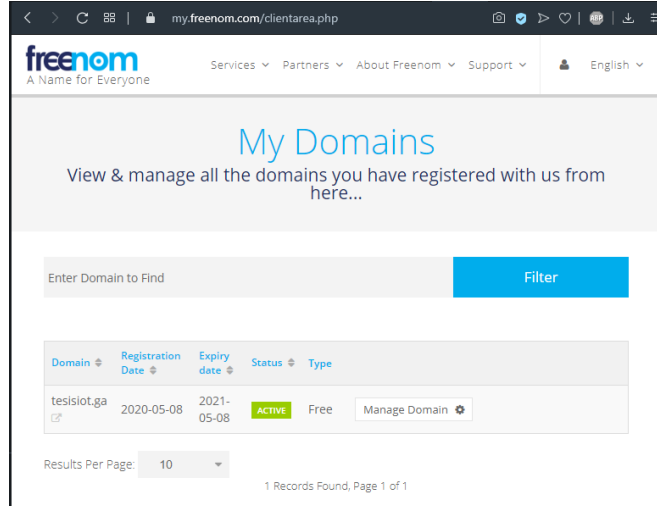


Figura 25. Estatus del dominio creado con freenom.

Se creó el DNS pasando a ser de 3.15.255.39 a tesisiot.ga bajo una cuenta propia con nuestros datos. El servicio dura un año al igual que AWS.

Una vez configurado, se aseguró y se confirmó a través de otra plataforma web el rastreo y cobertura de nuestro DNS alrededor del mundo. whatsmysdns.net es el responsable de esta acción la cual dos demostró la siguiente imagen.

4.9 Certificado de seguridad SSL.

Posteriormente se hizo cargo en cuestiones de seguridad, VestaCP permite un certificado de seguridad SSL el cual se activó pasando de obtener un ingreso seguro a través de cualquier navegador WEB, como se presenta en la Figura 26.

WEB DNS MAIL DB CRON

EDITING DOMAIN

☒ SSL Support

☒ Lets Encrypt Support

SSL Home
public_html

SSL Certificate
9w0BAQEFAAOCAg8AMIICGgKCAgEau/PRNpmlxs/STxsSvAk
tyWZupilYznirwPqM
1DZ1oTB8HXEAuFzhcD2PAQzJiJaCXyqw43DIwtITCFZWuxsW
TkPqSI7mnhLOGPjq
OhG/Jena32Sv3L2Yn5Ud1hIrF6PCERfE1dUZvQH7ZopKailieG

SSL Key
-----BEGIN RSA PRIVATE KEY-----
MIIEKQIBAAKCAgEau/PRNpmlxs/STxsSvAktyWZupilYznirwPq
M1DZ1oTB8HXEA
uFzhcD2PAQzJiJaCXyqw43DIwtITCFZWuxsWTKPqSI7mnhLO
-----END PRIVATE KEY-----

SSL Certificate Authority / Intermediate (optional)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEKjCCA3qgAwIBAgIQCGFBQgAAVOfc2oLheynCDANBgk
-----END CERTIFICATE-----

SUBJECT: tesisiot.ga
ALIASES: tesisiot.ga, www.tesisiot.ga
NOT_BEFORE: May 8 00:14:51 2020 GMT
NOT_AFTER: Aug 8 00:14:51 2020 GMT
SIGNATURE: sha256WithRSAEncryption
PUB_KEY: 4096 bit
ISSUER: C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3

Figura 26. Certificado SSL

Una vez se aseguró que nuestro DNS tenga una buena propagación, podemos proceder a configurar nuestra seguridad WEB en este caso SSL se encarga de que http:// a https://, de esta manera se puede verificar la conexión y el funcionamiento. Así podemos lograr una autenticación solida encriptando información dando mayor confianza al cliente o usuario.

4.10 Servidor FTP

Con el propósito de tener una edición mucho más cómoda y eficiente entre el desarrollador y el panel de control que va a permitir subir y transferir archivos al hosting de manera remota hacemos uso de este protocolo FTP (File Transfer Protocol) que están bajo el puerto 20 y 21., y lo ya hecho previamente, la configuración de los puertos en nuestro servidor VPS.

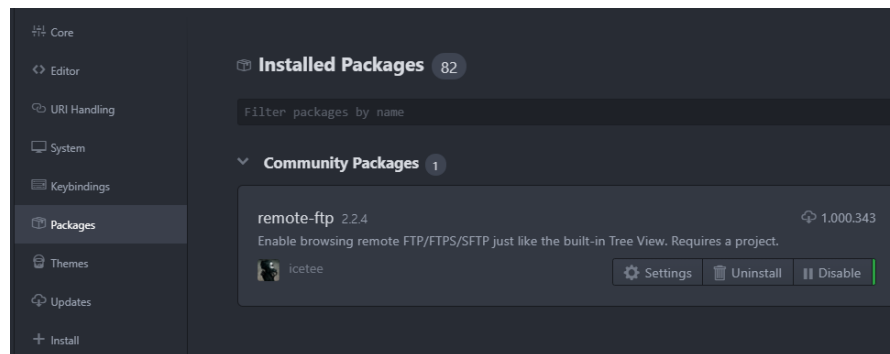


Figura 27. Paquete de herramientas instaladas en Atom.

Nuestro editor de código (Atom) cedió un plugin o un paquete de herramientas para trabajar con este protocolo.

A screenshot of the VestaCP 'Additional FTP' configuration panel. At the top, there's a checkbox 'Additional FTP' which is checked. Below it, 'FTP #1' is shown with a '(DELETE)' link. The 'Username' field contains 'tesisiot', with a note 'Prefix admin_ will be automatically added to username' and a preview 'admin_tesisiot'. The 'Password' field is empty, with a 'generate' link and an eye icon. The 'Path' field contains '/public_html', with a preview '/home/admin/web/tesisiot.ga/public_html'. At the bottom, there's a link 'ADD ONE MORE FTP ACCOUNT' and two buttons: 'Save' (green) and 'Back' (grey).

Figura 28. Panel de configuración de FTP en VestaCP.

Atom y su herramienta FTP facilitó el código por defecto el cual tenemos que configurarlo en nuestro panel de control editando solamente los datos de nuestro dominio e insertando el código en los espacios correspondientes. Así logramos una conexión remota entre nuestra maquina local y el panel de VestaCP.

4.11 Creación de página web.

El diagrama para la creación de la página Web se presenta en la Figura 29.

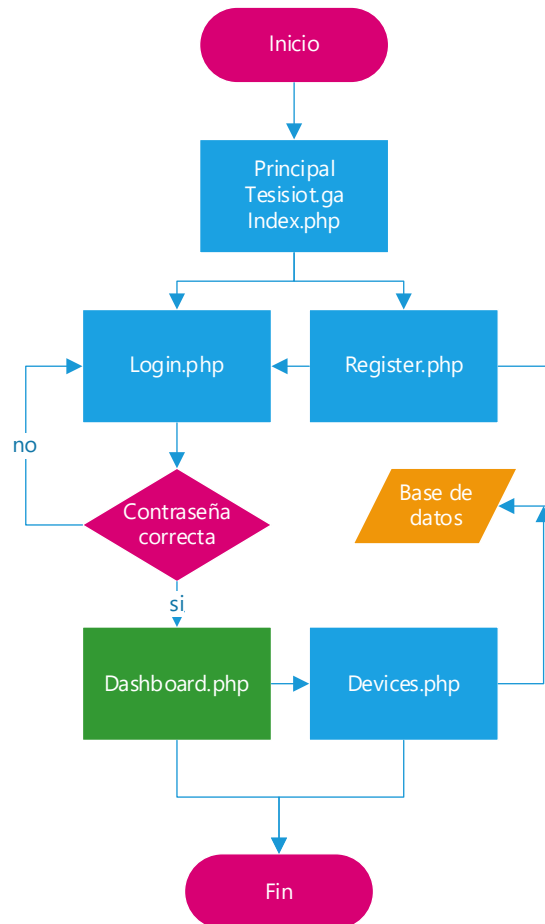


Figura 29. Diagrama de la estructura de la página web.

El usuario al ingresar la URL en el navegador web, directamente lo lleva a la página index.php la cual despliega la página principal, esta trae dos botones principales, “log in” y “register” los cuales tienen el mismo nombre de archivo con extensión “.php”. Al ingresar la cuenta creada de manera exitosa, se direcciona directamente a la página “dashboard”, esta tiene integrada la opción de “devices en ella”. “Devices” y “Register” tienen consulta directa con la base de datos, se quedan registrados las cuentas y contraseñas por parte de “register” y los dispositivos con su número de serie y alias por parte de “devices”.

4.12.1 Página web principal

Al tener un dominio funcionando bajo un DNS sobre el panel de control de VESTA y sobre el poder de control en edición de FTP, se obtuvo una página de simple mensaje, la cual representó el éxito del proceso, como se muestra en la Figura 30.

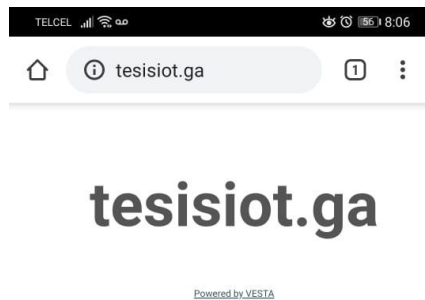


Figura 30. Página por “default” de entrada al dominio sin diseño gráfico.

La Figura 31 demuestra el punto de partida sobre el cual se modificó para su mejora visual.



Figura 31. Página web con modificación grafica.

Las modificaciones realizadas se presentan en la Figura 31. Observamos una mejora significativa. Se creó la página web principal en base a una plantilla prefabricada que da una breve bienvenida e introducción hacia el proyecto y a la cual se le adjuntaron los

botones principales en los que funciona el sistema web y botones de contacto e información de manera extra.

4.12.2 Sistema de registro de cuentas

Para poder tener un control de usuarios y sus datos, fue conveniente brindarle a cada usuario su integridad, a manera que solo ellos puedan visualizar su información. Por ende, se creó el sistema de cuentas donde registras cierta información al sistema (“Sign Up”) y donde al ya tener el sistema tu información puedes acceder de manera autentica (“Log In”).

4.12.2.1 “Sign up”

Al hacer “click” en “Sign up” en la página principal, esta te redirecciona al archivo “register.php” de los archivos web. Dicha página te muestra la ventana que se aprecia en la Figura 32.

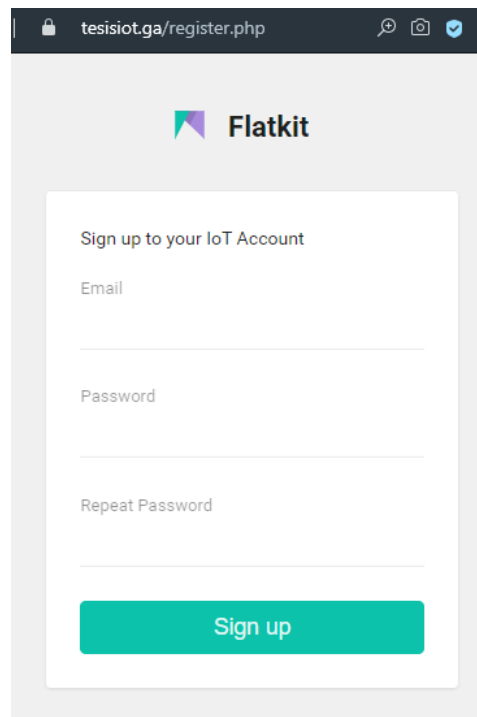
The image shows a web browser window with the address bar displaying 'tesisiot.ga/register.php'. The page features the 'Flatkit' logo at the top. Below the logo is a registration form titled 'Sign up to your IoT Account'. The form contains three input fields: 'Email', 'Password', and 'Repeat Password'. At the bottom of the form is a teal-colored button labeled 'Sign up'.

Figura 32. Interfaz de registro “Sign up”.

La interfaz de registro te despliega tres espacios para ingresar datos: correo electrónico, contraseña, repetir contraseña. Así como también el botón para avanzar. Si el usuario se equivoca en hacer coincidir las contraseñas, se ejecuta la acción de rechazo.

4.12.2.2 “Log in”

Una vez ingresada información a través de la página de registro, esta se queda guardada en la base de datos de manera permanente, como se presenta en la Figura 33.

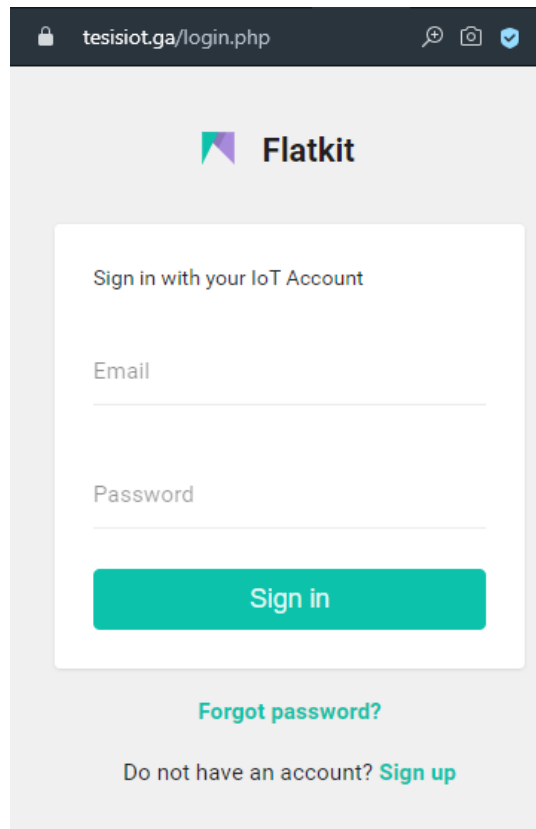
The image shows a web browser window with the address bar displaying 'tesisiot.ga/login.php'. The page features the Flatkit logo at the top. Below the logo is a white rectangular form with the heading 'Sign in with your IoT Account'. Inside the form, there are two input fields labeled 'Email' and 'Password'. A teal 'Sign in' button is positioned at the bottom of the form. Below the form, there is a teal link 'Forgot password?' and a teal link 'Do not have an account? Sign up'.

Figura 33. Interfaz de acceso “Login”.

En esta interfaz solo se ingresa el email registrado y la contraseña correspondiente, una vez ingresado se puede dar clic en el botón “Sign in” y se permite tener acceso. Esta página también puede redirigir a la página de registro si es que no se tiene una cuenta hecha o se requiere hacer otra. El archivo que se encarga de hacer funcionar esta página es “login.php”.

4.12.3 “Dashboard”

El acceso a la interfaz gráfica del usuario se pensó en aplicaciones donde se puede visualizar información de diferentes sensores, tener a la vista distintas graficas de historial, así como también se puede tener diferentes paneles de controles o botones para accionar un actuador si así se desea. El código que respalda este servicio es “dashboard.php” de los archivos de desarrollo. El “dashboard” le da esa facilidad al usuario de visualizar sus datos más importantes con un toque agradable al ojo. Para aplicaciones médicas se pueden configurar parámetros de interés por el proveedor de salud o por el mismo paciente, por lo que se tienen tres variables ajustables de prueba. Y dos botones “check in” los cuales se ponen temporalmente para probar la comunicación bidireccional.

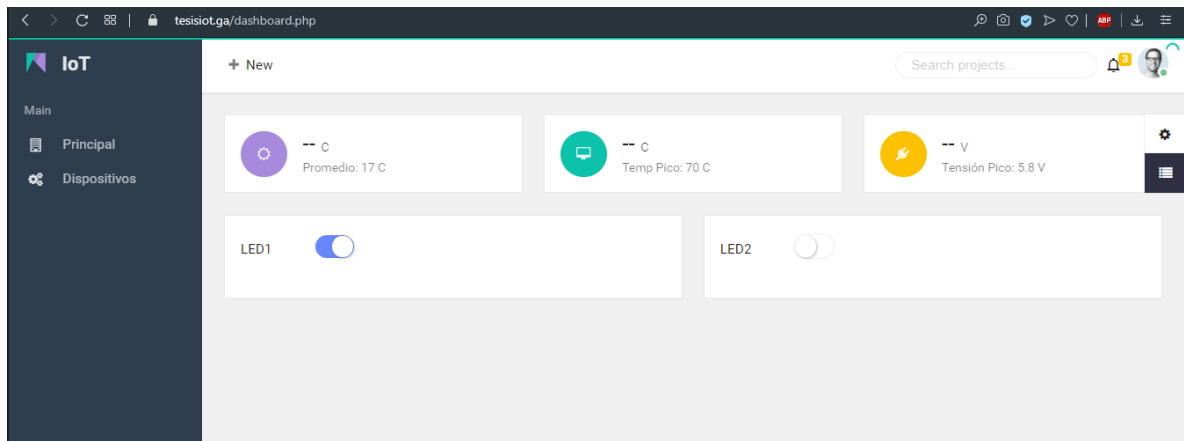


Figura 34. Interfaz “Dashboard” del usuario

Se creó una interfaz presentable donde se visualizan los parámetros mencionados y adicionalmente un menú a la izquierda donde dónde podemos manipular dispositivos, estos dispositivos son los que el usuario puede manejar, así mismo se tiene que registrar para su gestión.

4.12.3.1 Dispositivos

Agregación de dispositivos. Se propone una tabla dinámica donde el usuario puede ingresar los datos de sus dispositivos o sensores con su alias y número de serie.

The screenshot shows the 'Agregar Dispositivo' (Add Device) form in the IoT application. The form has two input fields: 'Alias' with the placeholder 'Ej: Casa Campo' and 'Serie' with the placeholder 'Ej: 777222'. Below these fields is a 'Registrar' button. At the bottom of the form is a table titled 'Dispositivos' with columns 'Alias', 'Fecha', and 'Serie'. The table contains two rows: 'Sensor2' with date '2020-05-13 01:38:18' and serial '124', and 'Sensor1' with date '2020-05-13 01:38:41' and serial '123'. Below the table is a section titled 'Eliminar Dispositivos' with a dropdown menu showing 'Sensor2<-->124' and an 'Eliminar' button.

Alias	Fecha	Serie
Sensor2	2020-05-13 01:38:18	124
Sensor1	2020-05-13 01:38:41	123

Figura 35. Agregación de dispositivos del usuario.

The screenshot shows the 'Eliminar Dispositivos' (Delete Devices) form in the IoT application. It features a dropdown menu with the selected item 'Sensor1<-->123'. Below the dropdown is a red 'Eliminar' button.

Figura 36. Eliminación de dispositivos del usuario

En las Figura 35 y Figura 36 se mostraron las tablas donde se probaron dos dispositivos, en la casilla alias se introdujo el carácter: Sensor1 y un número de serie en la casilla Serie: 123, en esta prueba. Sin embargo, si el usuario quiere eliminar o cometió un error en la asignación se puede remediar seleccionando en la pestaña Eliminar dispositivos, el dispositivo correspondiente y presionando clic en el botón eliminar se queda guardada la acción automáticamente.

4.13 Base de datos

Para la creación de nuestra base de datos, configuramos nuestros datos tales como nombre de base de datos: “tesisiot”, usuario: tesisiot, contraseña: *****, tipo: mysql: Host: local, Charset: UTF8. De esta manera creamos la configuración compatible para trabajar de manera remota y de manera local, usando “phpMyAdmin” y Heidi. El lenguaje de la base de datos es SQL.



Figura 37. Base de datos “PhpMyadmin”.

Esta herramienta permitió visualizar y editar de manera remota ingresando los datos proporcionados al panel de control. Dicha herramienta fue brindada por VESTA.

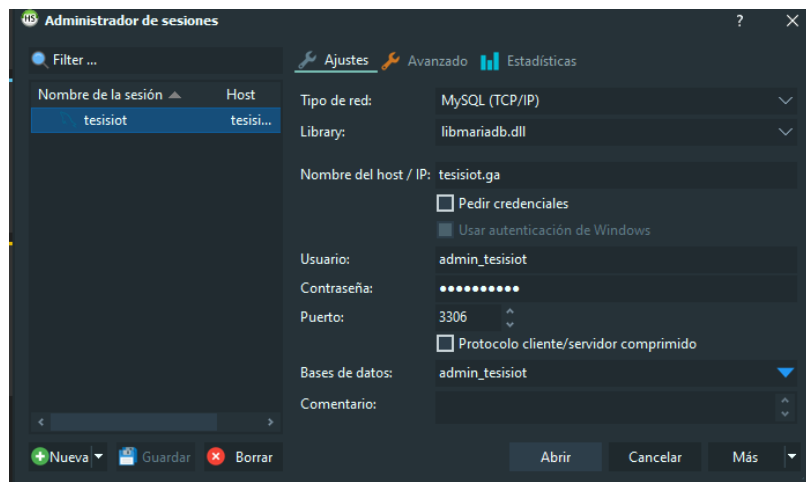


Figura 38. Configuración Heidi para trabajar con nuestra base de datos.

Configurando conexión a través del puerto 3306 y nuestra dirección del host ingresando el mismo usuario y contraseña brindado por VESTA. Es a partir de este programa donde se desarrolló toda creación de tablas y base de datos, como se muestra en la Figura 39.

users		devices	
users_id	int(7)	devices_id	int(7)
users_date	timestamp	devices_date	timestamp
users_email	varchar(60)	devices_alias	varchar(50)
users_password	varchar(60)	devices_serie	varchar(50)
		devices_user_id	int(7)

data		mqtt_acl		mqtt_user	
data_id	int(11)	id	int(11)	id	int(11)
data_date	timestamp	allow	int(1)	username	varchar(100)
data_temp1	float(4,1)	ipaddr	varchar(60)	password	varchar(100)
data_temp2	float(4,1)	username	varchar(100)	salt	varchar(35)
data_volts	float(7,1)	clientid	varchar(100)	is_superuser	tinyint(1)
		access	int(2)	created	datetime
		topic	varchar(100)		

Figura 39. Esquema de base de datos SQL.

Se crearon las anteriores tablas, cinco en total: “users”, “devices”, “data”, “mqtt_acl”, “mqtt_user”. Las cuales “mqtt_acl” y “mqtt_user” son brindadas por el “bróker” MQTT EMQ de su página oficial. Las tablas “users” y “devices” están vinculadas directamente con la página web. La tabla “data” es la que se encarga de guardar la información del sensor, esta información es mandada por el dispositivo ESP32 a través de MQTT donde “nodejs” intercepta los datos para hacer la consulta hacia la base de datos logrando su captura.

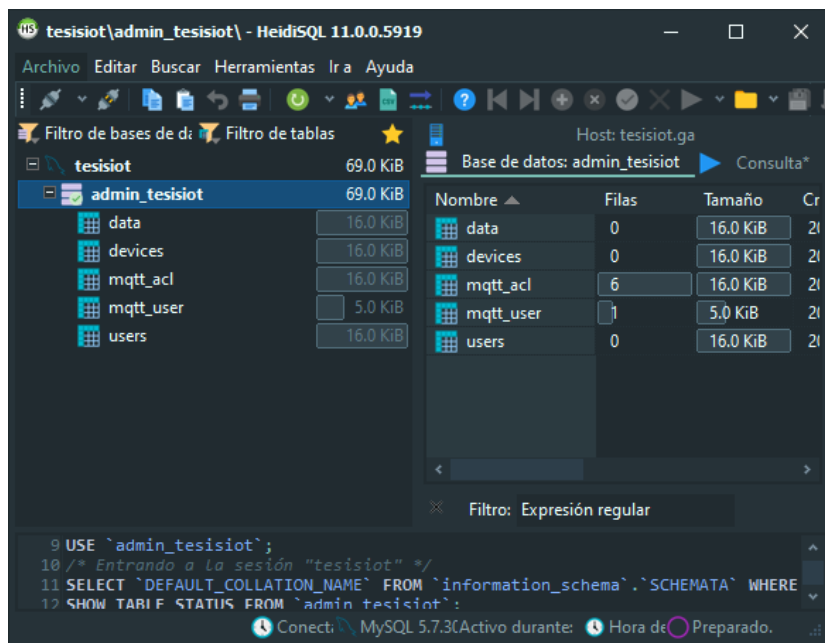


Figura 40. Dentro de la configuración de base de datos.

La que hace versátil esta herramienta es su interfaz, haciendo dinámica la edición donde se muestra de manera organizada los parámetros y las tablas realizadas, dichas tablas podemos modificarlas visualmente, así como también ingresando código SQL. Se necesitan variables del tipo VARCHAR, CURRENT TIME, INT, esto para almacenar diferentes parámetros que una base de datos estándar necesita tener. Que tenga la vinculación con nuestra base de datos alojada y poder editar en tiempo real automáticamente se guardan y se actualizan los cambios lo hace una herramienta muy poderosa y versátil para el desarrollo.

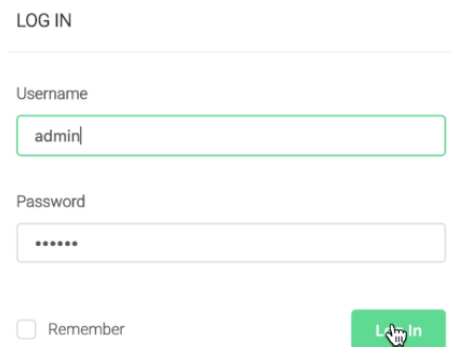
Una vez conectado nuestro editor Heidi a la base de datos de manera remota y el editor Atom de la misma manera. Se editó el archivo "register.php" en Atom para hacer coincidir las tablas creadas en Heidi, se edita a manera de que las variables con las que se estén trabajando sean las mismas y la conexión con el "hosting" sea de igual manera.

4.14 MQTT EMQ

Para establecer nuestro protocolo de mensajería MQTT, se hace valer de la empresa EMQ la cual facilita un bróker de código abierto con versión gratuita al igual que el popular

“broker” de MOSQUITTO, sin embargo, se tiene evidencia por terceros que MOSQUITTO falla en proyectos de gran escala. Por lo que se optó por instalar un “broker” de la empresa EMQ. Para añadir este bróker es necesario que se tenga la correcta apertura de puertos en el hosting y en el VPS, los cuales son 1883,8883,8093,8094,8090,18083,8080. De esta manera garantizamos que el panel que proporciona EMQ pueda tener entendimiento con nuestro sistema. Se hizo uso de la versión v4.0.6 dado que esta también estuvo probada en la versión de paga. Este bróker está diseñado para prototipos como de pequeña hasta gran escala, además puede ser escalable ya que tiene la capacidad de ser “cluster” de lo cual podemos tener potencia en paralelo si se necesita. Tiene Inter compatibilidad con cualquier versión instalada, esto garantiza integración sin problemas.

Instalación: Para poder hacer uso de este bróker necesitamos instalarlo a través de PUTTY nuevamente para poder instalarlo en sistema Ubuntu de nuestro VPS. Con los comandos proporcionados por el desarrollador EMQ ingresados correctamente en modo super usuario de la consola. Se configura para tener acceso al “log in” de la plataforma de EMQ, como se muestra en la Figura 41.



LOG IN

Username

admin

Password

☐ Remember

Login

Figura 41. “Log in” de administrador en EMQ

En este interfaz se ingresan los datos establecidos en la instalación de EMQ para poder entrar al panel de control que este brinda.

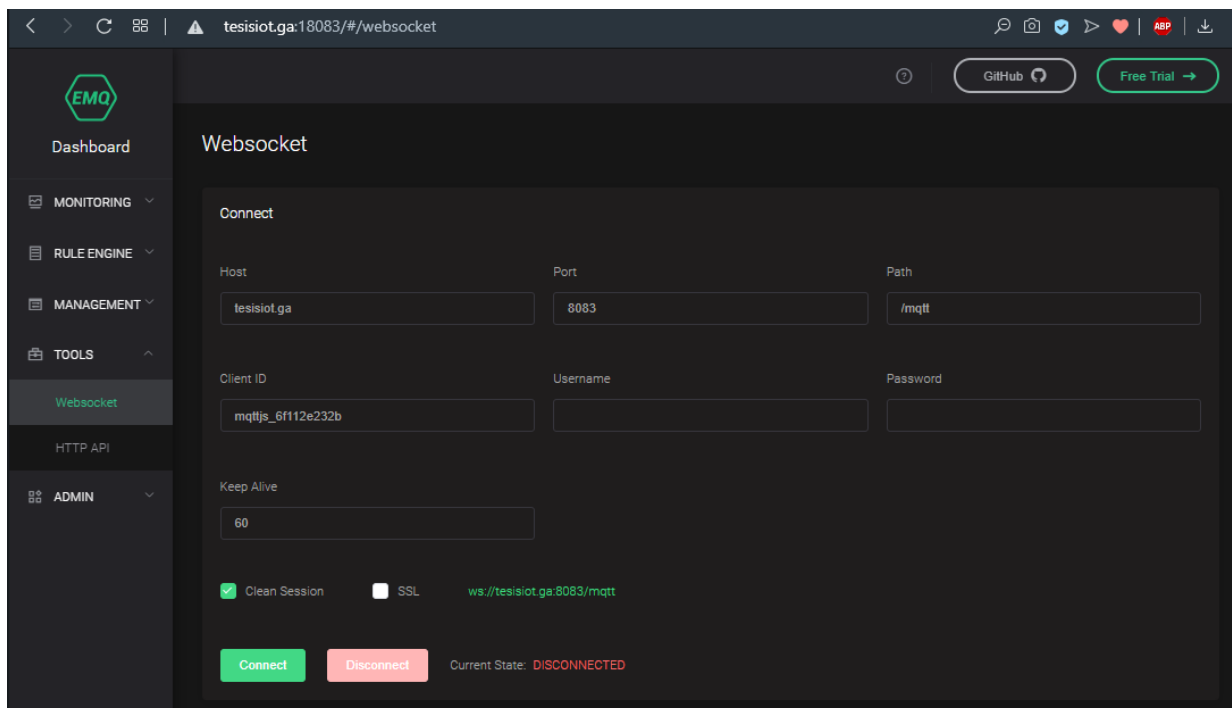


Figura 42. Panel de control EMQ de usuario

La interfaz de la Figura 42 brinda un servicio de visualización de las acciones ejecutándose entre otras cosas, muestra también la lista de puertos que la pagina está escuchando. En la pestaña de “Tools – Websocket”, da muestra del host configurado tesiot.ga. Como se encuentra en un navegador web la única manera de hacer conexión es través de un “websocket” mas no de TCP tradicional, este puede ser accedido por un Arduino o un programa aparte.

Configuración de seguridad SSH. Ya que EMQ tiene la opción de poder dar seguridad SSL, estas credenciales fueron configuradas porque las que trae por defecto no funcionan. Por lo cual se modificaron dos archivos de raíz “*cer.pem*” y “*key.pem*” por SSL. Una vez realizada esta acción se puede marcar la casilla SSL y cambiar el puerto de 8083 que era conexión insegura a 8084 el cual contiene la seguridad.

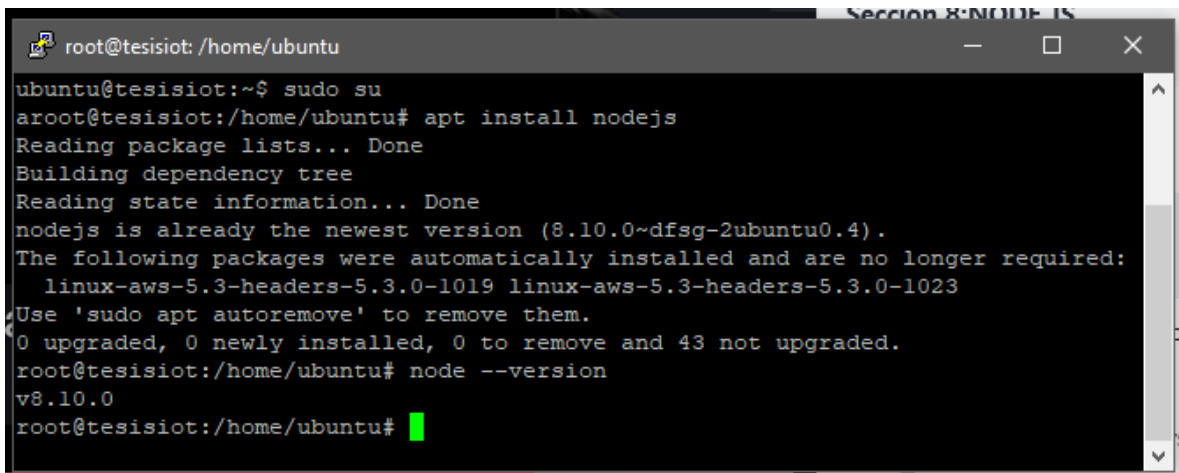
También se configuró el código de nuestra interfaz web, los archivos: “login.php”, y “dashboard.php”. a los cuales se les agrego algoritmo brindado por EMQ en sus páginas oficiales que comunica la interfaz del “dashboard” y la interfaz del panel de control EMQ.

4.15 Integración de NODEJS

La integración de NODEJS fue resultado de realizar las modificaciones que se presentan a continuación.

4.15.1 Instalación de NODEJS en VPS

Este servicio se integró para hacer suscripción a todos los tópicos, siendo cliente de MQTT con el propósito de ser cliente también a una base de datos, en particular a este proyecto es a “mysql”, al almacenar los datos en tiempo real. Se instaló directamente en la computadora donde está establecido el VPS.



```
root@tesisiot: /home/ubuntu
ubuntu@tesisiot:~$ sudo su
aroot@tesisiot:/home/ubuntu# apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
nodejs is already the newest version (8.10.0~dfsg-2ubuntu0.4).
The following packages were automatically installed and are no longer required:
  linux-aws-5.3-headers-5.3.0-1019 linux-aws-5.3-headers-5.3.0-1023
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 43 not upgraded.
root@tesisiot:/home/ubuntu# node --version
v8.10.0
root@tesisiot:/home/ubuntu#
```

Figura 43. Consola de instalación de “nodejs”.

Se requirió tener permisos de super usuario para poder instalar paquetes, con el comando “*sudo su*”, posteriormente con el comando “*apt install nodejs*” se ejecutó la instalación del paquete “nodejs”, al verificar la versión con el comando “*node --version*”, se demostró que la versión instalada fue la v8.10.0. Dentro de la dirección *root@tesisiot:/home/admin/web/tesisiot.ga/public_html/node*. Se creó el archivo llamado *index.js*, sobre el cual se escribió el código de funcionamiento.

4.15.2 Funcionamiento del programa

El diagrama utilizado para la integración del nodejs, se muestra en la Figura 44.

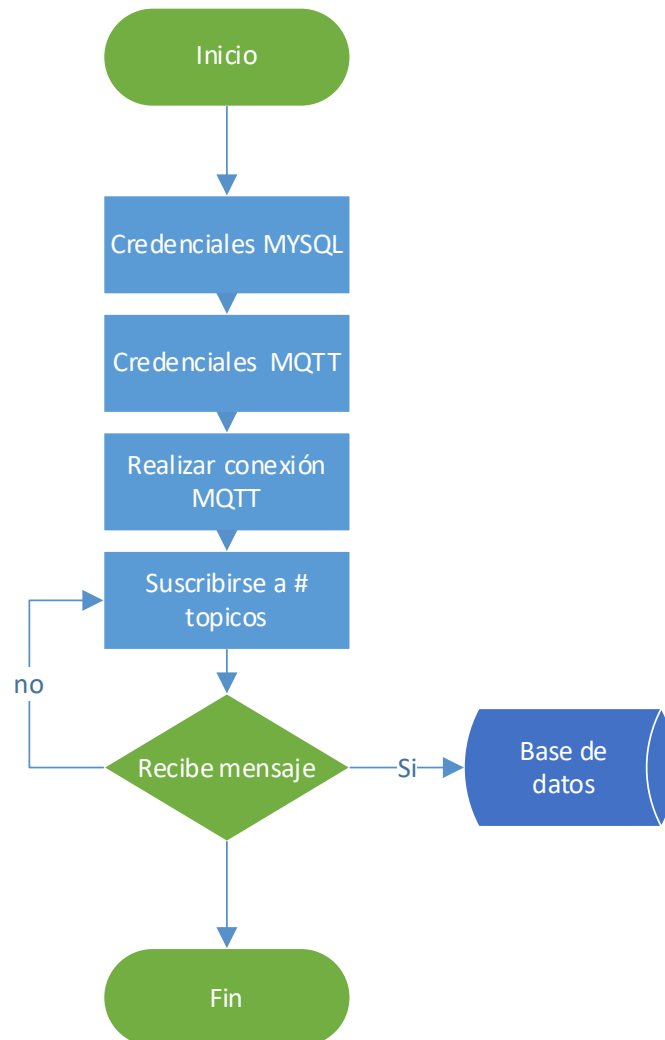


Figura 44. Diagrama de funcionamiento del programa “nodejs”.

El archivo que lleva el programa tiene por título “index.js”, el cual fue integrado a Atom, para su edición bajo el protocolo FTP. En parte se declaró el funcionamiento que lleva a cabo: Se declaran las credenciales MySQL para la base de datos y las credenciales MQTT para su conexión a “web socket” del bróker MQTT, se verifica una conexión MQTT para proceder a hacer el bucle infinito que le permite suscribirse a los tópicos de interés, en este caso es “values” el cual va almacenado en la base de datos en la tabla de nombre “data”.

4.16 Firmware del ESP32

Para poder trabajar con este dispositivo se hizo uso de la herramienta VSCODE como IDE de programación, la cual se le instalo dentro la extensión y herramienta PLATFORMIO. Esta nos permitió hacer modificaciones, compilar y crear código para que el dispositivo pueda ejecutarlo. ESP32 funciona bajo la librería Arduino, la cual fue agregada, así como las librerías correspondientes.

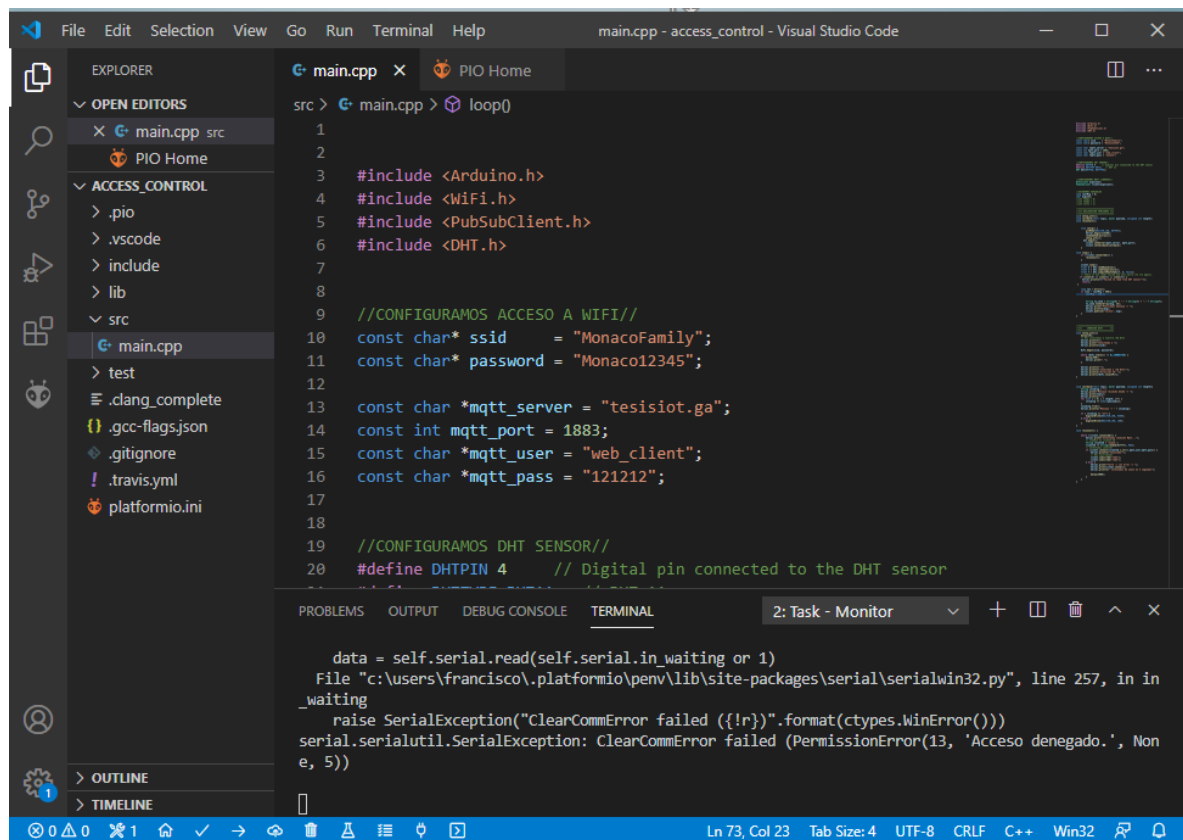


Figura 45. Herramienta VSCODE con PLATFORMIO integrado.

En la Figura 45 observamos la interfaz de desarrollo de software donde se escribió el código. Aparte de la librería Arduino antes mencionada, también se hicieron uso de las siguientes librerías:

Wifi.h: Se encargó de proporcionar una conexión eficiente a WIFI. *PubSubClient.h*: Proporcionó las herramientas adecuadas para hacer comunicación con “websocket” del “bróker” MQTT. *DHT.h*: Librería que se encargó de gestionar los datos del sensor DHT11.

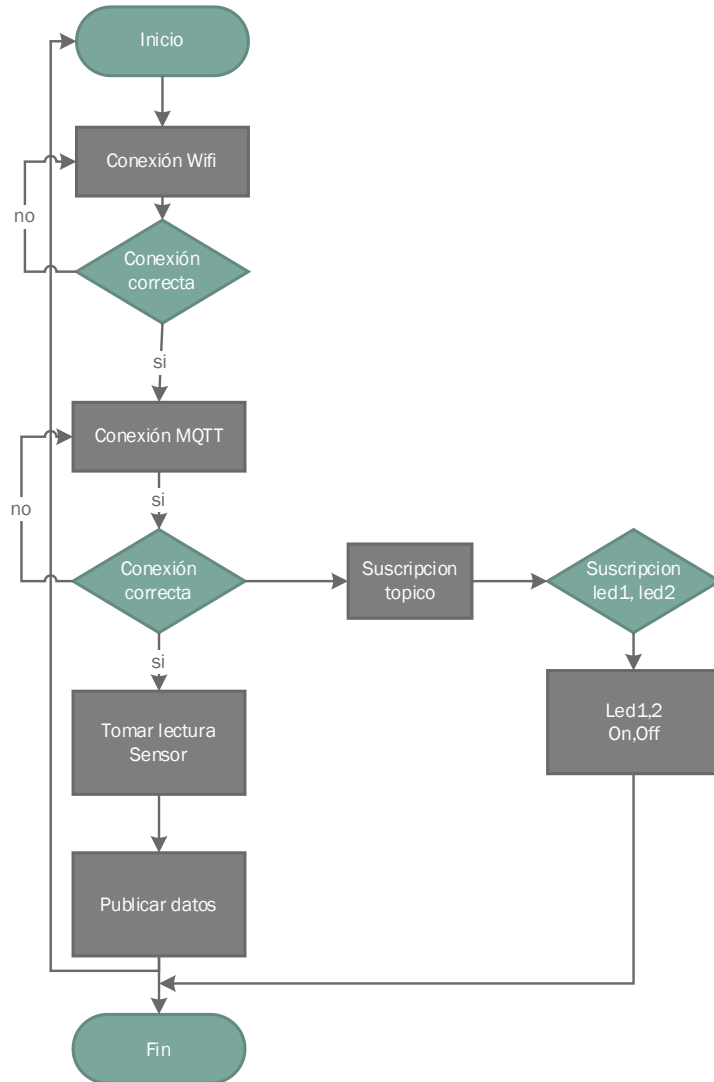


Figura 46. Esquema de funcionamiento “Firmware” ESP32.

Una vez establecidas las dos conexiones principales WIFI y MQTT, se ejecutan dos procesos, suscripción al tópico led1 y led2, los cuales están al pendiente de un cambio de estados para ser accionados por el LED integrado al dispositivo. Por otra parte, la publicación de los tres datos concatenados en un “string” para tomar un solo tópico: “values”.

CAPÍTULO V. RESULTADOS.

En este capítulo se tienen los resultados obtenidos a través de las diferentes etapas del desarrollo para lograr una arquitectura funcional de internet de las cosas. Se presentan los resultados de un sistema funcionando en conjunto de manera exitosa. Resultados del funcionamiento de la página web, comunicación MQTT, implementación en ESP32, funcionamiento de la base de datos, captura de datos de NODEJS a la base de datos.

5.1 Página web principal

Se obtuvo ingreso exitoso con seguridad SSL hacia la página web, el despliegue de la página principal es satisfactorio y de buen aspecto, el template fue bien adaptado y modificado, se adapta tanto en dispositivos móviles y computadoras. Funcionaron todos los botones.

Este paso representa el correcto funcionamiento como resultado de las configuraciones necesarias en los servicios utilizados como lo son: el panel de control VESTA, el servidor privado virtual por AWS, el servicio de dominio DNS, vinculación del protocolo FTP con el editor de código Atom.

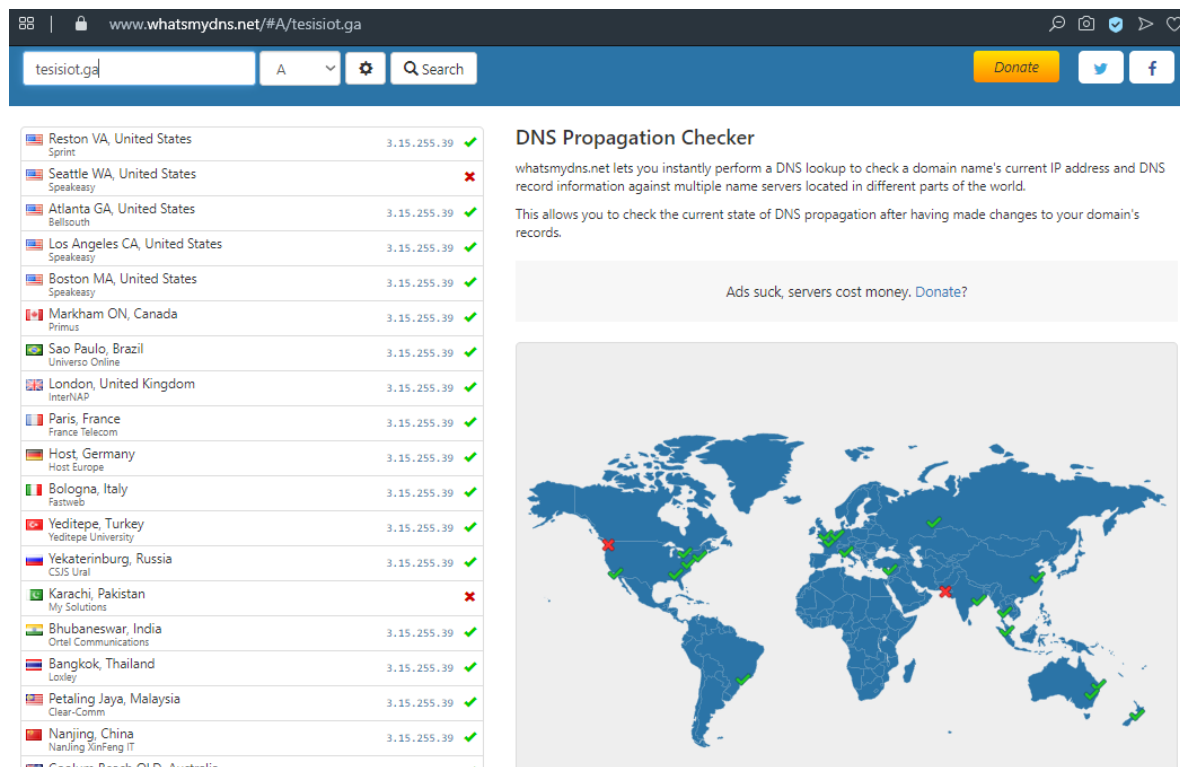


Figura 47. Propagación y alcance del DNS.

Se verificó el alcance y propagación del DNS, como se muestra en la Figura 47, se tiene cobertura mundial. Fue cuestión de minutos después del establecimiento del DNS para que se obtuviera esa cobertura. El servidor privado virtual tiene como ubicación el estado de Ohio, Estados Unidos. “Whatsmydns.net” fue la herramienta en línea que nos ayudó con esta información.

5.1.1 Control de usuarios

Compone el funcionamiento de los archivos “login.php” y “register.php” junto con la conexión trabajo en conjunto con la base de datos.

5.1.1.1 Funcionamiento “Login.php”

En “login.php” se configura de igual manera para establecer con conexión a la base de datos, este archivo se encarga de verificar y comprar los datos ingresados con los de la base de datos y así se logra dar el acceso verificado.

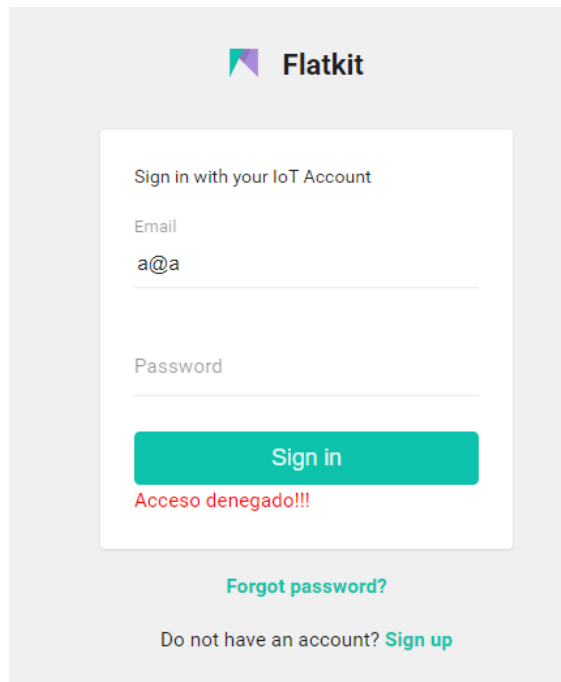


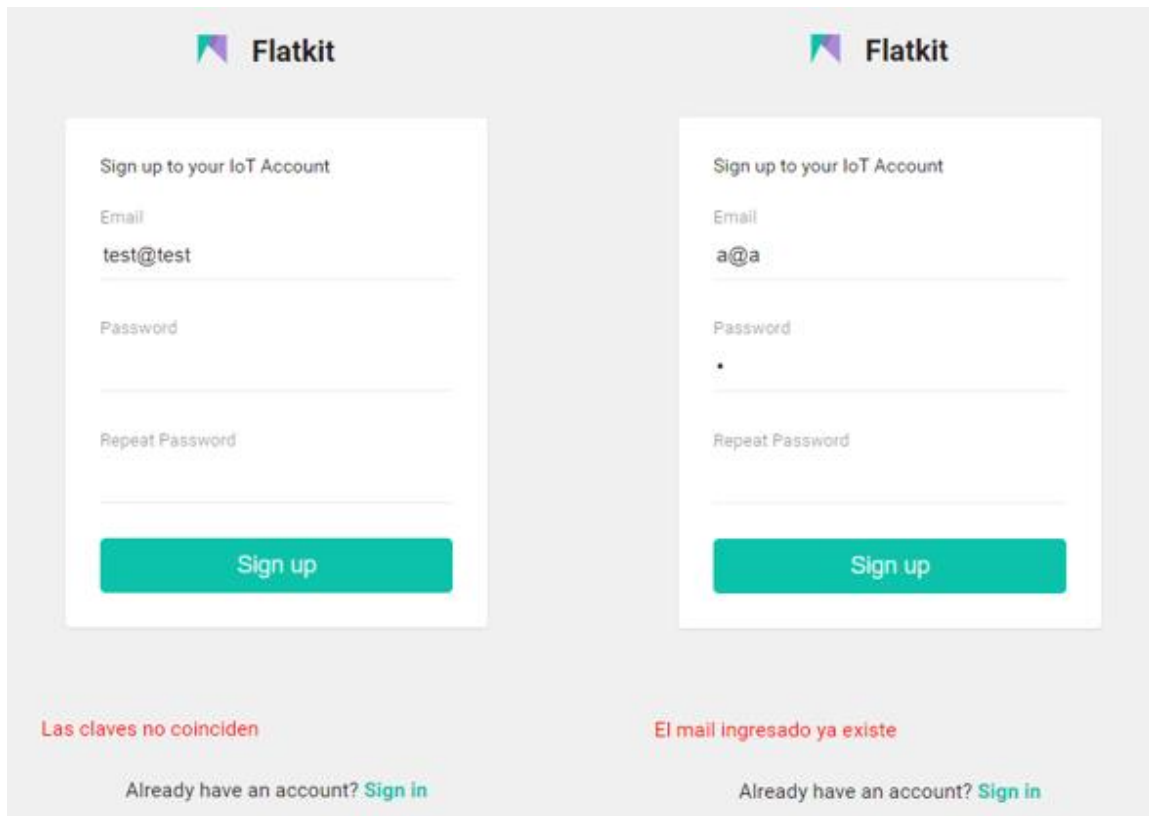
Figura 48. Acceso denegado a usuario en “Log In”.

En la Figura 48 vemos cómo responde la página ante la discrepancia entre la información ingresada en la interfaz y la base de datos una vez accionado el botón “Sign in”. ¡Se despliega el mensaje “Acceso denegado!” como resultado de dicha acción, de no ser así, se redirigirá con éxito a la pantalla principal de “dashboard” del usuario.

En la parte inferior se muestra el link que redirecciona al registro de nueva cuenta escrito en letra verde. Sin embargo, cabe resaltar que el botón “¿Forgot password?” quedó disfuncional al momento de este proyecto.

5.1.1.2 Funcionamiento “Register.php”

Al presionar el botón de “Sign Up”, se manda el mensaje: “Usuario creado correctamente, ingrese haciendo clic aquí”. Los datos quedan guardados en la base de datos automáticamente y se pudo proceder a hacer “click” en el texto en la parte inferior “Sign In”, el cual te redirige a la pantalla de “log in” que está respaldada en “login.php”. Al hacer pruebas se verificó que al ingresar una contraseña diferente en la casilla “Repeat Password” estuviera funcionando, esta se rechaza y vuelve a posicionarte si necesidad de volver a escribir el email



The image displays two side-by-side screenshots of the Flatkit registration form, titled "Sign up to your IoT Account". Both forms have fields for "Email", "Password", and "Repeat Password", and a teal "Sign up" button at the bottom. The left form shows the email "test@test" and has a red error message below it: "Las claves no coinciden" (Passwords do not match). The right form shows the email "a@a" and has a red error message below it: "El mail ingresado ya existe" (The email entered already exists). Both forms also include a link at the bottom that says "Already have an account? Sign in".

Figura 49. Errores al crear cuenta de usuario en “register.php”.

La Figura 49 ilustra dos eventos: las contraseñas ingresadas en la casilla “password” no son iguales, dando como mensaje “Las claves no coinciden” en letra roja. Y el correo electrónico ingresado en la casilla “email” ya está registrado en la base de datos dando como mensaje en letra roja también “El mail ingresado ya existe”.

5.2 Conexión entre MQTT y Pagina Web.

Se obtuvo una comunicación exitosa entre el panel de control de EMQ con el debido puerto de seguridad, esto se observó en tiempo real gracias a la herramienta de los navegadores web llamada “Inspeccionar elemento” la cual abre una consola de visualización de acciones en HTML.

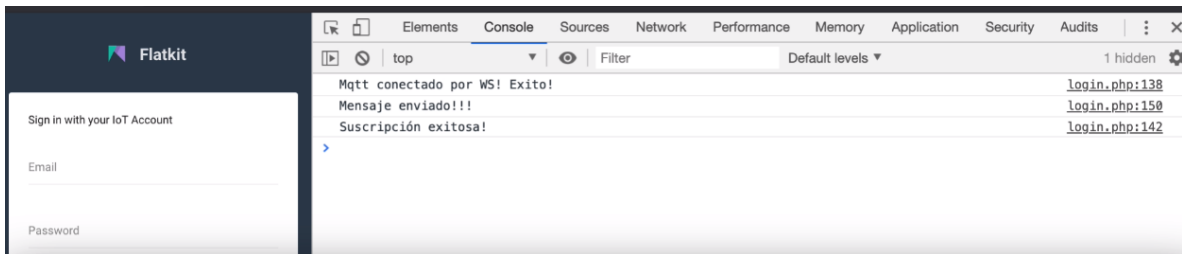


Figura 50. Consola del navegador web Opera.

Dado a la configuración del código en el archivo login.php. al establecer conexión, se mandó a dar mensaje para notificar esta acción, sin embargo, eso solo fue para dar una señal de funcionamiento para el desarrollador. Podemos ver que se imprimió en consola “¡Mensaje enviado”, “Suscripción exitosa” y “MQTT conectado por WS Éxito!”. Y también podemos observar en que líneas del archivo se están llevando a cabo las ejecuciones, en este caso fueron la 150, 142, 138, respectivamente.

5.3 Publicación de mensajes y suscripción MQTT entre EMQ y pagina Web.

En esta parte se puso a prueba el funcionamiento puro del bróker ya que se experimentaron dos eventos, publicación y suscripción bajo un tópico en específico. Estas pruebas fueron dadas sin la integración del ESP32 y el sensor.

5.3.1 Suscripción

Se configuro el código para que el panel de control EMQ fuera el publicador, se mandaron tres valores en cadena pudiendo simular cualquier parámetro. En “dashboard.php” se configuro el tópico “values” para recibir estos tres parámetros.

```
function process_msg(topic, message){
  // ej: "10,11,12"
  if (topic == "values"){
    var msg = message.toString();
    var sp = msg.split(",");
    var temp1 = sp[0];
    var temp2 = sp[1];
    var volts = sp[2];
    update_values(temp1,temp2,volts);
  }
}
```

Figura 51. “Dashboard.php” modo suscripción editado en Atom.

Estos variables de nombre temp1, temp2, volts, tienen interacción directa con los parámetros de la interfaz donde se visualizan los supuestos valores de los sensores.

The screenshot shows the EMQ Messages interface with the following configuration:

Topic	Messages	QoS	Retained	Action
values	31,21,12	1	<input type="checkbox"/> Retained	<button>send</button>

Figura 52. Valores configurados de prueba en EMQ.

Como ya se mencionó anteriormente en “topic” se puso el nombre del tópico al que la interfaz web esta suscrito y haciendo “click” en el botón “send” se mandan los valores. Se ajusta la misma calidad de servicio que se configuró en el archivo PHP, este valor quedó en “QoS” = 1.

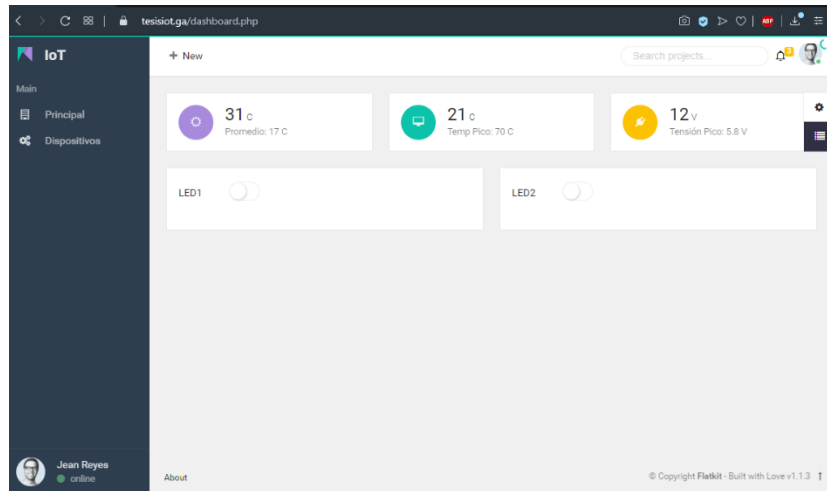


Figura 53. Recibimiento de valores en “dashboard” de página web.

Como se observa en la Figura 53 se desplegaron los valores que se dieron en la plataforma EMQ, respectivamente el 31, 21 y 12.

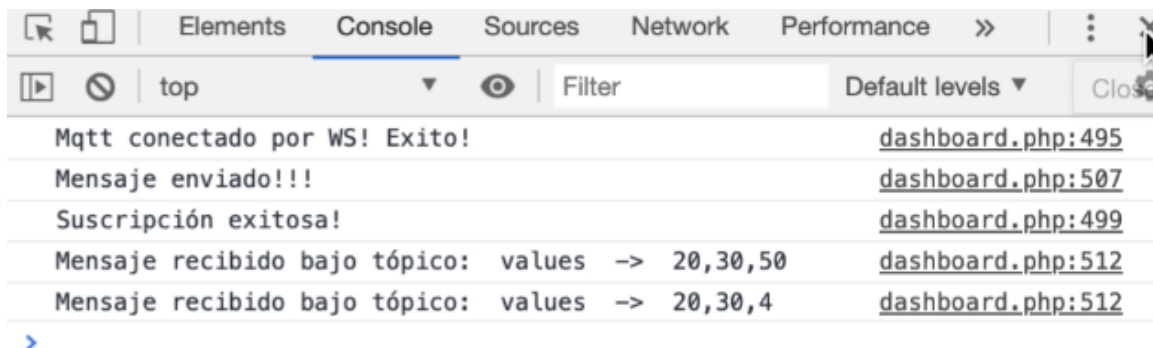


Figura 54. Respuesta de consola ante los mensajes recibidos.

Se configuró la consola para hacer notificación de funcionamiento, pero en esta vez se mandaron nuevos valores aleatorios para comprobar el correcto desempeño.

Se logró establecer completa suscripción con los parámetros correspondientes bajo el tópico específico. Lo que demostró un verdadero éxito.

5.3.2 Publicación

Para la prueba de este evento se estableció a la plataforma EMQ como suscriptor del tópico y a el “dashboard” de la página web como Publicador. Se hicieron pruebas con los dos botones simulando la acción de dos actuadores. Configurando el tópico en “dashboard.php” para que trabajara con “led1”, así como también se hizo lo mismo con “led2”

```
function process_led1(){
  if ($('#input_led1').is(":checked")){
    console.log("Encendido");

    client.publish('led1', 'on', (error) => {
      console.log(error || 'Mensaje enviado!!!')
    })
  }else{
    console.log("Apagado");
    client.publish('led1', 'off', (error) => {
      console.log(error || 'Mensaje enviado!!!')
    })
  }
}
```

Figura 55. “Dashboard.php” modo publicación editado en Atom.

De igual manera se hizo el código para process_led2 con casi los mismos parámetros.

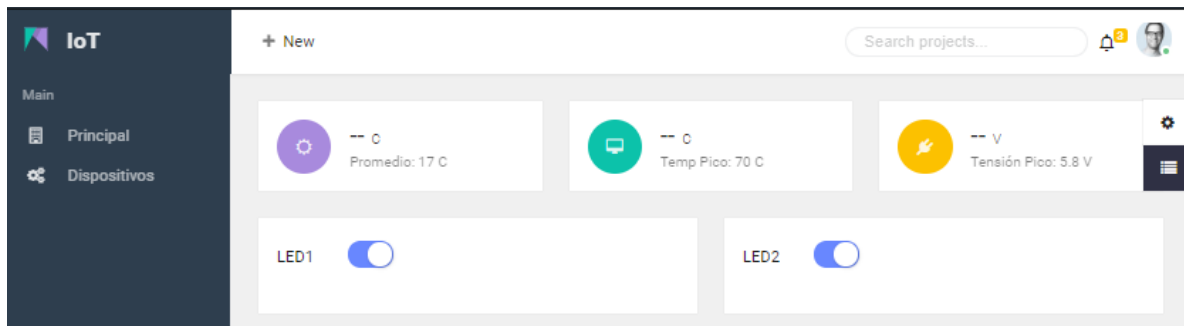


Figura 56. “Dashboard” accionando los dos interruptores.

Se aprecia en la Figura 56 como los dos interruptores LED1 y LED2 se cambiaron a estado activado lo que hace mandar la seña o mensaje MQTT a través del tópico “led1” y “led2”.

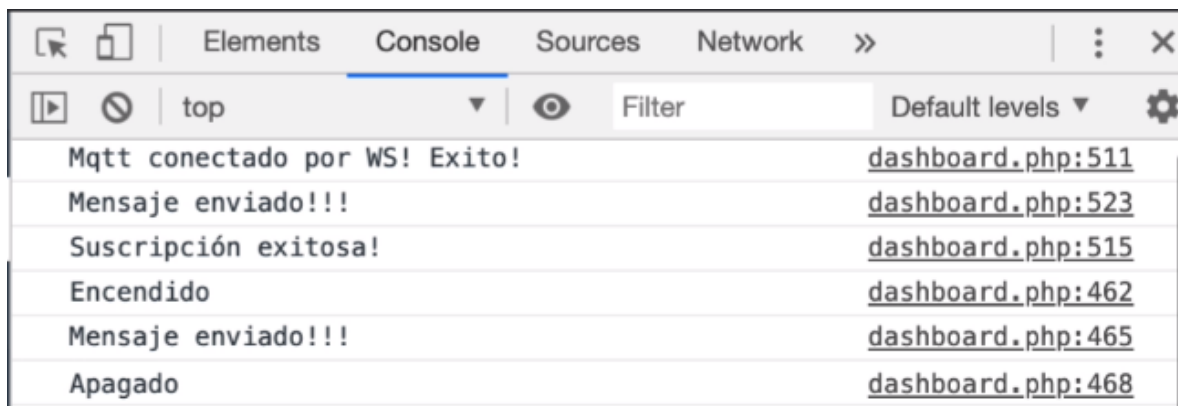


Figura 57. Respuesta de consola ante los mensajes publicados.

Cuando se tiene un cambio de estado en los interruptores la consola te notifica el manifiesto, como mensaje enviado y el estado tanto apagado como encendido.

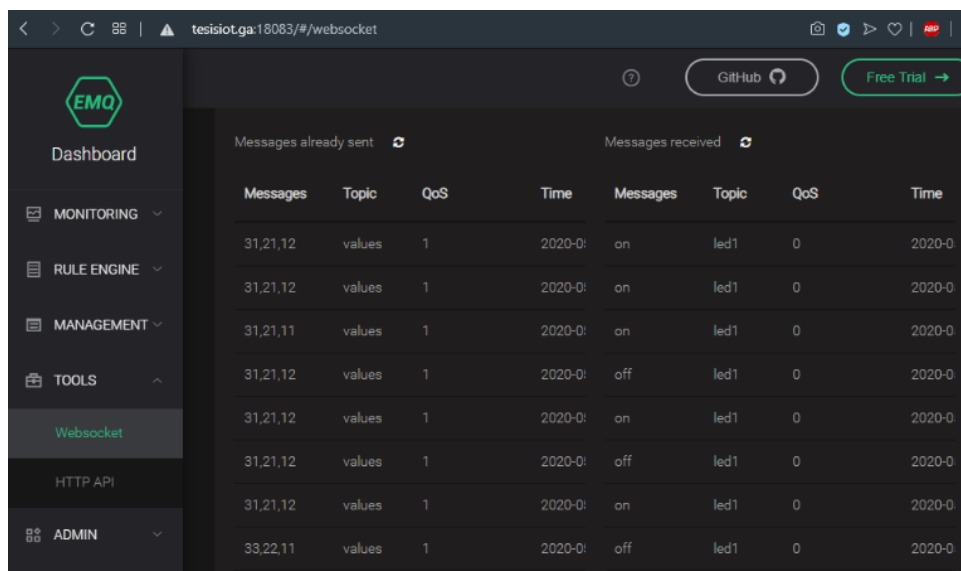
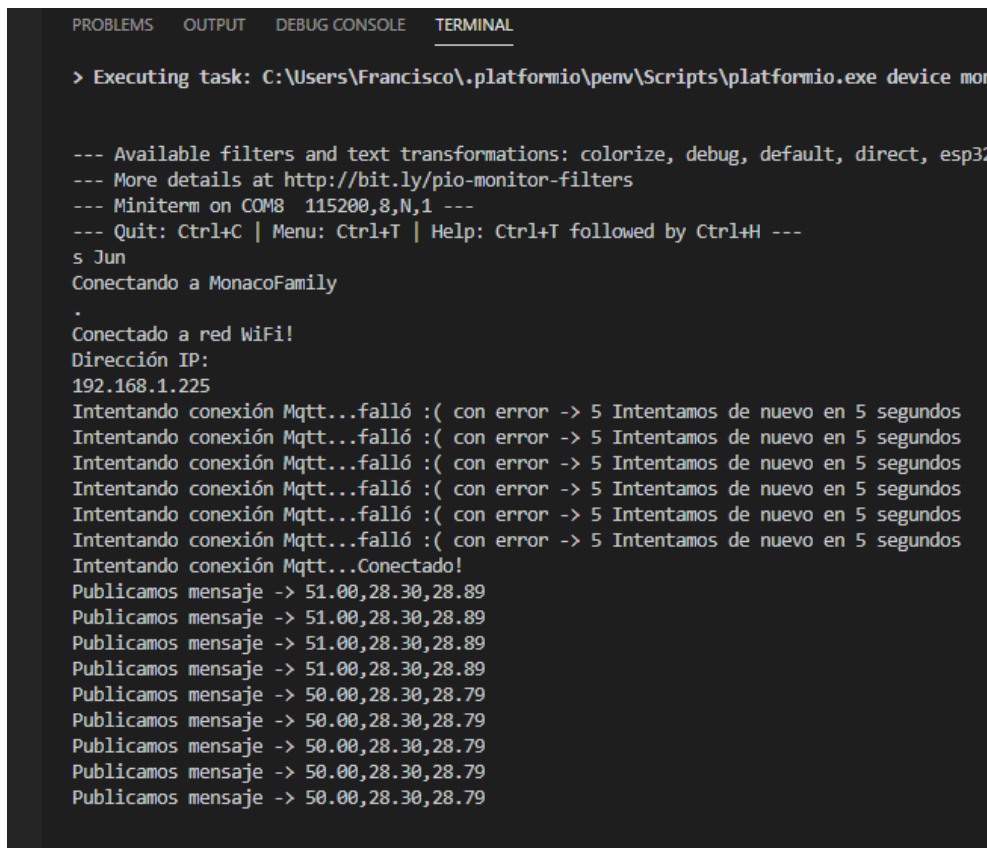


Figura 58. EMQ “dashboard” en modo suscripción.

En la parte derecha de la Figura 58 se muestra el historial de mensajes recibidos estando en modo de suscripción, con un QoS de cero, vemos como en Mensaje aparece tanto “on” y “off” y la fecha de acción. Es de llamar de atención el tiempo de respuesta que se tiene ya que es inmediato esto es un muy buen resultado métrico.

5.4 ESP32

Al integrarse el sensor DHT11, se obtuvo el funcionamiento deseado. Al momento de crear el código, se le añadieron ciertos indicadores que demostraban el correcto funcionamiento de cada proceso, esto se hizo ejecutando de manera serial mensajes que puedan ser leídos en una terminal o consola establecida una velocidad de 115200, en este caso el editor tiene integrada dicha consola.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

> Executing task: C:\Users\Francisco\.platformio\penv\Scripts\platformio.exe device mon

--- Available filters and text transformations: colorize, debug, default, direct, esp32
--- More details at http://bit.ly/pio-monitor-filters
--- Miniterm on COM8  115200,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
s Jun
Conectando a MonacoFamily
.
Conectado a red WiFi!
Dirección IP:
192.168.1.225
Intentando conexión Mqtt...falló :( con error -> 5 Intentamos de nuevo en 5 segundos
Intentando conexión Mqtt...falló :( con error -> 5 Intentamos de nuevo en 5 segundos
Intentando conexión Mqtt...falló :( con error -> 5 Intentamos de nuevo en 5 segundos
Intentando conexión Mqtt...falló :( con error -> 5 Intentamos de nuevo en 5 segundos
Intentando conexión Mqtt...falló :( con error -> 5 Intentamos de nuevo en 5 segundos
Intentando conexión Mqtt...Conectado!
Publicamos mensaje -> 51.00,28.30,28.89
Publicamos mensaje -> 51.00,28.30,28.89
Publicamos mensaje -> 51.00,28.30,28.89
Publicamos mensaje -> 51.00,28.30,28.89
Publicamos mensaje -> 50.00,28.30,28.79
Publicamos mensaje -> 50.00,28.30,28.79
Publicamos mensaje -> 50.00,28.30,28.79
Publicamos mensaje -> 50.00,28.30,28.79
Publicamos mensaje -> 50.00,28.30,28.79
```

Figura 59. Consola de monitoreo serial hacia ESP32.

En la Figura 59 se hizo notorio cada paso del proceso que siguió y ejecutó el dispositivo. Se estableció conexión exitosa en WIFI y MQTT cada vez que fue probado con las correctas credenciales, mientras que no se logren dichos procesos, el programa lo obliga a volverse a conectar hasta que pueda comunicarse. Una vez conectado, el dispositivo publica los tres datos leídos por el sensor y al mismo tiempo está siendo suscrito hacia el tópico que hace accionar el actuador.

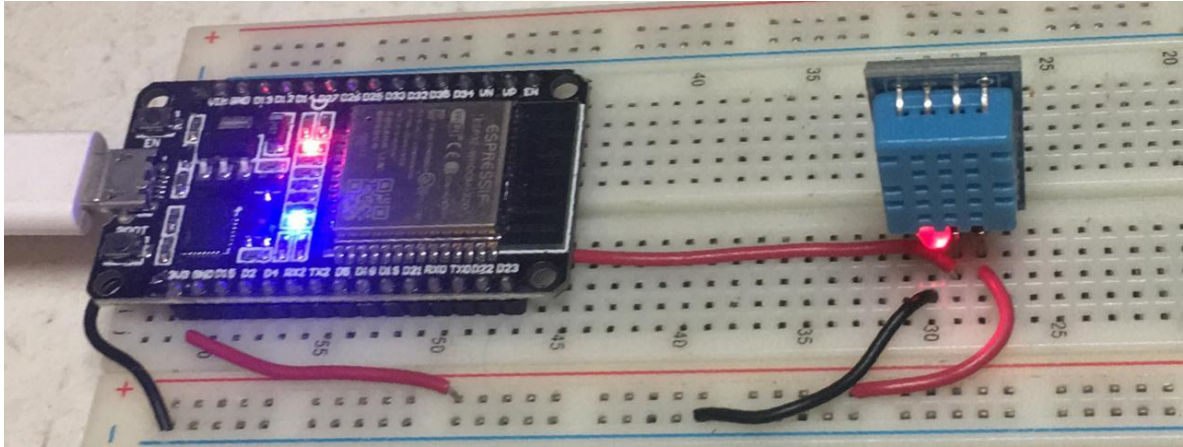


Figura 60. Implementación Esp32 con DHT11.

Se implementó la conexión del módulo sensor de temperatura y humedad hacia el módulo ESP32 en una tableta de prototipado (“protoboard”). Como se presenta en la Figura 60. Se utilizó el DHTPIN4 del ESP32 como entrada de información. Ambos módulos están alimentados por USB, el cual fue conectado al ordenador utilizando 3.3 volts.

El led rojo de ambos módulos significa que está alimentado, el led azul está indicando el cambio de estado en la casilla de LED1 o de LED2 en el “dashboard”. Este resultado representa el correcto funcionamiento del sistema que involucra la conexión MQTT entre la página web y el ESP32 conectado a WIFI.

5.5 Base de datos

Los resultados de la base de datos se demuestran en dos procesos que hacen consulta a la misma. Página web y “nodejs”

5.5.1 Página web

Dentro de la página web gestionada por PHP, el cual es el encargado de ejecutar las instrucciones en SQL para guardar información o pedirla, se obtuvieron los siguientes resultados de manera satisfactoria.

users_id	users_date	users_email	users_password
4	2020-07-18 11:38:14	test@test	51abb9636078defbf888d6
3	2020-07-14 05:35:50	test@test.com	a94a8fe5ccb19ba61c4c08
2	2020-07-12 17:14:58	antoniopalos03@gmail.com	0b55b2a9c18d6191bfdaa1
1	2020-07-10 02:59:07	a@a	48058e0c99bf7d689ce71c

Figura 61. Datos generados de las cuentas registradas en “register.php”

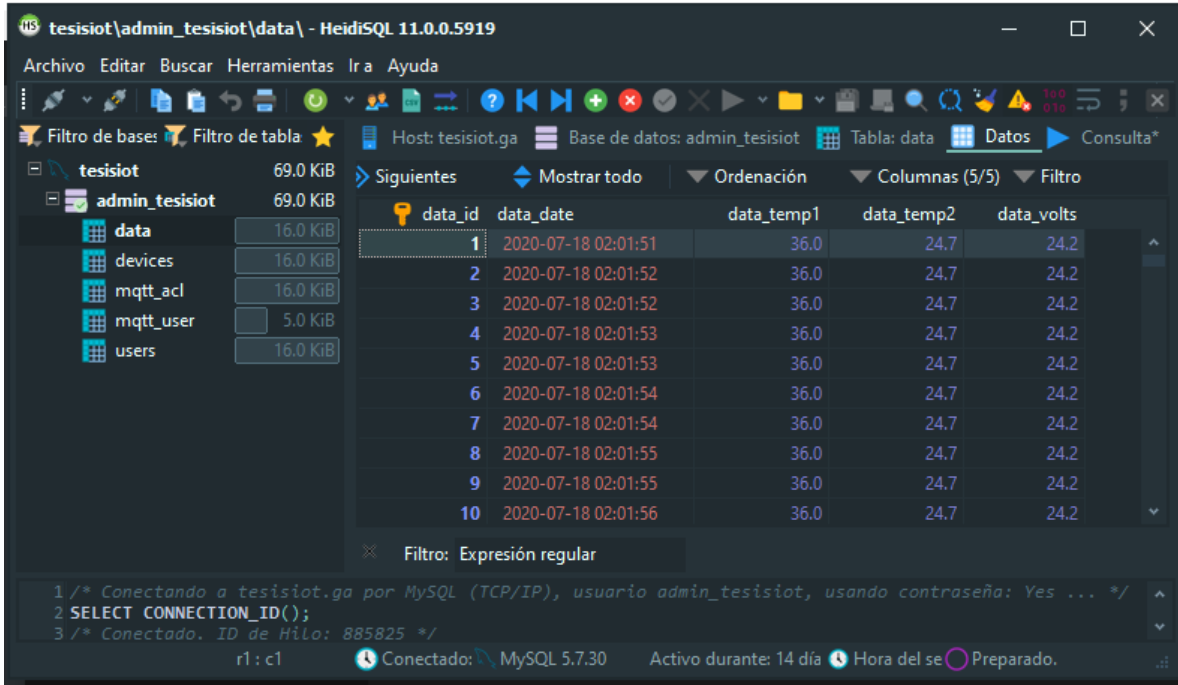
Se generaron cuatro cuentas de prueba con los datos ilustrados en la anterior imagen, se notó la correcta inserción de los tipos de datos establecidos. Cabe notar que la contraseña esta encriptada por seguridad.

devices_id	devices_date	devices_alias	devices_serie	devices_user_id
1	2020-07-18 00:04:58	Sensor1	0123456789	1
2	2020-07-24 07:38:01	Sensor2	1123456789	1

Figura 62. Datos generados de “devices.php”.

Se generaron dos dispositivos de prueba con su respectivo número de serie y alias, y como parámetro de prueba, se establecieron con el mismo identificador de usuario intencionalmente, esto significa que todas las cuentas pueden ver los mismos datos en el “dashboard” en tiempo real.

5.5.2 NODEJS



tesisiot\admin_tesisiot\data - HeidiSQL 11.0.0.5919

Archivo Editar Buscar Herramientas Ir a Ayuda

Filtro de bases: Filtro de tabla Host: tesisiot.ga Base de datos: admin_tesisiot Tabla: data Datos Consulta*

tesisiot 69.0 KiB

- admin_tesisiot 69.0 KiB
 - data 16.0 KiB
 - devices 16.0 KiB
 - mqtt_acl 16.0 KiB
 - mqtt_user 5.0 KiB
 - users 16.0 KiB

Siguientes Mostrar todo Ordenación Columnas (5/5) Filtro

data_id	data_date	data_temp1	data_temp2	data_volts
1	2020-07-18 02:01:51	36.0	24.7	24.2
2	2020-07-18 02:01:52	36.0	24.7	24.2
3	2020-07-18 02:01:52	36.0	24.7	24.2
4	2020-07-18 02:01:53	36.0	24.7	24.2
5	2020-07-18 02:01:53	36.0	24.7	24.2
6	2020-07-18 02:01:54	36.0	24.7	24.2
7	2020-07-18 02:01:54	36.0	24.7	24.2
8	2020-07-18 02:01:55	36.0	24.7	24.2
9	2020-07-18 02:01:55	36.0	24.7	24.2
10	2020-07-18 02:01:56	36.0	24.7	24.2

Filtro: Expresión regular

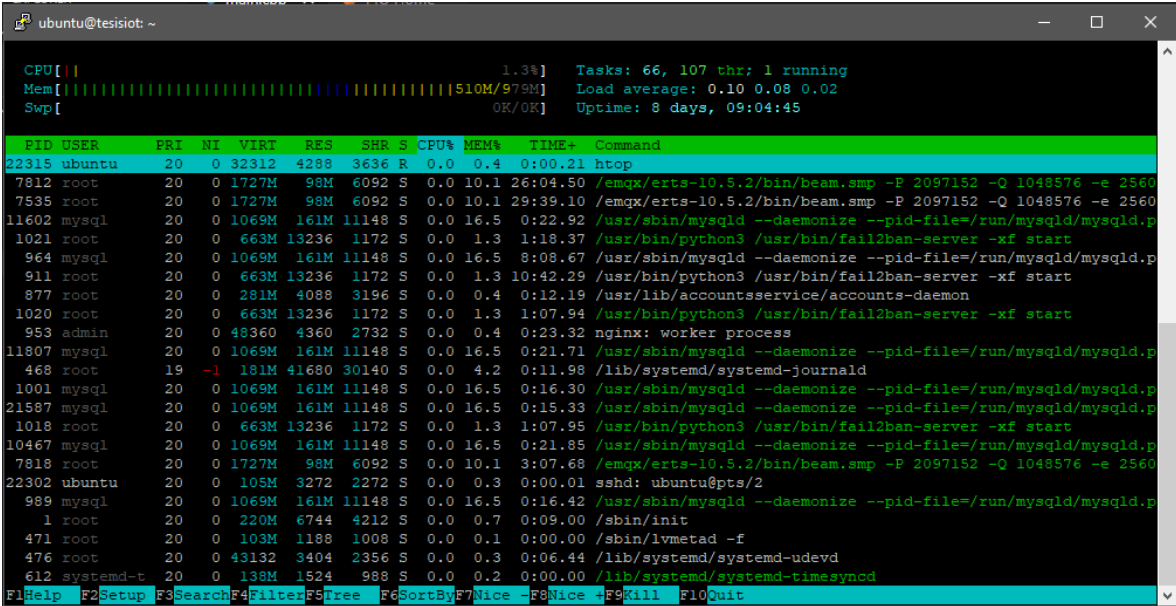
```
1 /* Conectando a tesisiot.ga por MySQL (TCP/IP), usuario admin_tesisiot, usando contraseña: Yes ... */
2 SELECT CONNECTION_ID();
3 /* Conectado. ID de Hilo: 885825 */
```

r1: c1 Conectado: MySQL 5.7.30 Activo durante: 14 día Hora del se Preparado.

Figura 63. Datos generados por el ESP32.

El dispositivo ESP32 brindó la información para ser enviada a través de MQTT donde NODEJS es cliente suscrito al tópic “values” el cual absorbió la información para después ser almacenada en la base de datos, y en la figura anterior nos hace evidencia del resultado obtenido. Se muestra una lista bastante larga resultado del tiempo de operación útil del ESP32. Se desplegaron los tres parámetros a medir junto con su respectiva fecha de captura.

5.6 Consumo de recursos en la máquina del VPS.



```
ubuntu@tesiot: ~  
CPU[ 1.3%] Tasks: 66, 107 thr; 1 running  
Mem[510M/979M] Load average: 0.10 0.08 0.02  
Swp[OK/OK] Uptime: 8 days, 09:04:45  
  
PID USER      PRI  NI  VIRT   RES   SHR  S CPU% MEM%   TIME+  Command  
22315 ubuntu    20   0 32312  4288  3636  R  0.0  0.4   0:00.21 httpd  
7812 root      20   0 1727M   98M   6092  S  0.0 10.1 26:04.50 /emqx/erts-10.5.2/bin/beam.smp -P 2097152 -Q 1048576 -e 2560  
7535 root      20   0 1727M   98M   6092  S  0.0 10.1 29:39.10 /emqx/erts-10.5.2/bin/beam.smp -P 2097152 -Q 1048576 -e 2560  
11602 mysql     20   0 1069M  161M  11148  S  0.0 16.5  0:22.92 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p  
1021 root      20   0 663M  13236  1172  S  0.0  1.3  1:18.37 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
964 mysql     20   0 1069M  161M  11148  S  0.0 16.5  8:08.67 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p  
911 root      20   0 663M  13236  1172  S  0.0  1.3 10:42.29 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
877 root      20   0 281M   4088  3196  S  0.0  0.4  0:12.19 /usr/lib/accountsservice/accounts-daemon  
1020 root      20   0 663M  13236  1172  S  0.0  1.3  1:07.94 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
953 admin     20   0 48360  4360  2732  S  0.0  0.4  0:23.32 nginx: worker process  
11807 mysql     20   0 1069M  161M  11148  S  0.0 16.5  0:21.71 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p  
468 root      19  -1 181M  41680 30140  S  0.0  4.2  0:11.98 /lib/systemd/systemd-journald  
1001 mysql     20   0 1069M  161M  11148  S  0.0 16.5  0:16.30 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p  
21587 mysql     20   0 1069M  161M  11148  S  0.0 16.5  0:15.33 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p  
1018 root      20   0 663M  13236  1172  S  0.0  1.3  1:07.95 /usr/bin/python3 /usr/bin/fail2ban-server -xf start  
10467 mysql     20   0 1069M  161M  11148  S  0.0 16.5  0:21.85 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p  
7818 root      20   0 1727M   98M   6092  S  0.0 10.1  3:07.68 /emqx/erts-10.5.2/bin/beam.smp -P 2097152 -Q 1048576 -e 2560  
22302 ubuntu    20   0 105M   3272  2272  S  0.0  0.3  0:00.01 sshd: ubuntu@pts/2  
989 mysql     20   0 1069M  161M  11148  S  0.0 16.5  0:16.42 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.p  
1 root       20   0 220M   6744  4212  S  0.0  0.7  0:09.00 /sbin/init  
471 root      20   0 103M   1188  1008  S  0.0  0.1  0:00.00 /sbin/lvmstat -f  
476 root      20   0 43132  3404  2356  S  0.0  0.3  0:06.44 /lib/systemd/systemd-udev  
612 systemd-t 20   0 138M   1524   988  S  0.0  0.2  0:00.00 /lib/systemd/systemd-timesyncd  
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit
```

Figura 64. Monitoreo de recursos del VPS.

En la Figura 64 se capturó al momento en que estaba funcionando todo el sistema en conjunto con el ESP32 transmitiendo datos en tiempo real. Se obtuvo un pico de 1.3% de uso en el CPU, consumiendo la mitad de la memoria en promedio. Se obtuvo la lista de tareas ejecutadas, lo cual nos da otro indicativo visual de funcionalidad.

5.7 Prueba de seguridad

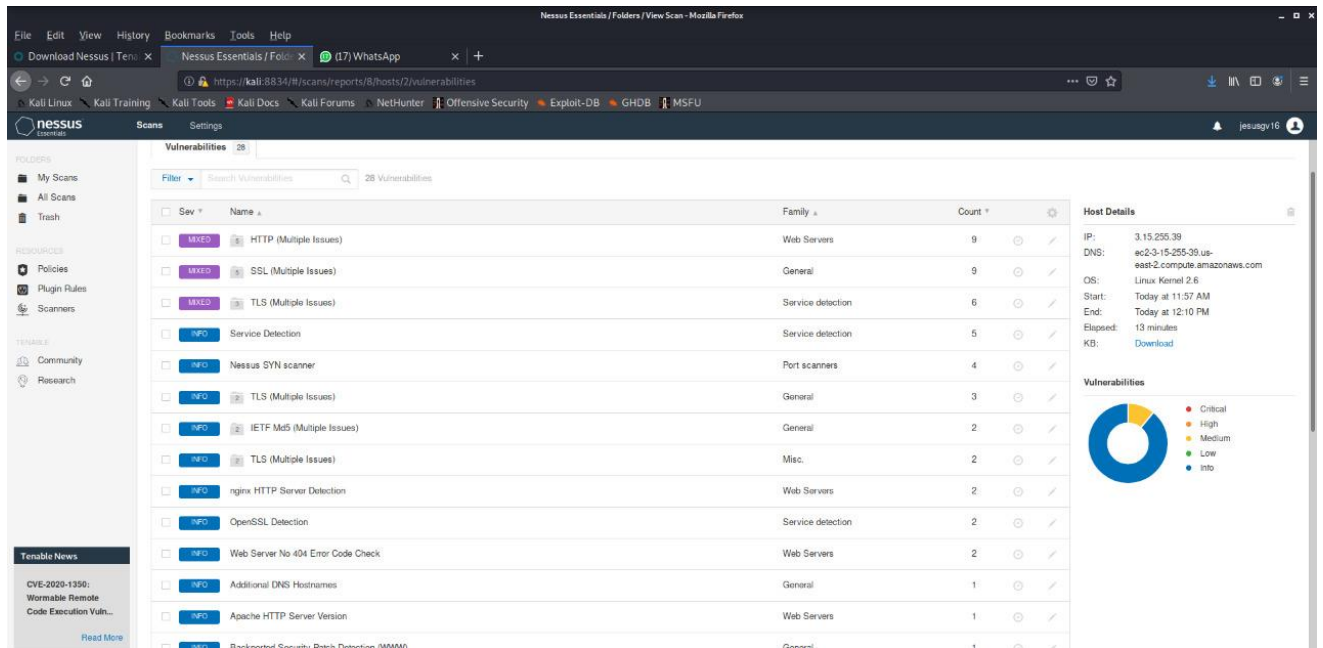


Figura 65. Prueba de seguridad en Nessus.

Se hizo una prueba de seguridad hacia el dominio *tesisiot.ga*, como lo muestra la Figura 65. El resultado nos demostró las debilidades en seguridad de la página web, y se detectó una carencia en la certificación de seguridad SSL, este servicio nos dio una sugerencia de dicha carencia en cuanto a su corrección, la cual es la que presenta la Figura 66.

Output

```
To disable these methods, add the following lines for each virtual host in your configuration file :
```

```
RewriteEngine on
RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
RewriteRule .* - [F]
```

```
Alternatively, note that Apache versions 1.3.34, 2.0.55, and 2.2 support disabling the TRACE method natively via the 'TraceEnable' directive.
```

[more...](#)

Figura 66. Diagnóstico de vulnerabilidad en seguridad SSL

Para solucionar esto, es necesario modificar el certificado SSL que está escrito en VESTA.

CAPÍTULO VI. CONCLUSIÓN

Como conclusión, se puede mencionar que la ejecución de la metodología planteada abrió las puertas a un desarrollo bien estructurado. Se atribuye el éxito en gran parte a la investigación plena de las mejores opciones y herramientas disponibles para lograr llevar a cabo la conclusión del proyecto.

Se cumplió el principal objetivo de desarrollar una propuesta para una arquitectura IoT que sea aplicable a 5G para aplicaciones de “M. Health”. Adicionalmente, se logró abarcar gran parte en el ámbito de red y conexión y cubrir los retos que se presentan en la actualidad.

Los parámetros en lo referente a censado fueron básicos y adaptados para una aplicación realista, sin embargo, esta arquitectura desarrollada da punto de partida a la implementación en dispositivos embebidos que puedan establecer conexión a una red y así lograr a gran manera el objetivo del Internet de las cosas.

La arquitectura es completa y robusta, lo que le permite estar preparada para altas demandas, teniendo la virtud de usar lo mejor de cada tecnología y servicios existentes actualmente, haciéndola muy eficiente y rentable. A pesar de que se utilizaron herramientas gratuitas y de código abierto, es más que suficiente para cumplir una alta demanda en cualquier escenario y no solo para aplicaciones de “M.Health”. Cabe mencionar que se

puede usar en otras aplicaciones si se desea en cualquier escenario realista ya que cuenta con la capacidad de comunicar miles de dispositivos de manera eficaz y todo el sistema es totalmente escalable, si así se requiere. Se pensó en el usuario y cliente de manera que el diseño sea amigable y al mismo tiempo útil.

Una característica muy importante se dio en la seguridad que la arquitectura presentó, ya que fue muy completa de acuerdo a las pruebas realizadas. Se trató de tener la mejor seguridad posible con certificados de encriptación respetados. Se logró dar la confianza al usuario o empresa potencial al saber que cuenta con autenticación en muchos aspectos. Es importante resaltar lo anterior ya que Internet de las cosas se vio muy vulnerable tiempo atrás, y esto le impedía avanzar. Gracias a los avances que se tienen en la actualidad, se presentó la oportunidad de aplicarlos en la arquitectura.

BIBLIOGRAFÍA

- [1] H. Rahimi, «A Novel IoT Architecture based on 5G-IoT and Next Generation Technologies,» *IEEE*, 2019.
- [2] P.Carlos, «Open IoT Architecture for Continuous Patient Monitoring in Emergency Wards,» *MDPI AG*, 2019.
- [3] P. Carlos, «Open IoT Architecture for Continuous Patient Monitoring in Emergency Wards,» *MDPI*, p. 15, 2019.
- [4] D. Hanes, *IoT Fundamentals*, Cisco Press, 2017.
- [5] A. McEwen y H. Cassimally, *Internet de las cosas la tecnologia que todo lo conecta*, Anaya multimedia, 2014.
- [6] H. M. Hasan, *IoT Protocols for Health Care Systems: A Comparative Study*, Iraq: H.M. Hasan, 2018.
- [7] D. K. Yan, «Network Protocols,» de *Javvin Technologies Inc.*, Saratoga CA, 2005, p. 268.
- [8] B. J. S. P. H. M. A.-M. Godfrey A. Akpakwu, «A survey on 5G networks for the internet of things communication technologies and challenges,» *IEEE*, p. 28, 2017.
- [9] P. S. a. S. R. Sarangi, «Internet of Things: Architectures, Protocols, and Applications,» *Hindawi*, p. 25, 2017.
- [10] A. S. A. F. N. P. Robert S .H. Istepanian, «INTERNET OF M-HEALTH THINGS “m-IOT”,» *IEEE*, 2012.
- [11] A. G. E. R. M. V. ., J. C. Francisco Valenzuela, «An IoT-Based Glucose Monitoring Algorithm to Prevent Diabetes Complications,» *MDPI*, 2020.

- [12] J. Z. Z. L. A. C. F. P. Ming Taoa, «Multi-layer Cloud Architectural Model and Ontology-based Security Service Framework for IoT-based Smart Homes,» *Future Generation Computer Systems*, p. 23, 2016.
- [13] R. S. Maanak Gupta, «Authorization Framework for Secure Cloud Assisted Connected Cars and Vehicular Internet of Things,» *SACMAT*, 2018.
- [14] R. S. Istepanian, A. Sungoor, A. Faisal y N. Philip, «INTERNET OF M-HEALTH THINGS “m-IOT”,» *IET*, 2011.
- [15] J. Flores, «National geographic,» Nationalgeographic.es, 22 Abril 2020. [En línea]. Available: https://www.nationalgeographic.com.es/ciencia/que-es-5g-y-como-nos-cambiara-vida_14449.
- [16] Huawei, «5G Network Architecture A high level perspective,» *Huawei*, 2016.
- [17] G. Americas, «LTE and 5G Technologies Enabling the Internet of Things,» *5G Americas White Paper*, vol. http://www.5gamericas.org/files/3514/8121/4832/Enabling_IoT_WP_12.8.16_FINAL.pdf 2016..
- [18] M. I. Grosser Hasenpfad, «NGMN View on 5G Architecture,» *IEEE*, p. 5, 2015.
- [19] M. Afaneh, «Bluetooth Low Energy, Broadcast, IoT Connectivity, Mesh, Point-to-Point, Range,» *Bluetooth*, 21 abril 2020. [En línea]. Available: <https://www.bluetooth.com/blog/wireless-connectivity-options-for-iot-applications-technology-comparison/>.
- [20] A. M. Dipa Soni, «A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT),» *resaerchgate*, p. 6, 2017.
- [21] «mqtt.org,» mqtt, [En línea]. Available: <http://mqtt.org/documentation>.

- [22] C. V. K. H. S. He, «Systems and methods of providing server initiated connections on a virtual private network,» *Google*.
- [23] R. E. S. Daniel J. Barrett, SSH, The Secure Shell: The Definitive Guide: The Definitive Guide, Mary Brady.
- [24] M. E. C. Hurtado, «Analysis of free SSL/TLS Certificates and their implementation as Security Mechanism in Application Servers.,» *Enfoque UTE*, 2017.
- [25] S. Bellovin, «Firewall-Friendly FTP,» *AT&T Bell Laboratories*, p. 4, 1994.
- [26] G. P. T. T. a. O. K. T. Mario Collotta, «Bluetooth 5: A Concrete Step Forward toward the IoT,» *IEEE*, p. 7, 2018.
- [27] G. C. Hillar, MQTT Essentials - A Lightweight IoT Protocol, Birmingham: Packt, 2017.
- [28] J. L. Molinero, Implementacion del protocolo MQTT'S sobre IEEE 802.15.4e en plataformas OpenMOTE, Creative Commons, 2018.
- [29] I. A. A. M. Z. b. S. A. F. R. M. A. A. Muhyi Bin Yaakop, «Bluetooth 5.0 Throughput Comparison for Internet of Thing Usability A Survey,» *IEEE*, p. 6, 2017.
- [30] B. L. E. B. A. D. Jevgenijus Toldinas, «MQTT Quality of Service versus Energy Consumption,» *IEEE*, p. 4, 2019.
- [31] S. Nicholas, «stephendnicholas.com,» Alfa (CHP Consulting), 12 Mayo 2012. [En línea]. Available: <http://stephendnicholas.com/posts/power-profiling-mqtt-vs-https>.

APÉNDICE

Archivos de página web:

https://github.com/fapa03/tesis/tree/master/tesis_webpage

Archivos de ESP32:

https://github.com/fapa03/tesis/tree/master/tesis_esp32