



**FACULTAD DE INGENIERIA, ARQUITECTURA Y
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS
TESIS**

**COMPARACIÓN DE PROTOCOLOS DE
COMUNICACIÓN EN INTERNET DE LAS COSAS,
DETERMINANDO EL NIVEL DE SEGURIDAD ANTE
ATAQUES EN DISPOSITIVOS BIOMEDICOS**

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

Autor (es):

Bach. Zeña Zeña, Edinson Omar

ORCID: <https://www.orcid.org/0000-0002-3774-1668>

Asesor:

Mg. Mejía Cabrera, Heber Iván

ORCID: <https://www.orcid.org/0000-0002-0007-0928>

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2021

APROBACIÓN DEL JURADO

COMPARACIÓN DE PROTOCOLOS DE COMUNICACIÓN EN INTERNET DE LAS COSAS, DETERMINANDO EL NIVEL DE SEGURIDAD ANTE ATAQUES EN DISPOSITIVOS BIOMEDICOS

Bach. Zeña Zeña, Edinson Omar.

Autor

Ing. Mejía Cabrera, Heber Iván.

Asesor

Ing. Tuesta Monteza, Victor Alexci.

Presidente de Jurado

Ing. Bravo Ruiz, Jaime Arturo.

Secretario de Jurado

Dedicatorias

A Dios.

Por haberme dado la fortaleza para seguir adelante, el empuje y valentía para perseguir y lograr mis objetivos, mis metas trazada.

A mi familia.

Por el apoyo constante que me brindan; a mis padres que de la gloria de Dios gozan, quienes me abrazaron y me motivaron para lograr este proyecto, a mi hermana quien me brinda tenacidad, firmeza y me impulsa a seguir adelante, siendo perseverante.

Agradecimientos

A Dios.

Por guiarme y orientarme en mi trayecto, a mi familia por ser mi soporte, mi estímulo e impulso para salir adelante, asimismo a mis docentes quienes me brindaron una excelente formación e instrucción, por último agradezco a todas las personas que coadyugaron para que pueda lograr mis metas, brindandome su apoyo.

Resumen

El presente trabajo tiene como problema general el ser atribuible a estar o no preparado para afrontar aquellos retos sustanciales y trascendentes en cuanto a la seguridad del Internet de las cosas, entorno a las tecnologías de la información, por cuanto se ha observado un crecimiento acelerado en la fabricación y utilización de dispositivos comunes conectados a internet; con respecto al problema de ingeniería se basa en la adecuada e idónea obtención y determinación del protocolo de capa de aplicación que implemente mayor seguridad ante ataques en internet de las cosas para proteger dispositivos biomédicos; siendo importante tener como método de ingeniería propuesto la comparación de los protocolos MQTT y COAP puesto que son los más utilizados en la industria de la internet de las cosas (IOT) considerando elementos esenciales tales como, la seguridad, confidencialidad y privacidad de la información, para una adecuada y eficiente seguridad que conlleve a una correcta protección de equipos biomédicos ante ataques cibernéticos que alteren el ideal funcionamiento, asimismo modifiquen o roben información, requiriéndose comparar protocolos de capa de aplicación en IOT, por lo que resulta ser una investigación cuantitativa consistente en cantidades numéricas de los indicadores los cuales están representados por porcentajes, siendo que los datos son recabados a través de mediciones representada mediante un adecuado análisis de métodos estadísticos, y a su vez continua, ya que se obtiene por medición o comparación de una unidad o patrón de medidas; los resultados obtenidos de esta investigación es que el protocolo de capa de aplicación COAP implementado en equipos biomédicos ofrece mayor seguridad ante ataques a dispositivos biomédicos conectados a internet de las cosas a comparación del protocolo MQTT, en internet de las cosas, utilizando equipos biomédicos; teniendo como conclusión que, el protocolo COAP presentó un índice de peticiones fallidas del 19.37%, mientras que MQTT arrojó 28.12%, el cual fue uno de los indicadores en estudio, por consiguiente COAP es el protocolo más seguro frente ha ataques DOS a comparación del protocolo MQTT.

Palabras clave

Internet de las cosas, protocolos, ataques, seguridad, latencia, dispositivos biomédicos y confidencialidad.

Abstract

The present work has as a general problem being attributable to being or not being prepared to face those substantial and transcendent challenges in terms of the security of the Internet of things, around information technologies, since accelerated growth has been observed in the manufacture and use of common devices connected to the internet; Regarding the engineering problem, it is based on the adequate and suitable obtaining and determination of the application layer protocol that implements greater security against attacks on the internet of things to protect biomedical devices; It is important to have as a proposed engineering method the comparison of the MQTT and COAP protocols since they are the most used in the Internet of Things (IOT) industry Considering essential elements such as security, confidentiality and privacy of information, for an adequate and efficient security that entails a correct protection of biomedical equipment against cyber attacks that alter the ideal operation, also modify or steal information, requiring comparison of application layer protocols in IOT, which is why it turns out to be a quantitative investigation consistent in quantities numerical indicators which are represented percentages, being that the data are collected through measurements represented by an adequate analysis of statistical methods, and in turn continuous, since it is obtained by measurement or comparison of a unit or pattern of measurements; The results obtained from this research is that the COAP application layer protocol implemented in biomedical equipment offers greater security against attacks on biomedical devices connected to the Internet of Things compared to the MQTT protocol, in the Internet of Things, using biomedical equipment; having as a conclusion that, the COAP protocol presented a rate of failed requests of 19.37%, while MQTT showed 28.12%, which was one of the indicators in the study, therefore COAP is the safest protocol for DOS attacks compared to the MQTT protocol.

Keywords

Internet of things, protocols, attacks, security, latency, biomedical devices and confidentiality.

INDICE

I. INTRODUCCIÓN	11
1.1. Realidad problemática.....	12
1.2. Trabajos previos.....	13
1.2.1. Estado del Arte.	15
1.3. Teorías relacionadas al tema	18
1.3.1. Modelo Open System Interconnection (OSI) ISO/IEC 7498-1.....	18
1.3.1.1. <i>Capa Física.</i>	19
1.3.1.2. <i>Capa de Enlace de Datos.</i>	19
1.3.1.3. <i>Capa de Red.</i>	20
1.3.1.4. <i>Capa de Transporte.</i>	20
1.3.1.5. <i>Capa de Sesión.</i>	21
1.3.1.6. <i>Capa de Presentación.</i>	21
1.3.1.7. <i>Capa de Aplicación.</i>	21
1.3.2. Modelo TCP / IP.....	22
1.3.3. Protocolos de capa de aplicación.	23
1.3.3.1. <i>Constrained Application Protocol (COAP).</i>	23
1.3.3.1.1. <i>Tipos de mensajes.</i>	25
1.3.3.1.2. <i>Librerías.</i>	27
1.3.3.2. <i>Message Queue Telemetry Transport (MQTT).</i>	27
1.3.4. Ataques.....	31
1.3.4.1. <i>Definición.</i>	31
1.3.4.2. <i>Categorías del ataque.</i>	31
1.3.4.3. <i>Tipos de ataques informáticos.</i>	32
1.3.4.4. <i>Ataques más practicados en los últimos tiempos.</i>	33
1.3.4.4.1. <i>Phishing.</i>	33
1.3.4.4.2. <i>Hombre en el medio.</i>	33

1.3.4.4.3.	<i>Ataques de denegación de servicio (D.O.S).</i>	33
1.3.4.4.4.	<i>Ataques de denegación de servicio distribuido Mirai (DDOS Mirai).</i>	33
1.3.5.	Tecnologías Biomédicas.	33
1.3.6.	Dispositivos biomédicos que pueden ser utilizados en internet de las cosas.	34
1.3.6.1.	<i>Tensiómetro.</i>	34
1.3.6.2.	<i>Marcapasos cardíaco inalámbrico.</i>	34
1.3.6.3.	<i>Audixi 10, audiómetro IOT.</i>	35
1.3.7.	Microcontroladores.	36
1.3.7.1.	<i>Arduino.</i>	36
1.3.7.2.	<i>Raspberry pi.</i>	37
1.3.8.	Definición de términos básicos.	38
1.3.8.1.	<i>Internet Of Things (IOT).</i>	38
1.3.8.2.	<i>Cloud Computing.</i>	38
1.3.8.3.	<i>RFID.</i>	38
1.3.8.4.	<i>Protocolo.</i>	39
1.3.8.5.	<i>Sensor.</i>	39
1.4.	Formulación del Problema	40
1.5.	Justificación e importancia del estudio	40
1.6.	Hipótesis	42
1.7.	Objetivos	42
1.7.1.	Objetivo General.	42
1.7.2.	Objetivos específicos.	42
II.	MATERIAL Y MÉTODOS.	42
2.1.	Tipo y diseño de la investigación	42
2.1.1.	Tipo de investigación.	42

2.1.2.	Diseño de investigación.....	42
2.2.	Población y muestra.....	43
2.3.	Variables, Operacionalización.....	44
2.3.1.	Variable independiente.....	45
2.3.2.	<i>Variable dependiente</i>	45
2.4.	Técnicas e instrumentos de recolección de datos, validez y confiabilidad.	45
2.4.1.	Abordaje Metodológico.....	45
2.4.2.	Técnicas de recolección de datos.....	45
2.4.3.	Instrumentos de recolección de datos.....	45
2.5.	Procedimientos de análisis de datos.....	45
2.5.1.	Análisis estadístico de los datos.....	46
2.6.	Criterios de Rigor Científico.....	46
2.7.	Criterios de rigor científico.....	47
III.	RESULTADOS.....	48
3.1.	Tablas y Figuras.....	48
3.1.1.	Descripción de resultados.....	48
3.1.1.1.	<i>Resultados respecto a:</i>	48
3.2.	Discusión de resultados.....	50
3.3.	Aporte Práctico.....	51
3.3.1.	Identificación de los ataques más efectuados.....	53
3.3.2.	Implementación de protocolos.....	54
3.3.2.1.	<i>Protocolo MQTT</i>	54
3.3.2.2.	<i>Protocolo COAP</i>	58
3.4.	Evidenciar resultados en base a las pruebas realizadas.....	58
3.4.1.	Medición de porcentaje de peticiones fallidas de los protocolos MQTT y COAP.....	58

3.4.2. Medición de la latencia media con respecto MQTT y COAP respectivamente.	59
IV. CONCLUSIONES Y RECOMENDACIONEs.	61
4.1. Conclusiones.....	61
4.2. Recomendaciones.....	62
REFERENCIA	63
ANEXOS	66

I. INTRODUCCIÓN

En los últimos años se ha hecho de conocimiento lo que podría significar la cuarta revolución, un tanto parecida y hasta superada al internet tradicional en muchos aspectos según señalan ciertos investigadores y consultores expertos en temas informáticos, dicha revolución ha sido denominada como: “Internet de las cosas” o Internet of Things (IOT).

Al mencionar uno de los puntos principales como es internet de las cosas, es hacer referencia a diversos dispositivos comunes conectados a internet, esto se debe gracias al desarrollo e implementación de IPv6, lo que generará según, Agencia Internacional de Asignación de Números de Internet (IANA), con 340 sextillones en direcciones de Internet Protocolo (IP), en donde desde objetos cotidianos hasta equipos sofisticados para el cuidado de la salud estarán conectados y en constante comunicación a través de internet.(Chavéz, 2019)

El gran reto que deben asumir las organizaciones, fabricantes, desarrolladores y usuarios entorno al uso de dispositivos conectados, es el tema de seguridad, ya sea en el caso de los fabricantes por ser eslabón principal, que a su vez requieren tener en cuenta estándares, normativas y estudios realizados a cerca de cómo implementar y respaldar la integridad en el análisis y tratamiento de la información, puesto que es de carácter sensible, con el fin de no dejarse acarrear por la emoción de ser los primeros en llegar al mercado, por el lado de los desarrolladores así mismo se requiere de estudios que permitan mejorar e implementar de forma progresiva la seguridad, ya que es un tema novedoso y con mucha popularidad. En el caso de las organizaciones deben prestar la atención requerida para establecer estándares, normativas y legislaciones que deben ser respetadas y puestas en práctica tanto para la fabricación de los dispositivos y comercialización, así como para el uso de los mismos, por el lado de los usuarios y/o consumidores finales, prestando la atención debida a los parámetros establecidos a fin de mitigar ataques.

1.1. Realidad problemática

En los últimos tiempos, se experimenta la creciente demanda de conectar dispositivos comunes a internet, para acceder a su manipulación e información desde cualquier parte del mundo. Con la revolución del IPV6, se estimó que para el 2020 existirán por arriba de los 50 mil millones de equipos y dispositivos conectados a la red, superando la cantidad de habitantes en el mundo, eso revela la importancia de la conectividad de cualquier dispositivo. (Cisco, 2011).

El internet de las cosas, en adelante (IOT), se ha convertido en un componente integral de muchos sistemas entre ellos la producción industrial, cuidado de la salud, ciudades y casas inteligentes, además, de muchos otros campos, mejorándose en gran medida por la adopción generalizada de sensores, actuadores, sistemas integrados, tecnologías de identificación y dispositivos móviles.(Fotiou et al., 2016)

Siendo que, uno de los campos de la IOT donde se desarrolla progresivamente es el campo de la medicina, ya que permite un mayor control de pacientes con serios problemas de salud, como por ejemplo: pacientes con enfermedades cardiovasculares, Alzheimer, epilepsia, entre otros; esto conlleva a reducir las estadísticas de mortalidad y acrecentar la calidad y condición de vida de las personas en la sociedad.

Asi mismo en la actualidad se fabrican dispositivos IOT los cuales tienen serios problemas de seguridad, ya que no existen estándares ni legislación pertinente que conlleve a la protección de personas que utilicen dispositivos biomédicos, poniendo en peligro la privacidad y aún más la vida del usuario a través de la integridad de este, por lo que se requiere asegurar la información a través de mecanismos oportunos, pues se trata de dispositivos que carecen de suficiente capacidad de procesamiento, almacenamiento y protección, es por ello, que es de necesidad conocer los principales protocolos de seguridad y además identificar su comportamiento de acuerdo a aquellos ataques que existen.

Por consiguiente, es trascendental identificar qué protocolo de comunicación implementa mayor, adecuada y eficiente seguridad que conlleve a una correcta protección de equipos biomédicos ante ataques cibernéticos que alteren el ideal funcionamiento, asimismo modifiquen o roben información.

En consecuencia, se considera que es trascendental estudiar y analizar la seguridad en cuanto a protocolos de comunicación en capa de aplicación para internet de las cosas, teniendo como conocimiento que la IOT utiliza el modelo TCP/IP, siendo que una de las capas comúnmente expuesta a los ataques es la capa de aplicación, tal como lo menciona, (Lavinia, 2017), quien indica, que la capa de aplicación es la puerta hacia el exterior y es donde se debe de prestar mayor atención, ya que los atacantes pueden tener opción a ejecutar ciertos ataques, tal como denegación de servicio (DOS), man on the middle o en español hombre en el medio, entre otros, para acceder a la información y/o alterar el correcto funcionamiento de los dispositivos conectados a internet, lo cual traería consigo múltiples consecuencias.

1.2. Trabajos previos

Olano. (2016), según su investigación, titulada “Desarrollo de un sistema de detección de tráfico anómalo en el contexto de la internet de las cosas”, utilizó el instrumento de análisis y diseño de método de detección de tráfico para la adecuación del protocolo de red distribuido versión 3 (DNP3), que sirve como complemento a otros sistemas de protección para conseguir el incremento de la seguridad en IOT, estando existente la preocupación en los temas de seguridad en la internet de las cosas.

Para desarrollar el trabajo de investigación el autor obtuvo información sobre el protocolo DNP3, diseñar y preparar un entorno de trabajo que permitió la simulación virtual de una red de inspección, registro y obtención de datos (SCADA), adquiriendo conocimientos del desarrollo de instrumentos para la detección de ataques y manipulación de paquete, además se planteó un escenario experimental donde aplicó el sistema previamente desarrollado para la seguridad de una red IOT. Finalmente escribió un apartado en el que dejó constancia de los procesos realizados y los conocimientos adquiridos.

A partir de las pruebas realizadas el autor llegó a la conclusión que se cumplieron con los requisitos funcionales de monitorización de tráfico en la interfaz, procesamiento de tráfico paquete a paquete sin pérdidas de tasas medias de procesamiento de paquetes y extracción de datos de interés para la aplicación.

Cuzme. (2015), Impulsó el trabajo de indagación, el cual lleva por nombre: “Internet de las cosas y consideraciones de seguridad”, en la cual indica el autor que, analizó

aquellos debilitamientos y ataques que obtiene la invención tecnológica (IOT) y plantea establecer dispositivos de seguridad que se debe de considerar al instante en que se implementa la IOT a nivel hardware, software, red y seguridad en la nube, conocida como Cloud Computing.

Además, se analizaron costos que deben tener en cuenta para desarrollar instrumentos de seguridad, en cada dispositivo que se encuentre conectado a internet, de igual manera se determinaron diseños para obtener objetos inteligentes con altos niveles de seguridad admisibles, considerandose los siguientes aspectos: recolectar, registrar, procesar, analizar e interpretar la información recabada a través de entrevistas realizadas a personalidades con cargos importantes en el ámbito de la informática.

Finalmente el autor llegó a la conclusión, que los desarrolladores y fabricantes, tienen la responsabilidad de valorar la seguridad en el transcurso de la concepción y desarrollo de tecnologías implementadas con IOT, debido a que no sólo se habla de pérdida o robo de información, si no al control de dispositivos remotamente, denegación de servicio, etc., que conllevaría al menoscabo de vidas humanas o ocasionar la detención colectiva.

Castro. (2016), desarrolló el trabajo, titulado "Internet de las cosas. Privacidad y seguridad", quien abordó el tema de los posibles problemas que puede causar en el futuro el uso de internet de las cosas y además analizar a detalle las distintas amenazas a la seguridad, como es en el nivel de software y/o hardware.

También propuso desarrollar soluciones a posibles amenazas, analizando resultados obtenidos a través de encuestas que se realizaron a usuarios.

Bautista y Juárez. (2017), desarrollaron el trabajo de investigación, titulada "Análisis comparativo de los protocolos inalámbricos de redes AD HOC, Zigbee y Bluetooth, aplicados en tecnologías médicas", basando su investigación en temas de comunicación de redes AD HOC inalámbricas aplicadas en tecnologías biomédicas, con el fin de definir cual es el protocolo de comunicación más eficaz y eficiente al comparar los protocolos Zigbee y Bluetooth, teniendo en cuenta factores como eficiencia en la calidad de servicio y gestión de la energía que afectan directamente a la transmisión de información.

Ellos realizaron un diagnóstico de las tecnologías biomédicas, identificar factores que influyen en ellas, analizar los protocolos de comunicación e implementación de

una red Ad Hoc tanto para ZigBee y Bluetooth, utilizando dos escenarios: el primero ubicando los nodos a una distancia de 8 metros con 30 centímetros, sin obstáculo alguno, y el segundo ubicando los nodos a una distancia de 6 metros con 10 centímetros pero con obstáculos; la red constó de un nodo receptor y uno emisor. Aplicando métodos para medir los indicadores para cada protocolo y comparar los resultados teniendo en cuenta el tiempo de respuesta y consumo de energía. En suma llegaron a la conclusión de que el protocolo de comunicación ZigBee es el más eficiente en cuanto a la transmisión de datos y el bajo consumo de energía, con respecto a Bluetooth, en condiciones de uso con obstáculos como las mencionadas anteriormente.

1.2.1. Estado del Arte.

“Internet de las cosas: los protocolos existentes y tecnológicos Desafíos en Seguridad”, Harshal señala que los factores importantes que se consideran en IOT, son la eficiencia, fiabilidad, seguridad, velocidad y conectividad a internet y para satisfacer estos parámetros se debe desarrollar tecnologías de comunicación específicos.

Se describieron los protocolos de IOT teniendo en cuenta que los dispositivos están equipados con cantidad de memoria limitada, lo que representan un problema de seguridad al crear nuevos algoritmos, por otra parte cierto tipo de sensores carecen de capacidad para la aplicación de soluciones criptográficas anticipadas; finalmente el autor concluye que el esquema de seguridad para IOT debe desarrollarse de manera tal que se necesiten cantidades mínimas de energía y potencia de cálculo además de memoria, sin afectar los niveles de seguridad del dispositivo. (Harshal Sardeshmukh, 2017).

Un “control de acceso a internet de las cosas”, en la que Nikos considera que uno de los mayores problemas en la IOT es el bajo procesamiento y reducida extensión de respaldo de información, por lo que se plantea utilizar un sistema de control de acceso, que mediante protocolo de comunicación (canal que se implementa usando criptografía de clave pública) se pueda enviar información desde los dispositivos hacia un centro de confianza o más conocido como “proveedor de control de acceso” (ACP) y así evaluar y comunicar resultados a los objetos inteligentes. (Nikos, 2016)

Se señaló además, un protocolo de comunicación para asegurar la autenticidad de dispositivos de borde en la IOT, mediante el uso de Physical Unclonable Function SRAM PUF, pudiéndose crear dispositivos que permitan la autenticación, y generar bits únicos y no clonables para la identificación y autenticación de circuitos integrados, ya que un adversario puede crear un acceso o eludir la seguridad con el fin de filtrar información por un canal de comunicación. (Ujjwal Et Al, 2017)

Lavinia (2017), señala que, realizó una encuesta sobre protocolos de capas de aplicación resguardando la seguridad en internet de las cosas, indicando que estos protocolos denominados capa de aplicación que son utilizados en la actualidad son los mencionados a continuación :

- a) Constrained Application Protocol (COAP),
- b) Message Queue Telemetry Transport (MQTT), y
- c) Extensible Messaging and Presence Protocol (XMPP).

El autor Lavini (2017), justificó su investigación en base a que la capa de aplicación es la puerta hacia el exterior de la red, y en la que pudieran existir muchas amenazas como: ataque por D.D.O.S., suplantación de identidad, modificación de datos; sumado a eso, es donde se realiza la autenticación del usuario y el acceso a datos confidenciales.

En conclusión, el mayor desafío es el desarrollo de nuevas soluciones para la IOT, las empresas desarrolladoras de hardware y software, es de necesidad considerar ampliamente los temas de seguridad y privacidad desde un inicio, así como también educar al usuario a utilizar procedimientos de seguridad adecuados durante el uso de un sistema de IOT. (Lavinia, 2017)

“Security analysis of Simple Network Management Protocol based IEEE P21451 Internet of Things”. (Xinzheng Feng. et, al 2017) según los autores indican que realizaron un análisis acerca de los problemas de privacidad y seguridad, el cual definen al problema como el mayor reto que tendrá que afrontar la IOT; también reconocen que actualmente la IOT no cuenta con un estándar uniforme y definido a nivel mundial, lo que representa una de las limitaciones para el desarrollo de la tecnología en cuestión.

A partir de los problemas mencionados líneas arriba, los autores consideran que para realizar mejoras en la IOT, es necesario implementar seguridad en la capa

física, en cuanto a la comunicación de los sensores, para lo cual identifican las amenazas más resaltantes detalladas en la **Tabla N° 1**.

Tabla 1:

Amenazas a dispositivos IOT y su definición.

AMENAZAS DE SEGURIDAD	DEFINICIÓN
Agresión física	Los atacantes destruyen o roban el terminal de la IOT.
Nodo sensor	Los atacantes se disfrazan como un nodo terminal para unirse a la percepción de la red, luego reportan información falsa o emiten una orden falsa.
El reemplazo del sensor	Los atacantes reemplazan ilegalmente el dispositivo sensor, dando como resultado el reconocimiento de datos fallida o anormal.
Ataque agotado	Los atacantes envían correo no deseado a los terminales de la IOT con el fin de agotar la energía de los terminales.
Intercepción, manipulación, falsificación y repetición.	Los atacantes interceptan, falsifican y reproducen los datos que se transfieren a la IOT, luego acceden a la información sensible del usuario para causar error de transmisión de la información.

Fuente: (Xinzheng, F. 2017)

Con base a los antecedentes mencionados, la IEEE propuso el estándar IEEE P21451-1-5 (estándar para interfaces de sensor inteligente de sensores, actuadores y dispositivos de protocolo simple de administración de red o reconocido por las siglas SNMP, las cuales son reconocidas en inglés; para la comunicación de dispositivos de red).

Los investigadores analizan la estructura del SNMP, detallan el mecanismo de seguridad y finalmente analizan las amenazas que necesitan ser resueltas, para

concluir que habrá que enfrentar, resolver los problemas que se vayan presentando y alcanzar los objetivos en cuanto a la gestión de redes, para proporcionar estabilidad y seguridad en la IOT.

De igual manera en el denominado “Análisis de seguridad de los protocolos del IOT: Un enfoque en COAP” (Reem & Babar, 2016), plantean el uso del IOT como una de las herramientas para aplicarla en el ámbito de la salud, para contribuir al descubrimiento y supervisión de enfermedades, adicional a ello el bienestar de los seres humanos.

Sin embargo, se indica que el tema de la seguridad se ha transformado en un contenido trascendente, crucial e indispensable, debido a que se trata de datos sensibles y sumamente importantes; uno de los protocolos abordados en su investigación es el protocolo COAP, ya que fue examinado básicamente por su arquitectura, la seguridad y la aplicación.

Siendo COAP considerado como uno de los protocolos que será el futuro de todos los protocolos de aplicación y se utilizará DTLS para optimizar la seguridad y respaldar la integridad, certeza y confidencialidad de la información.

Gran parte de autores concluyeron que aunque muchos protocolos incluyendo COAP se han estudiado, todavía se requiere de una investigación profunda para su posterior análisis en asunto de seguridad.

1.3. Teorías relacionadas al tema

1.3.1. Modelo Open System Interconnection (OSI) ISO/IEC 7498-1.

Es de trascendental importancia mencioanr que el modelo OSI es conceptual, ya que no precisa ni determina protocolos y interfaces, solamente establece métodos genericos sobre como se puede concebir las redes de comunicaciones de datos. Asi se tiene que en el debido y correcto proceso de traspaso de datos, en el que participan los componentes denominados software y hardware. Debido a este desarrollo, los procedimientos se distribuyen en niveles o capas, como se puede ver en la **Tabla N° 2.** (Stallings, 2007)

Tabla 2:

Niveles de modelo OSI.

7	Aplicación
6	Presentación
5	Sesión
4	Transporte
3	Red
2	Enlace de Datos
1	Física

Fuente: (Teoría de las capas de internet, 1980)

1.3.1.1. Capa Física.

Según Stallings define a la capa física como la capa de determinaciones funcionales, eléctricas y mecánicas para la correspondiente activación, mantenimiento, y desactivación del enlace físico entre sistemas, siendo la comisionada para prestar los servicios respectivos a la correspondiente capa de enlace de datos.

Por lo que se describe cuatro tipos de propiedades fundamentales en la presente capa:

- 1) Mecánica: es aquella que describe las propiedades físicas de la interfaz y el medio de comunicación;
- 2) Eléctrica: en este apartado se señala que se enfoca la capa en la representación de bits y en la velocidad de transmisión de datos;
- 3) Funcional: la presente esa relacionada con aquellas funciones que ejecutan los conectores; y
- 4) Procedimiento: se indica que esta referido a la secuencia de acontecimientos que se efectúan en el intercambio del flujo de bits a través del medio físico. (Stallings, 2007)

1.3.1.2. Capa de Enlace de Datos.

Es la encargada de transmitir los bits como cadenas sin significado ni estructura, las divide en tramas y les proporciona una adecuada estructura, agregando información de control (por ejemplo numeración de tramas) e información destinada

a la corrección de errores (CRC), además puede establecer mecanismos de control de flujo para evitar la saturación del receptor, tal como se indica. (Anzar, 2005)

Siendo además la encargada de la topología y del acceso a la red. Dentro de los cuales podemos mencionar algunos ejemplos de protocolos, siendo estos: high level data link control (HDLC) o protocolo de enlace de alto nivel, logical link control (LLC) y point to point protocol (PPP) o protocolo punto a punto). (Boranat & Montagud, 2012)

1.3.1.3. Capa de Red.

Se determina que, es aquella que encamina los paquetes de datos a sus respectiva asignación por aquellas rutas que resultan ser más óptimas en ese momento o instante. A su vez domina la conexión y diligencia aquella congestión que podría mostrarse en la red cuando un nodo obtiene mucho más tráfico del se puede procesar. Por lo que podemos observar que en esta capa se localizan los routers y el protocolo IP. (boronat & Montaud, 2012)

Se advierte que los routers operan en la capa de red, usando protocolos de enrutamiento o encaminamiento, para la respectiva determinación de la ruta que los paquetes de datos deben seguir.

1.3.1.4. Capa de Transporte.

Se observa que en la capa de transporte se admiten los paquetes de la capa de sesión, que posteriormente se tratará; siendo sustancial señalar que si es necesario esta capa los fracciona en unidades de menor dimensión y los enumera para seguidamente enviarlos al nivel de red con la finalidad de garantizar que lleguen sin equivocaciones al receptor, por ello se puede deducir que la capa de transporte del emisor proporciona a cada uno un valor calculado en base a la información del paquete completo o checksum. (Stallings, 2007)

Esta capa de transporte por parte del receptor es quien ordena todos los paquetes que fueron recibidos, verificando a su vez que el valor arrojado sea el mismo, antes de ser enviado a la capa de sesión, permitiendo así solicitar el posterior reenvío de los paquetes que no hayan llegado o que contengan algunos errores, no siendo necesario el reenvío del mensaje completo. Señalando que en esta capa se encuentran tres tipos de protocolos desiguados, denominados:

1. User datagram protocol (UDP),

2. Internet control message protocol (ICMP) y,
3. Transmission control protocol (TCP).

1.3.1.5. Capa de Sesión.

Es aquella capa que establece, organiza, sincroniza y finaliza la conexión entre los usuarios. Proporciona además puntos de comprobación para controlar el intercambio de datos realizados; por lo que si se presenta una detención de transmisión u otro algún tipo de fallo, se podrá proseguir la transmisión desde el último punto de verificación y no tener que reiniciar. Por lo que se puede concluir que en esta capa de sesión se otorgan servicios de control de diálogo, asimismo de agrupamiento y de recuperación. (Stallings, 2007)

1.3.1.6. Capa de Presentación.

Transforma la estructura que obtiene de los datos a un formato que la capa de aplicación pueda asimilar. Siendo interpretados correctamente los datos de interpretación por el receptor, pese a que los dispositivos posean distintas representaciones internas de caracteres, números, imágenes y sonido.

Sin embargo la capa de presentación procesa aquellos datos que admite de la capa subsiguiente denominada capa de aplicación para que esta deba ser transmitida por la red sin inconvenientes. Por ello se puede mencionar algunos ejemplos de servicios a realizar, como son los de compresión y cifrado de datos. (Stallings, 2007)

1.3.1.7. Capa de Aplicación.

Se define como aquella que admite el acceso a la información a las aplicaciones que lo soliciten. Estando destinados a tareas específicas, incorporando a aquellos procesos que brindan servicio al usuario, por lo que también se le denomina "servicios" a estos protocolos; mencionando también que están bajo su control directo; tenemos como ejemplo, compartición de ficheros, gestión de colas de trabajo de impresión, correo electrónico, gestión de base de datos, etc. (Stallings, 2007)

En la capa de aplicación se tienen protocolos, tales como:

1. SMTP (simple mail transfer protocol o protocolo simple de transferencia de correo), permitiendo el correcto envío y distribución de correo electrónico;

2. POP (post office protocol) posibilita el reparto de correo al usuario final;
3. FTP (file transfer protocol) realiza la transferencia de archivos;
4. TELNET (conexión remota) y;
5. HTTP.

1.3.2. Modelo TCP / IP.

Se señala que el modelo OSI, contiene un marco teórico completo de la interconexión de redes sin embargo este modelo resulta ser mucho mas tedioso que el modelo TCP/IP el cual esta destinado a su funcionalidad útil y directa, resultando ser este más práctico, es preciso señalar que la red se apoya en el modelo TCP/IP, surgiendo su trascendente relevancia en la actualidad.

Se desarrollo primero el modelo TCP/IP para luego trabajar en el modelo OSI, en el cual se aplicó los conceptos principales sobre gestión y diseño de redes. Así mismo se especificó que el modelo OSI analiza y desgloza el proceso de comunicación en varias capas o niveles, obteniendo una funcionalidad atribuida que ofrece a las capas superiores. Se indica que le corresponde a la capa física y de enlace los niveles más bajos, los cuales no están detallados, pues el protocolo se pensó para funcionar sobre cualquier tipo de red.

Los protocolos ARP, por sus siglas en inglés (Address Resolution Protocol) y RARP (Reverse Address Resolution Protocol), es aquella encargada de enlazar los sistemas de direccionamiento IP, así también el de la red física que es utilizada. Es imprescindible señalar que la base de la familia de protocolos es el nivel de Red, la cual importa un protocolo verdaderamente simple (Internet Protocol o IP) de prototipo datagrama para que de esta manera se consiga poner en practica o llevar a cabo en cualquier tipo de máquina.

Cabe indicar que la arquitectura TCP/IP se organiza y configura en las capas físicas de enlace, red, transporte y capa de aplicación. Por ello, al efectuar un nexo entre las capas del modelo OSI y TC P/IP, obteniendo las siguientes semejanzas.

Tabla 3

Comparación modelo OSI y TCP/IP

OSI		TCP/IP							
Aplicación		T	F	H	D	R	T	D	
Presentación		E	T	T	N	I	F	H	
Sesión		L	P	T	S	P	T	C	Aplicación
		N		P			P	P	
		E							Transporte
		T							Internet
Transporte		TCP				UDP			
Red		ARP		IP		ICMP			Intefaz de Red
Enlace de datos		Ethernet, Token Rink F.R., FDDI							
Física									

Fuente: (Anzar 2004)

1.3.3. Protocolos de capa de aplicación.

1.3.3.1. *Constrained Application Protocol (COAP).*

Conocido también como Protocolo de Aplicación Restringida, es uno de los protocolos nivelado a capa de aplicación que surgió en base a los mecanismos de HyperText Transfer Protocol, o Protocolo de Transferencia de Hipertexto (HTTP), con la diferencia que COAP es un protocolo más ligero debido a que los dispositivos fabricados para internet de las cosas tienen recursos limitados, tanto en capacidad de almacenamiento como de procesamiento, especialmente porque dichos dispositivos cuentan con baterías de poca duración.

COAP fue desarrollado por el equipo de trabajo *Constrained RestFul Environments* (CORE), de la *Internet Engineering Task Force* (IETF), definido como un protocolo de transferencia web, basada en Representational State Transfer (REST), permitiendo a los clientes y servidores consumir servicios web como el protocolo Simple Object Access Protocol (SOAP), en la figura 1 se puede visualizar un diagrama básico de su funcionamiento.

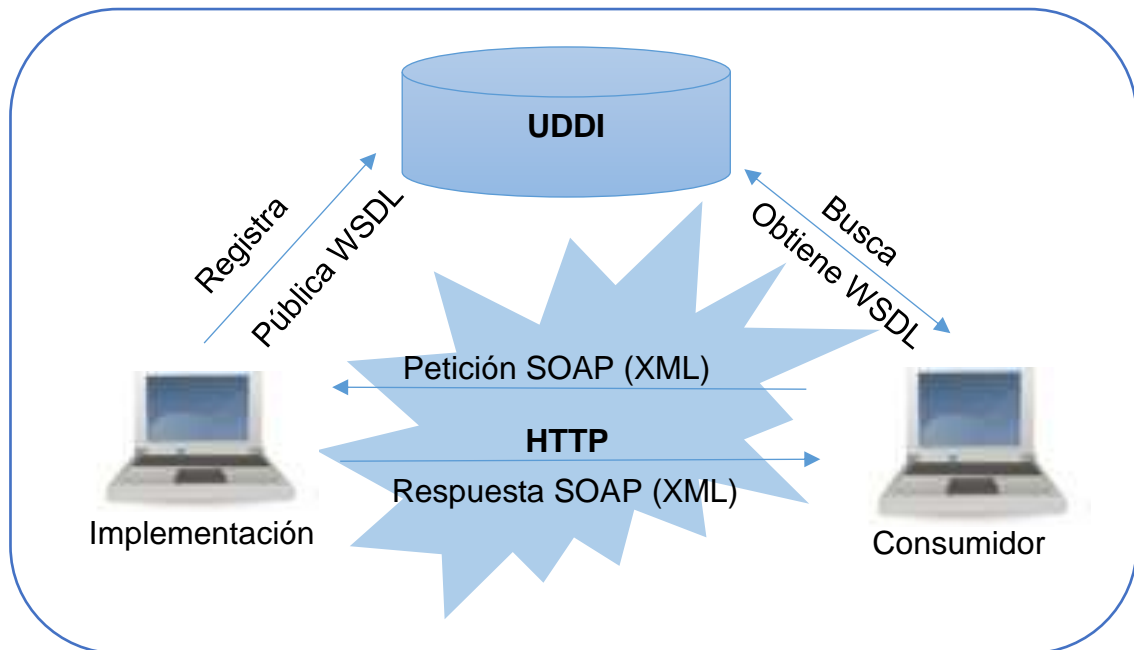


Figura 1. Representación SOAP.

Fuente: (Dejana & Gordana, 2015)

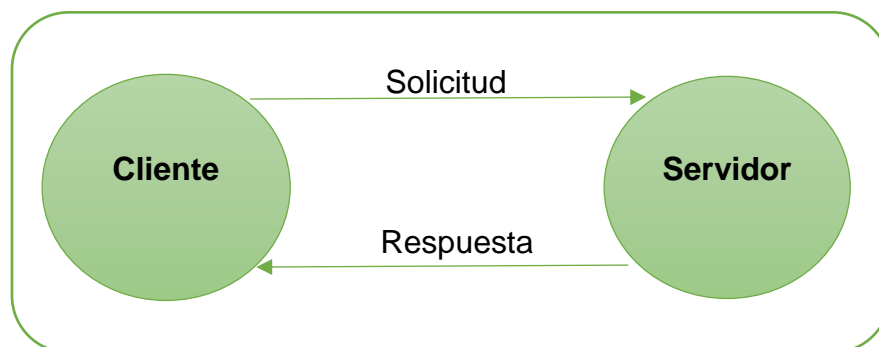


Figura 2. Comunicación cliente servidor.

Fuente: (Dejana & Gordana, 2015)

COAP utiliza User Datagram protocol (UDP) por defecto en la capa de transporte tal como persibe en la siguiente **Figura 3.**

HTTP		RTP		Aplicación
TCP	UDP	ICMP		Transporte
IP				Red
Ethernet MAC				Enlace de datos
Ethernet PHY				Física

Figura 3. Pila de protocolos para IOT.

Fuente: (Dejana & Gordana, 2015)

De esta forma se puede verificar que el protocolo COAP detalla cuatro tipos de avisos con una configuración idéntica. Estos mensajes son constituidos por el encabezamiento de dimensión estable, en seguida un número variable de alternativas y finalmente lo que corresponde a el contenido del mensaje o conocido como payload, en la **Figura 4** se puede visualizar la estructura.

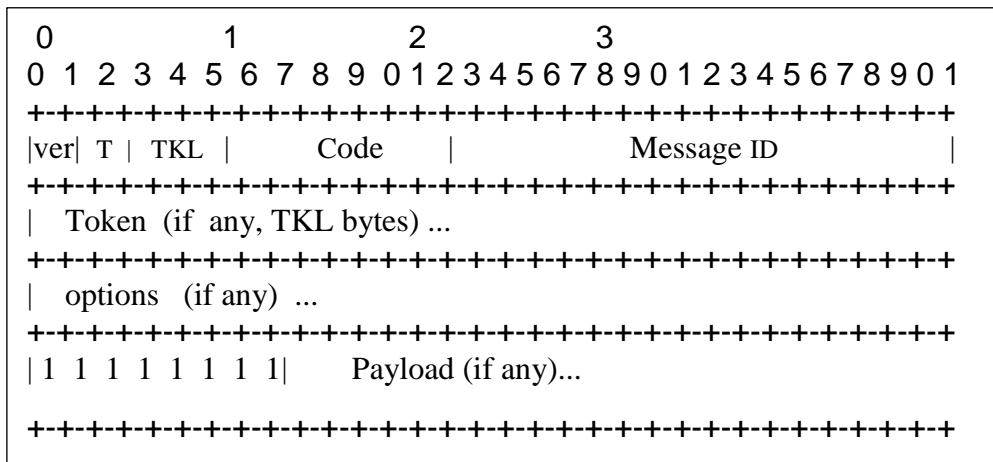


Figura 4. Formato de trama de un mensaje COAP.

Fuente: (González, 2017)

1.3.3.1.1. Tipos de mensajes.

- 1. Confirmable (CON):** Se indica que, los mensajes de tipo confirmable necesitan de una corroboración de recepción de parte del receptor. (Gimeno, 2017).

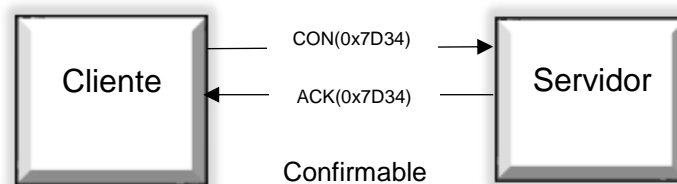


Figura 5. Mensaje de tipo confirmable.

Fuente: (González, 2017)

- 2. Non-confirmable (NON):** se señala que estos mensajes se envían cuando no es exigencia que el destinatario reciba el mensaje, por lo que no se requiere convalidación. Por lo que suele localizarse en aplicaciones con transmisiones regulares de mensajes, teniendo como ejemplo las repetidas lecturas del valor de un asesor. (Gimeno, 2017)

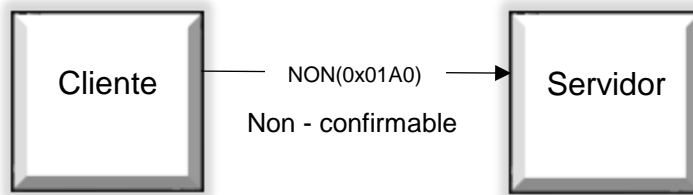


Figura 6. Mensaje de tipo Non-confirmable.

Fuente: (González, 2017)

3. **Acknowledgement (ACK):** se observa que este tipo de mensajes se envían para la respectiva convalidación de recepción de mensaje de tipo CON o también para responder a peticiones de tipo GET. (Gimeno, 2017)

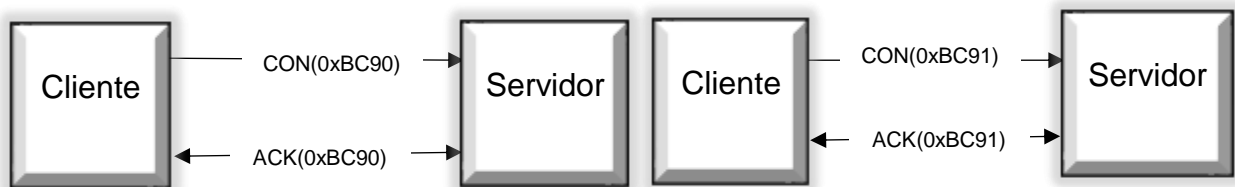


Figura 7. Mensaje de tipo ACK.

Fuente: (González, 2017)

4. **Reset (RST):** se advierte que se emite este mensaje como respuesta a un mensaje (CON O NON) el cual fue recibido pero que el receptor fue incapaz de procesar aun teniendo en cuenta que el contenido enviado era correcto. Esta postura suele oacontecer cuando alguno de los dispositivos se reinicia y pierde información sobre algún estado que pues le permitiría interpretar el mensaje que fue recibido de forma apropiada. (Gimeno, 2017).

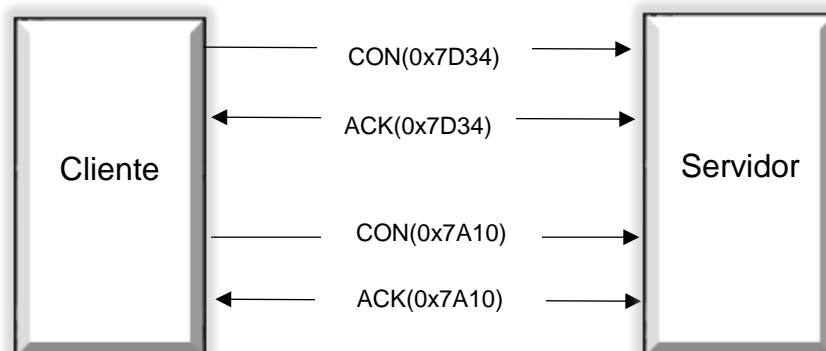


Figura 8. Mensaje de tipo Reset.

Fuente: (González, 2017)

1.3.3.1.2. Librerías.

Para implementar el protocolo COAP se pueden utilizar librerías de acuerdo a las funcionalidades brindadas o al lenguaje con el cual fueron desarrolladas, siendo las más utilizadas:

Tabla 4.

Librerías para COAP.

	Libcoap	Californium	txThings	Etri Coap
Desarrollado en	C	Java	Python	C
Permite el uso de cliente servidor.	Si	Si	Si	Si
Tipo de licencia.	Libre	Libre	Libre	Comercial
Implementa la opción Observe	Si	Si	Si	Si
Incluye la capa DTLS de seguridad.	No	Si	No	No
Uso de proxy.	No	En su última versión.	No	No

Fuente: (Elaboración propia)

1.3.3.2. Message Queue Telemetry Transport (MQTT).

Este protocolo es de mensajería de publicación–suscripción, desarrollado originalmente por International Business Machines, desde ahora IBM en el año 1999.

Se distingue por ser sencillo y ligero, de fácil implementación y además es de libre uso u open source, como generalmente se le conoce, dichas características permiten hacer de éste protocolo el ideal para su uso en entornos con limitaciones como lo es el internet de las cosas y la comunicación de maquina a máquina (M2M). En cuanto a la seguridad, la implementa en la capa de transporte, la cual proporciona un usuario y contraseña del sistema para la autenticación basado en TLS / SSL para el cifrado de datos. A partir del año 2010 es de código abierto y

muchas de las grandes corporaciones lo están utilizando recientemente tales como, Amazon y Facebook Messenger para Android e iPhone.

Además es un protocolo de mensajería de publicación suscripción diseñado específicamente para permitir comunicaciones ligeras máquina a máquina (M2M), cuenta con un servidor central conocido como broker (negociador) a través de TCP, tal cual se observa en la **figura 09**.

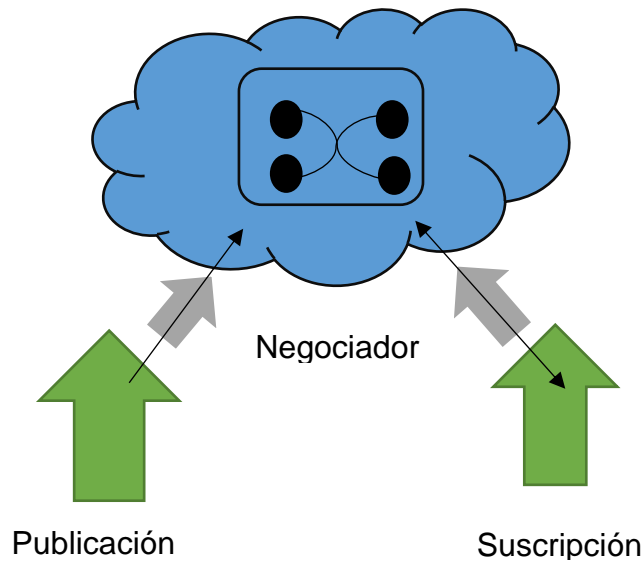


Figura 9. Esquema de mensajería publicación suscripción.

Fuente: (Oasis standard)

El modelo publicación - suscripción permite comunicarse a los clientes:

- Uno a uno.
- Uno a muchos.
- Muchos a uno.

El protocolo MQTT tolera tres niveles de calidad de servicio con lo que conlleva a una mejor comunicación entre dispositivos y son los siguientes:

- Envía y olvida o envía como mucho una vez (QoS 0)

El mensaje llega al broker una vez o ninguna, y si llega, llegará a cada uno de los suscriptores una vez o ninguna.

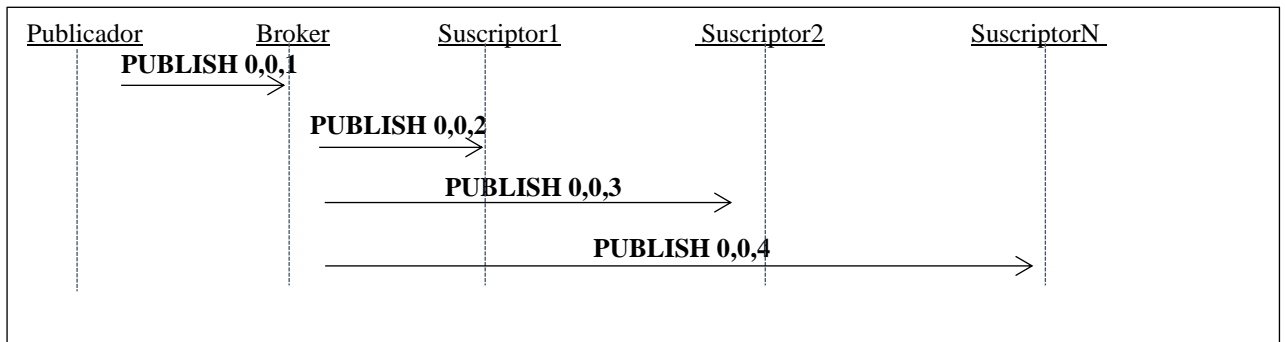


Figura 10. Diagrama de secuencia de publicación MQTT con QoS 0.

Fuente: (Oasis standard)

- Entrega al menos una vez (QoS 1)

En este caso se debe garantizar que el mensaje llegue al receptor al menos una vez. QoS 1 tiene un identificador de paquete en la cabecera y se debe confirmar con un paquete PUBACK. El lado que envía debe asignar un identificador de paquete único cada vez que publica un nuevo mensaje de aplicación.

Hasta que se reciba el paquete PUBACK del receptor el paquete se trata como “sin confirmar”.

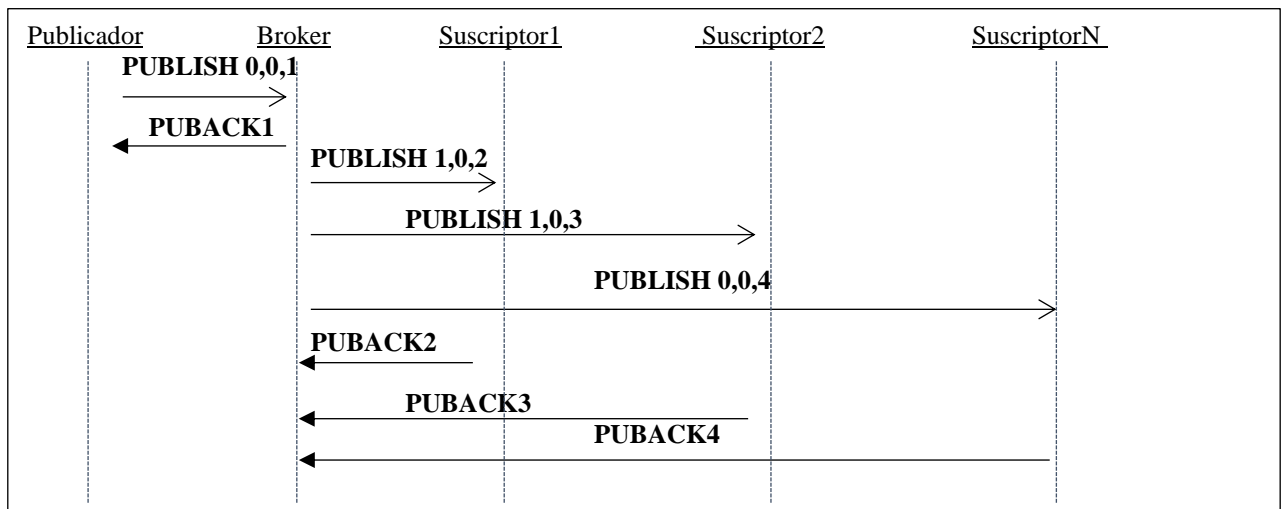


Figura 11: Diagrama de secuencia de publicación MQTT con QoS 1.

Fuente: (Oasis standard)

- Entrega exactamente una vez (QoS 2)

QoS 2 considerado como la mejor calidad de servicio, en este caso no se permiten pérdidas ni duplicación.

El lado emisor debe asignar también un identificador de paquete único, tratando como “sin confirmar” un paquete si no ha recibido un paquete PUBREC con el mismo identificador desde el receptor.

También debe enviar un paquete PUBREL que contenga el mismo identificador de paquete una vez recibe el PUBREC, tratando a este como “sin confirmar” hasta la recepción del paquete.

Una vez recibe el paquete PUBREL no debe reenviar el paquete PUBLISH original. Por otro parte, el lado receptor debe confirmar cada paquete PUBLISH con el mismo identificador con un paquete PUBREC hasta que recibe el paquete PUBREL. Una vez enviado el paquete PUBCOMP, el receptor debe tratar cualquier paquete

PUBLISH con el mismo identificador como una nueva publicación.

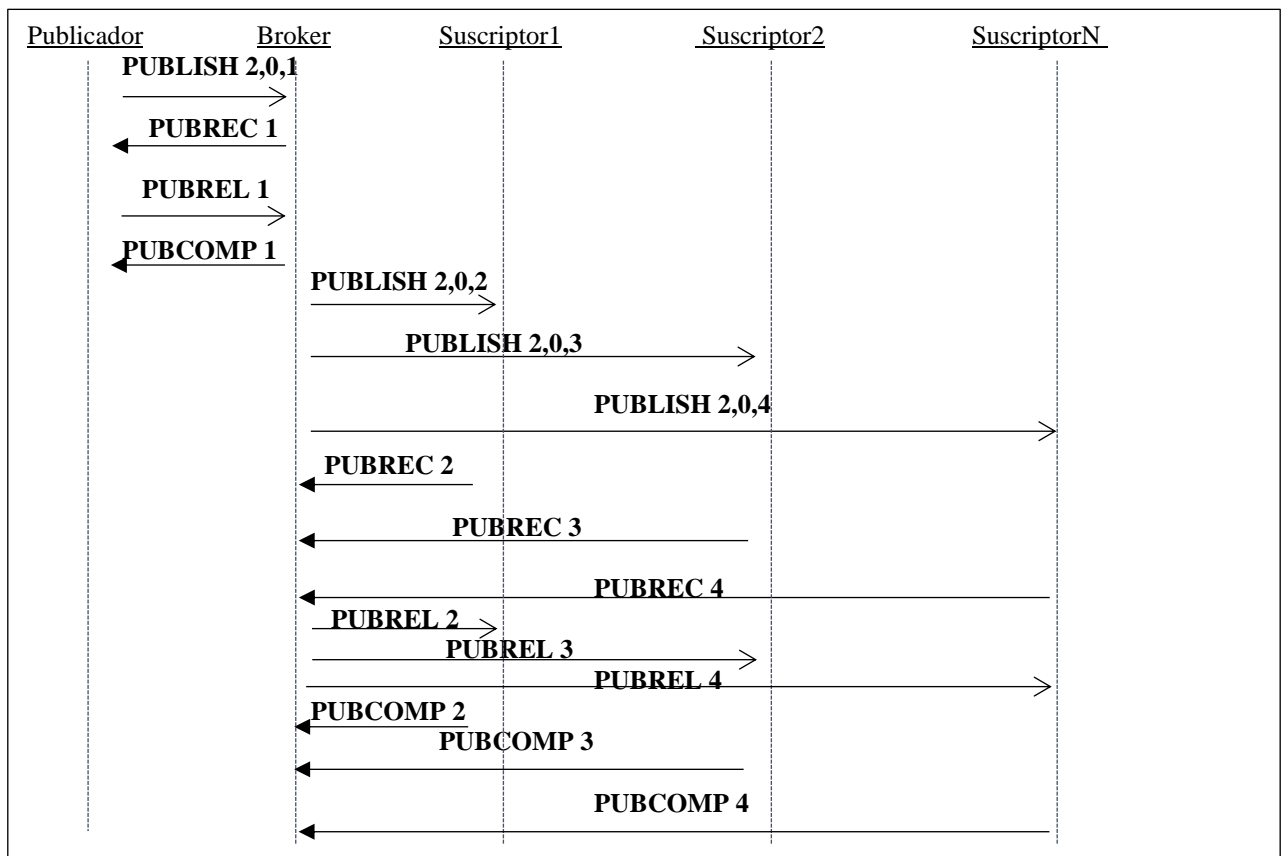


Figura 12. Diagrama de secuencia de publicación MQTT con QoS 2.

Fuente: (Oasis standard)

1.3.4. Ataques.

1.3.4.1. Definición.

Es importante indicar que los ataques informáticos se van incrementando conforme pasa el tiempo con mayor fuerza, de la mano del uso de la tecnología; por múltiples razones, ya sea por:

- a) Ventajas financieras, corporativas,
- b) Empleados disgustados, fraudes o extorsiones,
- c) Ataques a bases militares de estado, entre otros. (Izquierdo & Cabrera, 2017).

1.3.4.2. Categorías del ataque.

Son aquellos aprovechamientos de vulnerabilidades de aquellos sistemas que se localizan en la red de datos, por lo que requieren seguir una serie de categorías para lograr los objetivos trazados, obteniendo los siguientes puntos:

- i. **Interrupción:** esta categoría afecta y atenta directamente contra la disponibilidad, puesto que el impacto causado por el ataque deja un pequeño daño en alguna parte del sistema, siendo esta detección inmediata, por ejemplo la destrucción del disco duro y borrado de información.
- ii. **Intersección:** este ataque afecta la confidencialidad, obteniendo acceso a variada información personas no autorizadas. Este descubrimiento es un tanto complicado pues no suele dejar huella, por ejemplo escucha en línea de datos y copias ilícitas de programas.
- iii. **Suplantación:** a diferencia de la anterior, este ataque es tomado como delito de falsificación de información llegando a violar la autenticación, por lo que es aún más difícil su identificación, por ejemplo adulteración de mensajes a la red, registrar datos incorrectos en bases de datos, entre otros.
- iv. **Modificación:** requiere que la información obtenida sea modificada sin permiso del administrador del sistema y/o administrador de base de datos, ocasionando alteración de datos, siendo así difícil de detectar. López y Quezada (Izquierdo & Cabrera, 2017).

Pese al desarrollo e investigación de técnicas y/o herramientas en seguridad informática que se emplean e implementan en las empresas tanto públicas como privadas, podemos ver que a diario se descubren nuevos ataques ya sea en redes de datos como en el ámbito de la internet de las cosas, los cuales son capaces de interrumpir el buen funcionamiento del sistema, servidores, computadores, sistemas embebidos.

1.3.4.3. Tipos de ataques informáticos.

(Ramos & Ribagorda, 2004), señala que, los principales tipos de ataques son los que se describen a continuación:

- a) Ataques externos:** Son aquellos ataques realizados por aquellas personas que no tienen autorización dentro de la empresa, no teniendo los permisos correspondientes para el ingreso o modificación de sistema o red de dicha empresa, realizándolos por medio del internet o servidores.
- b) Ataques internos:** Son ataques realizados por las personas con los permisos correspondientes situados dentro de la organización o grupos de trabajo, teniendo este usuario los accesos para poder ingresar a la red de datos.
- c) Ataques a nivel de sistema:** Este tipo de ataques se basa en la vulnerabilidad en cuanto a la configuración de las políticas de acceso al servidor por un servicio mal configurado, llegando a atacar directamente al sistema operativo del servidor, obteniéndose privilegios a través de un administrador o root, mediante un terminal remoto, como por ejemplo los servicios TELNET y SSH.
- d) Ataque a nivel de aplicación:** Se basa principalmente en la modificación de información de datos acumulados en la base de datos por repositorios, realizándose sin necesidad de llevarse a cabo la ejecución del código del sistema operativo.
- e) Ataques pasivos:** Este ataque se basa en el control y monitoreo de tráfico de red, capturando gran parte de información que se transmite en la red de datos.
- f) Ataques de fuerza bruta:** Consiste en adivinar la clave secreta a través de las combinaciones para tener acceso ya sea a las aplicaciones y/o servidores. La única manera de descifrarla es utilizando programas que interceptan las comunicaciones y registra las contraseñas como por ejemplo el uso de

sniffers, puesto que las contraseñas se almacenan de modo encriptado y sin embargo cuando esta falla, los hackers recurren a la fuerza bruta. (Ramos & Ribagorda, 2004).

1.3.4.4. Ataques más practicados en los últimos tiempos.

1.3.4.4.1. Phishing.

Según (Unuth, 2014), define este apartado como aquel ataque que consiste en engañar a la víctima a través de suplantación de identidad de sitios o fuentes confiables, con el fin que proporcione información sensible de forma voluntaria.

La información obtenida previamente se utiliza de diversos fines, entre ellos robo y manipulación de información, tales como operaciones en nombre de la víctima, etc.

1.3.4.4.2. Hombre en el medio.

Según (Reyes, 2012), define como la interceptación de la comunicación entre dos partes, puede ser cliente - servidor, el host malicioso intercepta dicha comunicación, lo que le permite alterar, o robar información confidencial.

1.3.4.4.3. Ataques de denegación de servicio (D.O.S).

Según (Reyes, 2012), expresa tal contenido como uno de los ataques más utilizados, con pocas opciones de defensa, ya que los crackers se dedican a consumir ancho de banda y recursos, lo que les permite saturar a la víctima con tantos paquetes les sea posible durante un periodo. Para asegurar la efectividad del ataque, falsifican direcciones IP para evitar que la detección del ataque sea lo más complicado posible.

1.3.4.4.4. Ataques de denegación de servicio distribuido Mirai (DDoS Mirai).

A finales del año 2016 se registró una sucesión de ataques distribuidos de denegación de servicio (DDoS), dirigido a sitios web que son conectados a DYN, es una empresa de gestión del rendimiento de Internet basada en la nube; indicando que estaban entre esos sitios los de Amazon, Twitter, Reddit, Spotify y PayPal, probablemente se trate de un punto determinante en lo que concierne a la historia de los ataques. (Stephen Cobb de ESET, 2018).

1.3.5. Tecnologías Biomédicas.

Se tiene por tecnologías biomédicas a las aplicaciones de principios y técnicas de la ingeniería en la disciplina de la medicina. Por tanto, se afirma esencialmente en diseño y elaboración de los productos tecnológicos en equipos , dispositivos médicos, prótesis, etc. (Moore y Zouridakis, 2014)

Para la presente investigación es de necesidad estudiar y conocer las tecnologías biomédicas que se puedan utilizar en el internet de las cosas, así mismo que dichas tecnologías puedan monitorear parámetros fisiológicos de los pacientes, por ejemplo: ritmo cardiaco, presión arterial, temperatura, respiración, etc.

1.3.6. Dispositivos biomédicos que pueden ser utilizados en internet de las cosas.

1.3.6.1. *Tensiómetro.*

(Parra y La Torre, 2003) definen como un instrumento médico empleado para evaluar la presión arterial en pacientes, el cual proporciona unidades físicas de presión, estandarizadas en milímetros de mercurio.

Se compone esencialmente por un brazalete inflable, adicionalmente un manómetro y un estetoscopio para auscultar de manera clara el intervalo de los sonidos de korotkoff (sistólico y diastólico).

1.3.6.2. *Marcapasos cardíaco inalámbrico.*

(Cambridge Consultants, 2016) EBR Systems, desarrolló un dispositivo que permite la estimulación endocárdica inalámbrica, denominada WISE.

Actualmente es uno de los pioneros en desplegar un sistema que permite la terapia de re-sincronización cardíaca (TRC).

El sistema consiste en implantar un dispositivo (marcapasos) que permite ampliar la eficiencia de bombeo del corazón, equiparándolo con ambos ventrículos, tanto derecho como izquierdo, según investigaciones realizadas, se concluye que la referida tecnología podría favorecer a 1.5 millones de pacientes que presenten deficiencia cardíaca en el mundo, reduciendo las hospitalizaciones, insuficiencia cardíaca, e incluso la pérdida de vidas humanas.

El desarrollador utilizó el diseño de circuitos integrados y el procesamiento de señales para instaurar dos innovadores circuitos integrados, específicos de la

aplicación (CIEA) de señal mixta, administrados por un poderoso procesador XAP4, para la formación del núcleo del sistema WISE. Teniendo presente que el circuito interactúa con transductores ultrasónicos y con sensores de monitorización cardíaca.

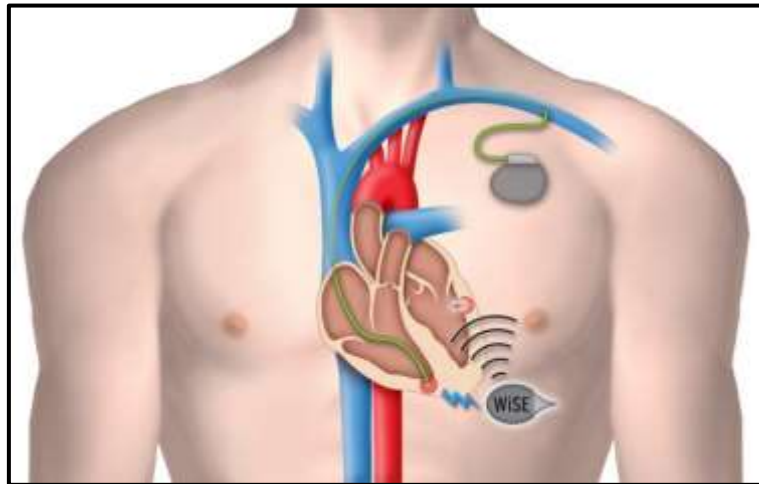


Figura 13. *Marcapasos Wise*

Fuente: *(wise)*

1.3.6.3. Audixi 10, audiómetro IOT.

Es un dispositivo auditivo con conectividad a internet, permitiendo el suministro y la integración de los datos en el correspondiente sistemas de información, cumple con las siguientes características:

- ✓ Conexión a plataformas y envío de información a la nube para almacenar y exportar datos de manera fácil.
- ✓ Análisis y estudio de datos obtenidos.
- ✓ Intercambio de datos con diferentes dispositivos.
- ✓ Monitorear al paciente en tiempo real.
- ✓ Asistencia de medicación a pacientes.
- ✓ Conexión o nexo médico-paciente para mejorar la experiencia del usuario.

Asimismo, es importante integrar un sistema de calibración que posibilite que siempre esté operativo. Señalando además, que su acceso remoto permite conocer el uso verídico del producto y simplifica la interconexión con sistemas y dispositivos externos.



Figura 14: Audiómetro IOT.

Fuente: (Audixi 10)

1.3.7. Microcontroladores.

1.3.7.1. Arduino.

Se trata de una placa microcontroladora muy conocida y utilizada por estudiantes, investigadores y expertos en sistemas electrónicos y embebidos, son de costo cómodo relativamente y factible de conseguir.

Consta de un entorno de desarrollo integrado (IDE) que así mismo es una interfaz gráfica (GUI) de código abierto u open source comúnmente conocido, es multiplataforma ya que puede descargarse e instalarse libremente ya sea en: Linux, Mac o Windows; consta de entradas analógicas (para recibir información de sensores) y digitales, apto para ejecutar órdenes sencillamente grabadas en su memoria, del mismo modo provee de librerías desarrolladas en lenguaje C++ que pueden ser utilizadas para el impulso de proyectos con mayor rapidez, permite escribir, compilar y depurar código, además de realizar modificaciones sin mayores inconvenientes, siempre y cuando se consideren las atenciones básicas sobre su uso y programación.

Usualmente se alimenta de entre 3.3V y 5V de acuerdo al modelo, ya sea mediante Universal Serial Bus (USB) o por medio de un transformador, actualmente existen varias versiones de arduino las que son utilizadas de acuerdo a las características que poseen y al proyecto que se requiera desarrollar.

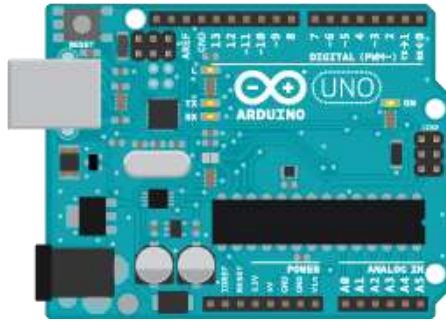


Figura 15: *Arduino UNO.*

Fuente: *(Arduino)*

1.3.7.2. Raspberry pi.

Es un microcomputador de precio accesible desarrollado por una organización con sede en Reino Unido, inicialmente diseñada para la enseñanza de ciencias básicas, gradualmente despertó el interés de aficionados, los que no vacilaron en utilizarla tanto para tareas ligeras así como para temas educativos, permitiendo aprender a programar lenguajes como Python y Scratch, a diferencia de Arduino, Raspberry Pi es una computadora totalmente funcional con consumo de energía mínimo pudiéndose conectar a un televisor u monitor, por el contrario Arduino es un microcontrolador capaz de ejecutar bloques de código específicos.

Posee de puertos USB para la conexión de periféricos, salida de audio y video mediante High-Definition Multimedia Interface (HDMI), General Purpose Input/Output (GPIO) entrada y salida de usos múltiples desde 26 hasta 40 pines (dependiendo del modelo) ver figura 16, Bluetooth, Random Access Memory (RAM) de 2 GB en el modelo Raspberry Pi 1 B, ranura para la conexión de una memoria flash, puerto Ethernet y los últimos modelos ofrecen conectividad WiFi.



Figura 16: Pines GPIO Paspberry Pi modelo A 26 pines y modelo B 40 pines.

Fuente: (Raspberry-pi)

1.3.8. Definición de términos básicos.

1.3.8.1. Internet Of Things (IOT).

Se indica que, IOT o internet de las cosas, señalado por Kevin Ashton en el año 1999, en una conferencia impartida en Procter & Gamble, pero fue en 2009 cuando el profesor Kevin Ashton utilizó la expresión de forma pública y es entonces donde surge la expectativa alrededor del término; el cual indica que, es una de las áreas prometedoras que está en pleno desarrollo y será accesible a muchas personas en el mundo, en un futuro no muy lejano, esto significa que las cosas estarán conectadas a internet pudiendo comunicarse entre ellas y tendrá un gran impacto en muchos sectores de nuestra vida, como la industria, economía, medicina, entre otros.

1.3.8.2. Cloud Computing.

Conocida como Computación en la nube, es una tecnología que permite el acceso remoto de dispositivos para su pertinente gestión y procesamiento de información (datos), en tiempo real.

La computación en nube a través de su flexibilidad y facilidad para acceder a los recursos compartidos y la infraestructura común de manera general y universal es una medida prometedora para la gestión eficiente de los datos de salud generalizados.

1.3.8.3. RFID.

Conocida por sus siglas en inglés "Radio Frequency Identification", que en español significa identificación por radio frecuencia.

Tecnología existente desde los años 40 y que gracias a la IOT está siendo utilizada en los últimos años. Se habla de aquella tecnología de identificación remota e inalámbrica, en la que un dispositivo lector conectado a un equipo de cómputo se comunica con un transponder siendo conocido como tag o etiqueta, utilizando ondas de radio, mediante antena.

1.3.8.4. Protocolo.

Según (Perez & Gardey, 2013) definen el protocolo como un estándar o conjunto de normas para establecer o guiar una conducta o acción, de este modo se puede establecer el intercambio de información entre dispositivos.

1.3.8.5. Sensor.

Es un componente electrónico, mayormente localizado al inicio del sistema eléctrico, se encargan de detectar magnitudes físicas o químicas, a través de estímulos externos para posteriormente enviar esa información por medio de pulsaciones a un módulo central de procesos y/o otro dispositivo que sea capaz de interpretar dicha información.

Con base a lo expuesto se tiene: Sensor de intensidad luminosa, distancia, temperatura, presión, desplazamiento, torsión, movimiento, fuerza, humedad y en el ámbito de la medicina se utilizan para detectar procesos biológicos de entre los cuales se encuentran los siguientes: sensor de presión sanguínea, pulso cardíaco, temperatura corporal, glucosa, posición del paciente, flujo de aire (respiración), entre otros, algunos de ellos se encuentran en la parte exterior del cuerpo, mientras que otros están implantados internamente para obtener información, analizarla o almacenarla en la nube para luego tomar decisiones con respecto a datos obtenidos.

Se muestra en la figura 17, un sensor adhesivo que es utilizado para monitorear el estado de signos vitales del paciente, se puede insertar en el corazón o en los pulmones y utilizan energía generada a través de los movimientos físicos del paciente para cargar las pequeñas baterías fabricadas de materiales biodegradables.

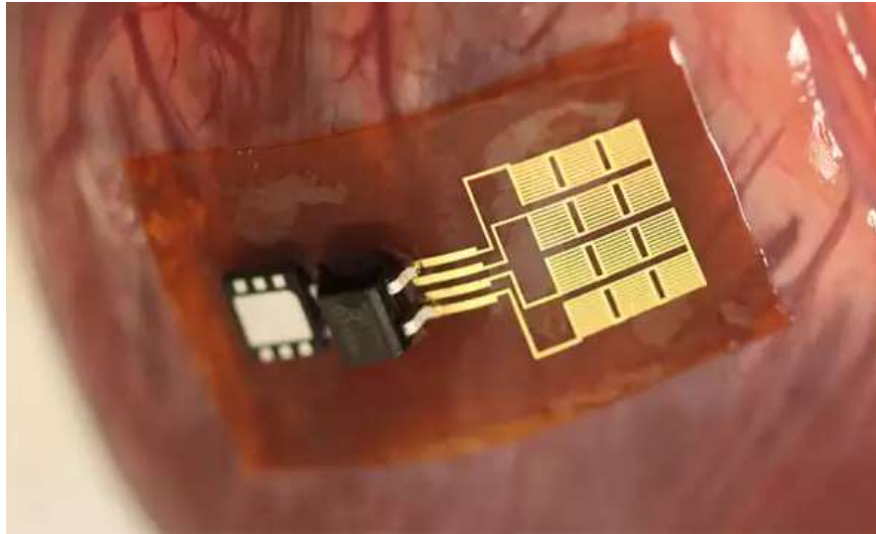


Figura 17. Sensor adhesivo para IOT.

Fuente: *(Technology for health and fitness pplications)*

1.4. Formulación del Problema

¿Qué protocolo de capa de aplicación implementa mayor seguridad ante ataques en internet de las cosas para proteger dispositivos biomédicos?.

1.5. Justificación e importancia del estudio

Conforme a los estudios realizados por muchas empresas consultoras e investigadoras, entorno a las tecnologías de la información, exponen el crecimiento acelerado en la fabricación y utilización de dispositivos comunes conectados a internet, es por ello que es de vital importancia obtener conocimientos transversales y estar preparados para afrontar retos sustanciales y trascendentales en cuanto a la seguridad del IOT.

Surgiendo la importancia de poder investigar y conocer temas de seguridad, a cerca de la nueva tendencia tecnológica basada en internet de las cosas, para determinar cuáles son las medidas de protección adecuadas a considerar en la capa de aplicación para el respectivo uso en dispositivos biomédicos.

“Según investigación se estima que la salud y las ciencias de la vida aumentarán de \$520 B en 2014 a \$1,335 T en 2020, logrando una tasa de crecimiento anual compuesto del 17%.” (Departamento de Investigación de Statista,2014)

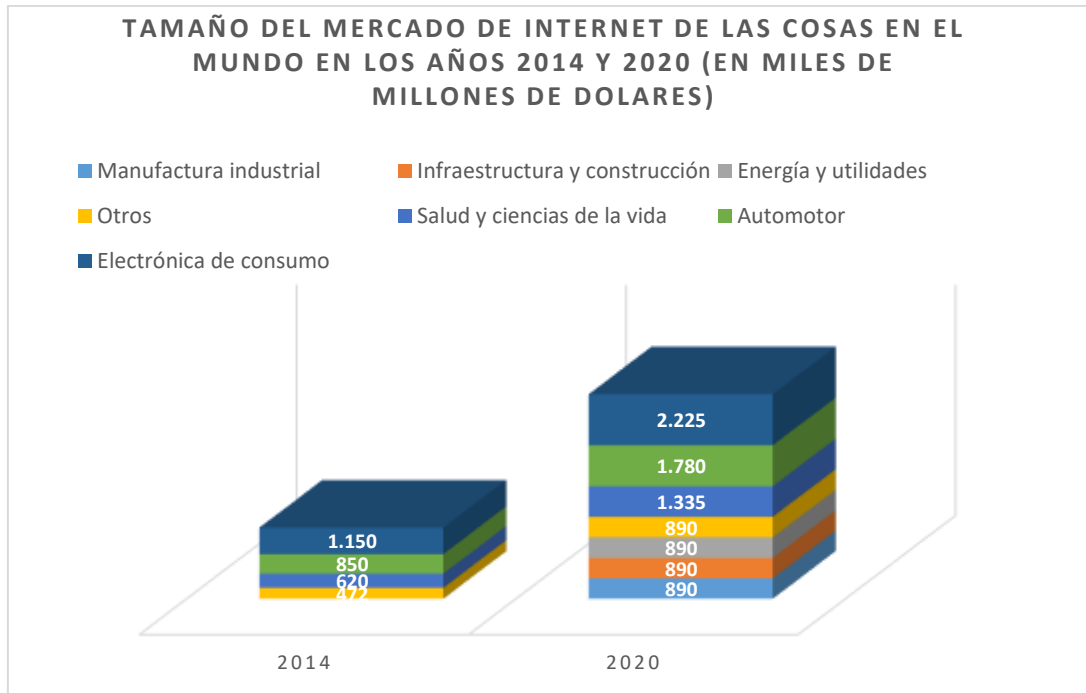


Figura 18: *Internet de las cosas en el mundo en 2014 y 2020.*

Fuente: (Statista, 2014)

Por lo consiguiente se seleccionaran e implementaran protocolos de IOT, para realizar las pruebas correspondientes y poder llevar a cabo la investigación, considerando elementos esenciales tales como, la seguridad, confidencialidad y privacidad de la información, ya que son temas primordiales para el uso de equipos biomédicos conectados a internet, ya que es fundamental conocer los tipos de ataques a los que están expuestos dichos equipos.

Siendo necesario comparar los protocolos MQTT y COAP puesto que son los más utilizados en la industria de la internet de las cosas según las encuestas realizadas por el autor (Lavinia, 2017), quien indica que los protocolos arriba mencionados son importantes para una adecuada investigación en el campo de la internet de las cosas.

En base a lo expuesto, esta investigación contribuirá a la línea de investigación de Tecnologías de la información, de la Facultad de ingeniería, Arquitectura y Urbanismo - Escuela de Ingeniería de Sistemas, por que conlleva a un pertinente estudio para determinar los niveles de seguridad que implementan los protocolos de capa de aplicación para internet de las cosas, contribuyendo a incentivar a otras

personas investiguen a cerca de la creciente tecnología en el ámbito del internet de las cosas.

1.6. Hipótesis

El protocolo de capa de aplicación COAP implementado en equipos biomédicos ofrece mayor seguridad ante ataques a dispositivos biomédicos conectados a internet de las cosas.

1.7. Objetivos

1.7.1. Objetivo General.

Comparar protocolos de capa de aplicación en IOT para medir el nivel de seguridad ante ataques en dispositivos biomédicos.

1.7.2. Objetivos específicos.

- I. Seleccionar protocolos más utilizados en la industria de la IOT.
- II. Identificar el ataque que más se efectúa en internet de las cosas.
- III. Implementar protocolos y efectuar ataques para estudiar su comportamiento.
- IV. Evidenciar resultados en base a las pruebas realizadas.

II. MATERIAL Y MÉTODOS.

2.1. Tipo y diseño de la investigación

2.1.1. Tipo de investigación.

Cuantitativa, por que consiste en cantidades numéricas de los indicadores los cuales están representados por porcentajes, siendo que los datos son recabados a través de mediciones representada mediante un adecuado análisis de métodos estadísticos, y a su vez continua, ya que se obtiene por medición o comparación de una unidad o patrón de medidas.

2.1.2. Diseño de investigación.

Cuasi- Experimental, ya que se manipulan las variables independientes para que luego de ello se observe sus efectos en lo que corresponde a las variables dependientes.

2.2. Población y muestra

2.2.1. Población.

La población de los protocolos de comunicación se ha considerado de acuerdo a la investigación de (Naik, 2017). En la que señala que a la actualidad existen muchos protocolos en capa de aplicación y que los más utilizados en la industria del IOT son: MQTT, COAP, AMQP y HTTP, referenciados debidamente en la **Tabla 6, 2.8.**

2.2.2. Muestra.

Para esta investigación es necesario definir la muestra de los protocolos que fueron seleccionados, de acuerdo a investigaciones realizadas por autores, como es el caso de (Hedi, Speh, & Sarabok, 2017), que consideran que los protocolos más frecuentes son MQTT y COAP considerándolos como los protocolos del futuro del IOT, donde los protocolos seleccionados son MQTT y COAP, el cual tiene amplia acogida en el uso de dispositivos e investigaciones recientes.

2.3. Variables, Operacionalización

Variable	Dimensiones	Indicadores	Unidad de medida.	Formula	Descripción.	Técnicas de recolección de datos
Variable independiente Protocolos MQTT y COAP.	-Protocolos.	Número de peticiones fallidas.	%	$FRT = \left(\frac{Ft}{Tt}\right) * 100$	FRT= Tasa de fracaso de las transacciones (respecto a los ataques). FT = Transacciones	Guía de observación
Variable dependiente Verificación de los ataques con respecto al rendimiento en seguridad.	Disponibilidad del dispositivo	Latencia media.	Segundos	$AL = \sum_{i=1}^N \left(\frac{T_{resp} - T_{req}}{N}\right)$	T req = Representa el momento en que comienza la transacción y Tresp = Representa el momento para obtener una respuesta completa. N = número de pruebas.	Guía de observación

2.3.1. Variable independiente.

Protocolos de comunicación en capa de aplicación.

2.3.2. Variable dependiente.

Determinar el nivel de seguridad.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

2.4.1. Abordaje Metodológico.

Hernández, Fernández & Baptista (2014) describe el abordaje metodológico cuasi – experimental, como el diseño que manipula la variable independiente para luego observar las consecuencias sobre una o más variables dependientes, por lo consiguiente, esta investigación cumple con la descripción citada y como variable independiente tenemos a los protocolos de comunicación que implementan seguridad, mientras que como variable dependiente tenemos los niveles de seguridad para cada protocolo.

2.4.2. Técnicas de recolección de datos.

Como técnica se considera la observación científica por lo que en un primer momento observaremos la realidad exterior con la finalidad de obtener datos que sirvan de fundamento para la investigación.

Como el presente trabajo es de tipo orientación científica e informativa, se eligió utilizar y/o aplicar el análisis documental basándose en el material bibliográfico como lo son los libros, artículos y tesis leídas.

2.4.3. Instrumentos de recolección de datos.

Guía de observación: ya que permitirá realizar un registro de datos importantes para su posterior análisis.

2.5. Procedimientos de análisis de datos

- a) En una primera fase se deben evidenciar los protocolos de capa de aplicación para el internet de las cosas con los que se realizarán las pruebas para la investigación.

- b) En una segunda fase se procede a implementar dichos protocolos utilizando una placa (microcomputadora) de bajo costo, Raspberry Pi, un módulo ESP8266, AUDUINO y otros componentes electrónicos, además de las librerías correspondientes.
- c) En una tercera fase se simula una red interna, se procede a realizar ataques utilizando software libre y luego se utilizan herramientas para la captura de paquetes.
- d) En una cuarta fase se procede a almacenar los datos en fichas de observación.
- e) Finalmente en la quinta fase se procede al análisis de los datos, calculando la latencia media y el porcentaje de paquetes caídos.

2.5.1. Análisis estadístico de los datos.

Se utilizaran hojas de cálculo de Excel para realizar un análisis estadístico con la determinación de analizar e interpretar los resultados alcanzados de la investigación, basados en los indicadores previamente descritos en la operacionalización de variables.

2.6. Criterios de Rigor Científico .

Los principios éticos de la presente investigación son:

1. **Honestidad e integridad**, puesto que no se manipulará la información, revelando los resultados de la misma manera como se obtuvieron.
2. **Confiabilidad**, puesto que las fuentes tomadas para realizar la investigación fueron rigurosamente verificadas.
3. **Responsabilidad**, porque se asumió un gran compromiso al desarrollar el la investigación, obteniendo como resultado el cumplimiento de las obligaciones, teniendo en cuenta muchas seriedad y prudencia.
4. **Respeto**, en este trabajo se tuvo en consideración las opiniones, comentarios y posiciones vertidas de otros autores, considerandose como valor supremo de toda sociedad.
5. **Autenticidad**, se es sincero y coherente al llevar a cabo la investigación puesto que se plasmó las ideas de acuerdo con la indagación realizada.

6. **Veracidad**, ya que el contenido fue constatado bajo un discernimiento valido, fidedigno, veráz y autentico, teniendo en cuenta normas de calidad, siendo muy cuidadoso con los conocimientos que se recabaron y actuando de buena fe.
7. **profesionalidad**, característica innata de una persona al desarrollar un trabajo, practicando la honradez, seriedad, aplicación, pericia y eficacia.

2.7. Criterios de rigor científico

Los principios que se consideran y practican son:

1. **Propiedad intelectual:** Guarda intereses de los autores de un producto intelectual en relación a teorías importantes y relevantes para a su respectiva protección.
2. **Derecho de autor:** Es el derecho que les corresponde a los investigadores, fundándose en derechos morales patrimoniales y éticos.
3. **Derecho de cita:** Fragmentos de los trabajos realizados con anterioridad se incluyen con la correspondiente cita para ayudar a fundamentar la investigación que se está realizando, también facilita la comprensión de los enunciados.
4. **Consentimiento informativo:** Son los aportes voluntarios que se realizan al investigar, recolectar y estudiar información para finalmente concluir en resultados fehacientes que avalen la investigación.

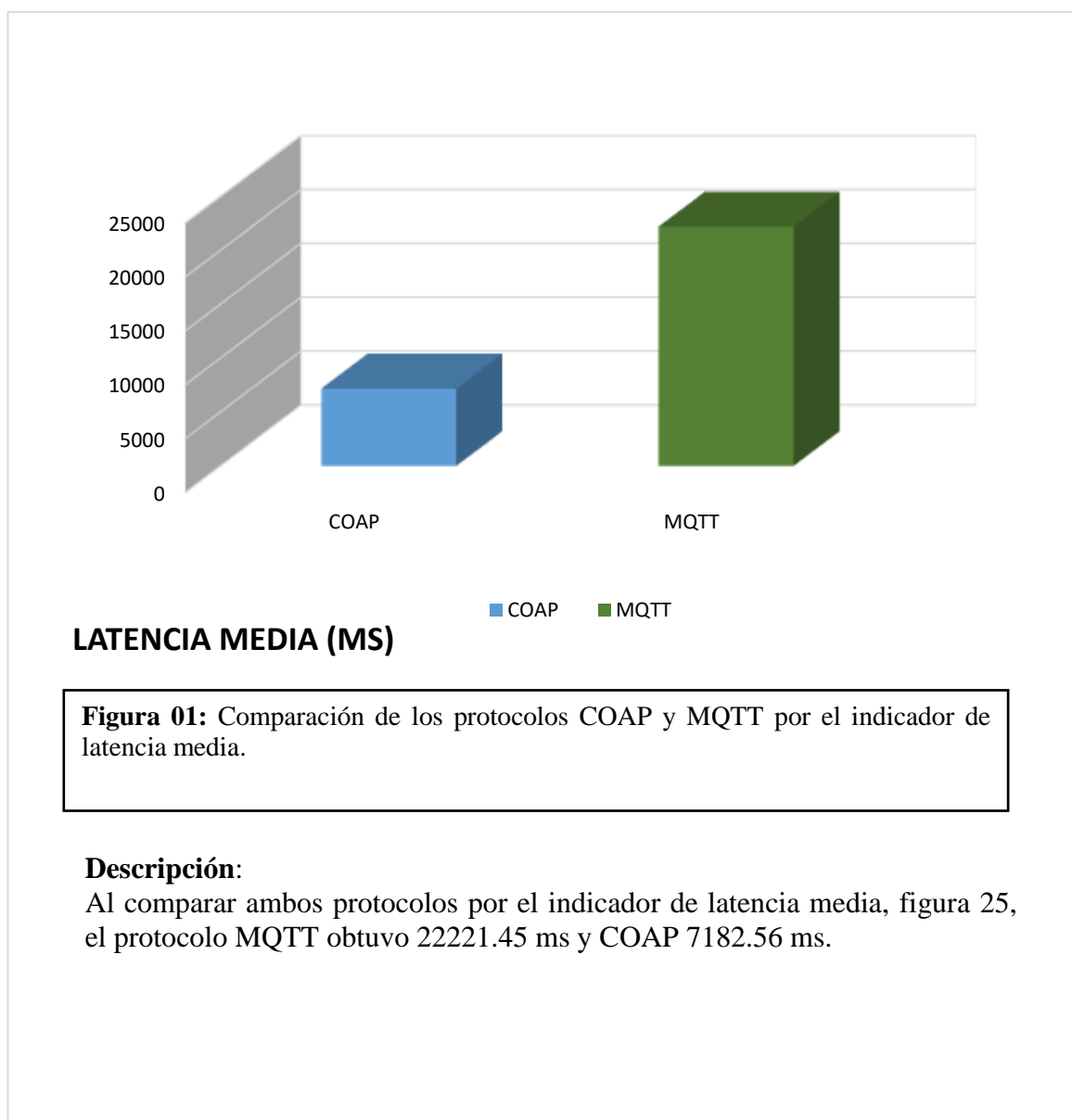
III. RESULTADOS

3.1. Tablas y Figuras

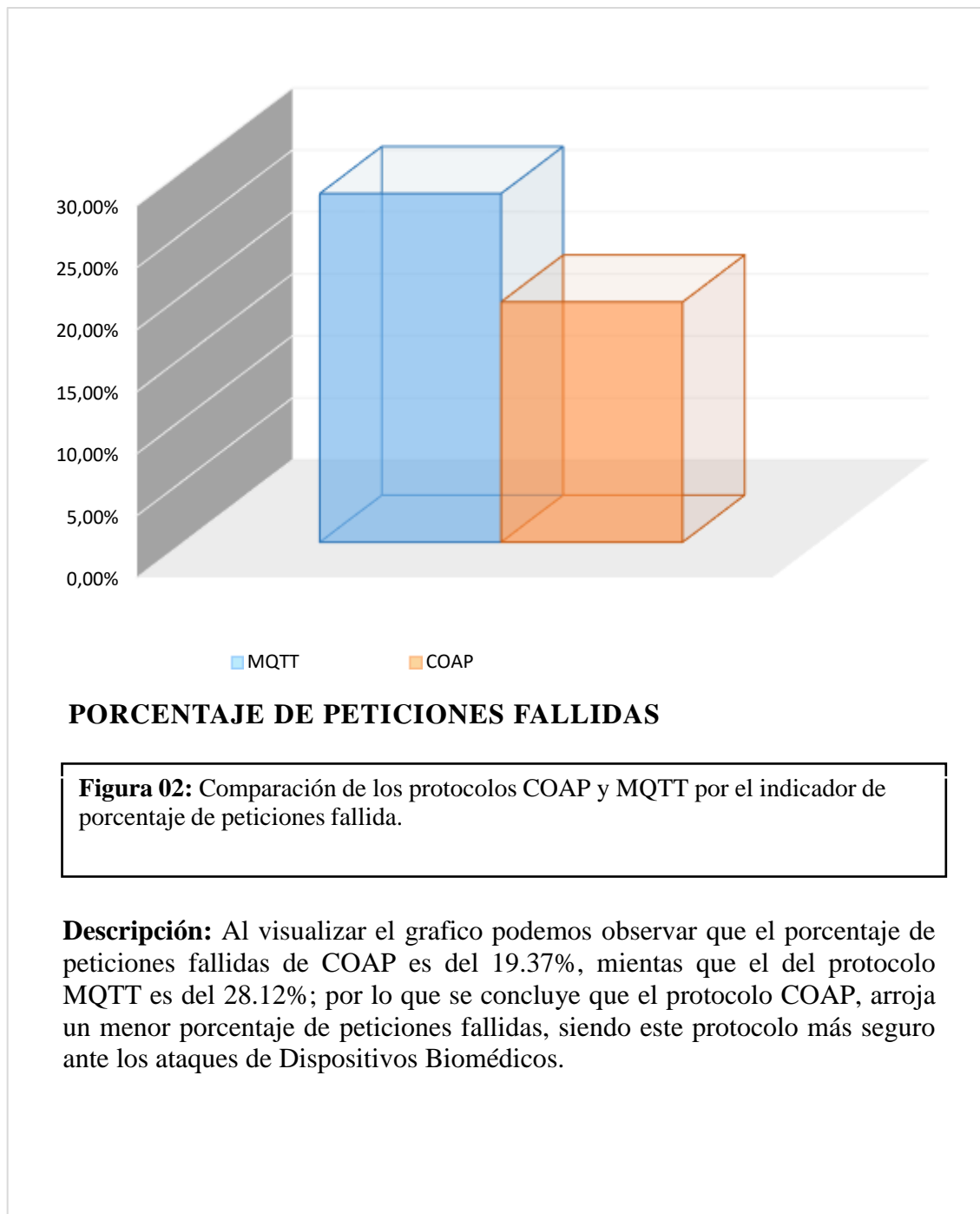
3.1.1. Descripción de resultados.

3.1.1.1. Resultados respecto a:

Al comparar los protocolos MQTT y COAP por el indicador de latencia media, figura 24, el protocolo MQTT obtuvo una latencia media de 22221.45 ms. mientras que COAP 7182.4 ms, por consiguiente el protocolo COAP presenta menor tiempo de respuesta en las peticiones frente a MQTT durante ataques de DOS.



3.1.1.2. Resultados respecto a :



3.2. Discusión de resultados

Al seleccionar los protocolos más utilizados en la Industria de la IOT, se encontró a los protocolos COAT y MQTT; sin embargo uno de ellos brinda mayor seguridad, confidencialidad y privacidad al implementarlo en los equipos biomédicos, por lo que se requiere identificar el ataque que más se efectúa en el ámbito del internet de las cosas.

Conforme a los resultados obtenidos podemos observar que, el protocolo COAP implementado en equipos biomédicos ofrece mayor seguridad ante ataques a dispositivos biomédicos conectados a internet de las cosas, ya que, al comparar los protocolos de capa de aplicación en IOT tales como COAT y MQTT por el indicador de latencia media, el protocolo MQTT obtuvo 22221.45 ms y COAP 7182.56 ms; siendo que al tener una menor latencia este último, se obtiene una mayor especificidad y eficiencia.

Asimismo, al evidenciar el porcentaje de peticiones fallidas se obtiene de esta que el protocolo COAP arroja un porcentaje del 19.37%, mientras que el protocolo MQTT un 28.12%; por lo que concluye que el protocolo COAP, arroja un menor porcentaje de peticiones fallidas, siendo este más seguro ante los ataques de Dispositivos Biomédicos, conforme lo señalado líneas arriba, brindando una ventaja amplia en cuanto al MQTT.

En este sentido, los resultados obtenidos confirman la hipótesis formulada, la cual señala que, el protocolo de capa de aplicación COAP implementado en equipos biomédicos ofrece mayor seguridad ante ataques a dispositivos biomédicos conectados a internet de las cosas, siendo ello así, se tiene que, al implementar los protocolos y efectuar los ataques se llegó a la conclusión que el protocolo COAP, es aquel que brinda mayor seguridad frente a ataques DOS, respecto al protocolo MQTT en internet de las cosas, utilizando equipos biomédicos.

3.3. Aporte Práctico

En la industria del IOT existen muchos protocolos, los más utilizados según Nitin 2016, son los que se describen en la **Tabla 5**.

Para esta investigación se estudiaron e implementaron dos protocolos, MQTT y COAP de acuerdo a la clasificación realizada previamente en la etapa de selección de la muestra.

Como se puede observar, MQTT es un protocolo que fue desarrollado por IBM y que posteriormente en el año 2011, fue publicado sin derechos de autor, donando el código fuente al proyecto Eclipse, seguidamente en el 2013 se estableció como un estándar de Organization for the Advancement of Structured Information Standards (OASIS). En comparativa, COAP fue publicado en el año 2014 por el IETF basado en HTTP para ser utilizado libremente en proyectos de redes restringidas y comunicación machine to machine (M2M).

El patrón de mensajería para MQTT es de publicación - suscripción y en su capa de transporte utiliza TCP, mientras que COAP utiliza cliente - servidor y UDP.

Tabla 5: *Protocolos de capa de aplicación en IOT.*

	MQTT	COAP
Establecido	1999 (estándar OASIS 2013)	2014
Patrón de mensajería	Publicar / suscribirse a través de intermediario de mensajes.	Solicitud / respuesta (cliente-servidor).
Transporte	Protocolo de Control de Transmisión (TCP).	Protocolo de datagramas de usuario (UDP).
Seguridad	SSL / TLS sobre TCP.	DTLS sobre UDP.
Fortalezas	La arquitectura del corredor puede simplificar la gestión; TCP y las opciones de calidad de servicio permiten una entrega robusta de mensajes.	Rápido y eficiente debido a la dependencia de UDP de baja sobrecarga; El modelo REST es acogedor para los desarrolladores y proporciona un descubrimiento de recursos integrado.

Fuente: (Naik, 2017).

En la **tabla 6** se realizó una comparativa de los protocolos MQTT y COAP, debido a que son los protocolos seleccionados como muestra para realizar la presente investigación, de acuerdo a: Su arquitectura, el tamaño de cabecera (en bytes), calidad de servicio (Qos), protocolos a nivel de capa de transporte UDP y/o TCP, año de publicación y finalmente acuse de recibo (ACK).

Tabla 6: Comparación de protocolos MQTT y COAP.

PROTOCOLO	ARQUITECTURA	TAMAÑO DE CABECERA	QoS	UDP /TCP	AÑO DE PUBLICACIÓN	ACK
MQTT	Publicación suscripción	2 bytes	Si	TCP	1999	No
COAP	cliente servidor	4 bytes	Si	UDP	2014	Si
AMQP	Publicación suscripción	8 bytes	si	TCP	2004	Si
XMPP	Publicación suscripción	0 bytes	No	TCP	1999	Si
DDS	Data-Centric Publish-Subscribe (DCPS) y Data-Local	0 bytes	Si	TCP/UDP	2004	si

Fuente: (Elaboración Propia)

3.3.1. Identificación de los ataques más efectuados.

En la tabla 7 se muestran los ataques en internet de las cosas por capas:

CAPA	Amenazas principales
Nivel de Aplicación	Fuga de datos DOS Attacks Inyección de código malicioso
Nivel de Transporte	Ataques de enrutamiento DOS Attacks Ataques de tránsito de datos
Nivel de Percepción	Ataques físicos Interpretación DOS Attacks Ataques de enrutamiento (e.g en WSN, RSN) Ataques de tránsito de datos (e.g en WSN, RSN)

Tabla 7: Amenazas de ataques según capas en IOT.

Fuente: (Frustaci, Pace, Aloï, & Fortino, 2018)

Un atacante puede inundar el servidor o el dispositivo IoT con un gran número de peticiones falsas o enviar información para desencadenar diversas peticiones, lo que causa el agotamiento debido a restricción de recursos, ya sea memoria, ancho de banda, CPU, espacio en disco, etc. y tal como lo ha demostrado (Daud et al., 2017), el que comprueba que el internet de las cosas es susceptible a ataques DOS.

En cuanto a la selección de la muestra de ataques, se consideraron los ataques (denegación de servicio) DOS que de acuerdo a la investigación realizada por (Daud et al., 2017), considera que los dispositivos de internet de las cosas son susceptibles a ataques DOS.

Mientras que (Reyes, 2012), expresa que es uno de los ataques más utilizados, con pocas opciones de defensa, ya que los crackers se dedican a consumir ancho de banda y recursos, lo que les permite saturar a la víctima con tantos paquetes les sea posible durante un periodo.

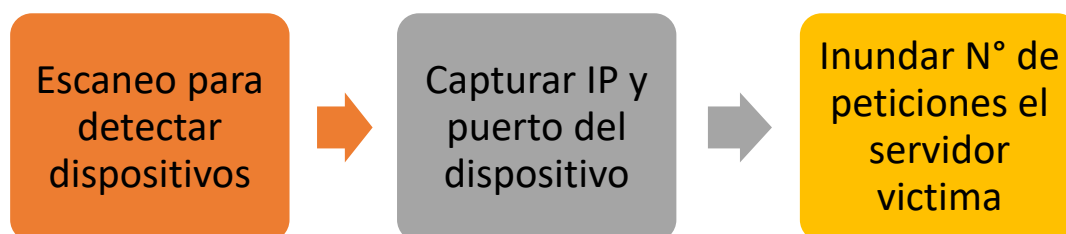


Figura 19 Flujo de ejecución de ataques DOS

Fuente: *Elaboración propia*

3.3.2. Implementación de protocolos.

3.3.2.1. Protocolo MQTT.

Para la implementación de los protocolos antes seleccionados en internet de las cosas, se utilizaron los siguientes componentes electrónicos:

- Placa Arduino nano,
- Microcontrolador ESP8266,
- Microcomputador Raspberry Pi modelo B,

- Sensor de pulsos cardiacos

Cabe señalar que los dispositivos utilizados en la presente investigación, son ampliamente requeridos por los investigadores e industria para construir e implementar dispositivos con internet de las cosas, por lo consiguiente es de necesidad describir sus características técnicas, las que se encuentran debidamente detalladas en el anexo 4.2, capítulo IV. Ya que los dispositivos e incluso la integridad de quien los manipulen puedan verse afectados debido a un mal uso o al no tener los cuidados necesarios del caso.

Por otro lado se utilizaron librerías como: Mosquito MQTT para la implementación del bróker (servidor) y cliente, Adafruit Sensor Master para capturar la información del sensor, también se utilizó el IDE de Arduino en su versión 1.6.10, donde se incluyeron los drivers tanto de Arduino nano y ESP8266 para su programación.

Lo primero que se debe realizar es la configuración del bróker, utilizando una memoria micro SD para la instalación del sistema operativo en la Raspberry Pi, **ver anexo 1, capítulo IV.**

Los dispositivos que se utilizan en la investigación se basan en el uso y la seguridad en dispositivos biomédicos (marcapasos), que uno de los precedentes se dio a finales de 2017, casi medio millón de personas usuarias de los dispositivos de la marca 'St Jude Medical' (SJM) fueron alertados por la Administración de Alimentos y Medicamentos de los Estados Unidos (FDA) que se encontraron serios agujeros de seguridad.

Teniéndose configurado el broker con la librería Mosquitto MQTT, seguido de ello se debe configurar y publicar un topic para obtener los valores que envíe el sensor a través del microcontrolador ESP8266.

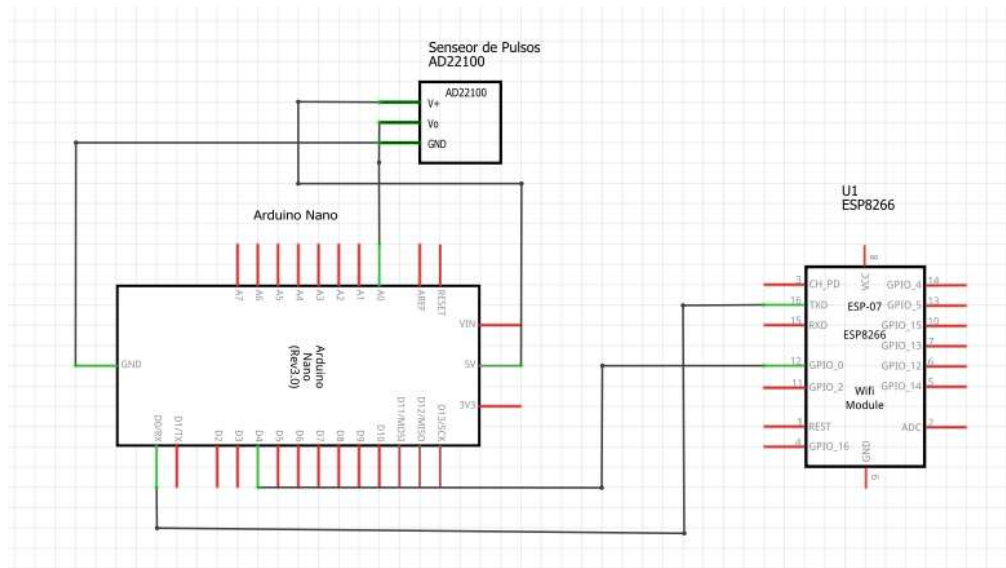


Figura 20 Diagrama de conexión para MQTT

Fuente: Elaboración propia

En la Figura 21 se logra apreciar la conexión de los dispositivos antes mencionados, los cuales están conectados a una placa de pruebas o también conocida como protoboard, los valores que se obtienen a través del sensor, enviándose al ESP8266 a través de Arduino modelo Nano y posteriormente se pueden apreciar los valores obtenidos, tanto en la consola del IDE de Arduino como de cualquier suscripción, en este caso dentro de los equipos conectados a la red como se puede ver en las figuras 22 y 23, para lograr acceder a los valores captados previamente por el sensor, se debe suscribir mediante el comando, para este caso: ***“mosquitto_sub localhost -t outTopic”***, ejecutado mediante línea de comandos desde una microcomputadora (Raspberry pi) y también desde la consola de Arduino IDE.

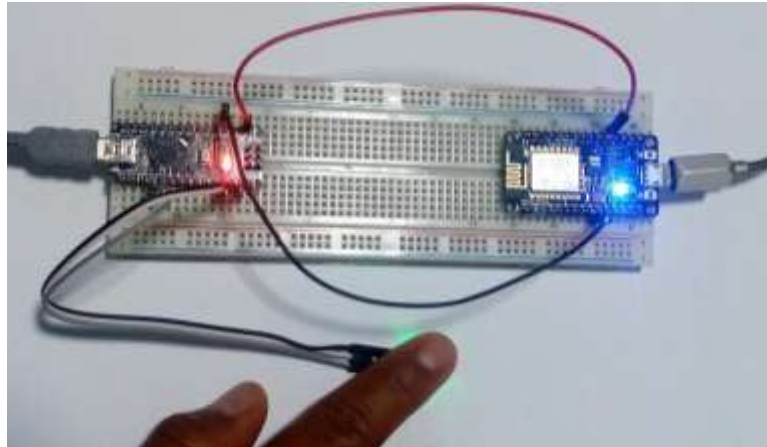


Figura 21 Pruebas de pulso cardiaco.

Fuente: Elaboración propia

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ mosquitto_sub -h localhost -t outTopic  
69  
70  
71  
73  
74  
73  
72  
70  
70  
70  
69
```

Figura 22 Mostrando los datos del sensor en la consola del servidor Raspberry

Fuente: Elaboración propia

```
COM4  
69  
70  
70  
69  
69  
70  
71  
73  
74  
73  
72  
70  
70  
70  
69
```

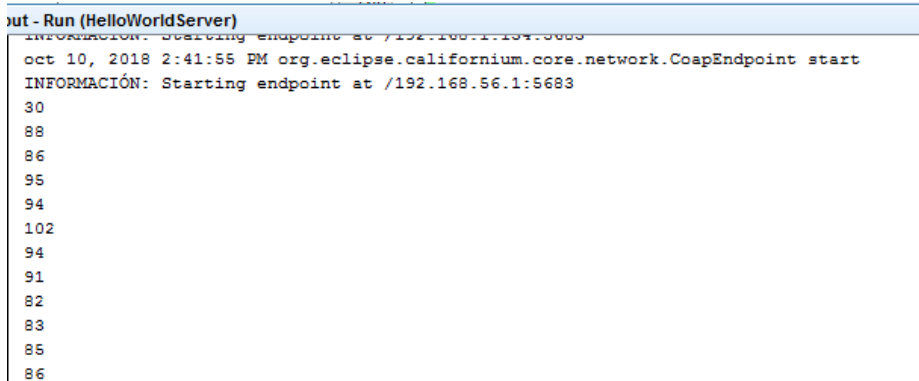
Figura 23 Mostrando los datos del sensor a través de la consola de Arduino.

Fuente: Elaboración propia

3.3.2.2. Protocolo COAP.

Para implementar el protocolo COAP, se utilizaron los mismos componentes electrónicos que en MQTT, para este caso se utilizó la librería californium que pertenece a Eclipse, es de uso libre y permite la implementación de cliente - servidor, entre otras ventajas, se procedió a descargar desde el repositorio y luego a configurar.

A continuación se programa Arduino modelo Nano, utilizando su IDE en su versión 1.6.10, para que lea la señal del sensor de pulsos cardiacos, además de realizar las conexiones físicas correspondientes, en seguida se programó y configuró el microcontrolador ESP8266 para finalmente obtener resultados como se aprecia en la figura 24



```
out - Run (HelloWorldServer)
INFORMACIÓN: Starting endpoint at /192.168.1.104:5683
oct 10, 2018 2:41:55 PM org.eclipse.californium.core.network.CoapEndpoint start
INFORMACIÓN: Starting endpoint at /192.168.56.1:5683
30
88
86
95
94
102
94
91
82
83
85
86
```

Figura 24 Mostrando los datos del sensor a través de la consola de Arduino.

Fuente: Elaboración propia

3.4. Evidenciar resultados en base a las pruebas realizadas

3.4.1. Medición de porcentaje de peticiones fallidas de los protocolos MQTT y COAP.

Se realizaron un total de 160 pruebas con diferentes números de paquetes enviados, obteniendo un promedio con respecto a estos. De esta forma se obtienen los datos a través de la ficha de observación y seguidamente se clasificaron por medio de:

- a) FRT= Tasa de fracaso de las transacciones.
- b) FT= Transacciones fallidas.
- c) Tt= Número total de transacciones

$$FRT = \left(\frac{Ft}{Tt}\right) * 100$$

Resultados respecto a COAP (ver Anexo III)

$$FRT = \left(\frac{31}{160}\right) * 100$$

$$FRT = 19.37\%$$

Resultados respecto a MQTT (ver Anexo IV)

$$FRT = \left(\frac{45}{160}\right) * 100$$

$$FRT = 28.12\%$$

3.4.2. Medición de la latencia media con respecto MQTT y COAP respectivamente.

T req = Representa el momento en que comienza la transacción y

Tresp = Representa el momento para obtener una respuesta completa.

N = número de pruebas.

Se reemplazan datos de acuerdo a los resultados obtenidos en los anexos:

$$AL = \sum_{i=1}^N \left(\frac{Tresp - Treq}{N}\right)$$

Resultados con respecto a COAP (ver Anexo III)

$$AL = \sum_{i=1}^{31} \left(\frac{488.91 - 0}{31}\right)$$

$$AL = \sum_{i=1}^{31} (14.48)$$

$$AL = \sum_{i=1}^{31} (14.48)$$

$$AL = 496(14.48)$$

$$AL = 7182.56 \text{ ms}$$

Resultados con respecto a MQTT (ver Anexo IV)

$$AL = \sum_{i=1}^{45} \left(\frac{996.50 - 0}{45} \right)$$

$$AL = \sum_{i=1}^{45} (21.47)$$

$$AL = \sum_{i=1}^{45} (21.47)$$

$$AL = 1035(21.47)$$

$$AL = 22221.45 \text{ ms}$$

IV. CONCLUSIONES Y RECOMENDACIONES.

4.1. Conclusiones

1. Es de precisar que el autor Naik en su investigación Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP, señala que, *“los protocolos en capa de aplicación MQTT y COAP son los más utilizados en la industria del internet de las cosas”*; arrojando resultados optimos que, ayudaron a la elección de los mismos, los cuales fueron implementados en la presente investigación, para determinar la seguridad con respecto a la latencia media e índice de peticiones fallidas en dispositivos biomédicos para IOT, por lo consiguiente es necesario dicha implementación y/o utilización de tecnología.
2. Se analizaron los ataques que más se efectúan en IOT, de acuerdo a las estadísticas publicadas por organismos internacionales relacionados a investigaciones en seguridad, es así que se eligió el ataque DOS para su implementación con respecto a los protocolos seleccionados y medir la seguridad de cada uno, respectivamente.
3. Para la implementación de protocolos se utilizaron dispositivos electrónicos y herramientas, así como librerías propias de cada protocolo. También se utilizó software libre (Kali Linux) para efectuar los ataques y herramientas para analizar y estudiar el comportamiento, rendimiento de cada protocolo, por lo consiguiente se utilizaron fichas de observación para posteriormente analizar los resultados de acuerdo a los indicadores.
4. Finalmente se concluye que el protocolo COAP presentó un índice de peticiones fallidas del 19.37%, mientras que MQTT arrojó 28.12%, el cual fue uno de los indicadores en estudio. Así también se ensayó la latencia que se genera luego de efectuar los ataques a ambos protocolos, lo que dio como resultado que el protocolo COAP presenta una latencia media de 7182.56 mili segundos a diferencia de 22221.45 mili segundos para MQTT, por consiguiente COAP es el protocolo que de acuerdo a los resultados de esta investigación es más seguro frente a ataques DOS que el protocolo MQTT.

4.2. Recomendaciones.

1. Con base a la investigación realizada, se pone en conocimiento que el uso de tecnologías IOT trae consigo muchos retos en cuanto a la seguridad, los que a su vez deben ser tomados en cuenta, tanto desde los fabricantes hasta los usuarios y/o consumidores finales, siendo realmente importante realizar estudios posteriores sobre la tecnología de la Internet de las Cosas, trayendo consigo indagaciones en cuanto a su seguridad e importancia de este en tanto se presenten ataques que afecten una adecuada conservación de información.
2. Con la creciente demanda en el uso de los objetos conectados a internet y específicamente en el campo de la medicina, tal como lo señalan las investigaciones previamente realizadas, es de gran importancia elegir la mejor opción en cuanto a la seguridad, esta es el protocolo COAP, el cual brinda mayor protección frente a ataques que se puedan sufrir por cibercriminales.
3. Un aspecto que también debe tomarse en cuenta es el adoptar buenas prácticas en el uso de dispositivos conectados a internet, ya sea el asignar un usuario y contraseña distinta al de fábrica con el cual viene el dispositivo, actualizar periódicamente el firmware del dispositivo en caso tuviera el soporte correspondiente.
4. Se recomienda COAP para utilizarlo en capa de aplicación en dispositivos biomédicos, obteniendo mayor seguridad frente a ataques cibercriminales, que se dan con ataques DOS, para resguardar la data y velar por bienestar de las personas que utilizan los dispositivos biomédicos; por lo que se requiere asegurar la información a través de este mecanismo, obteniendo mayor capacidad de procesamiento, almacenamiento y protección, es por ello que es de necesidad conocer el protocolo COAP y además identificar su comportamiento de acuerdo a aquellos ataques frecuentes.

REFERENCIA

1. Abouzakhar, N. S., Jones, A., & Angelopoulou, O. (2017). *Internet de las cosas de seguridad: Una revisión de los riesgos y Las amenazas a la Salud Sector*, 373–378.
2. *Bilgisayar Et Al (2017) Security attacks on IoT.*
3. Boronat, S. y Montagud, C. (2012). *El nivel de red en el modelo de interconexión de redes basado en capas*. Recuperado <http://ebookcentral.proquest.com/lib/bibsipansp/detail.action?docID=3206252&query=capas+del+modelo+osi>
4. Chavéz, Jorge. (2019). *¿Ya conoces el protocolo IPV6?*. Treves. Recuperado de <https://treebes.com/ya-conoces-el-protocolo-ipv6/>
5. Daeyoung, H. (2017). *Funciones de seguridad de red SDN-basado de La mitigación eficaz ataque DDoS.*
6. Daniel, C. (2018). *An evaluation of the internet safety of popular things The protocols for manufacturers.*
7. Dejana Et. Al (2015) *COAP protocol for Web-based monitoring in IOT healthcare applications.*
8. Departamento de Investigación de Statista. (2014). *Tamaño del mercado de Internet de las cosas en todo el mundo en 2014 y 2020, por industria (en miles de millones de dólares estadounidenses)*. Statista. Recuperado de <https://www.statista.com/statistics/512673/worldwide-internet-of-things-market/>
9. Eastman, D. (2017). *A simulation study to detect internet attacks of Things.*
10. Fotiou, N., Kotsonis, T., Marias, G. F., Polyzos, G. C., Informática, D. De, & Fotiou, G. (2016). *Control de acceso a la Internet de las cosas*, 29–38.
11. Harshal, S. (2017). *Internet of things: existing protocols and technological challenges in security.*
12. Izquierdo & Tafur (2017) *mecanismos de seguridad para contrarrestar ataques informáticos en servidores web y bases de datos.*
13. Moore y Zouridakis, (2014). *Biomedical Technology and devices. Washington.*

14. Mukrimah N. Et. Al. (2016) *Internet of Things (IO): Security Taxonomy attacks .Tailandia*. Fotiou, N., Kotsonis, T., Marias, G. F., Polyzos, G. C., Informática, D. De, & Fotiou, G. (2016). Control de acceso a la Internet de las cosas, 29–38.
15. Frustaci, M., Pace, P., Aloï, G., & Fortino, G. (2018). Evaluating critical security issues of the IoT world: Present and future challenges. *IEEE Internet of Things Journal*, 5(4), 2483–2495. <https://doi.org/10.1109/JIOT.2017.2767291>
16. Hedi, I., Špeh, I., & Šarabok, A. (2017). IoT network protocols comparison for the purpose of IoT constrained networks. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings*, 501–505. <https://doi.org/10.23919/MIPRO.2017.7973477>
17. Internacional, C., Electr, H. S., & Electr, D. A. (2017). Internet de las cosas : los protocolos existentes y tecnológicos Desafíos en Seguridad.
18. Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings*. <https://doi.org/10.1109/SysEng.2017.8088251>
19. Parra, M., Arias, R., y La Torre, A. (2003). *Procedimientos y Técnicas en el paciente Crítico. Barcelona*.
20. Ramos A. y Ribagorda G. (2004). *Avances en criptología y seguridad de la información. España*. Recuperado https://books.google.com.pe/books?id=ibSu6896I_YC&printss=frontcover&hl=es#v=onepage&q&f=false.
21. Sana B. Et. Al (2017). *A survey on attacks in internet of Things based networks*.
22. Tara S & Raj J. (2017). *Networking protocols and standards for internet of thing*.
23. United, K. (2017). *Defence School of Communications and Information Systems, Ministry of Defence*.
24. Vijay, S (2016). *Lightweight safety protocol for RFID chip removal on the Internet of objects (IO) Applications*.

25. Xinzheng, F. (2017), *Security analysis of Simple Network Management Protocol based IEEE P21451 Internet of Things*.

ANEXOS

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N° 1638-2020/FIAU-USS
Pimentel, 3 de agosto de 2020

VISTO:

El Acta de reunión N°1207-2020, de fecha 12 de julio de 2020 del Comité de Investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS, para la ejecución de la Tests: "COMPARACIÓN DE PROTOCOLOS DE COMUNICACIÓN EN INTERNET DE LAS COSAS, DETERMINANDO EL NIVEL DE SEGURIDAD ANTE ATAQUES EN DISPOSITIVOS BIOMÉDICOS", presentado por ZEÑA ZEÑA EDINSON OMAR, del Programa de estudios INGENIERÍA DE SISTEMAS, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48º que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21º señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24º señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 23º señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, en el Acta de reunión N°1207-2020 de fecha 12 de julio de 2020, del Comité de Investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS, se indica entre los acuerdos la aprobación del tema de la Tests denominado "COMPARACIÓN DE PROTOCOLOS DE COMUNICACIÓN EN INTERNET DE LAS COSAS, DETERMINANDO EL NIVEL DE SEGURIDAD ANTE ATAQUES EN DISPOSITIVOS BIOMÉDICOS" de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de ZEÑA ZEÑA EDINSON OMAR en condición de egresado, del Programa de estudios INGENIERÍA DE SISTEMAS.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1º: APROBAR, el tema del Tests denominado "COMPARACIÓN DE PROTOCOLOS DE COMUNICACIÓN EN INTERNET DE LAS COSAS, DETERMINANDO EL NIVEL DE SEGURIDAD ANTE ATAQUES EN DISPOSITIVOS BIOMÉDICOS", perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de ZEÑA ZEÑA EDINSON OMAR, del Programa de estudios INGENIERÍA DE SISTEMAS.

ARTÍCULO 2º: ESTABLECER, que la inscripción del Título del Tests se realice a partir de emitida la presente resolución y tendrá una vigencia de dos (02) años.

ARTÍCULO 3º: DEJAR SIN EFECTO, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE



Dr. Mario Fernando Salas Hualde
Decano - Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPAC



MSc. María Estela Ballester
Decana Académica / Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPAC

1. Materiales para la implementación de protocolos de aplicación en IOT

Para implementar los protocolos arriba mencionados se utilizaron componentes electrónicos, los cuales son más conocidos y trabajados en la industria del internet de las cosas:

1.1. Microcontrolador Arduino Nano.

Arduino Nano es una pequeña y completa placa basada en el ATmega328 (Arduino Nano 3.0) se utiliza conectándolo a un protoboard. Posee de más o menos la misma funcionalidad que el Arduino uno, pero con una presentación diferente. No posee conector para alimentación externa, y se puede programar a través de un cable USB Mini-B

Especificaciones técnicas:

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Voltaje de alimentación (Recomendado): 7-12V
- I/O Digitales: 14 (6 son PWM)
- Memoria Flash: 32KB
- EEPROM: 1KB
- Frecuencia de trabajo: 16MHz
- Dimensiones: 0.73" x 1.70"

1.2. Microcomputador.

Conocido como Raspberry, es un microcomputador de bajo costo, diseñado para múltiples aplicaciones, tales como prototipos, proyectos estudiantiles, automatizaciones, robótica, servicios web, o para ser el centro de control de un sistema remoto, etc..

Esta tarjeta integrada requiere de una memoria micro SD para instalar el sistema operativo basado en linux, y guardar los datos de las aplicaciones. Posee puertos USB donde se conectan el teclado, mouse, cuenta con un puerto HDMI para conectar un monitor, la conexión de la red se puede realizar mediante su puerto Ethernet o por wifi de acuerdo al modelo.

Se alimenta de energía mediante con una fuente de voltaje DC de 5 V a 2.5 Amperios.

Posee un puerto conocido como GPIO, de 40 pines, donde se tiene acceso para conectar entradas y salidas, para manejar directamente otro tipo de dispositivos electrónicos o sensores.

Especificaciones técnicas:

- Procesador a 1,2 GHz de 64 bits con cuatro núcleos ARMv8.
- 1GB de Memoria.
- 802.11n Wireless LAN.
- Bluetooth 4.1.
- Bluetooth Low Energy (BLE).
- 4 puertos USB.
- 40 pines GPIO.
- Puerto Full HDMI.
- Puerto Ethernet.
- Conector combo compuesto de audio y vídeo de 3,5 mm.
- Ranura para tarjetas microSD (push-pull).
- Núcleo de gráficos VideoCore IV 3D.
- Dimensiones de placa de 8.5 por 5.3 cm.

1.3. Placa de pruebas o protoboard.

El Protoboard sirve como base para montar componentes electrónicos y formas circuitos sin tener que fabricar un placa impresa, cuenta con 2 líneas de energía, 30 columnas y 10 filas - un total de 400 puntos.

Todos los pines están espaciados por un estándar de 0,1". Los dos conjuntos de cinco filas están separadas por aproximadamente 0,3". Posee un auto-adhesivo en la parte posterior y pueden conectarse varios protoboards juntos, tantos como se requiera.

Consta de tres partes:

A) Canal central: Es la región localizada en el medio del protoboard, se utiliza para colocar los circuitos integrados.

B) Buses: Los buses se localizan en ambos extremos del protoboard, se representan por las líneas rojas (buses positivos o de voltaje) y azules (buses

negativos o de tierra) y conducen de acuerdo a estas, no existe conexión física entre ellas. La fuente de poder generalmente se conecta aquí.

C) Pistas: La pistas se localizan en la parte central del protoboard, se representan y conducen según las líneas rosas.

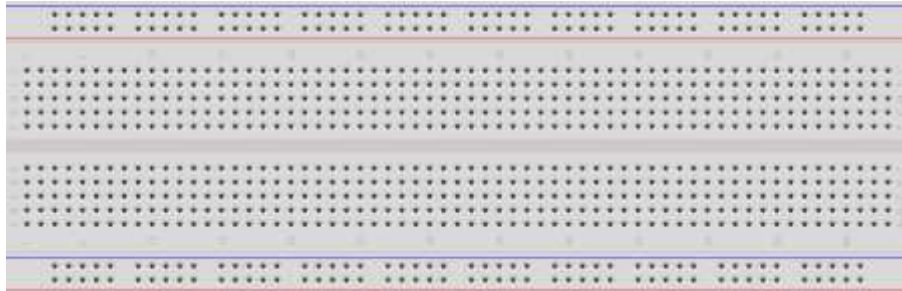


Figura 26: Placa de pruebas

1.4. Sensor de pulsos cardiacos



Figura 27: Sensor de pulsos

El sensor de pulsos cardíacos utiliza un voltaje entre 3V a 5V DC para su alimentación.

El dispositivo funciona del siguiente modo: un led de color verde emite luz que al entrar en contacto con nuestro dedo índice o el lóbulo del oído refleja cierta cantidad de luz, el flujo de sangre hace que la cantidad de luz reflejada cambie de acuerdo al pulso cardíaco. La luz reflejada es detectada por el sensor de luz APDS-9008, que convierte el flujo de luz en un voltaje analógico.

Sus características son:

- Voltaje de Operación: 3.0V – 5.5V DC
- Consumo corriente: 20mA máx.
- Sensor: APDS-9008

- Opamp: MCP6001
- Longitud de cable: 20cm
- Cables: GND, VCC, Señal

1.5. NodeMCU ESP8266

NodeMCU es una tarjeta de desarrollo similar a un Arduino, especialmente diseñada para el uso en el Internet de las cosas (IoT). Consta de un chip altamente integrado, diseñado para las necesidades de un mundo conectado. Integra un potente procesador con Arquitectura de 32 bits y conectividad Wifi.

Para el desarrollo de aplicaciones se puede elegir entre los lenguajes Arduino y Lua, además de hacer uso de toda la información sobre proyectos y librerías disponibles en internet.

NodeMCU viene con un firmware pre-instalado el cual nos permite trabajar fácilmente y está diseñada especialmente para trabajar en placa de pruebas (protoboard). Posee un regulador de voltaje en placa que le permite alimentarse directamente del puerto USB. Los pines de entradas/salidas trabajan a 3.3V. El chip CP2102 se encarga de la comunicación USB-Serial.

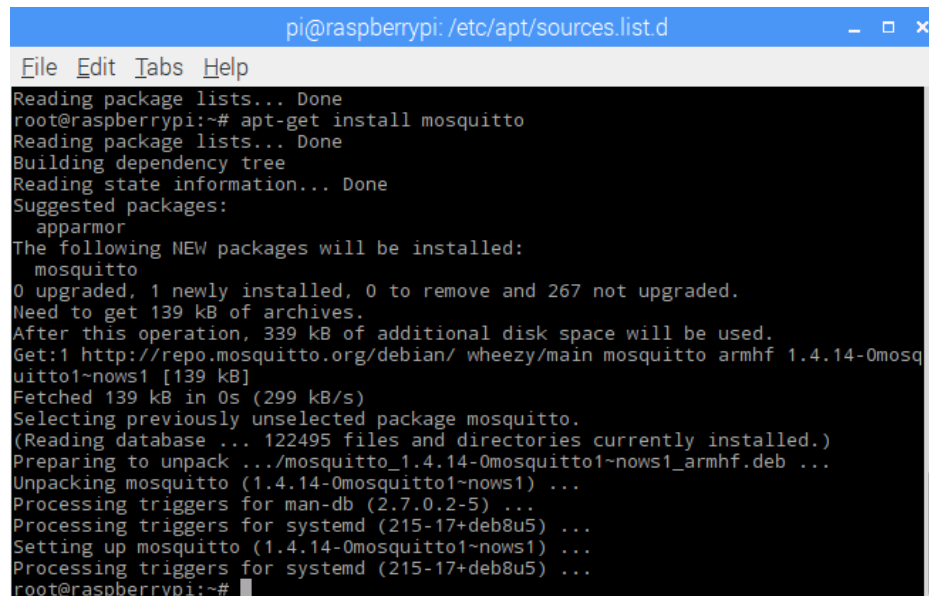
ESPECIFICACIONES TÉCNICAS

- Voltaje de Alimentación (USB): 5V DC
- Voltaje de Entradas/Salidas: 3.3V DC
- SoC: ESP8266 (Módulo ESP-12)
- CPU: Tensilica Xtensa LX3 (32 bit)
- Frecuencia de Reloj: 80MHz/160MHz
- Instruction RAM: 32KB
- Data RAM: 96KB
- Memoria Flash Externa: 4MB
- Pines Digitales GPIO: 17 (pueden configurarse como PWM a 3.3V)
- Pin Analógico ADC: 1 (0-1V)

- UART: 2
- Chip USB-Serial: CP2102
- Certificación FCC
- Antena en PCB
- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Stack de Protocolo TCP/IP integrado
- PLLs, reguladores, DCXO y manejo de poder integrados
- Potencia de salida de +19.5dBm en modo 802.11b
- Corriente de fuga menor a 10uA
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Consumo de potencia Standby < 1.0mW (DTIM3)
- INTERFACE CORRESPONDIENTE
- SDIO 2.0, SPI, UART
- Integra RF switch, balun, 24dBm PA, DCXO y PMU
- Posee un procesador RISC, memoria en chip e interface para memoria externa
- Procesador MAC/Baseband integrado
- Interface I2S para aplicaciones de audio de alta calidad
- Reguladores de voltaje lineales "low-dropout" en chip
- Arquitectura propietaria de generacion de clock "spurious free"
- Módulos WEP, TKIP, AES y WAPI integrados

2. Instalación de mosquito - MQTT.

Como siguiente paso se debe instalar el mosquito bróker con el siguiente comando: `apt-get install mosquitto`

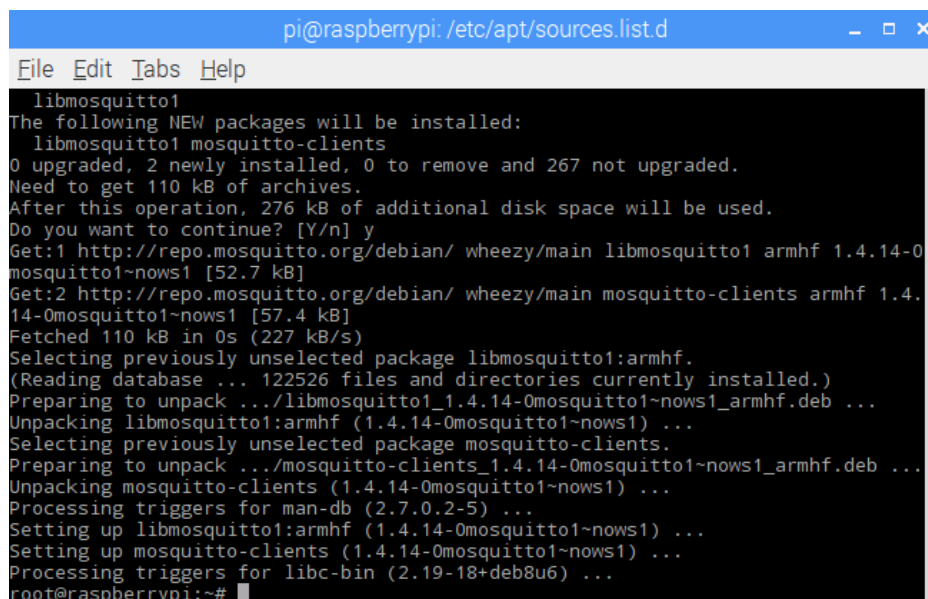


```
pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
Reading package lists... Done
root@raspberrypi:~# apt-get install mosquitto
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  apparmor
The following NEW packages will be installed:
  mosquitto
0 upgraded, 1 newly installed, 0 to remove and 267 not upgraded.
Need to get 139 kB of archives.
After this operation, 339 kB of additional disk space will be used.
Get:1 http://repo.mosquitto.org/debian/ wheezy/main mosquitto armhf 1.4.14-0mosquitto1-nows1 [139 kB]
Fetched 139 kB in 0s (299 kB/s)
Selecting previously unselected package mosquitto.
(Reading database ... 122495 files and directories currently installed.)
Preparing to unpack ../mosquitto_1.4.14-0mosquitto1-nows1_armhf.deb ...
Unpacking mosquitto (1.4.14-0mosquitto1-nows1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Processing triggers for systemd (215-17+deb8u5) ...
Setting up mosquitto (1.4.14-0mosquitto1-nows1) ...
Processing triggers for systemd (215-17+deb8u5) ...
root@raspberrypi:~#
```

Figura 28 *instalar mosquito MQTT*

Fuente: *Elaboración propia*

Para que después se pueda instalar el mosquito cliente con el comando: `apt-get install mosquitto-clients`



```
pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
libmosquitto1
The following NEW packages will be installed:
  libmosquitto1 mosquitto-clients
0 upgraded, 2 newly installed, 0 to remove and 267 not upgraded.
Need to get 110 kB of archives.
After this operation, 276 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://repo.mosquitto.org/debian/ wheezy/main libmosquitto1 armhf 1.4.14-0mosquitto1-nows1 [52.7 kB]
Get:2 http://repo.mosquitto.org/debian/ wheezy/main mosquitto-clients armhf 1.4.14-0mosquitto1-nows1 [57.4 kB]
Fetched 110 kB in 0s (227 kB/s)
Selecting previously unselected package libmosquitto1:armhf.
(Reading database ... 122526 files and directories currently installed.)
Preparing to unpack ../libmosquitto1_1.4.14-0mosquitto1-nows1_armhf.deb ...
Unpacking libmosquitto1:armhf (1.4.14-0mosquitto1-nows1) ...
Selecting previously unselected package mosquitto-clients.
Preparing to unpack ../mosquitto-clients_1.4.14-0mosquitto1-nows1_armhf.deb ...
Unpacking mosquitto-clients (1.4.14-0mosquitto1-nows1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libmosquitto1:armhf (1.4.14-0mosquitto1-nows1) ...
Setting up mosquitto-clients (1.4.14-0mosquitto1-nows1) ...
Processing triggers for libc-bin (2.19-18+deb8u6) ...
root@raspberrypi:~#
```

Figura 29 *instalación de mosquito client*

Fuente: Elaboración propia

Siendo importante indicar que previamente se debe instalar el IDE de Arduino e importar las librerías del microcontrolador ESP8266 y del sensor de pulsos, tal como se aprecia en la siguiente figura.

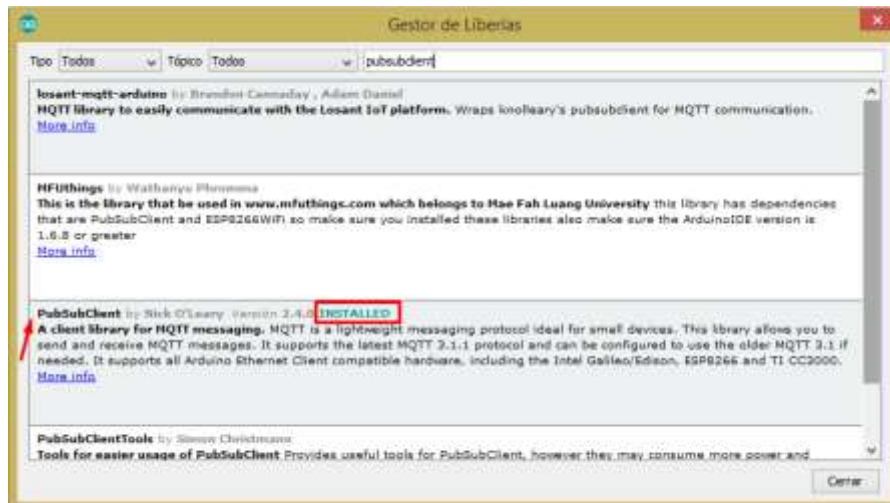


Figura 30 Importar librerías

Fuente: Elaboración propia

Incluyéndose las librerías, se declaran las variables, para luego proceder a programar la conexión, puerto, ip del dispositivo.

```
Archivo Editar Programa Herramientas Ayuda
[Icons]
mqtt_esp8266_edinson_prueba $

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

const char* ssid = "WifiZedin";
const char* password = "zedinson16";
const char* mqtt_server = "192.168.1.178";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;

char msgg='g';
int value = 0;
int a=0;

void setup() {
  pinMode(BUILTIN_LED, OUTPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
```

Figura 31 Programación en Arduino

Fuente: Elaboración propia

```
Archivo Editar Programa Herramientas Ayuda
mqt_esp8266_edinson_prueba5

void setup_wifi() {

  delay(10);

  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);
  IPAddress ip(192,168,1,191);
  IPAddress gateway(192,168,1,1);
  IPAddress subnet(255,255,255,0);
  WiFi.config(ip, gateway, subnet);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Figura 32 Configuración de red

Fuente: Elaboración propia

Finalmente, se carga el proyecto al microcontrolador, para luego verificar que se ha subido correctamente.

```
Archivo Editar Programa Herramientas Ayuda
mqt_esp8266_edinson_prueba Arduino 1.6.10

WiFi.begin(ssid, password);
IPAddress ip(192,168,1,191);
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);
WiFi.config(ip, gateway, subnet);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.println("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

// Subscribe on the MQTT server
//
```

Figura 33 Subir configuración

Fuente: Elaboración propia

Instalación de librería californium.

```
20 import Classes.PeticionConfiguration;
21 import com.google.gson.Gson;
22 import java.net.InetAddress;
23 import java.net.InetSocketAddress;
24 import java.net.SocketException;
25 import java.net.SocketException;
26 import java.util.List;
27
28 import org.eclipse.californium.core.CoapResource;
29 import org.eclipse.californium.core.CoapServer;
30 import org.eclipse.californium.core.coap.OptionSet;
31 import org.eclipse.californium.core.network.CoapEndpoint;
32 import org.eclipse.californium.core.network.EndpointManager;
33 import org.eclipse.californium.core.network.config.NetworkConfig;
34 import org.eclipse.californium.core.server.resources.CoapExchange;
35
36
37 public class HelloWorldServer extends CoapServer {
38
39     private static final int COAP_PORT = NetworkConfig.getStandard().getInt(NetworkConfig.Keys.COAP_PORT);
40
41     /*
42     * Application entry point.
43     */
44     public static void main(String[] args) {
45
46         try {
47
48             // create server
49             HelloWorldServer server = new HelloWorldServer();
50
51         }
52     }
53 }
```

Figura 34 Importar librerías.

Fuente: Elaboración propia

Adicionalmente se requiere de un complemento llamado copper que es compatible con la versión portable de Mozilla v.55

```
45     try {
46
47         // create server
48         HelloWorldServer server = new HelloWorldServer();
49         // add endpoints on all IP addresses
50         server.addEndpoints();
51         server.start();
52     }
53 } catch (SocketException e) {
54     System.err.println("Failed to initialize server: " + e.getMessage());
55 }
56
57
58 /**
59 * Add individual endpoints listening on default CoAP port on all IPv4 addresses of all network interfaces.
60 */
61 private void addEndpoints() {
62     for (InetAddress addr : EndpointManager.getEndpointManager().getNetworkInterfaces()) {
63         // only binds to IPv4 addresses and localhost
64         if (addr instanceof Inet4Address && !addr.isLoopbackAddress()) {
65             InetSocketAddress bindToAddress = new InetSocketAddress(addr, COAP_PORT);
66             addEndpoint(new CoapEndpoint(bindToAddress));
67         }
68     }
69 }
```

Figura 35 Programación de servidor COAP

Fuente: Elaboración propia

Se verifica que el servidor se haya iniciado e ingresamos desde el navegador (Mozilla Firefox v 55) mediante: `coap://localhost:5683/senialsensor`.

Como se puede apreciar, el servidor se ha iniciado correctamente.

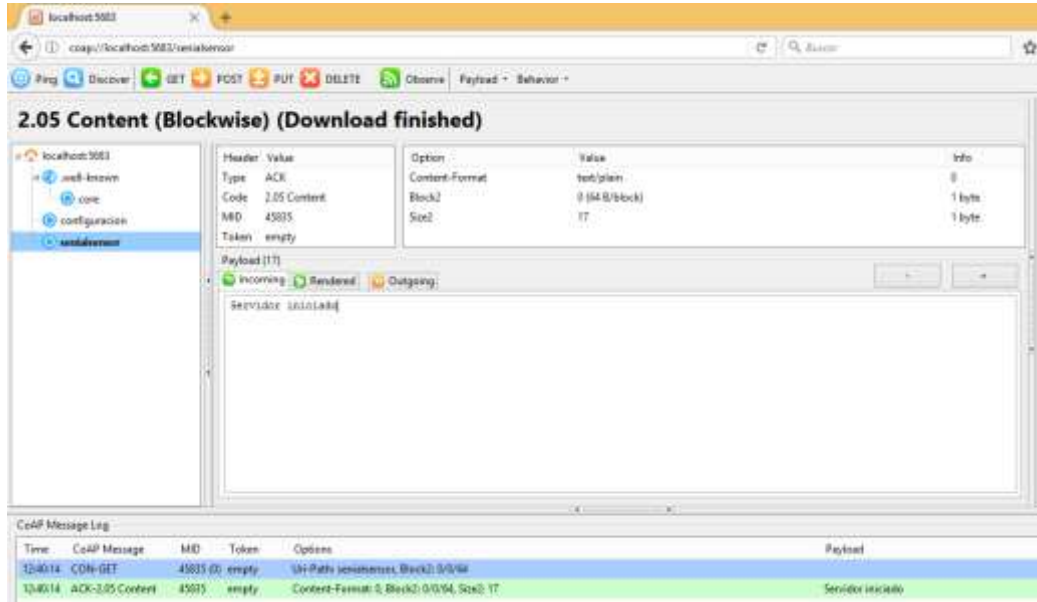


Figura 36 Verificar servidor COAP

Fuente: *Elaboración propia*

En seguida se programa Arduino, utilizando su IDE para que lea la señal del sensor de pulsos cardiacos, además de configurar el microcontrolador ESP8266 y el microcontrolador arduino para finalmente realizar las conexiones físicas correspondientes.

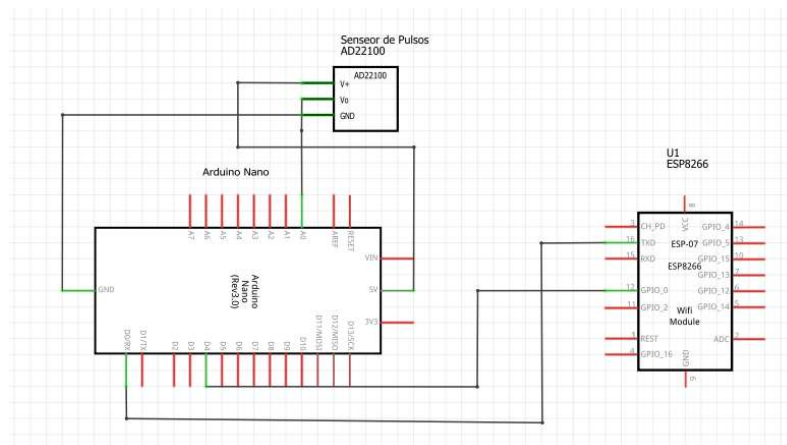


Figura 37 bosquejo de conexión de los componentes para implementar COAP

Fuente: *Elaboración propia*

Ahora se muestran los valores obtenidos a y través del sensor, tanto en la aplicación implementada con la librería californium líneas arriba configurada, así como también se muestran los resultados en el monitor serie de Arduino, seleccionando el puerto 115200.

```
Output - Run (HelloWorldServer)
INFORMACIÓN: Starting endpoint at /192.168.56.1:5683
oct 10, 2018 2:41:55 PM org.eclipse.californium.core.network.CoapEndpoint start
INFORMACIÓN: Starting endpoint at /192.168.56.1:5683
30
88
86
95
94
102
94
91
82
83
85
86
88
87
86
94
93
92
```

Figura 38 Valores mostrados por consola

Fuente: *Elaboración propia*

```
COM4
Connecting to WifiZedin
.....
WiFi connected
192.168.1.191
30
88
86
95
94
102
94
91
82
83
85
86
88
87
86
94
93
92
```

Figura 39 Valores mostrados por consola del IDE de Arduino

Fuente: *Elaboración propia.*

Anexo III:

Ficha de observación: Ataque de al protocolo MQTT utilizando Kali Linux

Nº de pruebas	Nº paquetes	Tiempo de respuesta	Afectó = 1 No Afectó = 0
Prueba 01	300	1010	0
Prueba 02	300	1167	0
Prueba 03	320	1125	0
Prueba 04	330	1160	1
Prueba 05	350	1128	0
Prueba 06	360	1175	0
Prueba 07	365	1104	0
Prueba 08	370	1116	1
Prueba 09	375	1110	0
Prueba 10	380	1080	0
Prueba 11	385	1045	0
Prueba 12	390	1050	0
Prueba 13	395	1006	1
Prueba 14	400	970	0
Prueba 15	400	935	0
Prueba 16	410	930	0
Prueba 17	420	896	0
Prueba 18	430	1050	1
Prueba 19	440	1030	0
Prueba 20	300	1176	0
Prueba 21	400	1100	1
Prueba 22	450	915	0
Prueba 23	500	830	0
Prueba 24	520	858	0
Prueba 25	300	980	1
Prueba 26	310	892	0
Prueba 27	350	860	0
Prueba 28	340	867	0

Prueba 29	450	850	0
Prueba 30	300	1030	1
Prueba 31	320	1148	1
Prueba 32	360	836	0
Prueba 33	380	807	0
Prueba 34	320	1096	1
Prueba 35	310	1160	0
Prueba 36	300	1020	0
Prueba 37	340	830	0
Prueba 38	400	1100	0
Prueba 39	380	783	0
Prueba 40	450	870	0
Prueba 41	480	836	0
Prueba 42	420	887	0
Prueba 43	430	883	0
Prueba 44	490	843	0
Prueba 45	300	1020	0
Prueba 46	320	1170	1
Prueba 47	350	820	0
Prueba 48	200	1500	1
Prueba 49	320	1160	0
Prueba 50	460	1120	0
Prueba 51	465	880	0
Prueba 52	500	800	0
Prueba 53	300	1070	0
Prueba 54	420	1130	1
Prueba 55	520	750	0
Prueba 56	460	1100	0
Prueba 57	300	1030	0
Prueba 58	310	870	0
Prueba 59	480	1050	0
Prueba 60	490	1070	0
Prueba 61	420	1110	1

Prueba 62	470	1130	1
Prueba 63	480	1130	0
Prueba 64	490	1120	0
Prueba 65	430	1120	0
Prueba 66	300	1090	0
Prueba 67	390	1100	0
Prueba 68	380	1170	0
Prueba 69	480	1130	0
Prueba 70	480	1090	0
Prueba 71	300	1050	0
Prueba 72	480	1087	0
Prueba 73	465	1130	0
Prueba 74	490	1100	0
Prueba 75	340	1154	0
Prueba 76	500	1020	0
Prueba 77	550	1027	0
Prueba 78	520	970	0
Prueba 79	580	1010	0
Prueba 80	320	1175	0
Prueba 81	600	880	0
Prueba 82	700	664	1
Prueba 83	400	1120	0
Prueba 84	420	1076	0
Prueba 85	720	620	1
Prueba 86	640	877	0
Prueba 87	460	1110	0
Prueba 88	465	1130	0
Prueba 89	340	808	0
Prueba 90	300	1190	0
Prueba 91	550	1045	1
Prueba 92	355	815	0
Prueba 93	200	1256	0
Prueba 94	520	970	0

Prueba 95	550	1019	0
Prueba 96	720	640	1
Prueba 97	520	958	0
Prueba 98	340	850	0
Prueba 99	550	1060	0
Prueba 100	355	792	0
Prueba 101	550	1075	0
Prueba 102	460	1140	0
Prueba 103	490	1105	0
Prueba 104	460	1103	0
Prueba 105	550	965	0
Prueba 106	480	1023	0
Prueba 107	720	700	1
Prueba 108	340	840	0
Prueba 109	700	700	1
Prueba 110	600	883	1
Prueba 111	520	950	1
Prueba 112	460	1075	0
Prueba 113	340	840	0
Prueba 114	550	1050	1
Prueba 115	550	1050	1
Prueba 116	355	900	0
Prueba 117	600	760	1
Prueba 118	340	830	0
Prueba 119	550	1050	1
Prueba 120	520	908	1
Prueba 121	550	1060	1
Prueba 122	480	1102	1
Prueba 123	340	882	0
Prueba 124	720	670	1
Prueba 125	490	1050	1
Prueba 126	720	708	1
Prueba 127	300	1210	0

Prueba 128	700	710	1
Prueba 129	600	890	1
Prueba 130	550	1050	0
Prueba 131	340	864	0
Prueba 132	600	750	0
Prueba 133	520	1082	0
Prueba 134	600	876	0
Prueba 135	600	850	1
Prueba 136	720	714	1
Prueba 137	200	1290	0
Prueba 138	340	838	0
Prueba 139	700	784	1
Prueba 140	520	1070	1
Prueba 141	550	905	0
Prueba 142	300	780	0
Prueba 143	340	830	0
Prueba 144	720	630	1
Prueba 145	600	780	0
Prueba 146	600	760	0
Prueba 147	300	916	0
Prueba 148	550	910	0
Prueba 149	310	850	0
Prueba 150	600	736	1
Prueba 151	300	825	0
Prueba 152	550	890	0
Prueba 153	720	680	1
Prueba 154	700	665	0
Prueba 155	200	1270	0
Prueba 156	600	980	1
Prueba 157	550	880	1
Prueba 158	340	550	0
Prueba 159	720	615	1
Prueba 160	520	914	1

Promedio	451.75	966.50	45
----------	--------	--------	----

Figura 41: *Ataque de DOS y captura de latencia.*

Fuente: Elaboración propia

Anexo II

Ficha de observación: Ataque de al protocolo COAP utilizando Kali Linux

N° de pruebas	N° paquetes	Tiempo de respuesta	Afectó = 1 Afectó = 0	No
Prueba 01	300	505	0	
Prueba 02	300	590	0	
Prueba 03	320	570	0	
Prueba 04	330	580	0	
Prueba 05	350	590	0	
Prueba 06	360	590	0	
Prueba 07	365	570	0	
Prueba 08	370	550	0	
Prueba 09	375	550	0	
Prueba 10	380	540	0	
Prueba 11	385	520	0	
Prueba 12	390	520	0	
Prueba 13	395	500	0	
Prueba 14	400	480	0	
Prueba 15	400	470	0	
Prueba 16	410	470	0	
Prueba 17	420	450	0	
Prueba 18	430	530	0	
Prueba 19	440	530	0	
Prueba 20	300	600	0	
Prueba 21	400	560	0	
Prueba 22	450	470	0	
Prueba 23	500	420	0	
Prueba 24	520	440	0	
Prueba 25	300	520	0	

Prueba 26	310	430	0
Prueba 27	350	420	0
Prueba 28	340	430	0
Prueba 29	450	440	0
Prueba 30	300	530	0
Prueba 31	320	580	0
Prueba 32	360	420	0
Prueba 33	380	410	0
Prueba 34	320	560	0
Prueba 35	310	600	0
Prueba 36	300	520	0
Prueba 37	340	420	0
Prueba 38	400	560	0
Prueba 39	380	400	0
Prueba 40	450	440	0
Prueba 41	480	430	0
Prueba 42	420	450	0
Prueba 43	430	440	0
Prueba 44	490	420	0
Prueba 45	300	520	0
Prueba 46	320	600	0
Prueba 47	350	420	0
Prueba 48	200	750	0
Prueba 49	320	590	0
Prueba 50	460	570	0
Prueba 51	465	450	0
Prueba 52	500	410	0
Prueba 53	300	530	0
Prueba 54	420	580	0
Prueba 55	520	400	0
Prueba 56	460	560	0
Prueba 57	300	520	0
Prueba 58	310	430	0

Prueba 59	480	530	0
Prueba 60	490	540	0
Prueba 61	420	550	0
Prueba 62	470	560	0
Prueba 63	480	570	0
Prueba 64	490	570	0
Prueba 65	430	580	0
Prueba 66	300	530	0
Prueba 67	390	600	0
Prueba 68	380	600	0
Prueba 69	480	570	0
Prueba 70	480	570	0
Prueba 71	300	530	0
Prueba 72	480	560	0
Prueba 73	465	570	0
Prueba 74	490	570	0
Prueba 75	340	580	0
Prueba 76	500	520	0
Prueba 77	550	500	0
Prueba 78	520	490	0
Prueba 79	580	470	0
Prueba 80	320	640	0
Prueba 81	600	450	0
Prueba 82	700	350	1
Prueba 83	400	570	0
Prueba 84	420	550	0
Prueba 85	720	320	1
Prueba 86	640	440	0
Prueba 87	460	570	0
Prueba 88	465	570	0
Prueba 89	340	420	0
Prueba 90	300	600	0
Prueba 91	550	530	1

Prueba 92	355	410	0
Prueba 93	200	640	0
Prueba 94	520	500	0
Prueba 95	550	520	0
Prueba 96	720	320	1
Prueba 97	520	490	0
Prueba 98	340	420	0
Prueba 99	550	540	0
Prueba 100	355	410	0
Prueba 101	550	540	0
Prueba 102	460	570	0
Prueba 103	490	560	0
Prueba 104	460	550	0
Prueba 105	550	490	0
Prueba 106	480	520	0
Prueba 107	720	350	1
Prueba 108	340	420	0
Prueba 109	700	350	1
Prueba 110	600	450	1
Prueba 111	520	490	1
Prueba 112	460	550	0
Prueba 113	340	420	0
Prueba 114	550	530	1
Prueba 115	550	530	1
Prueba 116	355	410	0
Prueba 117	600	390	1
Prueba 118	340	420	0
Prueba 119	550	530	1
Prueba 120	520	470	1
Prueba 121	550	530	1
Prueba 122	480	560	1
Prueba 123	340	430	0
Prueba 124	720	340	1

Prueba 125	490	540	1
Prueba 126	720	350	1
Prueba 127	300	610	0
Prueba 128	700	360	1
Prueba 129	600	450	1
Prueba 130	550	530	0
Prueba 131	340	430	0
Prueba 132	600	390	0
Prueba 133	520	550	0
Prueba 134	600	450	0
Prueba 135	600	430	1
Prueba 136	720	350	1
Prueba 137	200	650	0
Prueba 138	340	420	0
Prueba 139	700	360	1
Prueba 140	520	540	1
Prueba 141	550	460	0
Prueba 142	300	400	0
Prueba 143	340	420	0
Prueba 144	720	350	1
Prueba 145	600	400	0
Prueba 146	600	390	0
Prueba 147	300	450	0
Prueba 148	550	460	0
Prueba 149	310	420	0
Prueba 150	600	390	1
Prueba 151	300	420	0
Prueba 152	550	450	0
Prueba 153	720	350	1
Prueba 154	700	340	0
Prueba 155	200	650	0
Prueba 156	600	390	1
Prueba 157	550	450	1

Prueba 158	340	280	0
Prueba 159	720	320	1
Prueba 160	520	460	1
Promedio	451.75	488.91	31

Ejecutar captura

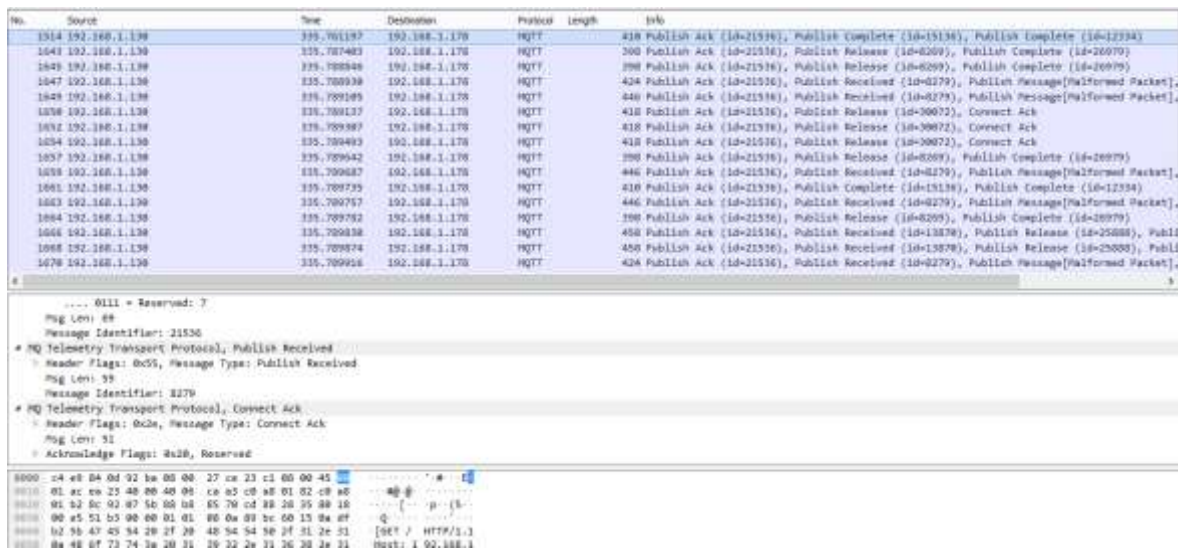


Figura 42: captura de paquetes utilizando Wireshark

Fuente: Elaboración propia ura de paquetes utilizando la herramienta wireshark

