



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Estudios de Postgrado

Maestría en Artes en Tecnologías de la Información y la
Comunicación

**ESTUDIO DE EFICIENCIA DE *MQTT* (ISO/IEC 20922:2016) Y *AMQP* (ISO/IEC 19464:2014)
EN EL MANEJO DE MENSAJES DE UN SISTEMA DE SENSORES PARA MONITOREO DE
CONDICIONES PRECURSORAS DE ESCARAS**

Ing. Roberto García López

Asesorado por el Ing. Edwin Estuardo Zapeta Gómez, MsC

Guatemala, octubre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ESTUDIO DE EFICIENCIA DE *MQTT* (ISO/IEC 20922:2016) Y *AMQP* (ISO/IEC 19464:2014)
EN EL MANEJO DE MENSAJES DE UN SISTEMA DE SENSORES PARA MONITOREO DE
CONDICIONES PRECURSORAS DE ESCARAS**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

ROBERTO GARCÍA LÓPEZ

ASESORADO POR EL ING. EDWIN ESTUARDO ZAPETA GÓMEZ, MSC

AL CONFERÍRSELE EL TÍTULO DE

**MAESTRO EN TECNOLOGÍAS DE LA INFORMACIÓN
Y LA COMUNICACIÓN**

GUATEMALA, OCTUBRE DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton De León Bran
VOCAL IV	Br. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

JURADO EVALUADOR QUE PRACTICÓ EL EXAMEN DE DEFENSA

DECANA	Inga. Aurelia Anabela Cordova Estrada
DIRECTOR	Ing. Edgar Darío Álvarez Cotí
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADOR	Ing. Edwin Estuardo Zapeta Gómez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**ESTUDIO DE EFICIENCIA DE *MQTT* (ISO/IEC 20922:2016) Y *AMQP* (ISO/IEC 19464:2014)
EN EL MANEJO DE MENSAJES DE UN SISTEMA DE SENSORES PARA MONITOREO DE
CONDICIONES PRECURSORAS DE ESCARAS**

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Postgrado, con fecha 18 de marzo de 2020.

Ing. Roberto García López



DTG. 505.2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Estudios de Postgrado, al Trabajo de Graduación titulado: **ESTUDIO DE EFICIENCIA DE MQTT (ISO/IEC 20922:2016) Y AMQP (ISO/IEC 19464:2014) EN EL MANEJO DE MENSAJES DE UN SISTEMA DE SENSORES PARA MONITOREO DE CONDICIONES PRECURSORAS DE ESCARAS**, presentado por el **Ingeniero Roberto García López**, estudiante del programa de **Maestría en Artes en Tecnologías de la Información y la Comunicación**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:


Inga. Anabela Cordova Estrada
Decana



Guatemala, octubre de 2021

AACE/cc



Guatemala, octubre de 2021

LNG.EEP.OI.024.2021

En mi calidad de Director de la Escuela de Estudios de Postgrado de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor, verificar la aprobación del Coordinador de Maestría y la aprobación del Área de Lingüística al trabajo de graduación titulado:

“ESTUDIO DE EFICIENCIA DE MQTT (ISO/IEC 20922:2016) Y AMQP (ISO/IEC 19464:2014) EN EL MANEJO DE MENSAJES DE UN SISTEMA DE SENSORES PARA MONITOREO DE CONDICIONES PRECURSORAS DE ESCARAS”

presentado por **Roberto García López** quien se identifica con carné **8512197** correspondiente al programa de **Maestría en artes en Tecnologías de la información y la comunicación** ; apruebo y autorizo el mismo.

Atentamente,

“Id y Enseñad a Todos”


Mtro. Ing. Edgar Darío Álvarez Cotí
Director



**Escuela de Estudios de Postgrado
Facultad de Ingeniería**



Guatemala, 01 de agosto 2020.

M.A. Edgar Darío Álvarez Cotí
Director
Escuela de Estudios de Postgrado
Presente

M.A. Ingeniero Álvarez Cotí:

Por este medio informo que he revisado y aprobado el **TRABAJO DE GRADUACIÓN** titulado: "ESTUDIO DE EFICIENCIA DE MQTT (ISO/IEC 20922:2016) Y AMQP (ISO/IEC19464:2014) EN EL MANEJO DE MENSAJES DE UN SISTEMA DE SENSORES PARA MONITOREO DE CONDICIONES PRECURSORAS DE ESCARAS" del estudiante Roberto García López quien se identifica con número de carné 008512197 del programa de **Maestría en Tecnologías de la Información y la Comunicación**.

Con base en la evaluación realizada hago constar que he evaluado la calidad, validez, pertinencia y coherencia de los resultados obtenidos en el trabajo presentado y según lo establecido en el *Normativo de Tesis y Trabajos de Graduación aprobado por Junta Directiva de la Facultad de Ingeniería Punto Sexto inciso 6.10 del Acta 04-2014 de sesión celebrada el 04 de febrero de 2014*. Por lo cual el trabajo evaluado cuenta con mi aprobación.

Agradeciendo su atención y deseándole éxitos en sus actividades profesionales me suscribo.

Atentamente,

MARLON ANTONIO PEREZ TURK
INGENIERO EN CIENCIAS Y SISTEMAS
COLEGIADO No. 4492

MSc. Ing. Marlon Antonio Pérez Türk
Coordinador

Maestría en Tecnologías de la Información y la Comunicación
Escuela de Estudios de Postgrado



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Postgrado
Coordinador de Área

Guatemala, 04 de junio de 2020.

M.A. Ing. Edgar Dario Álvarez Cotí
Director
Escuela de Estudios de Postgrados
Presente

Estimado M.A. Ing. Álvarez Cotí:

Reciba un cordial y atento saludo, a la vez aprovecho la oportunidad para hacerle de su conocimiento que he revisado y aprobado el trabajo especial de graduación titulado: “ESTUDIO DE EFICIENCIA DE MQTT (ISO/IEC 20922:2016) Y AMQP (ISO/IEC 19464:2014) EN EL MANEJO DE MENSAJES DE UN SISTEMA DE SENSORES PARA MONITOREO DE CONDICIONES PRECURSORAS DE ESCARAS” del estudiante Roberto García López del Programa de Maestría en Tecnología de la Información y Comunicación, identificado con número de carné: 008512197.

Agradeciendo su atención y deseándole éxitos en sus actividades profesionales me suscribo.

“Id y enseñad a todos”

MSc. Ing. Edwin Estuardo Zapeta Gómez
Colegiado No. 12767
Asesor

Cc: Archivo/LA

Doctorado: Sostenibilidad y Cambio Climático. **Programas de Maestrías:** Ingeniería Vial, Gestión Industrial, Estructuras, Energía y Ambiente Ingeniería Geotécnica, Ingeniería para el Desarrollo Municipal, Tecnologías de la Información y la Comunicación, Ingeniería de Mantenimiento. **Especializaciones:** Gestión del Talento Humano, Mercados Eléctricos, Investigación Científica, Educación virtual para el nivel superior, Administración y Mantenimiento Hospitalario, Neuropsicología y Neurociencia aplicada a la Industria, Enseñanza de la Matemática en el nivel superior, Estadística, Seguros y ciencias actuariales, Sistemas de información Geográfica, Sistemas de gestión de calidad, Explotación Minera, Catastro.

ACTO QUE DEDICO A:

Yeshúa (Jesucristo)

Porque con tu sacrificio nos diste todas las cosas.

Mi familia

Han estado allí siempre en las buenas y en las malas, sin duda alguna Yehová me rodeo de ustedes para mostrarme Su eterno amor. Gracias Eleonora Escribá Mazariegos de García, Sara Marianne García Escribá y Ana Lucía García Escribá.

Mi madre

Espero que estos logros muestren que hiciste un buen trabajo conmigo, que llenen tu corazón de alegría y que tanto esfuerzo a lo largo de los años valió la pena.

Mis amigos

A los que están presentes y a los que han partido, gracias por la amistad y el cariño de tantos años, agradezco su ayuda, su tiempo, sus palabras de motivación, las largas caminatas, la diversión y los enojos. Espero haber sido un buen amigo para ustedes también.

AGRADECIMIENTOS A:

Mis catedráticos

Everest Medinilla, Héctor Mendía, Pablo Rosales, Estuardo Zapeta, Mildred Caballeros, Leandro Yax, Estuardo Echeverría, Luis Espino, Carlos Pérez, Lenin Rodríguez, Sasha Palencia y Marlon Pérez Türk. Gracias por compartir con nosotros conocimientos académicos y vivencias profesionales, fueron de gran inspiración.

Compañeros

Compartir con ustedes ha sido un refresco para mi vida profesional y personal. Aprendí mucho de cada uno. Gracias por los recuerdos que quedarán.

Revisores y correctores de estilo

Maestro Jorge Gerardo García González, Ingeniero Gerson Absalon Ortiz Corona y Licenciada Jéssica Melgarejo: gracias por haber realizado la revisión del formato, redacción y ortografía del contenido del trabajo y por sus acertados aportes.

Universidad de San Carlos de Guatemala

Gracias querida universidad por todos los años de estudio y formación académica. Que tus egresados te sigamos haciendo *inter caeteras orbis conspicua*.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
RESUMEN.....	VII
PLANTEAMIENTO DEL PROBLEMA Y FORMULACIÓN DE PREGUNTAS ORIENTADORAS	IX
OBJETIVOS.....	XI
MARCO METODOLÓGICO	XIII
INTRODUCCIÓN	XIX
1. ANTECEDENTES	1
2. JUSTIFICACIÓN	9
3. ALCANCES.....	11
4. MARCO TEÓRICO.....	15
4.1. Internet del todo (<i>Internet of Everything – IoE</i>).....	15
4.2. El modelo <i>TCP/IP</i>	16
4.2.1. Descripción de las capas del modelo <i>TCP/IP</i>	17
4.2.2. Modelo <i>TCP/IP</i> aplicado a <i>IoT</i>	18
4.3. Tipos de dispositivos en ecosistemas <i>IoT</i>	19
4.3.1. Sensores y actuadores	20
4.3.2. <i>Gateways</i> y controladores	20
4.3.3. Almacenamiento	21
4.3.4. Manejadores	21

4.3.5.	<i>Message-brokers</i> empleados en <i>IoT</i>	22
4.3.5.1.	<i>AMQP</i>	22
4.3.5.2.	<i>MQTT</i>	23
4.3.5.3.	<i>CoAP</i>	24
4.4.	Factores que afectan las comunicaciones	24
4.4.1.	Latencia	25
4.4.2.	<i>Lags</i>	25
4.4.3.	<i>Jitters</i>	26
4.4.4.	Atenuación	26
4.4.5.	Calidad de servicio	27
4.4.6.	Tipo de servicio	27
4.4.7.	Servicios diferenciados - <i>differentiated services</i> (<i>DS - diffserv</i>)	28
4.5.	Escaras	29
4.5.1.	Etiopatogénesis de las escaras	29
4.5.2.	Escalas de medición de riesgo	31
5.	PRESENTACIÓN DE RESULTADOS	35
5.1.	Diseño del prototipo	35
5.1.1.	Condiciones de frontera	35
5.1.2.	Componentes del prototipo	38
5.1.3.	Modelo de control	40
5.1.4.	Parametrización de pruebas	42
5.2.	Realización de pruebas	44
5.3.	Módulo de captura de datos	45
5.4.	Tabulación y análisis de información	48
5.5.	Evaluación estadística de resultados	50
5.6.	Comparación de desempeño/eficiencia de <i>MQTT</i> y <i>AMQP</i>	55
5.6.1.	Tiempos de respuesta (t)	55

5.6.2.	Integridad, manejo de <i>payload</i> y contenido (I)	56
5.6.3.	Factores de aseguramiento de entrega de mensajes o calidad en el servicio (Q)	60
5.6.4.	Manejo de mecanismos de seguridad del protocolo (S)	63
5.6.5.	Facilidad de implementación (F).....	65
5.6.6.	Complejidad de uso (C)	65
5.6.7.	Resumen de la comparación	66
6.	DISCUSIÓN DE RESULTADOS	69
6.1.	Impacto y evaluación del entorno del prototipo en los resultados	69
6.2.	Evaluación del <i>hardware</i>	70
6.3.	Evaluación del <i>software</i>	72
6.4.	Evaluación de la propuesta de estudio y el método	73
6.5.	Aplicabilidad al problema de escaras	74
	CONCLUSIONES	77
	RECOMENDACIONES	81
	REFERENCIAS	83

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Factores comunes de incidencia de escaras en pacientes hospitalizados	2
2.	Tiempos de hospitalización extra debido a las escaras	3
3.	Ejemplos de protocolos y tecnologías <i>IoT</i> utilizados por capa en el modelo <i>TCP/IP</i>	19
4.	Arquitectura propuesta para el prototipo, patrón <i>publish/subscribe</i> uno-a-uno.....	36
5.	Pre-muestreo para la obtención de la cantidad óptima de lecturas.....	41
6.	Ejemplo de sincronización <i>NTP</i> /micro-controlador previo a las ruebas	44
7.	<i>Time-plot</i> de la serie 09 de lecturas de humedad utilizando <i>AMQP</i>	50
8.	Distribución de tiempos de respuesta de la serie 09 de lecturas de humedad relativa utilizando <i>AMQP</i>	51
9.	Distribución de tiempos de respuesta de <i>MQTT</i>	54
10.	Distribución de tiempos de respuesta de <i>AMQP</i>	54
11.	Diagrama de contraste de la dispersión de las pruebas <i>MQTT</i> / <i>AMQP</i> (en segundos).....	56
12.	Análisis de <i>payload/overhead</i> <i>Mosquitto-MQTT</i>	57
13.	Análisis de <i>payload/overhead</i> de <i>RabbitMQ-AMQP</i>	58

TABLAS

I.	Normativa UNE 100713-2005 para construcciones hospitalarias en España	4
II.	Escala de Braden-Q para medición de riesgos de úlceras de piel.....	32
III.	Escala Emina para medición de riesgos de úlceras de piel	33
IV.	Descripción de variables utilizadas en el experimento.....	43
V.	Procesamiento de datos del módulo de suscripción a <i>MQTT</i> – publicaciones sensor cafet cada 15 segs	46
VI.	Resultados <i>AMQP</i> , prueba 09 – Sensor #2 (Azul) / Temperatura	49
VII.	Resultados <i>MQTT</i> , prueba 09 – Sensor #1 (Café) / Humedad	49
VIII.	Resultados de las pruebas con <i>MQTT</i> , en sensores AZUL y CAFÉ	52
IX.	Resultados de las pruebas con <i>AMQP</i> , en sensores AZUL y CAFÉ	53
X.	Relación entre el tamaño del mensaje y sus correspondientes <i>frames</i>	59
XI.	Comparación entre <i>MQTT</i> y <i>AMQP</i> , ponderación sobre 5 puntos por ítem.....	67

RESUMEN

El aparecimiento de *IoT* expuso los altos requerimientos de su instauración debido a la ubicuidad y cantidad exorbitante de sus dispositivos. Linthicum (2017) refiere que las exigencias van desde la implementación de infraestructuras más apropiadas (e.g. *fog*, *edge*, *cloudlets* u otros orientados a llevar el poder computacional más cerca de los sensores), y otros aspectos importantes como seguridad, consumo eficiente de energía, manejo óptimo de latencia, filtrado y *data-wrangling*, entre otros.

A pesar de los inconvenientes mencionados, el Internet de las Cosas ha ganado paulatinamente terreno, y es que estos sistemas son conexiones inteligentes de máquinas, personas, procesos, datos y en general de cosas. Lo que convierte información en acciones que a su vez crean nuevas capacidades, experiencias más ricas y oportunidades económicas sin precedentes para países, empresas y personas individuales (Evans, 2012).

Esto brinda grandes oportunidades y hacen que *IoT* tenga un nicho de aplicación en prácticamente todas las esferas humanas: desde agricultura hasta aplicaciones médicas. Y es, en este último donde se hace imprescindible contar con sistemas, sensores, controladores, *gateways*, actuadores, entre otros, eficientes, seguros y rápidos, debido a que se desea proteger la vida o la integridad de las personas.

Considerando lo anterior, el presente trabajo propone una sistematización a través del monitoreo de los factores de temperatura y humedad relativa para la prevención del aparecimiento de escaras en pacientes en estado de

inmovilidad, afecciones que de acuerdo a la *National Center for Biotechnology Information* (2015) son padecimientos recurrentes que causan una gran erogación de dinero al sistema de salud debido a la inversión extra en medicinas, terapias y estancia adicional en hospitales.

Por otra parte, desde el punto de vista de eficiencia tecnológica, se propone la utilización de *message-brokers* como elementos intermediarios entre los componentes de la arquitectura de la solución, debido a que estos incorporan grandes bondades a los sistemas *IoT*, particularmente por el estupendo procesamiento de *streaming* de las comunicaciones asíncronas, buen manejo de latencia y un alto grado de desacoplamiento.

En este sentido, el trabajo plantea de forma explícita la adopción del uso de *message-brokers* como componentes *de facto* en los diseños de sistemas *IoT*. Sin embargo, al año 2020 existen muchos de ellos disponibles en el mercado con sus aplicaciones o usos particulares, por lo que es necesario contar con métodos o procedimientos para realizar un análisis de contraste que permita determinar cuál o cuáles son los más adecuados para una situación determinada.

El documento ofrece a manera de *exempli gratia*, un estudio comparativo de dos *message-brokers* a través de la implementación de un prototipo para efectuar las pruebas de eficiencia. Concretamente, el autor seleccionó las especificaciones de *MQTT* y *AMQP*.

Se espera que el presente estudio ofrezca una panorámica del uso y la forma de evaluar estos mecanismos como parte de la solución de un problema particular o brinde una guía para la selección apropiada de estos, especialmente en entornos *IoT*.

PLANTEAMIENTO DEL PROBLEMA Y FORMULACIÓN DE PREGUNTAS ORIENTADORAS

A pesar de la existencia de programas de profilaxis y prevención de úlceras por presión o escaras, este tipo de lesiones aún posee una alta frecuencia. Lamentablemente, Guatemala no posee registros estadísticos de estas: se buscó información al respecto en el Instituto Nacional de Estadística, el Instituto Guatemalteco de Seguridad Social y el hospital Roosevelt, con resultados negativos.

Se debe indicar que existen elementos causales internos y externos, sin embargo, para delimitar el ámbito del trabajo, se considerará solamente el monitoreo de los factores físicos exógenos de temperatura y humedad relativa, que bajo la conjetura *a priori*, se estima que poseen una alta incidencia en el padecimiento descrito.

Derivado de lo anterior, deben encontrarse mecanismos que permitan monitorear dichos factores con la finalidad de anticiparse a sus efectos. Sistemas que por lo *ad hoc* pueden ser extraídos de tecnologías *IoT*; es decir, utilizando sensores, *gateways*, protocolos manejadores de mensajes, entre otros, debido a que el sistema de vigilancia requiere de una alta eficiencia, especialmente en el manejo tráfico y latencia, tópicos donde los *message-brokers*, constituyen una piedra angular.

De lo presentado previamente, surge la pregunta central:

- ¿Son *MQTT* y *AMQP* protocolos eficientes para el manejo de mensajes en un sistema de sensores de temperatura y humedad para aplicaciones médicas como el monitoreo de las condiciones precursoras de escaras?

Y se establecen en forma complementaria, las preguntas auxiliares siguientes:

- De acuerdo con las investigaciones y escalas de medición de riesgos, ¿son la temperatura y humedad factores precursoras de úlceras o escaras?
- ¿Qué descripción y especificaciones técnicas deben cumplir los sensores, controladores y servicios presentados en la propuesta de solución?
- ¿Cuáles son las ventajas del uso de *message-brokers* en esquemas del Internet de las Cosas, en ámbitos como el de monitoreo por sensores de temperatura y humedad para la prevención de escaras?
- ¿Cómo se puede determinar qué protocolo, entre *AMQP* y *MQTT*, es el más eficiente en el manejo de mensajes para el monitoreo por sensores de temperatura y humedad relativa?

OBJETIVOS

Objetivo general

Determinar si *AMQP* (ISO/IEC 19464:2014) y *MQTT* (ISO/IEC 20922:2016) son protocolos eficientes en el manejo de mensajes provenientes de sensores de temperatura y humedad, para establecer su aplicabilidad como solución al monitoreo de las condiciones precursoras de escaras.

Objetivos específicos

1. Valorar el riesgo o influencia de los factores de temperatura y humedad relativa en el padecimiento de escaras o úlceras basados en las investigaciones o escalas.
2. Describir las características, especificaciones y propiedades de los sensores, controladores, *gateways* y servicios que podrían ser utilizados para monitorear las condiciones físicas precursoras del padecimiento.
3. Exponer el funcionamiento de los *message-brokers* en el Internet de las Cosas, especialmente en entornos médicos.
4. Implementar un prototipo que permita realizar un estudio comparativo de eficiencia de algoritmos basados en *AMQP* y *MQTT* en un sistema de monitoreo por sensores de temperatura y humedad.

MARCO METODOLÓGICO

- Tipo de estudio

Para la realización del presente documento fue requerido realizar una investigación empírica-descriptiva respecto a la frecuencia de ocurrencia, estimación de riesgo de las escaras así como de la incidencia de la temperatura y humedad en el padecimiento. Esto se logró a través de un pequeño cuestionario realizado en el Centro Universitario Metropolitano de la Universidad de San Carlos de Guatemala.

Por otra parte, para el diseño del prototipo fue indispensable realizar estudios documentales y técnicos para conocer las interioridades de los componentes así como de los requerimientos y dependencias de configuración e instalación de los mismos.

Lo anterior conllevó la adquisición de conocimientos técnicos particulares al escenario experimental propuesto, especialmente los relacionados a las implementaciones de las especificaciones de *AMQP* y *MQTT*, para conocer su funcionamiento e interacción con los componentes seleccionados para el prototipo tanto de *software* o entorno operativo como de *hardware* (i.e. sensores, controlador, servidor, *gateway*, clientes, entre otros) y su interrelación con el lenguaje de programación seleccionado.

- Diseño de la investigación

Debido a que una de las finalidades del trabajo fue la obtención de datos de comparación entre *MQTT* y *AMQP* y su adecuado desempeño como parte de la solución al monitoreo de la situación de escaras, estos *message-brokers* fueron instalados bajo la condición *ceteris paribus* o *cet. par.*, previo a la realización de las pruebas de rendimiento de estos.

Otras consideraciones previas a la realización del experimento fueron:

- Establecer las condiciones de frontera del prototipo: formas de conexión de los dispositivos, frecuencia de lectura de los sensores, establecimiento de los procedimientos de publicación y suscripción, entre otros.
- Establecer el modelo de control: definir los filtros de calidad de los datos para las consideraciones de válido/descartable, especificar el tamaño de las muestras (*sets* de lecturas), definir el tamaño y estructura de los mensajes *publish/subscribe* - *pub/sub*, establecer las medidas estadísticas (medias y desviaciones estándar particulares) para la realización de las comparaciones.
- Establecer las variables cuantitativas y cualitativas que serían evaluadas: facilidad de implementación, temperatura, humedad relativa, tiempos de respuesta, manejo del *payload/overhead* y seguridad, entre otros.

- Variables

Las variables y sus magnitudes escalares, considerados para el experimento fueron:

- Cuantitativas: temperatura – grados centígrados [10,50], temperatura relativa – porcentaje [15,75], sensibilidad de 8 dígitos significativos de lectura de los sensores [0.0000000, 0.99999999], tiempos de latencia – segundos [0.000001, 10.000000], entre otras.
- Cualitativas: facilidad de instalación e implementación (horas requeridas hasta la puesta en marcha), funcionalidad del producto evaluado (sencillo, complejo, difícil, etc.), seguridad (ninguna, baja, alta, fuerte), entre otras.

- Técnicas de recolección de información

La recolección de información se realizó a través de la observación y almacenamiento de los datos emitidos por los sensores y formateados por el controlador (en su contenido y estructura) y que fueron captados por los clientes o consumidores.

Dicha *data* fue tabulada, almacenada y analizada utilizando un módulo escrito en *Python* 3.7.3, lo que incluyó el proceso de obtención de medias y desviaciones estándar de cada corrida o experimento, las cuales fueron exportadas a *Excel* en archivos *.data* (.csv), para realizar las gráficas de comportamiento.

- Procedimiento metodológico

La construcción del trabajo se basó en tres fases:

- Fase I - Investigación documental

Esta cubrió la realización de la investigación empírica-cualitativa del padecimiento de escaras a través de encuestas y se complementó con la información de algunas fuentes secundarias disponibles.

En relación con la elaboración del prototipo fue requerido indagar acerca de las características de los sensores asequibles y disponibles en Guatemala, qué dispositivo podría utilizarse como microcontrolador y qué requerimientos computacionales eran necesarios para la instalación de los *message-brokers*. Lo anterior se realizó considerando las limitaciones presupuestarias establecidas previamente en el protocolo del trabajo.

Respecto a los *brokers* seleccionados se recurrió a una búsqueda e investigación de las implementaciones más apropiadas o utilizadas en el mercado y que fueran de carácter gratuito, llegándose a la conclusión de examinar a *MQTT* con el uso de *Mosquitto* 1.6.9 y a *AMQP* a través de la evaluación de *RabbitMQ* 3.8.3.

Esta fase se culminó con el contenido de varios cursos de la Maestría en Tecnologías de la Información y la Comunicación e interacciones con algunos profesionales de Informática y electrónica.

- Fase II – Modelado e implementación del prototipo

Teniendo todos los componentes disponibles se realizaron búsquedas en sitios *web* y manuales en línea para averiguar cómo se conectaban los sensores a los pines del *Raspberry-Pi* 3B+ (que fue el controlador seleccionado para el prototipo), cómo se instalaba y ejecutaba *Mosquitto* y *RabbitMQ* en los equipos utilizados y el desarrollo en *Python* de los módulos de publicación y suscripción para cada uno de estos.

Junto con lo anterior se afinaron las condiciones de frontera, modelado y parametrización de las variables y la definición de algunas tolerancias e inclusión de otros *constraints* del modelo.

- Fase III – Pruebas y resultados

Habiendo establecido las variables y funcionalidad del prototipo se procedió a realizar las pruebas para la obtención de resultados. Cabe indicar que el prototipo fue particularmente útil para evaluar el desempeño de los protocolos en sus tiempos de respuesta, integridad del mensaje y manejo del *payload/overhead*.

Respecto a la evaluación cualitativa de facilidad de implementación y características o funcionalidades adicionales de las implementaciones de *MQTT* y *AMQP* se requirió del estudio de los archivos de configuración y los manuales correspondientes.

Los resultados obtenidos fueron valiosos para el planteamiento de las conclusiones de eficiencia/rendimiento de *AMQP* y *MQTT*, así como del establecimiento de las recomendaciones respectivas.

- Técnicas de análisis de información

La obtención de líneas de comportamiento del fenómeno, se logró gracias al uso de herramientas de estadística descriptiva, las cuales fueron suficientes para abordar el estudio comparativo de eficiencia de los protocolos evaluados.

La evaluación se realizó utilizando las medidas de tendencia central y dispersión, es decir, un estudio de medias y desviaciones estándar de cada set de pruebas realizadas con *AMQP* y *MQTT*.

Finalmente, como se ha indicado previamente, se recurrió a la exportación de los datos recolectados en el experimento para utilizarlos en *Microsoft Excel* con el objetivo de procesar los mismos y obtener las medidas estadísticas y la representación gráfica del comportamiento de los escenarios planteados en el prototipo.

INTRODUCCIÓN

El trabajo se encauzó en la determinación de cuál *message-broker* entre *MQTT* y *AMQP*, ofrece la mejor eficiencia o rendimiento en la entrega de mensajes en un ecosistema de sensores para el monitoreo de temperatura y humedad relativa.

Con lo anterior se persiguió una sistematización para la atención médica temprana en pro de la minimización o prevención del aparecimiento de escaras o úlceras en la piel particularmente en pacientes en estado de inmovilidad.

En este contexto, se ha de mencionar que el uso de *brokers* trae grandes ventajas a los sistemas de atención temprana, como el indicado en el párrafo anterior, y es que el empleo de estos mecanismos intermediarios además de mejorar la latencia, permiten comunicaciones asíncronas, brindan un alto grado de desacoplamiento entre los publicadores/suscriptores y remedian los problemas de pérdidas por congestión, entre otros.

Existe una amplia variedad de *brokers* en el mercado, sin embargo, el proyecto se enfocó particularmente en los protocolos indicados, debido a que estos poseen especificaciones avaladas por las normas *ISO/IEC*, lo cual les brinda un alto grado de confiabilidad.

De acuerdo con lo anterior, se comparará la eficiencia o rendimiento en el manejo de mensajes de *MQTT* (*Message Queuing Telemetry Transport*, *ISO/IEC 20922:2016*) y *AMQP* (*Advanced Message Queuing Protocol*, *ISO/IEC 19464:2014*) en la modalidad *publish/subscribe*.

El documento está dividido en seis secciones, el capítulo 1 se enfoca en los antecedentes del padecimiento, específicamente en las causas por las cuales se manifiestan úlceras o escaras en pacientes en estado de inmovilidad, particularmente las exógenas de temperatura y humedad relativa.

Esta sección incluye datos estadísticos obtenidos de fuentes primarias, a través de una investigación cualitativa llevada a cabo en el Centro Universitario Metropolitano, CUM, y de fuentes secundarias recopiladas en la exigua bibliografía encontrada respecto a este tema.

Se muestran algunos datos de los factores que tienen incidencia en el padecimiento y cómo la falta de tratamiento o prevención de este provoca no solamente problemas serios de salud sino también problemas económicos.

Este capítulo subraya la importancia de la adopción de estándares internacionales considerados en otras regiones, como base para la implementación de normas en la construcción de hospitales o clínicas particularmente para mantener las condiciones ambientales de temperatura y humedad.

Por otra parte, esta sección también incluye algunos antecedentes del uso de *message-brokers* en sistemas de mensajería, así como del gran éxito que han tenido en el Internet de las Cosas, particularmente en aquellos entornos donde se utilizan sensores y micro controladores.

El capítulo dos presenta la justificación del porqué de la propuesta de un sistema de monitoreo para la prevención de úlceras o escaras a través del uso de la tecnología, concretamente dentro del ámbito del Internet de las Cosas.

Acá se manifiesta la necesidad de implementación de un sistema monitoreo o vigilancia continua de las causas físicas externas de temperatura y humedad, con la finalidad de ayudar o colaborar con los métodos tradicionales de tratamiento (e.g. terapias de movimiento) o la planificación de cambios de ropa o sábanas, entre otros, tal que minimicen o prevengan la ocurrencia de este tipo de lesiones.

En el capítulo tres se presentan y establecen los alcances del estudio desde las perspectivas investigativa, técnica y de resultados. Estos incluyen la delimitación de las condiciones de frontera del modelo a implementar.

Esta sección muestra las razones de la selección de los dispositivos y protocolos involucrados en el entorno de pruebas, así como las restricciones de estos, especialmente las debidas a factores económicos y de *networking*.

El capítulo cuatro muestra el marco teórico considerado idóneo para la comprensión del contenido del documento, acentuando los temas correspondientes a: escalas de medición de riesgo de úlceras o escaras; las ventajas de la tecnología del Internet de las Cosas, sus componentes – sensores, controladores, actuadores, *gateways*, protocolos de manejo de mensajes, entre otros-; los diferentes modelos de capas empleados en la descripción de las arquitecturas *IoT*; mecanismos de reducción de latencia o priorización de tráfico en entornos de redes; y demás.

El capítulo cinco comprende la presentación de resultados y cómo estos fueron conseguidos. Se muestran los esquemas, el modelado e implementación del prototipo y la forma de realización de las pruebas para la obtención de estadísticas de eficiencia o rendimiento respecto al manejo de mensajes y otros factores relevantes del funcionamiento tanto de *AMQP* como de *MQTT*.

En esta sección se establecen las escalas y los parámetros de medición de la recopilación de datos, y se plantean las restricciones, ventajas y desventajas del prototipo propuesto, así como el margen de tolerancia o error de las mediciones del modelo.

El capítulo presenta la tabulación de los datos recopilados de las pruebas, los gráficos asociados y el análisis derivado del experimento que facultó la obtención de las conclusiones y recomendaciones respectivas.

La discusión de resultados se presenta en el capítulo seis. En este se expone una evaluación de la influencia de los componentes y el diseño del prototipo en los resultados obtenidos. Lo anterior incluye la reflexión de experimentar y realizar otros estudios en otros entornos o plataformas *IoT*, que permitan complementar o comparar con los resultados presentados en el documento, particularmente en aspectos relacionados con otros *message-brokers*. Lo anterior es una derivación natural debido a la plétora de escenarios posibles y la variedad de arquetipos en el Internet de las Cosas.

De lo expuesto anteriormente, este capítulo presenta el impacto del entorno, la calidad y especificaciones del *hardware*, la selección del *software* (e.g. librerías de *Python*), la metodología empleada, particularmente la forma de evaluación de las variables cuantitativas y cualitativas, entre otros, y cómo esto pudo afectar o tener algún impacto en el estudio comparativo de los dos *message-brokers* analizados.

El contenido de este capítulo sirvió como piedra angular para el fundamento y elaboración de las conclusiones y recomendaciones del trabajo, especialmente considerando los resultados obtenidos y las experiencias adquiridas en la realización del experimento, sus aciertos y tropiezos.

1. ANTECEDENTES

Este documento presentará dos antecedentes relacionados con el tema seleccionado: 1) lo concerniente al padecimiento, y 2) lo relativo al uso de *message-brokers*.

- Antecedentes del padecimiento

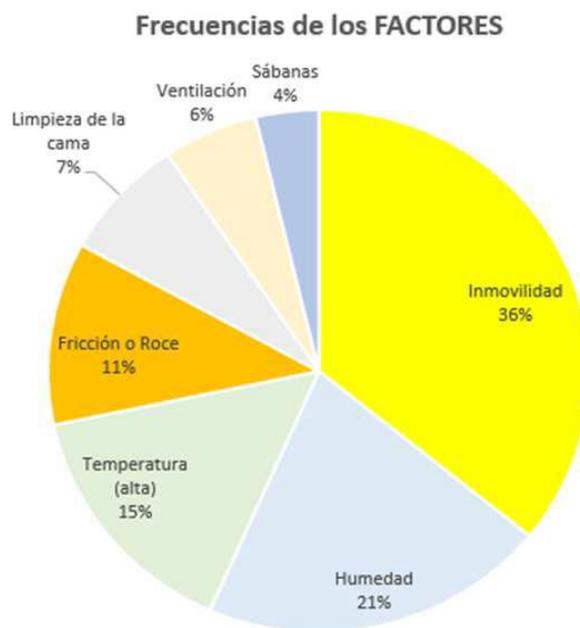
Muchas personas han tenido que recurrir a un centro médico por enfermedades o accidentes y se han visto en la necesidad de ser hospitalizados, en muchos casos por largo tiempo, lapso en el que se espera que, con los cuidados respectivos se alcance una recuperación total de la salud.

Sin embargo, la permanencia en dichos sitios, y muchas veces en posiciones de inmovilidad, hacen que, bajo las condiciones de presión corporal, humedad, temperatura, entre otros, promuevan el aparecimiento de escaras, las cuales si no son tratadas pueden llegar a complicarse y provocar daño permanente al tejido tisular. Lo que representa sufrimientos adicionales para el paciente, permanencia extra en los centros médicos, costos agregados por terapias y gastos en medicinas, entre otros.

A pesar de las metodologías y procedimientos para mitigar el aparecimiento de dichas lesiones, aún se erogán grandes cantidades de dinero y tiempo para aliviar esta condición, por ejemplo, en Estados Unidos se estima que ocurren unos 2.5 millones de casos, derivados de pacientes que permanecen hospitalizados en un rango de 6 o más días, a un costo anual de 11 millardos de dólares (Bauer *et al.*, 2016).

En Guatemala, el autor reconoce que estas lesiones son más frecuentes de lo que se discute acerca de las mismas, no solamente considerando los datos recopilados a través de la encuesta realizada en la Facultad de Ciencias Médicas de la Universidad de San Carlos (ver figuras 1 y 2), sino considerando el precario sistema de salud nacional, por tanto, el atender este tipo de lesiones constituyen una necesidad.

Figura 1. **Factores comunes de incidencia de escaras en pacientes hospitalizados**



Fuente: datos de encuesta de elaboración propia en la Facultad de Ciencias Médicas, Universidad de San Carlos de Guatemala. Centro Universitario Metropolitano (CUM).

Figura 2. **Tiempos de hospitalización extra debido a las escaras**



Fuente: datos de encuesta de elaboración propia en la Facultad de Ciencias Médicas, Universidad de San Carlos de Guatemala. Centro Universitario Metropolitano (CUM).

Los datos anteriores trataron de complementarse con la búsqueda de información relacionada en los sitios *web* del INE, hospital Roosevelt e IGSS, sin embargo, esto resultó infructuoso.

Algunos países cuentan con normas o estándares, los cuales se desconoce si son utilizadas en Guatemala por las empresas de diseño y construcción de clínicas u hospitales. El autor considera que debiera existir una unidad gubernamental o área del Ministerio de Salud Pública y Asistencia Social, que velara por el establecimiento, adopción y la vigilancia de estas normas, las cuales deberían tener carácter obligatorio.

Estas normas permitirían afinar el monitoreo de las condiciones ambientales de los pacientes para el establecimiento de programas de mantenimiento de sensores, controladores y actuadores del sistema de vigilancia y además configurar los umbrales o *thresholds* de las alarmas.

Tabla I. **Normativa UNE 100713-2005 para construcciones hospitalarias en España**

Área del hospital		Rango temperatura (-C)	Rango de humedad (%HR)
Quirófanos	Quirófanos tipo A y B, incluso accidentes y partos	22-26	45-55%
	Pasillos, almacén estéril, entrada y salida		
	Sala despertar		
	Otros locales		
Partos	Paritorios	24-26	
Medicina intensiva	Habitaciones con camas, incluso eventual antesala		
	Habitaciones para pacientes con riesgo de contraer infecciones		
	Para el resto de pacientes		
	Sala de urgencias		
Cuidados especiales	Habitaciones con camas		
	Sala de urgencias		
	Pasillos		
Infeciosos	Habitaciones con camas, incluso eventual antesala		
Prematuros	Habitaciones con camas		
Recién nacidos	Habitaciones con camas		
Otras	Habitaciones con camas para hospitalización		
Quemados	Unidad de quemados (Fuera UNE)	24-26	< 30%
Fisioterapia	Piscinas y baños terapéuticos	A definir	< 65%

Fuente: Asociación Española de Normalización. *UNE 100713-2005: Instalaciones de acondicionamiento de aire en hospitales*. Consultado el 13 de diciembre de 2019.

Recuperado de <https://fisair.com/controlling-humidity-in-hospitals/>

De la tabla I, por ejemplo, se pueden extraer los rangos 24 °C - 26 °C y 45 % - 55 % como niveles aceptables de temperatura y humedad relativa respectivamente para las habitaciones con pacientes en estado de inmovilidad.

Dicha tabla hace referencia a la norma AENOR-UNE 100713:2005 para las instalaciones de acondicionamiento de aire en hospitales españoles, la cual puede resultar útil para los propósitos del presente documento respecto a la observancia de los rangos de admisibilidad de los factores de temperatura y humedad relativa.

Particularmente debe notarse que existe una fuerte correlación entre temperatura y humedad relativa y que esta última posee una incidencia marcada en el apareamiento de escaras, concretamente cuando ocurre una exposición constante de humedad sobre la piel del paciente (ver tablas II y III – sección 4.5.2) provocada por condiciones ambientales o padecimientos como fiebres, incontinencias o carencia de ventilación adecuada.

El conocimiento de los estándares minimizará los problemas conexos o potenciales de hospitalización debido a un mal manejo de la humedad relativa: proliferación de bacterias, hongos, moho, gases, ácaros, entre otros.

- Antecedentes del uso de *message-brokers* (manejadores de mensajes)

El uso de mecanismos o programas intermediarios se remonta a las décadas 80 - 90 debido al auge del apareamiento de aplicaciones de mensajería o *chat*. Y es que los *brokers* ofrecen ventajas como el *casting* o formas de transmisión de datos entre emisores y receptores (e.g. entregas uno a uno, uno a muchos y muchos a muchos), persistencia, transformación, descomposición, ruteo basado en tópicos o contenidos, entre otros.

Lo anterior ha provocado que estas aplicaciones intermediarias se conviertan prácticamente en un patrón arquitectónico en soluciones en sistemas distribuidos.

Stipe, Eugen y Zeljko, en *State-Of-The-Art Of Messaging For Distributed Computing Systems*, indican que uno de los usos más efectivos de los *message-brokers* fue en la mensajería instantánea o *Instant Messaging (IM)* como entes de capa intermediaria o *Message Oriented Middleware (MOM)* debido a que estos aportaron eficacia, eficiencia y versatilidad (Stipe *et al.*, 2017).

Por otra parte, desde el apareamiento de la tendencia *IoT*, los *brokers* se han considerado piezas fundamentales, a tal grado que se consideran generalmente como elementos *sine qua non* debido a que aportan versatilidad, independencia, desacoplamiento en soluciones modulares *multi-vendor*, particularmente cuando se involucran diferentes marcas, tecnologías, dispositivos o estándares.

Adicionalmente, los manejadores de mensajes ofrecen ventajas en los aspectos de *routing* y formateo. Simon Holdsworth en *Selection of communication strategies for message brokers or publish/subscribe communications*, indica que dentro de un sistema de transmisión de mensajes, estos pueden entregarse en la red utilizando uno o más *brokers*, colocados como *hubs* en los canales de comunicación, los cuales proveerán los servicios *ad hoc* de ruteo y formateo de los datos (Holdsworth, 2003).

Prácticamente, los *message-brokers* se pueden emplear en diferentes arquitecturas, sean estas basadas en topologías cliente-servidor, *pipelines*, *peer-to-peer*, bus de servicios, modelos federados y otros más, en términos generales, donde haya productores y consumidores o publicadores y suscriptores (Kampuri, 2014).

La eficacia de los manejadores de mensajes ha hecho que estos sean muy populares debido a que su implementación en los sistemas de envío/recepción de datos continuos o *stream* ha mostrado tener un gran potencial y por tanto resultan útiles en aplicaciones *IoT* (Gerber, 2018).

Sin embargo, la selección de cuál protocolo – o *broker* – es el adecuado para determinada situación, dependerá de las características u orientación particular de los mismos, especialmente en detalles como manejo de latencia, implementación de seguridad, tolerancia a fallos y recuperación, escalabilidad, infraestructura subyacente de red y el tipo de problema a resolver.

2. JUSTIFICACIÓN

La tecnología ha resuelto muchos problemas y continúa siendo una herramienta empleada para mejorar la calidad de vida de las personas, sin embargo, las estadísticas indican que falta mucho por hacer especialmente en el área de la medicina.

Por esta razón, el presente trabajo se fundamenta en la línea de investigación de Tecnologías de la Información y la Comunicación para apoyo a la salud, debido a que planteará un sistema para el monitoreo de los factores de temperatura y humedad precursores de escaras.

Dicha vigilancia o monitoreo puede realizarse a través de la instalación de sensores *ad hoc* situados anatómicamente en las camas, salas o habitaciones de tal forma que puedan cumplir su función e integrarse al entorno sin provocar molestias al paciente.

Adicionalmente, si se desea un ecosistema eficiente, este debe incluir componentes de alta calidad (sensores, controladores, *gateways*, entre otros) y que estos cumplan las condiciones esenciales de conectividad, integridad, alta disponibilidad y baja latencia, requeridos para reacciones en tiempo real.

Por lo anterior, el diseño debe contemplar el uso de mecanismos de intermediación o *message brokers*, para que el sistema pueda manejar los mensajes en forma asíncrona, exista desacoplamiento o independencia de módulos, se minimicen las pérdidas de datos por congestión y brinden un manejo excepcional de latencia.

Se espera que los resultados y conclusiones obtenidas en el presente trabajo resulten útiles para el desarrollo de una solución real, considerando que el problema de las escaras, además de ser un problema de salud es también un problema económico, y que, por tanto, soluciones tecnológicas como la planteada puede contribuir paralelamente con los programas, terapias y protocolos existentes para la atención y reducción del padecimiento.

3. ALCANCES

De acuerdo con el enfoque exploratorio establecido para el presente trabajo y los objetivos específicos planteados, a continuación, se exponen los alcances logrados desde las perspectivas investigativa, técnica y de resultados.

- Perspectiva investigativa

Para lograr la determinación de cuál protocolo es más eficiente en el manejo de mensajes en un ecosistema de sensores de monitoreo de la temperatura y humedad, se requirió de la realización de diversas investigaciones acerca del padecimiento y de la determinación de la eficiencia de *MQTT* y *AMQP* en el entorno de pruebas adoptado.

En este sentido fue necesario:

- Realizar una investigación cualitativa a través de encuestas y de revisión documental de escalas de riesgo del padecimiento para determinar y valorar si efectivamente la temperatura y la humedad son factores precursores de escaras.
- Investigar acerca del enfoque de *IoT* en aplicaciones médicas, específicamente respecto al uso de sensores, controladores, *gateways* y en general de todos los dispositivos involucrados, características, propiedades, formas de conexión, especificaciones, entre otros, y su potencial aplicación al problema de escaras.

- Documentar y exponer las similitudes y diferencias del funcionamiento de los *message-brokers* utilizados en entornos *IoT*, específicamente, los relacionados con al manejo de mensajes en la capa de aplicación, así como los contextos donde se utilizan, especialmente de aquellos relacionados con la salud.
 - Estudiar las referencias documentales asociadas a *MQTT* y *AMQP*, las implementaciones particulares de sus especificaciones, así como los manuales de uso, instalación e interoperabilidad con la herramienta y las librerías del lenguaje de desarrollo seleccionado (i.e. *Python 3.7.3*).
- Perspectiva técnica

De acuerdo con los objetivos del trabajo, fue necesario realizar lo siguiente:

- Interpretar y ponderar las diferentes causas descritas en las escalas de riesgo del padecimiento de escaras y la documentación de estas, con el fin encontrar relaciones directas o indirectas con los factores de temperatura y humedad relativa.
- Conocer las especificaciones de voltaje, energía, sensibilidad y *sampling*, entre otros, de los dispositivos involucrados en el prototipo, así como el formato con el que los sensores y los protocolos manejan y transmiten los datos (e.g. *Plain, RAW, Frames, Stream, XML, JSON*, entre otros). Con la finalidad de estandarizar los mensajes a lo largo todos los *layers* de la cadena publicación-suscripción.

- Conocer las características de los *message-brokers* y como estos coadyuvan en *IoT*, particularmente en aquellos componentes que realizan envío y recepción de mensajes (transformación y traducción de datos, almacenamiento, uso de colas, ruteo, entre otros).
- Para la puesta en marcha del prototipo fue requerido comprender cómo *AMQP* y *MQTT* serían instalados e implementados en el entorno de pruebas, así como lograr que estos funcionaran.
- Perspectiva de resultados

Los resultados obtenidos a lo largo de la investigación y desarrollo del presente documento y experimento permitieron lo siguiente:

- Establecer que la temperatura y humedad efectivamente tienen una correlación con el padecimiento de escaras.
- Determinar que los sensores, controladores, módulos y servicios elegidos para la realización del prototipo, funcionaron bajo los niveles de tolerancia inherentes a la calidad de los mismos y que los resultados obtenidos se consideraron aceptables y admisibles para las conclusiones correspondientes.

Empero, en el caso de los sensores se hace la acotación que los utilizados en el prototipo son de calidad media y por tanto no serían recomendables para un escenario médico real, donde definitivamente, la solución debiera contemplar el uso de sensores de bajo consumo energético, alto desempeño y sensibilidad.

- Justificar que los resultados del experimento con *MQTT* y *AMQP* mostraron la importancia de los *message-brokers* en el uso de estos en entornos *IoT* orientados a la medicina.
- Mostrar que los resultados y conclusiones obtenidas derivadas de las pruebas realizadas, evidenciaron con claridad las virtudes y debilidades de *AMQP* y *MQTT*. Y además, afirmaron el satisfactorio rendimiento de estos en conjunto con los sensores, actuadores y las librerías de *Python* utilizadas en el prototipo propuesto, lo que permitió un adecuado estudio comparativo.

4. MARCO TEÓRICO

4.1. Internet del todo (*Internet of Everything – IoE*)

La expresión *Internet of Everything (IoE)* es un concepto extendido, y más abarcador que el tradicional *Internet of Things (IoT)*. El término *IoE* fue impulsado por *Cisco* (Evans, 2012) quienes lo definieron como “conexiones inteligentes de máquinas, personas, procesos, datos y en general de cosas”.

En este sentido, se amplía el término más allá de comunicaciones maquina a máquina (*M2M*). Sin embargo, en el presente trabajo se ha adoptado el uso de *IoT* por ser de uso más general y conocido.

Uno de los principios subyacentes del *IoT* es utilizar los datos generados por todas las cosas, con el objetivo de convertirlos en información y conocimiento para generar acciones, experiencias enriquecedoras de uso y mejoramiento sin precedentes de la calidad de vida a nivel individual y general.

Adicionalmente, el establecimiento de *IoT* trae consigo repercusiones económicas notables, considerando los ahorros derivados de respuestas más rápidas, reducción de tiempos de ocio o *downtime* de las máquinas, manejo de la cadena de suministros, control de inventarios, mantenimientos auto gestionados por los mismos equipos, ciudades inteligentes – *SmartCities* -, entre otros, es decir, una conexión de dispositivos con la finalidad de colaborar con las actividades sociales, médicas, educativas, comerciales e industriales del ser humano.

Lo anterior aunado a *Ubiquitous-Connectivity*, *Big-Data Analysis*, *Machine Learning*, avances en tecnologías de comunicación inalámbrica, telefonía 5G, entre otros, provocará una explosión de nuevos proyectos, tendencias, filosofías y ordenamientos jurídicos respecto a la privacidad, cuyos efectos o impactos ni aún los expertos han logrado medir con certeza, lo cual dará mucho que hablar en un futuro cercano.

4.2. El modelo TCP/IP

En los años setenta, el Departamento de Defensa de los Estados Unidos (*DOD*) a través de la agencia *DARPA* (*Defense Advanced Research Projects Agency*), encargo a Vinton Cerf y Robert E. Kahn la elaboración de una descripción de guías generales y protocolos de operación que serían implementados en *ARPAnet*. Esto dio origen a este modelo, el cual es mantenido actualmente por la *Internet Engineering Task Force* (*IETF*).

A pesar de que el modelo de *OSI* de siete capas es más utilizado en el ámbito de redes, el presente documento propone el uso del modelo *TCP/IP* por dos razones: 1) debido a que este posee una mejor orientación para los propósitos del contenido del documento; y 2) algunos protocolos en la abstracción de su funcionamiento, no se ajustan solamente a una sola capa del modelo *OSI*.

4.2.1. Descripción de las capas del modelo *TCP/IP*

Rayes y Salam (2016) en su libro *Internet of Things from Hype to Reality*, establecen que el modelo *TCP/IP* se divide en 4 capas o *layers*. Las cuales se describen a continuación.

- Capa 4 – capa de aplicación

Es el estrato más alto del modelo, este equivale a los *layers* de Aplicación, Presentación y Sesión del modelo *OSI*. En esta capa se llevan a cabo los procesos de inicio/final de una comunicación, en este nivel en una relación bidireccional con la capa 3.

- Capa 3 – capa de transporte

Su propósito principal es permitir que los dispositivos fuente y destino lleven a cabo la ‘conversación’, en la capa de transporte se definen los niveles de servicio y se vigila el estado de las conexiones punto a punto.

- Capa 2 – capa de red (Internet)

Este *layer* es el responsable del empaquetado de los datos y se agrega información de la fuente y destino (direcciones *IPv4* o *IPv6*) la cual es utilizada para el manejo del datagrama a lo largo del enrutamiento en la red.

- Capa 1 – capa de acceso al medio

Esta capa es la base del *stack* y combina los *layers* 2 y 1 del modelo *OSI*, *data-link* y físico, respectivamente. Aquí se define cómo los *bits* son enviados a través de la red en sus distintas tecnologías (coaxial, fibra, ondas de radio, entre otros). En esta capa los dispositivos son identificados por direcciones *MAC* (Control de acceso al medio o *Media Access Control*).

4.2.2. Modelo TCP/IP aplicado a IoT

La comunicación es un aspecto central para *IoT*, las diferentes tecnologías empleadas les permiten a los dispositivos, las aplicaciones y los servicios comunicarse entre sí. En un mundo tan heterogéneo se requiere de protocolos que permitan que dicha comunicación se realice en forma segura y confiable.

De acuerdo con información del sitio de *IBM* y publicaciones relacionadas, existen diversidad de tecnologías y protocolos utilizados ampliamente y que han sido adoptados en el mundo de *IoT* (Gerber y Romeo, 2017).

Una muestra de las tecnologías y protocolos más conocidos se presentan en la figura 3.

Figura 3. **Ejemplos de protocolos y tecnologías *IoT* utilizados por capa en el modelo *TCP/IP***

Application	HTTPS, XMPP, CoAP, MQTT, AMQP
Transport	UDP, TCP
Internet	IPv6, 6LoWPAN, RPL
Network access & physical	IEEE 802.15.4 Wifi (802.11 a/b/g/n) Ethernet (802.3) GSM, CDMA, LTE

Fuente: IBM – Sección de Desarrollo. *Guía para la selección de tecnologías de red para enfrentar los retos de IoT*. Consultado el 3 de julio de 2019. Recuperado de <https://developer.ibm.com/articles/iot-lp101-connectivity-network-protocols/>

4.3. Tipos de dispositivos en ecosistemas *IoT*

Como en todo sistema, los diferentes componentes efectúan tareas particulares en su interacción para el logro de sus objetivos. En el caso de *IoT* se cuenta en forma general con los siguientes componentes.

4.3.1. Sensores y actuadores

Los sensores funcionan como los dispositivos de entrada del ecosistema y prácticamente constituyen el punto de inicio de la cadena de eventos de acuerdo con la circunstancia particular. Específicamente son dispositivos que pueden generar una señal eléctrica a partir de cierto evento o acontecimiento físico (Holdowsky, 2015).

Existen sensores orientados a diferentes aplicaciones: acelerómetros, humedad, temperatura, presión, entre otros; por su parte, un actuador recibe un impulso o entrada y generan una acción o evento, son por así decirlo dispositivos de salida. Los actuadores, de acuerdo con el escenario, pueden estar localizados en entornos locales o remotos.

Son entonces, tanto los sensores como los actuadores, dispositivos conocidos como transductores, porque estos reciben energía de alguna fuente y la convierten en otra de naturaleza diferente (Bishop, 2002; Sinha, 2017).

4.3.2. Gateways y controladores

Los *gateways* y los controladores son componentes de interfaz *middle-tier* que recolectan información de los terminales *front-end*, comúnmente sensores, que luego transmiten a componentes *back-end*. Es decir, son una combinación de componentes de *hardware* y *software* que conectan una red con otra (Holdowsky, 2015).

En muchos escenarios estos dispositivos pueden poseer cierto almacenamiento orientado a colas, y cierta capacidad de decisión o “discernimiento” respecto al contenido o *content-type* de los datos de acuerdo con ciertas políticas preestablecidas.

La ubicación de los *gateways* o controladores dependerá de la inteligencia integrada en el ecosistema, las restricciones tecnológicas particulares y/o los requerimientos de eficiencia del diseño del escenario particular.

4.3.3. Almacenamiento

De acuerdo con la arquitectura particular de un determinado sistema *IoT* será necesario contar con medios de almacenamiento para los datos generados: 1) sea para utilizarlos para disparar eventos, alarmas o acciones o bien para llevar a cabo una analítica en tiempo real; o 2) como repositorio para estudios de comportamiento posteriores, en casos donde los datos históricos sean relevantes (e.g. auditoría, contabilidad, rotación de inventarios, entre otros).

4.3.4. Manejadores

La gran cantidad de datos generados por los diferentes dispositivos de entrada pueden ser utilizados como fundamentos de inteligencia para la toma de decisiones en tiempo real o bien para un *batch-processing* posterior. En ambos casos los elementos del sistema encargados de realizar esta tarea se denominan manejadores.

4.3.5. Message-brokers empleados en IoT

Existe una gran variedad de manejadores o gestores de mensajes en uso en la comunidad de las comunicaciones (Vermesan y Friess, 2013), sin embargo, los más populares al año 2020 son los siguientes.

4.3.5.1. AMQP

Este protocolo fue desarrollado en el año 2004 y es mantenido por la organización OASIS (www.oasis-open.org), un consorcio de estándares abiertos. En el año 2014, la versión 1.0 de este protocolo fue aprobada como un estándar ISO/IEC 19464:2014 (Geyer, 2018).

AMQP (Advanced Message Queuing Protocol) provee un método que asegura el transporte de los datos en una forma segura entre aplicaciones en infraestructuras móviles o a través de la nube de una forma agnóstica (independiente de la tecnología).

Por la versatilidad y orientación en el manejo de mensajes (publicación-suscripción), *AMQP* es ampliamente utilizado en implementaciones para la observación oceánica, transporte, juegos en línea, finanzas, entre otros. Las implementaciones más populares de *AMQP* son: *RabbitMQ*, *Zyre*, *OpenAMQP* y *ApacheQpid*.

4.3.5.2. **MQTT**

MQTT (Message Queuing Telemetry Transport) constituye a partir del 2016 el estándar *ISO/IEC PRF 20922:2016* y aunque pertenece al cuerpo de especificaciones de *OASIS* (al igual que *AMQP*), originalmente era un producto de manejo de colas para mensajería creado por Andy Stanford-Clark (*IBM*) y Arlen Nipper (*Eurotech*), quienes inventaron el protocolo para administración de tuberías de petróleo en 1999.

La motivación principal atrás de *MQTT* era diseñar un protocolo ligero que no consumiera mucho ancho de banda con soporte multiplataforma y manejo de niveles de servicios (*QoS*). Características que se han mantenido en el tiempo, al año 2018 la versión impulsada por *OASIS* es la especificación de protocolo 3.11, la que, como se mencionó previamente, constituye un estándar.

Sin embargo, existen opiniones encontradas, por ejemplo, de acuerdo a Janakiram Msv: "... *MQTT* no es otra implementación de una infraestructura *MOM, Message Oriented Middleware*, ya que no es un protocolo relacionado con manejo de colas y por tanto no puede reemplazar a estos mecanismos más especializados" (Janakiram, 2016).

La diferencia principal entre las implementaciones *MOM* y *MQTT* es que este último no considera la durabilidad o persistencia de los mensajes y por tanto no podría ser utilizado en sistemas con enfoque persistente (*store-and-forward*) y por tanto no podría utilizarse como *middleware* en sistemas transaccionales, donde no sería tolerable la pérdida de mensajes.

Sin embargo, la característica de liviandad de *MQTT* lo hace muy eficiente en comunicaciones entre dispositivos a través del uso de *APIs*, adicionalmente, este puede utilizar otras infraestructuras *publish/subscribe*.

4.3.5.3. CoAP

El protocolo de aplicación restringida (*Constrained Application Protocol*) es un protocolo especializado en transferencias *web* en contextos de redes o nodos restringidos en el ámbito del Internet de las Cosas.

Originalmente, fue diseñado para aplicaciones de automatización de edificios, uso inteligente de la energía e interacciones máquina a máquina – *M2M* –. *CoAP* posee grandes similitudes con *HTTP* respecto a los métodos utilizados y el empleo de *REST* con formatos *XML*, *JSON* y *CBOR* (Bormann, 2016).

CoAP posee la virtud de trabajar con gran cantidad de nodos a un bajo costo debido a sus bajos requerimientos en especificaciones de memoria de micro controlador, razón por la cual es muy utilizado en *IoT* y actualmente continúa siendo desarrollado bajo documento estándar *RFC 7252*.

4.4. Factores que afectan las comunicaciones

La eficacia y eficiencia de la comunicación entre dos puntos puede ser afectada por gran cantidad de factores dependiendo de las características inherentes de los medios físicos de comunicación, las formas de transmisión, los protocolos empleados y la influencia de factores externos como campos magnéticos, temperatura, distancia, en otros.

A continuación, se provee una lista de términos relacionados con la comunicación dada particularmente en el contexto de redes de nodos, dispositivos informáticos o computadoras que afectan las transmisiones.

4.4.1. Latencia

En el ámbito de las comunicaciones, se conoce como latencia a la suma de los retardos temporales dadas en las conexiones de red. Dichos retardos son producidos por la demora en la propagación y transmisión de paquetes en el sistema de dispositivos y el medio utilizado, sea este inalámbrico o cableado.

Todos los sistemas poseen latencia por consideraciones inherentes a las diferentes tecnologías, algoritmos, distancias, entre otros, y aunque hipotéticamente se hable de cualidad de latencia cero, es solamente una cuestión de percepción

Un punto muy importante es que todos los sistemas de comunicaciones siempre van a poseer cierta cualidad de latencia y que cuando se trate de latencia cero, lo que realmente sucede es que esta es imperceptible en un rango por debajo de 100 milisegundos.

4.4.2. Lags

En el mundo de las comunicaciones existe una relación tan estrecha entre *lag* y latencia que muchas veces se utilizan en forma intercambiable, sin embargo, para ser más específico, se dirá que un *lag* es un retraso comúnmente ocurrido en sistemas de alta latencia (cualidad) debido a un mal desempeño de procesamiento, bajos anchos de banda, pérdida de paquetes, entre otros.

4.4.3. Jitters

Los *jitters* son datos que se envían con cierta secuencia o sincronización bajo un dominio de frecuencias marcadas por un reloj, pero llegan a su destino fuera de dicha especificación (de-sincronizados).

Este efecto de desplazamiento temporal es más fácil de percibir en transmisiones de datos que representan magnitudes análogas (v.g. audio y video) causando degradación del desempeño de la transmisión.

4.4.4. Atenuación

La atenuación es la degradación de una señal debido al tiempo o la distancia. Su ocurrencia se debe a que el cable en sí ofrece cierta resistencia al flujo de la señal, lo que provoca una reducción en la señal eléctrica conforme esta se desplaza en el medio utilizado (e.g. cableado).

Cuando la atenuación es muy grande, la proporción señal/ruido es tan inaceptable que el dispositivo receptor no podrá diferenciar entre ellas lo cual produce pérdidas o reenvíos, aumentando así la latencia.

Para resolver esto se han establecido límites máximos en las longitudes del cableado de acuerdo con el material del mismo, así como el uso de amplificadores de señal para mantener los niveles de aceptabilidad del sistema (Hill, 2002).

4.4.5. Calidad de servicio

Calidad de servicio (*QoS – Quality of Service*) es un artificio para generar prioridades dentro de las comunicaciones de red de un sistema para garantizar un ancho de banda aceptable para un determinado servicio el cual estaría compitiendo por esta de no existir dicho sistema.

QoS permite la asignación de ciertos *tags* para lograr la priorización de cierto tipo de servicios cuyo nivel de aceptación depende de una alta eficiencia de entrega y reducción de *jitters*, por ejemplo, en el caso de comunicaciones estándar de audio y video (e.g. *VoIP*, *IPTV*, juegos *on-line*, videoconferencia, entre otros).

La meta principal de QoS es, en resumen, la provisión de prioridades al tráfico de una red de acuerdo con su contenido para dedicar ancho de banda con la finalidad de minimizar los *jitters*, reducir las pérdidas de paquetes y proveer al sistema de una baja latencia a través de la identificación y el marcado (o *tagging*) que coordine y garantice calidad en el servicio de un punto a otro entre los elementos de una red.

4.4.6. Tipo de servicio

El mecanismo de *ToS (Type of Service)* se relaciona con la morfología de un datagrama debido a que es un campo de la estructura de su encabezado, específicamente el segundo byte. Su enfoque de uso es similar que el de QoS en el sentido que se emplea para establecer la prioridad del datagrama dentro de la comunicación a través del requerimiento de una ruta de bajo retraso, alta confiabilidad o alto *throughput*.

De acuerdo con el valor del campo *ToS* un paquete es priorizado dentro de las colas o redirigido a través de rutas con latencia baja. Los ocho *bits* de *ToS* son utilizados en redes *IPv4* e *IPv6* como indicadores de congestión explícita (*ECN*) y de diferenciación de servicios (*DS*).

4.4.7. Servicios diferenciados - *differentiated services (DS - diffserv)*

Los servicios diferenciados (*DS*) brindan un método para la garantía de calidad de servicio en redes como Internet. Para lograrlo, los paquetes *IP* se clasifican en clases, en función de diferentes términos de *QoS* que poseen especial relevancia para la conexión.

Uno de los usos más populares es el uso de la categoría *EF PHB (Expedited Forwarding Per-Hop Behavior)* empleado para la transmisión de contenido multimedia que por su requerimiento de tiempo real requieren bajo nivel de pérdidas de paquetes, así como de baja latencia.

Como se mencionó previamente, ahora se utilizan los *bits* de *ToS*: seis empleados para uso de los servicios diferenciados y dos para la Notificación Explícita de Congestión (*ECN, Explicit Congestion Notification*); este último utilizado para detectar la congestión en la red sin necesidad de utilizar paquetes específicos para ello; adicionalmente, algunos dispositivos pueden recurrir a *DSCP (Differentiated Services Code Point)* modificando algunos *bits* las cabeceras de los paquetes *IP* para diferenciar la calidad en la comunicación que tienen los datos que se transportan.

Para el caso particular de *IPv6* se ha ampliado el término a clase de tráfico, *Traffic Class*, y los ocho *bits* mencionados se utilizan en la misma forma (seis primeros para *DS*, y los dos últimos para *ECN*).

Debido a que los servicios diferenciados analizan varios flujos de datos en vez de conexiones únicas o reservas de recursos, son regularmente utilizados para establecer negociaciones dentro de los Acuerdos de Nivel de Servicio (*SLA – Service Level Agreement*). Estos *SLA* especifican qué clases de tráfico serán brindados, las garantías de cada clase y los datos considerados para cada clase.

4.5. Escaras

El conocimiento de la valoración de las condiciones que pueden provocar úlceras o escaras permitirá un mejor conocimiento del problema, así como los márgenes de variabilidad aceptables para el diseño de la solución. A continuación, se expone parte de la terminología respectiva y algunas escalas empleadas en su valoración.

4.5.1. Etiopatogénesis de las escaras

El vocablo Etiopatogénesis proviene de dos términos: Etiología y Patogénesis, ambos hacen referencia a las razones o causas del cómo se produce una enfermedad concreta, tanto es sus factores endógenos como exógenos.

De acuerdo al contexto geográfico o cultural de las diferentes regiones, los términos escara y úlcera son manejados como sinónimos, aunque algunos autores difieren al respecto, por ejemplo, la publicación del Centro Nacional de

Información de Ciencias Médicas de Cuba, denomina que: "... [una úlcera por presión o inmovilidad es] aquella lesión de origen isquémico producida por el daño tisular por presión, rozamiento o fricción prolongada (o una combinación de estos) de la piel entre una prominencia ósea y un plano duro" (Centro Nacional de Información de Ciencias Médicas de Cuba, 2017, p.1).

Considerando lo indicado en el párrafo anterior y para estandarizar el uso de la palabra, el documento trata el término úlcera como escara.

Dicha publicación también cita que estas afecciones de la piel son muy comunes particularmente: "... las encontradas en pacientes críticos hospitalizados o que requieran atención institucional a largo plazo [...], la incidencia varía mucho según el entorno clínico, siendo mayor en los países en vías de desarrollo, debido a deficiencias en el sistema sanitario" (Centro Nacional de Información de Ciencias Médicas de Cuba, 2017, p.2).

Dicho daño tisular complicarse aún bajo las condiciones de humedad y temperatura como es natural, debido a la estrecha relación entre calor y sudor o bien condiciones particulares del paciente como incontinencia urinaria.

La sección de antecedentes del presente documento muestra la influencia de la temperatura y humedad como condiciones precursoras del padecimiento.

4.5.2. Escalas de medición de riesgo

Para valorar las condiciones de predisposición o riesgo al padecimiento se han creado algunas escalas. Todas las analizadas (Norton, Braden, Nova5, Arnell, Emina, entre otras) consisten en un pequeño cuestionario o *scoreboard*, donde se obtiene un puntaje sumando los *ítems*, el que luego es cotejado con una tabla que indica, de acuerdo con el rango del punteo, el nivel o porcentaje de riesgo del paciente.

De las escalas mencionadas, solamente dos de ellas recalcan más la incidencia de factores físicos como la humedad. Por esta razón dentro del trabajo se considerarán solamente las escalas de Braden y Emina, las que se utilizarán para el establecimiento de las pruebas en el prototipo. Dicha humedad puede ser producida por el entorno ambiental o condiciones inherentes al paciente (e.g. incontinencia urinaria).

Las escalas mostradas en las tablas II y III son auto explicativas, se adjuntan como ilustración de los factores que tienen incidencia en este tipo de lesiones.

Tabla II. Escala de Braden-Q para medición de riesgos de úlceras de piel

ESCALA BRADEN Q					
<i>Intensidad y duración de la presión</i>					PTOS
MOVILIDAD Capacidad para cambiar y controlar la posición del cuerpo	1.- Completamente inmóvil: No realiza ningún movimiento con el cuerpo o las extremidades sin ayuda.	2.- Muy limitada: Ocasionalmente realiza ligeros cambios en la posición del cuerpo o de las extremidades, pero no es capaz de realizar cambios frecuentes o significativos sin ayuda.	3.- Ligeramente limitada: Realiza frecuentes aunque ligeros cambios en la posición del cuerpo o de las extremidades sin ayuda.	4.- Sin limitaciones: Realiza importantes y frecuentes cambios de posición sin ayuda.	
ACTIVIDAD Nivel de actividad física	1.- Encamado: Limitado/a a la cama.	2.- En silla: Capacidad para caminar severamente limitada o inexistente. No puede soportar su propio peso y/o hay que ayudarle para sentarse en una silla o en una silla de ruedas.	3.- Camina ocasionalmente: Camina ocasionalmente durante el día, pero distancias muy cortas, con o sin ayuda. Pasa la mayor parte de cada turno en la cama o en la silla.	4.- Todos los pacientes demasiado jóvenes para caminar O camina frecuentemente. Camina fuera de la habitación al menos dos veces al día y dentro de la habitación al menos una vez cada dos horas durante las horas de paseo.	
PERCEPCIÓN SENSORIAL Capacidad de responder adecuadamente, según su nivel de desarrollo, al discomfort relacionado con la presión	1.- Completamente limitada: No responde a estímulos dolorosos (no gime, no se estremece ni se agarra) debido a un bajo nivel de conciencia o a sedación O capacidad limitada para sentir dolor en la mayor parte del cuerpo.	2.- Muy limitada: Responde sólo a estímulos dolorosos. Incapacidad para comunicar malestar, excepto gimiendo o mostrando inquietud O tiene alguna alteración sensorial que limita la capacidad de sentir dolor o malestar en más de la mitad del cuerpo.	3.- Ligeramente limitada: Responde a órdenes verbales, pero no siempre puede comunicar la incomodidad o la necesidad de ser cambiado de postura O tiene alguna alteración sensorial que limita la capacidad de sentir dolor o malestar en una o dos extremidades.	4.- Sin limitaciones: Responde a órdenes verbales. No tiene ninguna alteración sensorial que limite su capacidad de sentir o comunicar dolor o malestar.	
<i>Tolerancia de la piel y la estructura de soporte</i>					
HUMEDAD Nivel de exposición de la piel a la humedad	1.- Piel constantemente húmeda: La piel se mantiene húmeda casi permanentemente debido a la transpiración, orina, drenaje, etc. La humedad es detectada cada vez que se gira o mueve al paciente.	2.- Piel muy húmeda: La piel está húmeda a menudo, pero no siempre. La ropa de cama debe ser cambiada al menos cada 8 horas.	3.- Piel ocasionalmente húmeda: La piel está húmeda de forma ocasional, necesitando cambio de la ropa de cama cada 12 horas.	4.- Piel raramente húmeda: La piel está casi siempre seca. Hay que cambiar los pañales de forma rutinaria; la ropa de cama solo hay que cambiarla cada 24 horas.	
FRICCIÓN Y CIZALLAMIENTO Fricción: ocurre cuando la piel se mueve contra las superficie de apoyo Cizallamiento: ocurre cuando la piel y la superficie ósea adyacente rozan la una contra la otra	1.- Problema significativo: La espasticidad, contracturas, picores o agitación le llevan a un movimiento y fricción casi constantes.	2.- Problema: Requiere ayuda de moderada a máxima para moverse. Es imposible levantarse completamente sin que se produzca un roce contra las sábanas. Frecuentemente se desliza hacia abajo en la cama o en la silla, necesitando recolocarlo con la máxima ayuda.	3.- Problema potencial: Se mueve débilmente o necesita una ayuda mínima. Mientras se mueve, es posible que la piel roce de algún modo con las sábanas, la silla u otros dispositivos. La mayor parte del tiempo mantiene una posición relativamente buena en la silla o cama, aunque ocasionalmente se desliza hacia abajo.	4.- Sin problema aparente: Es posible levantarlo completamente durante un cambio de postura. Se mueve independientemente en la cama y en la silla, y tiene suficiente fuerza muscular para levantarse completamente mientras se mueve. Mantiene una buena postura en la cama o en la silla en todo momento.	
NUTRICIÓN Patrón habitual de ingesta de alimentos	1.- Muy pobre: Está en ayunas; o a dieta líquida; o con sueroterapia más de 5 días O albúmina < 2.5 mg/dl O nunca come una comida completa. Raras veces come más de la mitad de cualquier comida ofrecida. La ingesta de proteínas incluye 2 o menos raciones de carne, pescado o productos lácteos al día. Toma pocos líquidos. No toma un suplemento dietético líquido.	2.- Inadecuada: Se le administra una nutrición enteral (SNG) o nutrición parenteral (IV) que le proporciona una cantidad inadecuada de calorías y minerales para su edad O albúmina < 3 mg/dl O raras veces come una comida completa y generalmente sólo come la mitad de cualquier comida ofrecida. La ingesta de proteínas incluye sólo 3 raciones de carne, pescado o productos lácteos al día. Ocasionalmente toma un suplemento dietético.	3.- Adecuada: Se le administra una nutrición enteral (SNG) o nutrición parenteral (IV) que le proporciona una cantidad adecuada de calorías y minerales para su edad O come la mitad de la mayoría de las comidas. Come un total de 4 raciones de proteínas al día (carne, pescado, productos lácteos). Ocasionalmente rechaza una comida, pero normalmente toma un suplemento si se le ofrece.	4.- Excelente: Toma una dieta normal que le proporciona las calorías adecuadas para su edad. Por ejemplo: come/bebe la mayor parte de cada comida/toma. Nunca rechaza una comida. Normalmente come un total de 4 o más raciones de carne, pescados o productos lácteos al día. Ocasionalmente come entre comidas. No necesita suplementos.	
PERFUSIÓN TISULAR Y OXIGENACIÓN	1.- Muy comprometida: Hipotensión (TA _{ec} <50mmHg; <40 en recién nacidos) O el paciente no tolera fisiológicamente los cambios posturales.	2.- Comprometida: Normotenso; el pH sérico es < 7.40; la saturación de oxígeno puede ser < 95% O la hemoglobina puede ser < 10mg/dl O el relleno capilar puede ser > 2 segundos.	3.- Adecuada: Normotenso; el pH de la sangre es normal; la saturación de oxígeno puede ser < 95% O la hemoglobina puede ser < 10mg/dl O el relleno capilar puede ser > 2 segundos.	4.- Excelente: Normotenso; la saturación de oxígeno es >95%; hemoglobina normal; y el relleno capilar < 2 segundos.	
					TOTAL

© Quigley S & Curley M. 1996.

© Versión en castellano: Quesada C. Departamento de Sanidad y Consumo. Gobierno Vasco. 2009.

CON RIESGO ≤ 16 puntos

SIN RIESGO > 16 puntos

Fuente: EstudyLib. Documento 5067281: Escala Braden-Q. Consultado el 11 de julio de 2019.

Recuperado de <https://studylib.es/doc/5067281/escala-braden-q>

Tabla III. Escala Emina para medición de riesgos de úlceras de piel

ESCALA EMINA

	Estado mental	Movilidad	Humedad R/C Incontinencia	Nutrición	Actividad
0	<u>Orientado</u> Paciente orientado y consciente	<u>Completa</u> Autonomía completa para cambiar de posición en la cama o en la silla	<u>No</u> Tiene control de esfínteres o lleva sonda vesical permanente, o no tiene control de esfínter anal pero no ha defecado en 24 horas	<u>Correcta</u> Toma la dieta completa, nutrición enteral o parenteral adecuada. Puede estar en ayunas hasta 3 días por prueba diagnóstica, intervención quirúrgica o con dieta sin aporte proteico. Albúmina y proteínas con valores iguales o superiores a los estándares de laboratorio	<u>Deambula</u> Autonomía completa para caminar
1	<u>Desorientado o apático o pasivo</u> Apático o pasivo o desorientado en el tiempo y en el espacio. (Capaz de responder a órdenes sencillas)	<u>Ligeramente limitada</u> Puede necesitar ayuda para cambiar de posición o reposo absoluto por prescripción médica	<u>Urinaria o fecal ocasional</u> Tiene incontinencia urinaria o fecal ocasional, o lleva colector urinario o cateterismo intermitente, o tratamiento evacuador controlado	<u>Ocasionalmente incompleta</u> Ocasionalmente deja parte de la dieta (platos proteicos). Albúmina y proteínas con valores iguales o superiores a los estándares de laboratorio.	<u>Deambula con ayuda</u> Deambula con ayuda ocasional (bastones, muletas, soporte humano, etc.)
2	<u>Letárgico o hipercinético</u> Letárgico (no responde órdenes) o hipercinético por agresividad o irritabilidad	<u>Limitación importante Siempre</u> necesita ayuda para cambiar de posición	<u>Urinaria o fecal habitual</u> Tiene incontinencia urinaria o fecal, o tratamiento evacuador no controlado	<u>Incompleta</u> Diariamente deja parte de la dieta (platos proteicos). Albúmina y proteínas con valores iguales o superiores a los estándares de laboratorio	<u>Siempre precisa ayuda</u> Deambula siempre con ayuda (bastones, soporte humano, etc.)
3	<u>Comatoso</u> Inconsciente. No responde a ningún estímulo. Puede ser un paciente sedado	<u>Inmóvil</u> No se mueve en la cama ni en la silla	<u>Urinaria y fecal</u> Tiene ambas incontinencias o incontinencia fecal con deposiciones diarreicas frecuentes	<u>No ingesta</u> Oral, ni enteral, ni parenteral superior a 3 días y/o desnutrición previa. Albúmina y proteínas con valores inferiores a los estándares de laboratorio	<u>No deambula</u> Paciente que no deambula. Reposo absoluto
<p>Sin riesgo: 0; Riesgo bajo: 1-3; Riesgo medio: 4-7; Riesgo alto: 8-15</p>					

Fuente: Ulceras.net. *ESCALA%20EMINA.pdf*. Consultado el 13 de julio de 2019.

Recuperado de <https://www.ulceras.net/userfiles/files/ESCALA%20EMINA.pdf>

5. PRESENTACIÓN DE RESULTADOS

5.1. Diseño del prototipo

Para lograr los objetivos propuestos en el presente documento se tuvo que recurrir a la experimentación a través de la implementación de un prototipo. Las pruebas fueron realizadas dentro de algunos márgenes o condiciones de frontera establecidos considerando los recursos tecnológicos disponibles y asequibles con el presupuesto financiero planificado.

A continuación, se muestran detalles de la arquitectura empleada, las características de los componentes del prototipo, el modelo de control, la parametrización de las pruebas, descripción de las variables y los resultados estadísticos obtenidos entre otros.

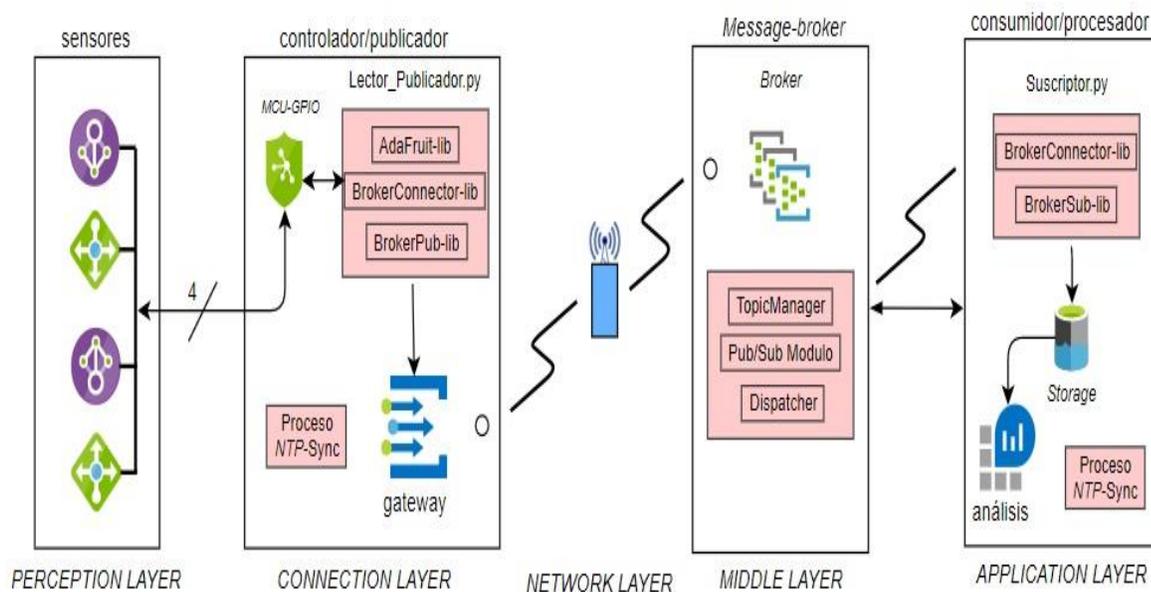
5.1.1. Condiciones de frontera

Como se estableció previamente, el prototipo está enfocado a investigar el desempeño de *MQTT* y *AMQP* desde varias aristas (e.g. características de funcionamiento, instalación, configuración, latencia, entre otros), evaluación que resulta muy útil como modelo extensible a implementaciones *IoT* locales, *edge* o *fog*, debido a que estas requieren buen procesamiento y respuestas rápidas como la del escenario médico.

Sin embargo, por disponibilidad de recursos presupuestarios e infraestructura fue necesario establecer las siguientes condiciones o *constraints* físicos del modelo:

- El prototipo propuesto trabajará sobre una red local con la arquitectura mostrada en la figura 4, la imagen muestra en forma abstracta la interrelación de las 5 diferentes capas utilizadas en el prototipo.

Figura 4. **Arquitectura propuesta para el prototipo, patrón *publish/subscribe* uno-a-uno**



Fuente: elaboración propia, realizado con *Google - Draw.io* (<https://app.diagrams.net/>).

- La comunicación entre los sensores y el controlador es alámbrica (a través de *jumper-cables*).
- La conexión entre el *message-broker* (*middle layer*) y el controlador (*connection layer*) se realizó a través de wifi (*network layer*).
- La figura 4 coloca las capas una a la par de la otra por motivos esquemáticos y facilidad de lectura, sin embargo, el vínculo entre los *layers* correspondientes al *broker* y al consumidor deben interpretarse

como conexiones a través de la capa *network* (la cual puede ser física o inalámbrica, y puede atravesar múltiples *routers* o *access-points*).

- A pesar de que en *IoT*, la medición del consumo energético es muy importante, este no fue considerado en el prototipo y por tanto no fue considerado en el estudio de eficiencia.
- La medición de los tiempos se realizó con una exactitud de 10^{-6} segundos.
- La frecuencia de lecturas de los sensores se realizó basándose en las especificaciones y restricciones de los mismos, a decir, el calentamiento del controlador y los sensores y otras condiciones inherentes de la tecnología wifi (e.g. *EMI*).
- Respecto a los *brokers* estudiados se centró en la observancia del modelo publicación/suscripción de mensajes únicamente y no se realizaron pruebas de persistencia de mensajes (envío a posteriori), alta disponibilidad, escalabilidad, recuperación de conexiones o *failbacks*, entre otros.
- Una restricción sobreentendida es que los protocolos estudiados trabajan por el momento exclusivamente sobre la capa de *TCP* – no *UDP*. Como lo establecen las normas *ISO/IEC* 19464:2014 y 20922:2016 en los incisos 2.8.19 y 4.2 respectivamente.

- La configuración de uso de ambos protocolos fue la *default*, no se realizó ningún cambio a los archivos de configuración, puertos o parámetros extra en los tiempos de corrida. No se emplearon *callbacks*, ni se cambiaron parámetros de *heartbeats* o *keep-alives* u otro *tweak* de funcionamiento. Tampoco se contempló el uso de *acknowledgements* o mecanismos de *reply* en las pruebas.
- Todas las pruebas *pub/sub* se realizaron uno-a-uno: un publicador, un suscriptor. No varios-a-varios, uno-a-varios o varios-a-uno.

5.1.2. Componentes del prototipo

Descripción de los diferentes dispositivos, equipos y protocolos empleados en el prototipo.

- Sensores digitales: se emplearon dos sensores *AM2302* montados sobre un *proto-board*, cada uno con una resistencia (*pull-up*) de 10k Ohms. Dichos sensores brindaron los datos de temperatura y humedad relativa, elementos que influyen en el aparecimiento de escaras.

Descripción técnica más importante de los sensores:

- Voltaje de funcionamiento: 3.3V – 5.0V, *DC*.
- *Sampling* o muestreo de lecturas: 2 segundos.

- Rango de medición: Temperatura: -40 °C a 80 °C y Humedad Relativa: de 0 % a 100 %.
- Precisión: Temperatura: +/- 0.5 °C y Humedad Relativa: +/- 2 %
- Microcontrolador (*MCU*): Como micro controlador se utilizó un *Raspberry-Pi 3 B+*. La conexión a los sensores se realizó como se describe a continuación:
 - *GPIOs* 5 y 6 (pines 29 y 31) para los conectores de *data*.
 - Pines 1 y 17 para la alimentación (voltaje) 3.3V.
 - Pines 6 y 14 para la conexión a tierra o *ground – GND*.
- El sistema operativo en el micro controlador fue *Raspbian Jesse*, en el cual se instalaron los siguientes componentes: 1) *Python 3.7.3*, lenguaje en el que se elaboraron la rutinas de lectura y publicación de los datos; 2) Librería *Adafruit-DHT* versión 1.4.0 como interfaz de captura de la *data* de los sensores conectados en los *GPIOs* indicados; 3) *Software-Publish: Mosquitto* versión 1.5.7 para la comunicación con el *broker MQTT* y *Kombu* versión 4.6.8 y *amqp* 2.5.2 para la comunicación al servidor correspondiente de *AMQP*.
- Entorno LOCAL: lo constituyó una red wifi (*WPAN*) 802.11bgn a 2.4 GHz y seguridad *WPA2-Personal (AES)*. Productor y consumidor estáticos en un radio de 3 a 3.5 metros al punto de acceso en línea-*vista* (i.e. sin obstrucciones).

- Estación de trabajo *Intel Core™ I7-3520M* (2 núcleos), *CPU@2.90 GHz*, *RAM: 16.0 GB*, *Windows 10 Pro x64* con *software broker Mosquitto v.1.6.9* y *RabbitMQ v.3.8.3*. Este equipo también fue utilizado para la instalación de los módulos o *scripts* consumidores, así como del almacenamiento de los datos provenientes de los sensores y de la analítica de comportamiento.

En la estación de trabajo también fue requerido afinar las reglas de *firewall* correspondiente para abrir los puertos utilizados por los dos *brokers*.

- *Network Time Protocol (NTP)* externo: para las pruebas se utilizó el servicio *0.north-america.pool.ntp.org* tanto para la estación de trabajo (en el *application layer*) como en el micro controlador. Este servicio fue requerido luego de observar las discrepancias en los datos de emisión/recepción y significó una condición *sine-qua-non* para medir la latencia de los protocolos. También se pudo optar por la implementación de un *NTP* dentro del ámbito de la arquitectura propuesta.

5.1.3. Modelo de control

Para evitar algún sesgo en los resultados inducido por la arquitectura adoptada (*hardware*, conexiones, sistema operativo, entre otros), el prototipo estableció condiciones de control *ceteris páribus - cet. par.* - para el modelo publicación/subscripción (*pub/sub*) de los algoritmos evaluados, de tal forma que lo único variable yació en el *message-broker* evaluado.

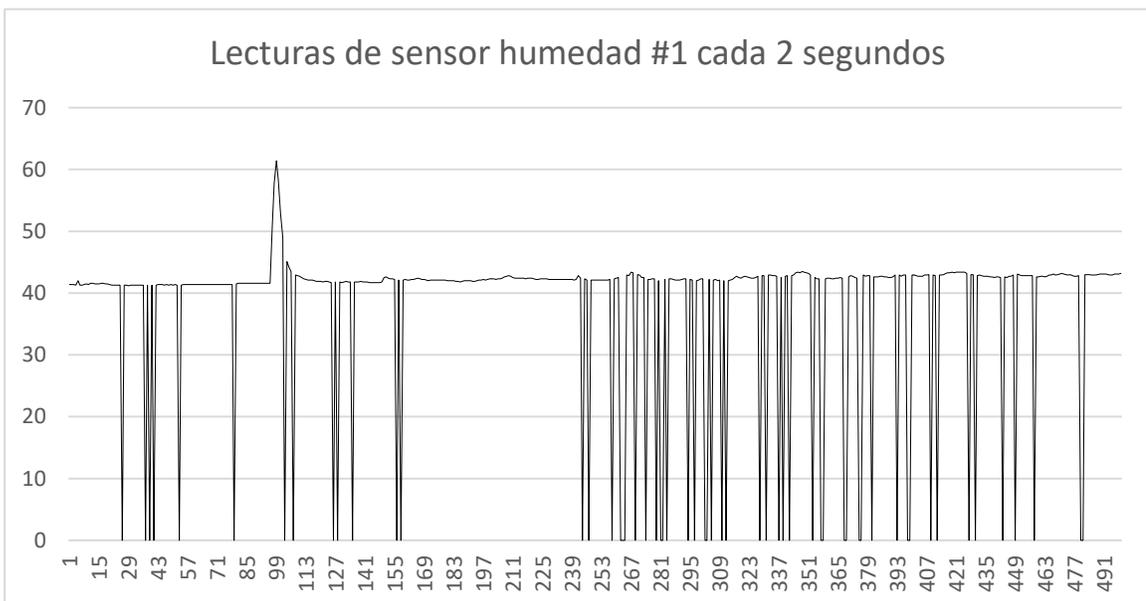
Como se indicó previamente en la sección 5.2.2 inciso 4, para la evaluación de los *brokers*, se seleccionaron las implementaciones de

especificación siguientes: para *MQTT: Mosquitto v.1.6.9* – de *Eclipse Foundation* –, y para *AMQP: RabbitMQ v.3.8.3* – de *Pivotal Software* -.

El modelo de control no solamente lo constituyeron los elementos, las conexiones y el *software* mencionado, sino también una evaluación de las fallas de lectura de los sensores para la afinación del funcionamiento del prototipo (datos con valor cero, *Outliers*, *Nones*, entre otros).

Para lograrlo se realizaron pruebas en *sets* de 500, 200 y 100 lecturas, para medir el funcionamiento de los sensores bajo estrés, considerando no solamente la baja calidad de estos, sino el natural calentamiento de los circuitos y/o termistores dados por la alta frecuencia de lecturas.

Figura 5. **Pre-muestreo para la obtención de la cantidad óptima de lecturas**



Fuente: diagrama de elaboración propia utilizando *MS-Excel*.

La figura 5 muestra la importancia del pre-muestreo: los picos indican fallas de lectura. Sin embargo, se logró obtener un estimado de fallas por sensor de aproximadamente 11 % en las lecturas.

En total se realizaron 5,500 lecturas de los sensores, de las cuales fueron descartadas 3,500 debido a diversas fallas (calentamiento, lecturas *out-of-sync* en tiempos envío/recepción, problemas de electricidad, *time-outs* excesivos, entre otros); finalmente, se conservaron 2,000 lecturas válidas que fueron las empleadas para la obtención de los resultados presentados.

5.1.4. Parametrización de pruebas

Para la realización del experimento, se utilizaron las variables indicadas en la tabla IV. Dichos elementos incluyen variables cuantitativas y cualitativas. Respecto a las primeras, el prototipo buscó evaluar los siguientes tópicos de eficiencia/desempeño importantes: tiempos de respuesta o latencia, integridad, manejo del *payload/overhead*, *content-type* y aspectos de seguridad (AAA y SSL).

Las variables F (Facilidad de implementación), C (Complejidad de implementación) y f (Funcionalidades adiciones) evaluadas son de carácter cualitativo y, por tanto, poseen un cierto grado de subjetividad y sesgo proveniente del autor.

Respecto a las variables, la tabla IV muestra sus dimensionales, así como el rango (valor máximo y mínimo esperado) y/o el conjunto de elementos que se les podría asignar a las mismas de acuerdo a la evaluación de los resultados del experimento.

Tabla IV. **Descripción de variables utilizadas en el experimento**

Variable	Definición (dimensionales) [escala de variabilidad]
T	Temperatura (grados centígrados) [10 - 50]
H	Humedad relativa (porcentajes) [15 - 75]
F	Facilidad de implementación (de acuerdo con el tiempo empleado para leer la documentación, manuales de referencia, pruebas, entre otros) hasta la puesta en marcha del modelo (horas) [24 - 120]
C	Complejidad del uso (disponibilidad de librerías, adaptabilidad a la plataforma de desarrollo de módulos <i>Python</i>) (horas) [24 - 120]
I	Integridad, tipo de contenido y manejo del <i>payload/overhead</i> de los datos Sensor/Consumidor: si estos se alteraron en tránsito, el tipo de contenido que se puede transportar - <i>content-type</i> , entre otros. (8 dígitos significativos de las mediciones de los sensores) [0 – 0.99999999]; porcentaje comparativo en manejo del <i>payload</i> [X.XX %]; <i>content-type</i> [contenido del mensaje]
Q	Mecanismos de aseguramiento o garantía de entrega de mensajes – <i>Quality of Service (QoS)</i> . Dependiendo de los niveles de garantía (niveles) [0, 1, 2, 3]
S	Seguridad [ninguna, baja, alta y fuerte] de acuerdo con el uso de estándares AAA y/o mecanismos criptográficos, infraestructura de llave pública, entre otros.
t	Tiempo de respuesta para obtención de datos – latencia (segundos) [0.001 – 10,000]
f	Funcionalidades adicionales del producto evaluado (características) [0 – 5]

Nota: los rangos de comportamiento aceptable esperado de las variables en el experimento se muestran entre corchetes.

Fuente: elaboración propia utilizando *MS-Excel*.

5.2. Realización de pruebas

En resumen y basados en la tabla IV, para la realización de pruebas de desempeño y eficiencia se optó por evaluar lo siguiente: 1) facilidad de implementación; 2) complejidad de configuración y disponibilidad de librerías o herramientas de desarrollo; 3) consistencia o integridad de los datos, manejo de *payloads* y tipos de contenido de los mensajes; 4) herramientas de calidad de servicio – QoS; 5) elementos de seguridad (i.e. AAA, SASL, TLS, entre otros); 6) tiempos de respuesta; y 7) Funciones adicionales al patrón publicación/suscripción.

Se recalca que para medir los tiempos de respuesta fue requerido el uso de un servidor *NTP* (*Network Time Protocol*) que proporcionara una sincronización de los productores y consumidores, tal que los datos de tiempo obtenidos estuvieran dentro de una dispersión admisible o error mínimo y permitieran una comparación lógica o coherente. Para lograrlo se realizaron varias sincronizaciones hasta lograr *offsets* y *jitters* menores a 2.000 ms., como el indicado en la figura 6.

Figura 6. **Ejemplo de sincronización *NTP*/micro-controlador previo a las pruebas**

```
pi@raspberrypi:~$ ntpq -pn
remote          refid          st t when poll reach  delay  offset jitter
=====
*206.108.0.132 .PTP0.         1 u  64  64  17  78.641  1.809  1.084
```

Nota: La figura muestra los diferentes parámetros de ejecución de la sentencia *ntpq -pn*.

Fuente: Elaboración propia. La captura de pantalla fue realizada con aplicación *ScreenCapturer*.

Respecto a la nomenclatura o nombre de los sensores, debe mencionarse que se empleó el dato de Sensor Café y Sensor Azul, por la facilidad de ubicación física de estos en el prototipo, debido a que fue el color del cable utilizado para la conexión al controlador. Sin embargo, esto pudo haber tenido otras connotaciones: sensor #1, habitación #1, dormitorio #1, cama#1, entre otros.

5.3. Módulo de captura de datos

Como se expresó previamente, se utilizó *Python* para la elaboración de los módulos correspondientes para las publicaciones, las suscripciones, el procesamiento y el análisis de datos. También, se recurrió al uso de algunas librerías para la invocación de las sentencias particulares de *MQTT* y *AMQP*.

Por ejemplo, la librería *kombu* fue empleada para lograr la conexión cliente-*AMQP*, aunque debe hacerse la aclaración que existe una cantidad considerable de otras herramientas tanto para *Python* como para otras plataformas (*Java*, *Erlang*, *C*, *Go*, entre otros).

Para el caso particular de *MQTT*, para la suscripción no se utilizaron librerías nativas de *Python* (e.g. *Paho-mqtt*), sino que estas se realizaron a través de llamadas al sistema operativo (similar a *system()* del lenguaje *C*) lográndose una especie de automatización del modelo *pub/sub* en consola y redireccionamiento a archivos de texto – en este caso particular con la extensión *.data* -. La sintaxis de la sentencia concreta empleada para la suscripción utilizando *Mosquitto*, fue:

```
mosquitto_sub -h [broker-host] -t [topic] [-F [options]] >> [output-file]
```

Ejemplo:

```
mosquitto_sub -h 192.168.0.17 -t "CAFE/Temp" -F @Y-@m-@d, @H: @M: @S: @N, %t, %p >>MQTT-20200413-CAFET-01.data
```

El significado de los *switches* (opción -F) puede consultarse en los manuales de referencia respectivos.

Tabla V. **Procesamiento de datos del módulo de suscripción a MQTT – publicaciones sensor cafet cada 15 segs**

FECHA_S	HORA_S	SENSOR	VALOR	FECHA_P	HORA_P	LECTURA	DIF_PS	MEDIA	DESV
13/04/2020	20:04:28:184000	CAFE/Temp	26.20000076	13/04/2020	20:04:27:887814	1	0:00:00.296186	0:00:00.446379	0.150193
13/04/2020	20:04:43:947000	CAFE/Temp	26.10000038	13/04/2020	20:04:43:601022	2	0:00:00.345978	0:00:00.446379	0.100401
13/04/2020	20:04:59:480000	CAFE/Temp	26.10000038	13/04/2020	20:04:59:300014	3	0:00:00.179986	0:00:00.446379	0.266393
13/04/2020	20:05:15:070000	CAFE/Temp	26.10000038	13/04/2020	20:05:14:894419	4	0:00:00.175581	0:00:00.446379	0.270798
13/04/2020	20:05:30:681000	CAFE/Temp	26.10000038	13/04/2020	20:05:30:481994	5	0:00:00.199006	0:00:00.446379	0.247373
13/04/2020	20:05:46:307000	CAFE/Temp	26.10000038	13/04/2020	20:05:46:098752	6	0:00:00.208248	0:00:00.446379	0.238131
13/04/2020	20:06:01:949000	CAFE/Temp	26.10000038	13/04/2020	20:06:01:720518	7	0:00:00.228482	0:00:00.446379	0.217897
13/04/2020	20:06:17:534000	CAFE/Temp	26.20000076	13/04/2020	20:06:17:366880	8	0:00:00.167120	0:00:00.446379	0.279259
13/04/2020	20:06:33:120000	CAFE/Temp	26.20000076	13/04/2020	20:06:32:952356	9	0:00:00.167644	0:00:00.446379	0.278735
13/04/2020	20:06:48:706000	CAFE/Temp	26.20000076	13/04/2020	20:06:48:539005	10	0:00:00.166995	0:00:00.446379	0.279384

Nota. El sufijo colocado _P y _S hace referencia a los eventos Publicación y Suscripción respectivamente.

Fuente: elaboración propia utilizando *MS-Excel* con los datos del archivo .data (datos provenientes del sensor).

Descripción de los campos (columnas):

- FECHA_S: Fecha de suscripción o consumo del mensaje.
- HORA_S: Hora de suscripción o consumo del mensaje.

- **SENSOR:** Indicador del color del cable y tipo de sensor conectado al *MCU*.
- **VALOR:** Contenido del mensaje recibido del sensor particular.
- **FECHA_P:** Fecha en que fue publicado el mensaje.
- **HORA_P:** Hora en que fue publicado el mensaje.
- **LECTURA:** Número de lectura.
- **DIF_PS:** Diferencia entre los tiempos de publicación y suscripción (en segundos y millonésimas de segundo).
- **MEDIA:** Valor promedio de la serie de DIF_PS.
- **DESV:** Desviaciones de cada valor DIF_PS respecto de la media.

Los *scripts* se elaboraron para trabajar con un formato de fecha/hora *ISO 8601*, sin embargo, como las aplicaciones lo permitían se consideraron hasta millonésimas de segundo – como se observa en la tabla V (columnas HORA_S y HORA_P).

El formato se adoptó de esta manera debido a la gran eficiencia en los tiempos de respuesta de ambos protocolos en la transferencia de datos y buscando la mayor exactitud posible; datos y tablas similares se obtuvieron para las pruebas con *AMQP*.

5.4. Tabulación y análisis de información

De acuerdo con las variables especificadas en la sección 5.1.4, se recurrió a hojas de verificación o control en forma de tablas para el análisis del comportamiento particular de la temperatura (T) y la humedad relativa (H). Esta tabulación brindará información en cuanto a: 1) la calidad y funcionamiento de los sensores; y 2) la veracidad o confiabilidad de los datos obtenidos.

Debido a las restricciones o condiciones de frontera establecidas para el prototipo, cuando se obtuvo datos “None” o absurdos, estos aparentes errores fueron considerados como anomalías, las cuales se analizaron empleando la herramienta *5W/1H* para la determinación de las causas, lo que permitió la afinación de modelo, así como la especificación de los lapsos o *thresholds* de las lecturas o mediciones de los sensores.

En las tablas VI y VII se presentan cinco registros de las pruebas realizadas para ambos *brokers*, los datos mostrados fueron editados para mostrar solamente el número de lectura, el dato emitido por el sensor y el tiempo de respuesta (diferencia entre los tiempos de publicación y suscripción).

Tabla VI. **Resultados AMQP, prueba 09 – Sensor #2 (Azul) / Temperatura**

Lectura	Dato (°C)	Tiempo Respuesta (seg.)
1	26.6000004	0.308717
2	26.6000004	0.283100
3	26.6000004	0.293956
4	26.6000004	0.354931
5	26.6000004	0.397917

Fuente: elaboración propia utilizando *MS-Excel*.

Tabla VII. **Resultados MQTT, prueba 09 – Sensor #1 (Café) / Humedad**

Lectura	Dato (°C)	Tiempo Respuesta (seg.)
1	49.500000	0.510618
2	49.400002	0.482280
3	49.700001	0.529696
4	49.700001	0.509288
5	49.799999	0.563088

Fuente: elaboración propia utilizando *MS-Excel*.

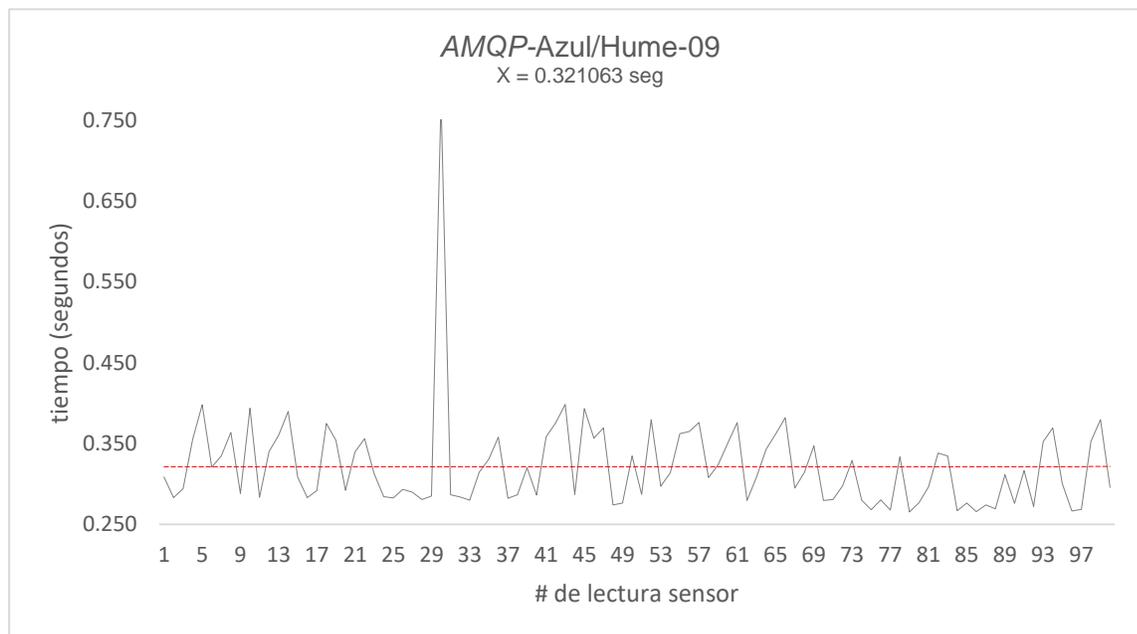
Luego de la recopilación de las pruebas y la tabulación de datos se procedió a realizar los cálculos de rango, valores máximos/mínimos, media y desviación estándar. Estos resultados fueron almacenados en diversas hojas electrónicas en la estación o computadora empleada en la capa de *application layer*.

5.5. Evaluación estadística de resultados

A pesar que se obtuvo varios indicadores, acá se hace énfasis en los datos de la media y desviación estándar, los cuales fueron los más representativos para evaluar la latencia de cada *set* de lecturas.

Respecto a los puntos que pudieran especularse como ‘anormales’ o *outliers* no se suavizaron ni se anularon, sino que fueron considerados como comportamientos aleatorios dentro de los márgenes de tolerancia aceptables debido al entorno del prototipo (saturación del procesador, variación en la calidad de la señal inalámbrica, entre otros).

Figura 7. ***Time-plot serie 09 de lecturas de humedad utilizando AMQP***



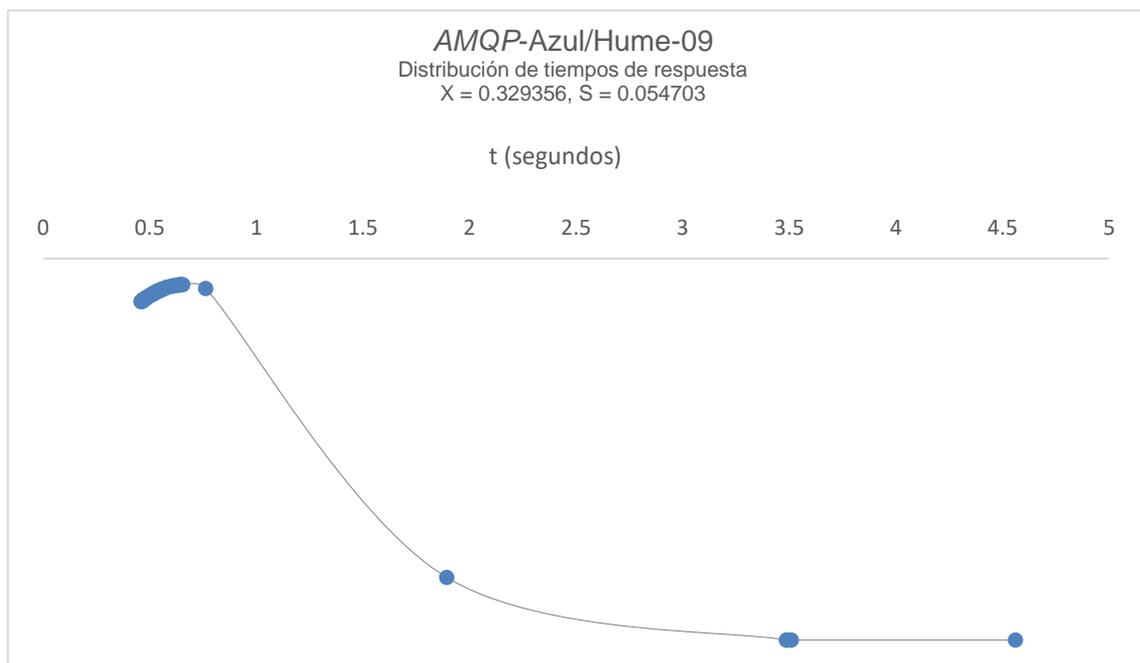
Nota: Para la realización del gráfico se utilizaron los datos de humedad relativa provenientes del sensor azul utilizado en el prototipo.

Fuente: Elaboración propia utilizando *MS-Excel*.

Por ejemplo, el pico representado por el punto tiempo (eje Y) igual 0.775370 correspondiente a la lectura número 30 de la serie de tiempo mostrada en la figura 7, se calificó como aceptable y, por tanto, este fue considerado en los cálculos estadísticos.

De acuerdo con lo anterior y en forma similar se obtiene la distribución proyectada en la figura 8.

Figura 8. **Distribución de tiempos de respuesta de la serie 09 de lecturas de humedad relativa utilizando AMQP**



Fuente: elaboración propia utilizando *MS-Excel*.

Durante todas las pruebas realizadas fue notoria la consistencia de los datos alrededor de la media, lo cual se refleja en la gráfica de distribución anterior, es decir, el fenómeno se comporta en forma normal bajo las consideraciones del prototipo adoptado respecto a los tiempos de respuesta.

Las tablas VIII y IX abstraen los datos de las 2,000 pruebas realizadas para *MQTT* como *AMQP*. Estas muestran el número de prueba, tópico utilizado para la publicación/suscripción, la media y la desviación estándar (*DS*) de cada *set*.

Tabla VIII. **Resultados de pruebas con *MQTT*, en sensores AZUL y CAFÉ**

N o	Tópico	Media	<i>DS</i>
1	AZUL/Hume05	0.312395	0.291662
2	AZUL/Hume06	1.035769	2.051305
3	AZUL/Hume07	0.661737	1.258121
4	AZUL/Hume08	0.697324	0.471430
5	AZUL/Hume09	0.616647	0.508357
6	AZUL/Temp05	0.290872	0.161302
7	AZUL/Temp06	1.326906	2.378697
8	AZUL/Temp07	0.817438	1.720899
9	AZUL/Temp08	0.650303	0.314378
10	AZUL/Temp09	0.621290	0.896561
11	CAFÉ/Hume05	0.426586	0.984655
12	CAFÉ/Hume06	0.801191	1.822161
13	CAFÉ/Hume07	0.564261	1.448081
14	CAFÉ/Hume08	0.646948	0.176039
15	CAFÉ/Hume09	0.601827	0.512987
16	CAFÉ/Temp05	0.357304	0.366168
17	CAFÉ/Temp06	1.422231	2.536484
18	CAFÉ/Temp07	0.533241	1.359471
19	CAFÉ/Temp08	0.622375	0.193281
20	CAFÉ/Temp09	0.666802	0.658747

Fuente: elaboración propia utilizando *MS-Excel*.

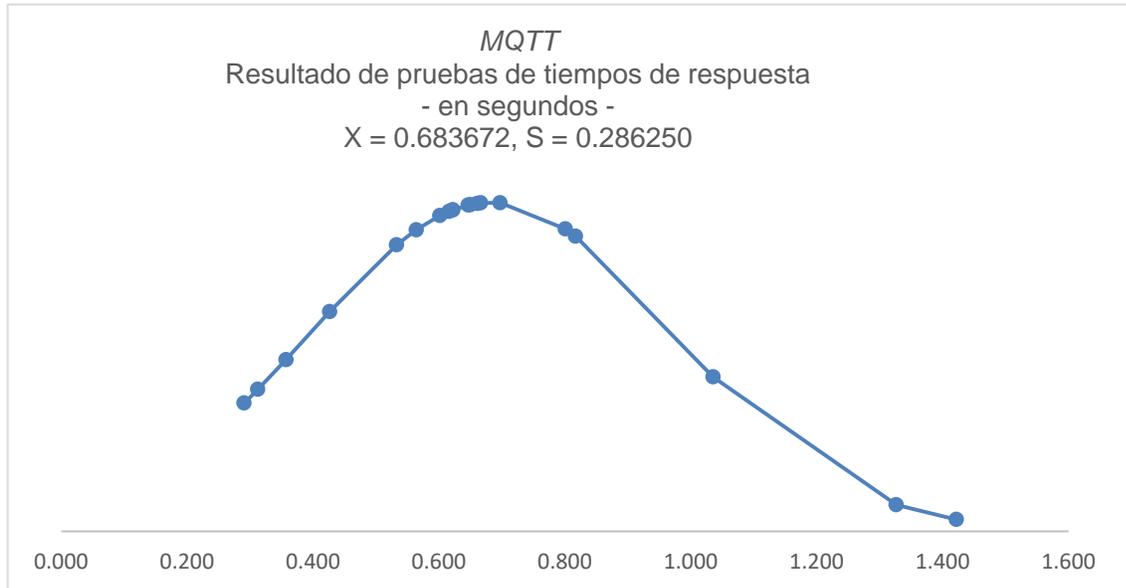
Tabla IX. **Resultados de pruebas con *AMQP*, en sensores AZUL y CAFÉ**

No	Tópico	Media	DS
1	AZUL/Hume05	0.412290	0.045059
2	AZUL/Hume06	0.399658	0.064658
3	AZUL/Hume07	0.349287	0.025850
4	AZUL/Hume08	0.321143	0.072607
5	AZUL/Hume09	0.329356	0.054703
6	AZUL/Temp05	0.410459	0.029735
7	AZUL/Temp06	0.393044	0.043252
8	AZUL/Temp07	0.356196	0.059021
9	AZUL/Temp08	0.312130	0.019784
10	AZUL/Temp09	0.321063	0.060734
11	CAFÉ/Hume05	0.416167	0.044465
12	CAFÉ/Hume06	0.393696	0.060451
13	CAFÉ/Hume07	0.368265	0.089062
14	CAFÉ/HUme08	0.315165	0.038882
15	CAFÉ/Hume09	0.328100	0.065097
16	CAFÉ/Temp05	0.411308	0.043623
17	CAFÉ/Temp06	0.401951	0.076466
18	CAFÉ/Temp07	0.346794	0.016123
19	CAFÉ/Temp08	0.315023	0.040103
20	CAFÉ/Temp09	0.332352	0.060155

Fuente: elaboración propia utilizando *MS-Excel*.

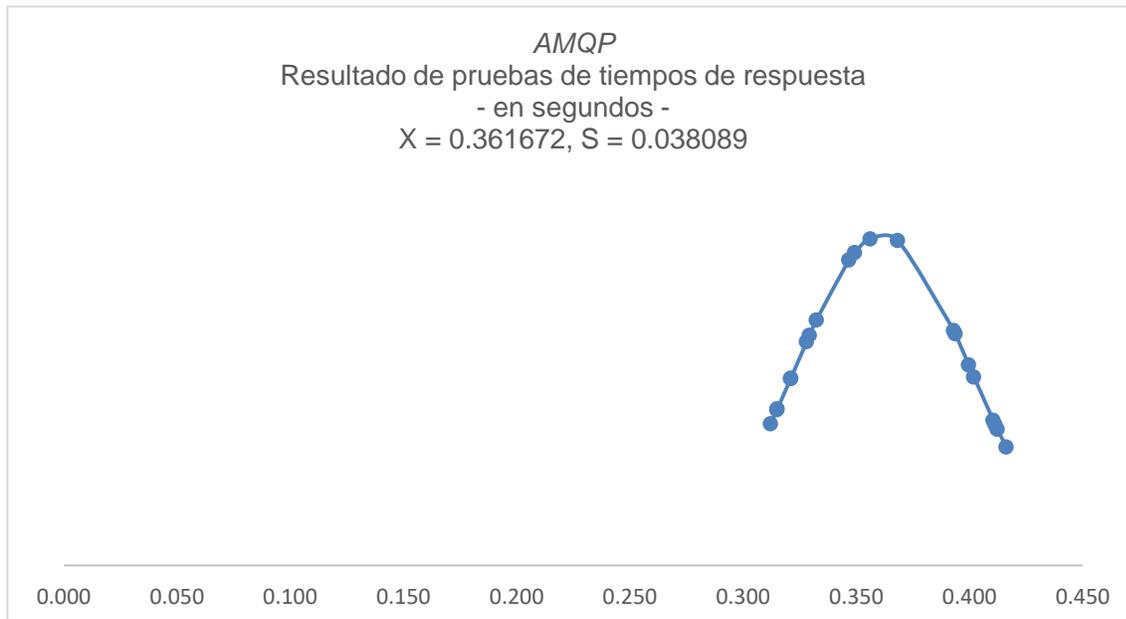
Posteriormente, se compararon los promedios (medias) de los resultados mostrados en las tablas anteriores para obtener la eficiencia en los tiempos de respuesta de *MQTT* y *AMQP*.

Figura 9. **Distribución de tiempos de respuesta de MQTT**



Fuente: elaboración propia, diagrama realizado utilizando *MS-Excel*.

Figura 10. **Distribución de tiempos de respuesta de AMQP**



Fuente: elaboración propia, diagrama realizado utilizando *MS-Excel*.

5.6. Comparación de desempeño/eficiencia de *MQTT* y *AMQP*

Como se estableció previamente, la evaluación del desempeño de los dos algoritmos seleccionados se basó en las variables estipuladas en la sección 5.1.4, parametrización de pruebas. De acá que los resultados de los exámenes estuvieran basados en estimaciones cuantitativas (t, I, Q, S y f) y cualitativas (F y C).

La evaluación de las variables para *MQTT* y *AMQP* es la siguiente:

5.6.1. Tiempos de respuesta (t)

El autor ha de admitir que los resultados en los tiempos de respuesta fueron una total sorpresa debido a que a priori flotaba en el aire una hipótesis de que *MQTT* debiera ser más rápido que *AMQP* debido a su bajo *footprint*; sin embargo, luego de las pruebas realizadas se tuvo que el comportamiento de las medias de los 20 *sets* de 100 lecturas cada uno arrojó que:

- *MQTT* obtuvo un valor promedio representativo de respuestas alrededor de 684 milésimas de segundo, mientras que su contra parte *AMQP* brindó tiempos más favorables con un promedio de 362 milésimas de segundo.
- *AMQP* resultó más consistente en sus respuestas al poseer una desviación típica de 0.038089 s., mientras que *MQTT* brindó una de 0.286250 s., comportamiento que se mantendría el 68 % de las veces bajo la premisa de un comportamiento normal del fenómeno.

Esto es fácilmente notable yuxtaponiendo las figuras 9 y 10 sobre el eje de abscisas y observando la apertura de ambas curvas; sin embargo, para proporcionar una visualización más adecuada de la dispersión de las medias, se provee la imagen de la figura 11.

Figura 11. **Diagrama de contraste de la dispersión de las pruebas MQTT / AMQP (en segundos)**



Nota: la primera columna representa los datos de MQTT y la segunda los de AMQP.

Fuente: elaboración propia utilizando MS-Excel.

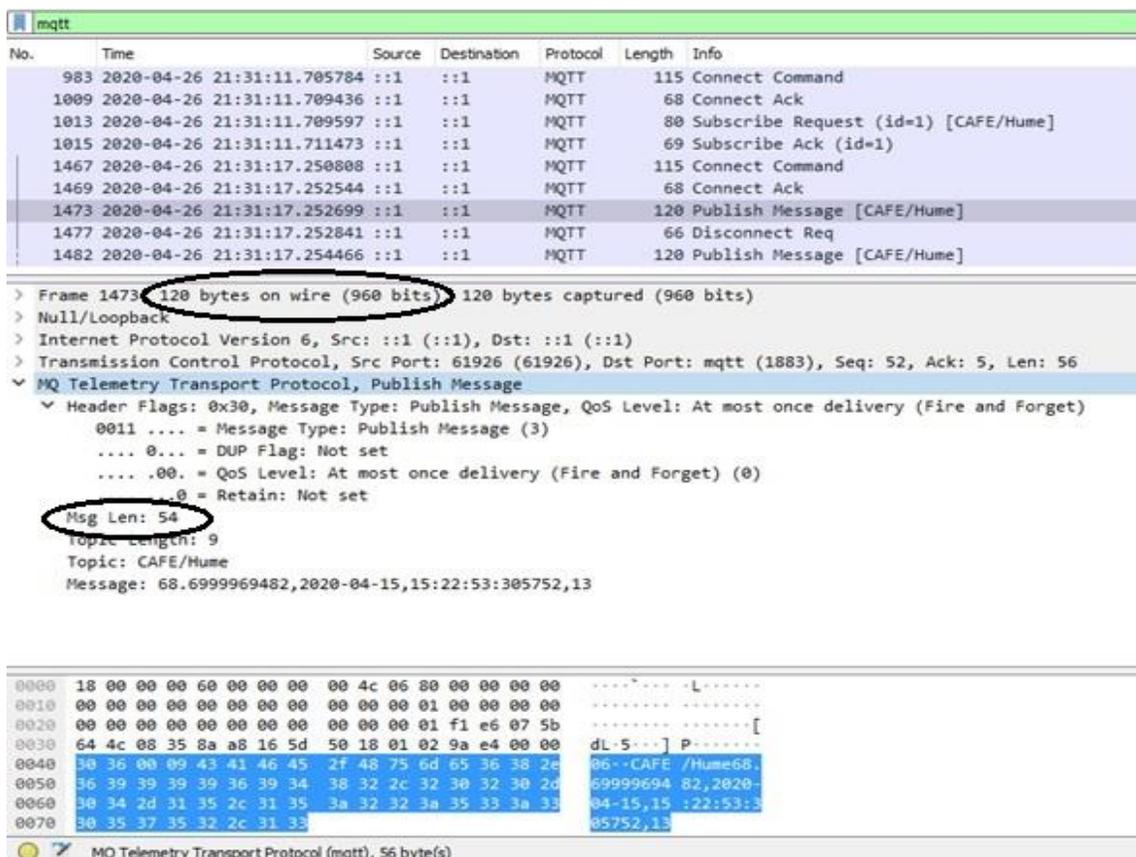
5.6.2. Integridad, manejo de *payload* y contenido (I)

En este rubro, ambos *brokers* brindaron una consistencia en tránsito productor/consumidor del 100 % en todas las lecturas de temperatura y humedad relativa enviada por los sensores, considerando los 8 dígitos significativos de dichos valores.

También, hay que apuntalar que las pruebas fueron realizadas sobre *TCP* lo cual benefició en forma tajante los resultados de integridad debido a los mecanismos anticorrupción del *checksum* que este posee.

Por otra parte, respecto al manejo del *payload/overhead*, los resultados observados utilizando el analizador de paquetes (*Wireshark*) muestran que *Mosquitto* efectivamente administra mejor el ancho de banda debido al bajo *footprint* o espacio empleado para la mensajería.

Figura 12. **Análisis de *payload/overhead* Mosquitto-MQTT**



Nota: Imagen producida por *Wireshark*.

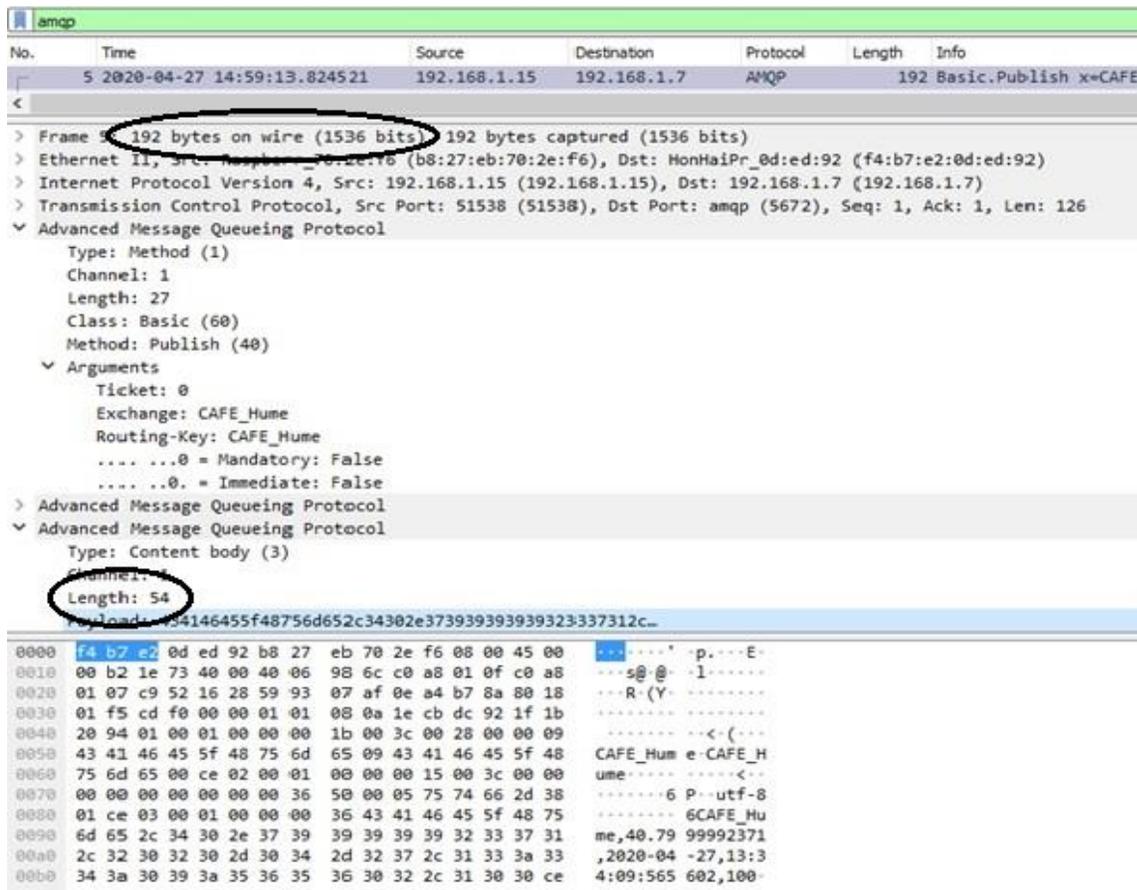
Fuente: elaboración propia, captura de imagen realizada con *ScreenCapturer*.

Las figuras 12 y 13 muestran el tratamiento del mensaje siguiente:

“CAFE/Hume,68.6999969482,2020-04-15,15:22:53:305752,13”

A través de las capas de aplicación, transporte, *internet* y acceso al medio para ambos protocolos. Dicho *string* tiene una longitud de 54 *bytes* (432 *bits*) y es similar al empleado en cada una de las pruebas realizadas.

Figura 13. **Análisis de *payload/overhead* de *RabbitMQ-AMQP***



Nota: Imagen producida por *Wireshark*.

Fuente: elaboración propia, captura de imagen realizada con *ScreenCapturer*.

Las imágenes indicadas muestran el tamaño de los *frames* (*bytes on wire*). Los datos obtenidos se plasman en la tabla X. En esta se puede apreciar el *over-head* agregado por cada protocolo previo al envío del mensaje:

Tabla X. **Relación entre el tamaño del mensaje y sus correspondientes *frames***

	<i>MQTT</i> (bytes)	<i>AMQP</i> (bytes)
Tamaño en capa de aplicación	54	54
Tamaño en capa de acceso al medio	120	192

Fuente: elaboración propia utilizando *MS-Excel*.

Finalmente, respecto al *content-type*, *Mosquitto* posee la filosofía con la que fue creado: ‘ser lo más ligero posible’ y en sus orígenes no se contemplaron mecanismos de identificación de qué es lo que va dentro del mensaje, el contenido se muestra, a la fecha en que fueron realizadas las pruebas, como *byte arrays* solamente.

Por otra parte, en la versión de *Mosquitto* evaluada, el tamaño puede acordarse previamente entre productores y consumidores, sin embargo, el *default* es utilizar el parámetro 0 que indica libertad en este aspecto, pero sin superar el máximo estipulado de 268, 435, 455 *bytes*.

Sin embargo, los implementadores pueden recurrir al ingenio tanto para la fragmentación del contenido como la descripción de este a través de la generación de alguna nomenclatura dentro del mensaje o el tópico, para el ensamblado y el posterior *parsing* en la recepción.

Por su parte, *RabbitMQ* posee un recomendado de 128 *MB*, aunque el tamaño teórico soportado es de 2 *GB*, pero estas dimensiones pueden producir problemas a nivel de manejo de memoria y fallas de aplicación.

Dependiendo de la implementación de *AMQP* que se utilice (e.g. 0-9, 0-9-1, 1.0, entre otros), *RabbitMQ* posee más recursos de identificación del *content-type*, por ejemplo, a través del uso de *headers* o alguna clase como *MessageProperties Class*, aunque estos siempre están sujetos a interpretación abierta. Sin embargo, podrían facilitar la tipificación del mensaje.

En resumen, en ambos protocolos los publicadores pueden enviar mensajes de cualquier índole, pero estos están sujetos a la interpretación y ensamblaje de los consumidores, lo cual puede resultar complejo, por ejemplo, al tratar de despachar diferentes tipos de archivo en segmentos o fragmentos; sin embargo, y a pesar del tamaño considerable del mensaje manejado por ambos protocolos, no es buena práctica ni recomendable enviar archivos en lugar de mensajes, aunque esto es factible.

5.6.3. Factores de aseguramiento de entrega de mensajes o calidad en el servicio (Q)

A la fecha, *Mosquitto-MQTT* posee 3 niveles de calidad de servicio:

- QoS 0 – enviar si mucho una vez, coloquialmente ‘disparar y olvidar’. Este nivel es el más básico y pretende que el emisor del mensaje confíe en que este siempre llegará al consumidor. Por no emplear paquetes de confirmación de lectura (*ACK*), QoS 0 es que es el método más rápido.

Este nivel – *default* - fue el adoptado en todas las pruebas realizadas debido a que previamente en las condiciones de frontera del prototipo, se estableció la condición a priori de confianza de mejor esfuerzo a través de la capa inferior de transporte con *TCP*.

- QoS 1 – Entregar por lo menos una vez. Este nivel persigue garantizar que el receptor del mensaje por lo menos lo obtenga una vez. Para lograrlo el sistema almacena el mensaje hasta recibir la confirmación a través de un paquete *PUBACK*.

Lo cual significa que por lo menos por cada publicación debe fluir en el canal dos señales: una de envío y otra de recepción.

A pesar de la ventaja de confirmación expuesta, puede ocurrir que cuando existan varios suscriptores al mismo tópico: estos pueden recibir mensajes duplicados, a pesar de la existencia de un *tag DUP (duplicated message)*.

- QoS 2 – Entregar exactamente una vez. Este nivel garantiza que cada mensaje es recibido por los consumidores solamente una vez. Esta es la razón por la que este nivel se constituye como el más seguro, aunque es el más lento debido a que para realizar un QoS tipo 2, se requiere un *handshake* de cuatro pasos entre el publicador y el consumidor.

Por su parte, *RabbitMQ-AMQP* ofrece varios métodos de *acknowledgement*: *basic.ack*, *basic.nack* y *basic.reject*. Los que se relacionan con un intrincado *set* de instrucciones, dependiendo del método de obtención del mensaje (e.g. *basic.get*, *basic.consume*, *basic.deliver*, entre otros).

Adicionalmente, *AMQP* por especificación posee encabezados con campos *reply_to* que pueden ser utilizados para enviar mensajes de respuesta, el único inconveniente es que debe especificarse una cola para estos *replies*.

Particularmente, *RabbitMQ* ha venido mejorando estos mecanismos de entrega *post-mortem* o relacionados a desperfectos constituidos por situaciones donde falla el consumidor, el *link* o el *broker* pero esto requiere de almacenamiento en disco a través de entrada a registros de la base de datos de *Erlang*, *Mnesia* (Videla y Williams, 2012).

Dichos mecanismos requieren de un estudio “*what-if*” particular y como fue indicado en las condiciones de frontera del prototipo (sección 5.1.1), estos no fueron considerados para las pruebas. Por tanto, si se desea más información al respecto se sugiere consultar las especificaciones y los manuales correspondientes de los productos evaluados.

5.6.4. Manejo de mecanismos de seguridad del protocolo (S)

El presente documento reconoce que el tema de seguridad es un tema muy importante para *IoT* debido a que por lo general la transmisión de mensajes atravesará la jungla del internet. Sin embargo, este apartado abarca este asunto en forma somera, debido al constante cambio por mejoras u obsolescencia de los protocolos de seguridad e infraestructuras.

Se incluye exclusivamente para motivos de evaluación de eficiencia/desempeño considerando el contexto temporal en que el trabajo se realizó (*circa* 2020) y principalmente como material de apoyo al lector para efectos de elección.

Por esta razón los siguientes párrafos incluyen las versiones y la fecha del *release* de ambos protocolos para referencia o futuras comparaciones de *MQTT* y *AMQP*.

Habiendo enmarcado el tema, se tiene que ambos protocolos poseen mecanismos *AAA* (*Authentication, Authorization, Accountability*). Por ejemplo, a nivel de autenticación, los dos tienen incorporado el uso clásico de usuario/*password*, utilizado comúnmente a la hora de conexión al *broker*.

Sin embargo, *RabbitMQ* 3.8.3 (*released* 2020-03-10) posee más herramientas de autenticación en forma natural, por ejemplo, validación por directorio activo (*LDAP plug-in*) y mecanismos *SASL*, comparado con su contraparte rival de evaluación *Mosquitto* 1.6.9 (*released* 2020-02-20).

De acuerdo con el *changelog* y documentación relacionada, *Mosquitto* requiere de la implementación de productos generados por iniciativas abiertas o aisladas, pero con poca colaboración y/o proyectadas para especificaciones posteriores del producto, esto particularmente considerando la referencia al inciso 5.4.1 (*Authentication of Clients by the Server*) de la norma *ISO/IEC 20922:2016*, respecto a uso de sistemas de autenticación externa.

Respecto a la autorización, *RabbitMQ* 3.8.3 distingue diversas acciones sobre sus recursos, por ejemplo, para los *exchanges* y las colas, los permisos para la creación o declaración, configuración, lectura y escritura, entre otros, los cuales pueden ser concedidos a los diferentes usuarios del *broker* y estas autorizaciones las debe realizar un ente administrador.

En el caso de *Mosquitto* 1.6.9, existen mecanismos de autorización, pero no son nativos y, por tanto, no son parte de la especificación.

En conclusión, en el caso de auditoría o monitoreo del sistema, *Mosquitto* 1.6.9 posee solamente un nivel básico de ‘verbosidad’ muy elemental, en tanto que *RabbitMQ* 3.8.3, al ser instalado viene preparado para levantar un servicio *web* que facilita no solamente el monitoreo o auditoría, sino también la creación de usuarios, *vhosts*, *exchanges*, colas, entre otros, lo cual provee un estupendo nivel visual de lo que está sucediendo tras bambalinas, aunque también se puede recurrir al modo carácter a través de la consola *rabbitmqctl*.

Finalmente, uno de los aspectos más importantes en el caso de la integridad o la seguridad en tránsito de los mensajes lo constituye que ambos protocolos pueden hacer uso de la infraestructura de llave pública, *PKI*, y por tanto, pueden utilizarla en los procesos de publicación/suscripción debido a que estos básicamente son interacciones de arquitectura cliente/servidor.

Esto lo logran recurriendo a las herramientas de protección o cifrado en la capa de transporte y dependen de está para su uso, por ejemplo, pueden utilizar *TLS*, a través del beneficio provisto por herramientas de generación de certificados y/o el par de llaves requeridas (e.g. *OpenSSL*).

5.6.5. Facilidad de implementación (F)

De acuerdo con la documentación, la configuración y el afinamiento realizado para el experimento realizado con el prototipo, se tiene la apreciación cualitativa que *Mosquitto (MQTT)* es más sencillo de instalar e implementar; mientras que *RabbitMQ (AMQP)* requiere no solamente el manejo de conceptos adicionales sino también posee una curva de aprendizaje e implementación más escarpada.

5.6.6. Complejidad de uso (C)

Considerando la disponibilidad de librerías, de acuerdo con el lenguaje de desarrollo seleccionado, y la adaptabilidad a los componentes de procesamiento del prototipo, se tiene que ambos protocolos poseen gran cantidad de herramientas para diferentes entornos de desarrollo. Por ejemplo, para el caso particular de *Python 3.7.3* fue sencillo encontrar *toolboxes* apropiados e intuitivos para ambas especificaciones.

Sin embargo, debido al auge que ha cobrado *AMQP* en los últimos años, este posee más variedad de *plug-ins* así como la disponibilidad de más librerías o módulos de *software* para conexión, publicación, monitoreo y suscripción.

Por ejemplo, *RabbitMQ* posee una consola *web* que se instala junto con el producto, donde se puede monitorear el comportamiento de los diferentes

componentes definidos en la estructura de la aplicación – i.e. *Exchanges*, *Queues*, *Routing Keys*, entre otros – (Videla y Williams, 2012). Además, dicha consola provee detalles de tamaño, frecuencia de emisión y lectura, estado de las colas en tiempo real, entre otras grandes virtudes.

Por otra parte, *Mosquitto* ofrece un nivel de ‘verbosidad’ elemental en su *output* de *logs* a consola, pero a pesar de que este muestra información concerniente a conexiones, publicaciones/suscripciones y detalles de los mensajes, este es muy básico y probablemente se queda corto al momento de realizar un *troubleshooting* profundo de lo que podría estar sucediendo en *background*.

Finalmente, la gran cantidad de características que posee *RabbitMQ* hacen que este sea más complejo de implementar, pero al mismo tiempo le brinda a una gran versatilidad a la hora del manejo y monitoreo de mensajes. Además, posee diversas herramientas que van más allá del modelo de publicación/suscripción.

5.6.7. Resumen de la comparación

La tabla XI muestra la evaluación para *MQTT* y *AMQP*, derivada de los resultados y la experiencia adquirida durante el experimento. La ponderación está basada en la escala de 0 a 5, donde 5 representa una mejor valoración y 0 lo contrario u opuesto.

Lo anterior se aplicó tanto a las variables cuantitativas como a las cualitativas (i.e. F y C), en este último caso, por ejemplo, para facilidad de implementación 5 significa que fue un proceso sencillo y rápido, mientras que 0 significa la experiencia contraria.

Tabla XI. **Comparación entre *MQTT* y *AMQP*, ponderación sobre 5 puntos por ítem**

Factor	<i>MQTT</i>	<i>AMQP</i>	Descripción
F	5.0	1.0	Facilidad de implementación
C	4.5	3.5	Complejidad del uso (menos complejo más puntaje)
I	4.5	5.0	Integridad, manejo del <i>payload/overhead</i> y <i>content-type</i>
Q	4.0	5.0	Mecanismos de priorización o garantía de entrega de mensajes
S	4.0	5.0	Seguridad
t	4.0	5.0	Tiempo de respuesta publicación/suscripción
f	1.0	5.0	Funcionalidades adicionales del producto evaluado
TOTALES	27.0	29.5	

Fuente: elaboración propia utilizando *MS-Excel*.

6. DISCUSIÓN DE RESULTADOS

El presente capítulo muestra la influencia de los elementos del prototipo y la metodología empleada en los resultados presentados. A continuación, se esbozan las razones del porqué se obtuvieron los datos y las implicaciones obtenidas en el experimento.

6.1. Impacto y evaluación del entorno del prototipo en los resultados

El prototipo fue montado prácticamente en un entorno controlado constituido por una red local conectada a través wifi. Esto hace que los datos *pub/sub* muestren una alta integridad, consistencia y disponibilidad debido a que los mensajes no atravesaron *routers* o equipos *ISPs* como sucedería en un escenario *cloud* o de *VPN*.

Lo mismo aplica a las conexiones de los sensores: en el modelo se enlazaron al controlador a través del uso de *jumper-cables*, sin embargo, en *IoT* existen muchos equipos conectados a través de diversas tecnologías inalámbricas basadas en los estándares de capa de acceso al medio, *IEEE 802* (e.g. *Bluetooth*, *Zigbee*, *MiWi*, entre otros) o las recientemente populares de la *LoRa Alliance*, que de haberse utilizado hubiera influido definitivamente en: 1) la forma de realizar el estudio, 2) la consistencia e integridad de los datos y 3) los tiempos de respuesta o latencia.

Adicionalmente, de haberse implementado en otros entornos (i.e. *cloud* o *VPN*), hubiera hecho más complejo la ponderación de las variables facilidad de implementación (F) y complejidad de uso (C), debido a que se tendría que

haber afinado los parámetros de calidad de servicio, los mecanismos de garantía de entrega (Q) y la seguridad en tránsito (S), tópicos muy particulares de los canales de comunicación y de los equipos activos utilizados.

6.2. Evaluación del *hardware*

Este es uno de los aspectos más influyentes en los resultados, iniciando por la baja calidad de los sensores y por el microcontrolador seleccionado, el cual podría considerarse no apropiado, por las razones que más adelante se exponen.

- Calidad de los sensores: a pesar que los utilizados en el prototipo no son los más *ad-hoc* para una implementación de estándar médico, los indicados en la sección 5.1.2 se utilizaron debido al presupuesto disponible para el proyecto, sin embargo, se prefirió utilizar sensores físicos para no recurrir al uso de un *set* predefinido o la generación aleatoria de datos a través de una plataforma de emulación como *node-red*. Esto último, aunque hubiera servido para cumplir con los objetivos enumerados, no hubiera considerado una arquitectura de prototipo de “mejor esfuerzo” apegado a la realidad como lo fue propuesto en la figura 4 respecto a la interacción *sensor-broker*.

Las repercusiones del prototipo elegido fueron notorias en la cantidad de tiempo invertido, particularmente, en la realización de todas las pruebas, tanto las exitosas como las fallidas, porque estas permitieron afinar el modelo y proveyeron el establecimiento de un escenario más realista.

Finalmente, sin el uso del prototipo adoptado, no se hubiera podido experimentar con el desarrollo de los módulos de publicación/suscripción en *Python* ni el uso de las librerías disponibles para *MQTT* y *AMQP*.

- Microcontrolador: respecto a la elección de *Raspberry-pi* como *MCU*, se considera que se pudo elegir otra plataforma más orientada a *IoT* en el sentido de mejor eficiencia energética, mejor relación costo/beneficio - más asequible -, un sistema embebido de codificación (microprogramación), entre otros, por ejemplo, se pudo optar por productos como *Arduino pro mini*, *ESP8236*, *Wemos ESP32*, *Beaglebone black*, *PIC-IoT* e inclusive con el más reciente *Raspberry-pi-pico*, entre otros.

Sin embargo, se optó por el *Raspberry-pi* indicado por la facilidad de uso y que este ya posee los elementos que se consideraban necesarios para el prototipo, a decir, facilidad para el autor de la gestión de sistema operativo y herramientas *SSH*, conexiones *GPIO*, módulo wifi integrado, facilidad de implementación del entorno de desarrollo de los *scripts* e instalación de librerías de conexión a los protocolos estudiados.

A pesar de lo previamente observado en los párrafos anteriores, se considera que para cumplir los objetivos planteados en el documento, el empleo de los sensores descritos y micro controlador seleccionado fue el adecuado para el proyecto y cumplieron con el propósito de evaluar la eficiencia/desempeño de *MQTT* y *AMQP* particularmente de las variables cuantitativas, es decir, medición de los tiempos de respuesta (*t*), manejo de integridad, *payload/overhead* y *content-type* (*l*), evaluación de los mecanismos de aseguramiento de entrega (*Q*) y los aspectos de seguridad (*S*) de ambos protocolos.

Por tanto, se considera que dichos resultados fueron obtenidos bajo las condiciones apropiadas en el entorno de pruebas adoptado y que los resultados obtenidos poseen una alta confiabilidad. Especialmente las relacionadas al aspecto de latencia y manejo de *payload/overhead*.

6.3. Evaluación del *software*

Definitivamente la elección del *software* jugó un papel importante en la obtención de resultados. Por ejemplo, la elección de otro microcontrolador, pudo haber provocado que las pruebas brindaran datos de respuesta más rápidos (por ejemplo, en entornos orientados a *Assembler* o C); sin embargo, para los motivos de comparación de rendimiento de *MQTT* y *AMQP*, la selección de *Python* y las pruebas realizadas con *paho-mqtt*, *pika* (*AMQP*), *kombu* (*AMQP*) y las de *output* a consola para *MQTT*, brindaron las pruebas adecuadas para la obtención de los resultados de confrontación.

Por tanto, respecto al cotejo de las variables cuantitativas expuestas en la tabla IV, se puede concluir que el *software* utilizado en las pruebas brinda resultados confiables y precisos.

Respecto a la evaluación del lenguaje de desarrollo, se puede exponer que personalmente se optó por *Python* 3.7.3 como herramienta de programación de los módulos y los *scripts*, aunque otros escritores posean diferente apreciación o inclinación hacia otros entornos de desarrollo (C, *Java*, *Erlang*, entre otros) bajo el argumento que son los lenguajes en los que se realizó el producto evaluado y por tanto “está homologado” o existe un grado mayor de “compatibilidad”.

6.4. Evaluación de la propuesta de estudio y el método

Se considera que la metodología planteada desde un inicio en el documento es la adecuada no solamente para modelar y estipular todas las condiciones y restricciones del prototipo, sino también para la obtención de los resultados logrados, particularmente de aquellas variables cuantitativas definidas para la medición de la eficiencia/desempeño tanto de *MQTT* como de *AMQP*.

Sin embargo, respecto la selección de *Mosquitto* y *RabbitMQ* para un estudio comparativo, al evaluar especialmente las variables cualitativas (F y C) relacionadas a la instalación, configuración y uso, pareciera que se estaba trabajando con elementos sin relación alguna. Es por tanto natural que exista entre los lectores opiniones divididas respecto a los algoritmos seleccionados para dicha comparación.

Especialmente porque *RabbitMQ* requiere no solamente del aprendizaje de muchos conceptos y términos (e.g. *Exchange*, *Binding*, *Routing*, entre otros); sino porque este posee una complejidad mayor, en el sentido del uso de abstracciones. Por su parte, *Mosquitto* no requiere de malabarismos técnicos elevados: este resultó muy fácil de instalar y usar.

También, se debe comentar que las siglas del protocolo *MQTT*, *Message Queuing Telemetry Transport*, generaran algún tipo de confusión o desconcierto debido al término *Queuing*, el cual puede evocar que este es un *broker* administrador de colas de mensajes, cuando en realidad no lo es el sentido estricto de funcionamiento interno, como lo es su antagonista de evaluación *AMQP*. Y es que *MQTT* posee algunas características de almacenamiento de estados, manejo de calidad en el servicio, sesión persistente, políticas de

retención, *Will-delay*, entre otros, que pareciera que desde ‘fuera de la caja’ existen colas *per se*.

Sin embargo, y dejando de lado la semántica del término *queuing*, lo que interesa acá es el manejo de mensajes de estos dos *brokers* y sus mecanismos de publicación/suscripción, y es, desde esta perspectiva previamente adoptada en los objetivos del documento, que los protocolos seleccionados cumplen lo estipulado.

Además, dentro del contexto de la solución propuesta, respecto al procesamiento de datos de sensores de temperatura y humedad relativa para el monitoreo y atención temprana de pacientes, *MQTT* y *AMQP* suplen y cumplen con la expectativa de un *message-broker* en este ámbito particular de *IoT*.

Finalmente, de todas las posibles opciones disponibles, *MQTT* y *AMQP* eran los únicos *message-brokers* que a la fecha de elaboración del presente documento poseían especificaciones avaladas dentro de una norma o estandarización *ISO*, lo cual le brindaba a la selección de dichos protocolos un alto nivel de certidumbre, peso y confianza.

6.5. Aplicabilidad al problema de escaras

Como corolario de lo expuesto en los párrafos anteriores y sellar los resultados del experimento, se puede concluir que *MQTT* y *AMQP* pueden ser utilizados como manejadores de mensajes de los datos provenientes de los sensores de temperatura y humedad relativa para el monitoreo efectivo de dichas condiciones para minimizar el apareamiento de escaras en un entorno local y por tanto su aplicabilidad es imprescindible a la solución planteada.

Ambos protocolos tienen grandes virtudes, algunas muy parecidas y otras muy particulares, por ejemplo, *MQTT* demostró una alta eficiencia en el manejo del ancho de banda, sin embargo, *AMQP* no solamente presentó estupendos tiempos de latencia sino también exhibió una gran versatilidad, desempeño y características adicionales.

CONCLUSIONES

1. De acuerdo con los resultados obtenidos, se establece que *MQTT* y *AMQP* son protocolos eficientes para el manejo de mensajes en escenarios *IoT* como el sistema de monitoreo planteado, y su aplicación particular dependerá mucho del requerimiento del sistema. Por ejemplo, si las condiciones predominantes son el manejo de latencia, el dinamismo en la administración de colas y roles, la versatilidad de intercambio productor/consumidor, el monitoreo del comportamiento del sistema, entre otros, prefíerese *AMQP*. Lo anterior se fundamenta no solamente en la evaluación general sino en lo sobresaliente que resultó *AMQP* al manejar los tiempos de respuesta con un promedio de 362 ms., y una desviación estándar de 38 ms., mientras que *MQTT* obtuvo 684 ms., y 286 ms., en dichos valores. Es decir, con una mejora de 322 ms., *AMQP* no solamente resultó mejor en el manejo de latencia sino también fue más consistente en sus respuestas.

Sin embargo, *MQTT* tuvo mejor desempeño en el uso del ancho de banda: considerando los resultados, los *bytes on wire* agregados al *payload* mostraron que *AMQP* utiliza 1.6 veces más en espacio *overhead* que *MQTT*. Esto podría tener altos beneficios si se consideraran millones de transacciones publicador/suscriptor, lo cual significaría un mejor rendimiento energético y costos de uso de infraestructura relativamente menores, especialmente en escenarios donde el ancho de banda es muypreciado o costoso (e.g. comunicaciones satelitales o móviles, conexiones *dial-up*, aplicaciones industriales máquina a máquina – *M2M*, entre otros).

2. A través del estudio documental y la encuesta realizada queda de manifiesto que la temperatura y la humedad relativa, tienen una incidencia marcada en el padecimiento de escaras, con un aproximado causal del 36 % entre ambos factores, ocupando así el tercer lugar en los precursores, solamente atrás de los causales de falta de movimiento, presión corporal y los factores endógenos o inherentes al paciente (incontinencia, edad, nutrición, enfermedades crónicas como la diabetes, entre otros).
3. La descripción y especificaciones de los sensores, controladores, *gateways* y servicios establecidos en el presente trabajo, satisfacen la propuesta de análisis y evaluación de los *brokers* en el prototipo y experimentación realizada, sin embargo, desde el punto de vista médico, particularmente los sensores y controladores no satisfacen los estándares de calidad requeridos para aplicaciones o implementaciones hospitalarias (i.e. alto rendimiento, durabilidad, confiabilidad, aval de certificación, entre otros). En este sentido, si se deseara una aplicación real se deben realizar los estudios de mercado correspondiente para evaluar los impactos económicos y financieros demandados por el proyecto.
4. El uso de mecanismos intermediarios como *brokers* de mensajería debiera ser un elemento *defacto* en el diseño de arquitecturas o sistemas *IoT* como el de monitoreo por sensores para atención temprana de pacientes, debido a que estos proveen eficiencia en el manejo de latencia, administración de la congestión, comunicación asíncrona y desacoplamiento entre componentes. Sin embargo, debido a su funcionamiento particular y orientación debe realizarse un estudio

minucioso respecto a cuál es el más apropiado entre todos los *brokers* disponibles, así como un análisis del entorno donde se desean utilizar.

5. La implementación del prototipo presentado permitió realizar un estudio comparativo de eficiencia de *MQTT* y *AMQP*, evidenciando que ambos poseen un estupendo manejo de tiempos de respuesta; sin embargo, *AMQP*, por su diseño y características adicionales, supera por mucho a *MQTT*; y por tanto es más *ad hoc* para establecer un sistema de monitoreo de las condiciones de temperatura y humedad precursoras de escaras. No obstante, dicha extensibilidad y versatilidad requiere de más consideraciones técnicas para su instalación, configuración y aprendizaje debido a que lo anterior implica la comprensión de más conceptos, abstracciones y funcionalidades que las mostradas por *MQTT*.

RECOMENDACIONES

1. Realizar pruebas en otras arquitecturas de microcontrolador (e.g. *Arduino ESP8236*, *Wemos ESP32*, *Beaglebone black*, *PIC-IoT*, entre otros). El autor considera que hay plataformas más *ad-hoc* en este aspecto, particularmente en la conectividad y ahorro de energía.
2. Experimentar con otras plataformas de código embebido y/o *firmware* para el desarrollo de los módulos, que puedan resultar más eficientes (i.e. *MicroPython*, *Assembler*, *C*, entre otras). Esto puede constituir una ventaja en los tiempos de reacción del sistema.
3. Experimentar en otras infraestructuras de comunicación (capa de acceso al medio) basadas en los estándares *IEEE 802* potencialmente útiles en *IoT*, como *wifi6*, *Bluetooth*, *Zigbee* y tecnologías de la *LoRa Alliance*, entre otros, y contrastar con los resultados del presente trabajo. Particularmente el autor recomienda el uso de la infraestructura apropiada de acuerdo al ámbito, envergadura o tamaño espacial (distancias de cobertura) del proyecto y los requerimientos de eficiencia energética, integridad de datos, consistencia y velocidad de respuesta.
4. Comparar la eficiencia/desempeño de los protocolos evaluados con otros manejadores de mensajes (e.g. *STOMP*, *MQTT-SN*, *CoAP*, entre otros) así como el desempeño de estos en otros entornos operativos más especializados (e.g. *RTOS*). Se prevé que en un futuro existan plataformas más especializadas y enfocadas a *IoT* que las presentes al año en que fue realizado el presente documento.

5. Experimentar con mezclas de *brokers* y extraer las virtudes particulares de cada uno, por ejemplo: *MQTT+AMQP*, *STOMP+AMQP*, *MQTT+CoAP*, entre muchas más, con lo que se puede aspirar a obtener lo mejor de los mundos y así lograr el nirvana en el problema de la eficiencia en la transmisión de mensajes en *IoT* u otros sistemas que necesiten las versatilidades de este tipo de mecanismos intermediarios.

La elección de la mezcla dependerá en gran medida del tipo de problema a resolver (características y restricciones) y se deberá estudiar a profundidad este *middleware* para verificar compatibilidades, formateo de mensajes, los mecanismos de aseguramiento de entrega, la integridad, la seguridad, entre otros más, particularmente si se tratara de entornos de computación distribuida.

REFERENCIAS

1. Bauer, K., Rock, K., Nazzal, M., Jones, O. y Qu, W. (2016). *Pressure Ulcers in the United States' Inpatient Population from 2008 to 2012: Results of a Retrospective Nationwide Study*. United States of America.
2. Bishop, R. H. (2002). *The Mechatronics Handbook*. Austin, Texas. United States of America: University of Texas at Austin, CRC Press.
3. Bormann, C. (2016). *CoAP, RFC 7252 Constrained Application Protocol*. Enfoque del protocolo CoAP. Recuperado de <http://coap.technology/>
4. Centro Nacional de Información de Ciencias Médicas, (2017). *Úlcera por presión*. Biblioteca Médica Nacional de Cuba. Bibliomed Vol. 24, No. 2. La Habana, Cuba · Febrero, 2017. Compilación de recursos en línea (links). Recuperado de: <http://files.sld.cu/bmn/files/2017/02/bibliomed-febrero-2017.pdf>
5. Evans, Dave (2012). *The Internet of Everything. How More Relevant and Valuable Connections Will Change the World*. Cisco Internet Business Solutions Group (IBSG).

6. Gerber, A. y Romeo, J. (2017). *Connecting all the things in the Internet of Things*. Publicación del centro de desarrollo de IBM. Guía para la selección de tecnologías de red para los retos de IoT. Actualizado en enero 2020. Recuperado de:
<https://www.ibm.com/developerworks/library/iot-lp101-connectivity-network-protocols/index.html>
7. Geyer, C. (2018). *ISO and IEC approve OASIS AMQP Advanced message Queuing Protocol*. Publicado por OASIS Press. Respecto al establecimiento de AMPQ como un estándar dentro de las normas ISO. Recuperado de: <https://www.oasis-open.org/news/pr/iso-and-iec-approve-oasis-amqp-advanced-message-queuing-protocol>
8. Hill, B. (2002). *CISCO – Manual de referencia*. Madrid, España: McGraw-Hill/Interamericana de España.
9. Holdowsky, J., Mahto, M., Raynor, M.E. y Cotteleer, M. (2015). *Inside de Internet of Things (IoT). A primer on the technologies building the IoT*. Texas, United States of America: DUPress, Deloitte University Press.
10. Holdsworth, A.J. (2003). *Selection of communication strategies for message brokers or publish/subscribe communications*. Anotaciones de background número [0003, 0004 y 0012] respecto a la funcionalidad de los message-brokers. Recuperado de:
<https://patents.google.com/patent/US20030135556A1/en>

11. Janakiram, M. (2016). *Get to Know MQTT: The Messaging Protocol for the Internet of Things*. Descripción del origen de MQTT y terminología relacionada. Recuperado de: <https://thenewstack.io/mqtt-protocol-iot/>
12. Kampuri, T. (2014). *Message brokers and RabbitMQ in action*. Thesis, Degree Programme in Media Engineering, Technology, Communication and Transport. Finlandia. JAMK University of Applied Sciences. Recuperado de: https://www.theseus.fi/bitstream/handle/10024/78263/Kamppuri_Tsuri.pdf?sequence=1
13. Linthicum, D. (2017). *Edge computing vs. Fog computing: Definitions and enterprise uses*. Respecto a definiciones y tendencias de uso de computación Edge en IoT. Recuperado de: <https://www.cisco.com/c/en/us/solutions/computing/what-is-edge-computing.html#~types-of-edge-computing>
14. Rayes, A., Salam, S. (2016). *Internet of Things: From Hype to Reality, The Road to Digitization*. Suiza: Springer, Switzerland.
15. Sánchez-Cruz, L.Y., Martínez-Villareal, A., Ashley, A., Lozano-Platonoff, A., Cárdenas-Sánchez, A. y Contreras-Ruiz, J. (2016). *Epidemiología de las úlceras cutáneas en Latinoamérica*. Colegio Ibero-Latino-Americano de Dermatología. Recuperado de: <https://www.medigraphic.com/pdfs/cutanea/mc-2016/mc163f.pdf>

16. Sinha, Professor G. (2017). *Introduction to Sensors, Actuators and Mechatronics*. Respecto a la relevancia de diferentes sensores/actuadores y sus aplicaciones. Recuperado de: https://www.researchgate.net/publication/321625410_Introduction_to_Sensors_Actuators_and_Mechatronics

17. Stipe, C., Eugen, M. y Zeljko, S. (2017). *State-Of-The-Art Of Messaging For Distributed Computing Systems*. Croacia: International Journal - VALLIS AUREA, Volume 3, Number 2.

18. (The) National Center for Biotechnology Information (2015). *Fabric-based Pressure Sensor Array for Decubitus Ulcer Monitoring*. Datos estadísticos de Estados Unidos respecto a personas afectadas por úlceras por presión y costos anuales derivados de las mismas. Recuperado de: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4606918/>

19. Videla, A. y Williams, J. (2012). *RabbitMQ in action, distributed messaging for everyone*. New York, United States of America: Manning Publications Co.