Artículos de Investigación

Protocolo de comunicación inteligente para una arquitectura enfocada en internet de las cosas como solución a desafíos de interoperabilidad

Intelligent communication protocol for an architecture focused on the internet of things as a solution to interoperability challenges

Ibarra Cuevas, Zazil Josefina; Martínez Pérez, Francisco

Zazil Josefina Ibarra Cuevas
a246224@alumnos.uaslp.mx
Universidad Autónoma de San Luis Potosí, México
Francisco Martínez Pérez
eduardo.perez@uaslp.mx
Universidad Autónoma de San Luis Potosí, México

Revista Ingenio
Universidad Francisco de Paula Santander, Colombia
ISSN: 2011-642X
ISSN-e: 2389-864X
Periodicidad: Anual
vol. 19, núm. 1, 2022
revistaingenio@ufpso.edu.co

Recepción: 20 Abril 2021
Aprobación: 04 Noviembre 2021

URL: http://portal.amelica.org/ameli/journal/814/8144200002/

DOI: https://doi.org/10.22463/2011642X.2783

Universidad Francisco de Paula Santander Ocaña

Resumen: El objetivo del Internet de las Cosas es contar con sensores inteligentes colaborando directamente sin participación humana, habilitando a los objetos físicos para ver,escuchar y pensar de forma que se les permita compartir información y coordinar decisiones.Actualmente existen protocolos de comunicación para Internet de las Cosas que cumplen con ciertos requisitos que éste demanda, sin embargo, pocos son los trabajos que permiten un procesamiento inteligente de la información, dejando este análisis a cargo de otros componentes de la arquitectura.En este trabajo se propone un protocolo de comunicación para una arquitectura enfocada en Internet de las cosas con base en la transferencia de estado representativo (REST, Representational State Transfer) combinado con la metodología publicar/suscribir con infraestructura de servicio de mensajería.El protocolo puede administrar una gran cantidad de dispositivos, cuenta con un formato de mensaje compacto para que requiera poca capacidad de procesado, utiliza el lenguaje de programación Java pensando en la interoperabilidad, usa sockets para hacer posible que haya un gran número de comunicaciones simultaneas e implementa agentes inteligentes para la construcción de mensajes significativos para la toma de decisiones.

Palabras clave: Agentes Inteligentes, Arquitectura de tres capas, Arquitectura Orientada a Servicios, Internet de las Cosas, Protocolo de Comunicación.

Abstract: The goal of the Internet of Things is to have smart sensors collaborating directly without human participation, enabling physical objects to see, hear and think in a way that allows them to share information and coordinate decisions.Currently, there are communication protocols for the Internet of Things that meet certain requirements that it demands, however, few are the jobs that allow intelligent information processing,leaving this analysis in charge of other components of the architecture.In this work,a communication protocol is proposed for an architecture focused on the Internet of Things based on Representational State Transfer (REST) combined with the publish/subscribe methodology with messaging service infrastructure.The protocol can manage

a large number of devices, has a compact message format so that it requires little processing capacity, uses the Java programming language with interoperability in mind, uses sockets to allow a large number of simultaneous communications and implements intelligent agents for the construction of meaningful messages for decision making.

Keywords: Intelligent Agents, Three-layer Architecture, Service Oriented Architecture, Internet of Things, Communication Protocol.

1. Introducción

En la actualidad existen definiciones de Internet de las Cosas (IoT,Internet of Things) como las presentadas en [1-2].Se puede ver a IoT como una infraestructura que conecta y habilita a una gran cantidad de dispositivos o "cosas" que se utilizan comúnmente para interactuar y coordinar decisiones reduciendo o incluso haciendo nula la intervención humana en tareas comunes de la vida diaria [1].Al hablar de IoT no se trata de un término aislado, integra múltiples tecnologías entre las cuales se encuentran; computo en la nube, análisis de big data, sistemas embebidos, protocolos de seguridad, servicios web y redes de sensores y/o dispositivos inalámbricos [2].

En [3-4-5-6] se resalta que uno de los principales retos y objetivos de IoT es lograr la interconexión de múltiples redes y dispositivos para que la recopilación de datos, el intercambio de recursos, análisis y gestión puedan llevarse a través de un medio heterogéneo. Se han propuesto múltiples soluciones para este reto, como las arquitecturas que se presentan en [7-8].El modelo básico de tres capas, la Arquitectura orientada a Servicios(SoA, Service Oriented Architecture),la Arquitectura orientada a Recursos (ROA, Resource Oriented Architecture), el proyecto AKARI, entre otras, son arquitecturas distribuidas de capas donde cada capa tiene su propia funcionalidad para lograr la interconexión. Comúnmente en muchas de estas implementaciones los protocolos que se utilizan en la capa de aplicación son; el Protocolo de aplicación restringida(CoAP, Constrained Application Protocol),el de Transporte de telemetría de cola de mensajes (MQTT, Message Queue Telemetry Transport), el Protocolo extensible de mensajería y presencia (XMPP, Extensible Messaging and Presence Protocol),entre otros [7-8].Sin embargo, ninguna de estas opciones comunica la información de forma inteligente, es decir, se limitan a transmitir todo dato que obtengan del exterior hacia una unidad central de procesamiento.

En [9] se describe una categorización de cuatro tipos de servicios que debe brindar IoT; (1) los Servicios relacionados con la Identidad son de los más básicos e importantes ya que se encargan de dar un mecanismo para poder identificar a los objetos del mundo real.(2) Los Servicios de Agregación de Información recopilan los datos provenientes del exterior.(3) Los Servicios de Colaboración Consciente se encargan de analizar los datos de los servicios de agregación para una toma de decisiones.Y finalmente los (4) Servicios Generalizados tienen como objetivo proporcionar la información obtenida por los servicios de colaboración consciente a cualquier persona, en cualquier momento y desde cualquier lugar.

La mayoría de los protocolos de aplicación anteriormente mencionados proporcionan únicamente servicios relacionados con la Identidad y Agregación de Información.Por lo que en el presente trabajo se desarrolla un protocolo de comunicación que conecte las entidades involucradas en una Arquitectura enfocada en Internet de las Cosas añadiendo servicios más valiosos.Dichas entidades además de encargarse de brindar los servicios básicos de Identidad y de Agregación de Información,también se busca que sea posible proporcionar Servicios de Colaboración Consciente y Servicios Generalizados dotando de cierta inteligencia a los objetos físicos.Con la combinación de estos servicios se busca que los objetos sean capaces de tomar decisiones para hacer que el sistema sea valioso proporcionando información útil a los usuarios.

El documento está organizado de la siguiente manera:en la sección dos se discuten trabajos y desafíos relacionados con la investigación.En la sección tres se describe la metodología utilizada,se abordan los antecedentes del proyecto y se describe a detalle los elementos de la arquitectura y las características del protocolo de comunicación desarrollado.La sección cuatro expone los resultados que se obtuvieron al implementar el protocolo de comunicación desarrollado usando la arquitectura previamente descrita.En la sección seis se discuten los resultados obtenidos.Finalmente,la sección siete concluye el documento.

2. Trabajo Relacionado

El cómputo en la nube es una tecnología madura que se utiliza para proporcionar servicios de procesamiento y/o almacenamiento de datos a través de internet [5].Sin embargo, al aplicarse en IoT se presentan algunas complicaciones, una de las más importantes es la latencia; con el computo en la nube todos los datos censados u obtenidos del exterior deben de cargarse en servidores centralizados y, después de cierto procesamiento, los resultados deben enviarse de vuelta a los sensores, actuadores o dispositivos finales. Este proceso crea una gran presión en la red, específicamente en los costos de transmisión de datos de ancho de banda y recursos, por consecuencia, el rendimiento de la red empeorará con el aumento del tamaño y flujo de los datos.

En el cómputo de borde y niebla los datos masivos generados por diferentes tipos de dispositivos de IoT se pueden procesar en el borde de la red en lugar de transmitirlos a una infraestructura de nube centralizada [5].Esto con el objetivo de que el proceso de análisis de la información esté más cerca del origen de ésta,es decir,de los dispositivos finales y de los usuarios (Figura 1).

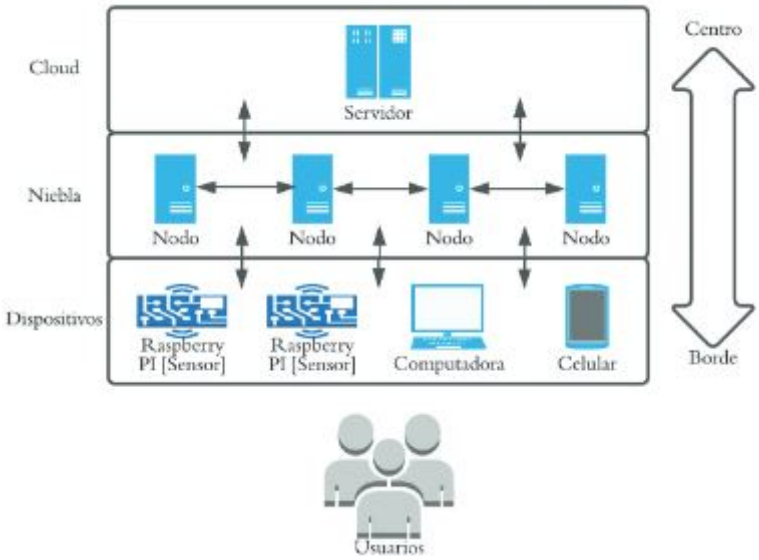

FIGURA 1.
Arquitectura de capas basada en la integración de cómputo de borde y niebla con IoT.

Dentro de los desafíos para IoT se identifican diversas áreas y perspectivas hacia futuras investigaciones como las descritas en [3].En [5] se plantea la integración de cómputo de borde y niebla con IoT, y en [10] se describen diferentes arquitecturas,modelos y aplicaciones de cómputo de borde en IoT.Nuestra investigación se centra en proponer una alternativa de solución para los desafíos de Interoperabilidad y para la integración de IoT con computo de borde y niebla.

2.1 Interoperabilidad

Se debe de tener en cuenta que las aplicaciones de IoT pueden contener una amplia gama de objetos/ dispositivos, redes y plataformas. Cada uno de estos elementos puede tener diferentes especificaciones y es para IoT todo un reto poder lograr una interconexión efectiva para poder interoperar y comunicarse de forma inteligente [3].En [11] se discuten diferentes perspectivas de interoperabilidad, la presente investigación se centra en tres de ellas; (1) Interoperabilidad de dispositivos: Brinda los mecanismos necesarios para la integración de nuevos dispositivos.(2) Interoperabilidad sintáctica: Brinda una interfaz para el intercambio de información, establece un formato y estructura de los datos.(3) Interoperabilidad semántica: Proporciona un modelo para las entidades del sistema intercambien información de manera significativa. Se parte de estos requisitos ya que cada uno corresponde a los Servicios que se desea que el protocolo y la arquitectura planteados en este trabajo proporcionen, como se muestran en la Tabla 1.

Tipo de interoperabilidad	Tipo de servicio
Interoperabilidad de dispositivos	Servicios relacionados con la Identidad
Interoperabilidad sintáctica	Servicios de Agregación de Información
Interoperabilidad semántica	Servicios de Colaboración Consciente

TABLA 1.
Relación de tipos de interoperabilidad con tipos de servicios seleccionados para la presente investigación

2.2 Integración de IoT con computo de borde y niebla

En cómputo de borde y niebla un nodo puede ser cualquier dispositivo de red con capacidad de almacenamiento, cómputo y conectividad de red [5].Para la presente investigación se elige trabajar con esta infraestructura ya que al ser una arquitectura distribuida se pueden brindar servicios con una respuesta más rápida y de mayor calidad. La arquitectura y el protocolo de comunicación de este trabajo tienen como objetivo brindar los mecanismos necesarios para realizar un preprocesamiento de toda la información obtenida por los sensores en sus mismas tarjetas controladoras (nodos) con la intención de crear objetos inteligentes con débil acoplamiento hacia una entidad centralizada.

3. Metodología

La presente investigación utiliza el paradigma empírico con enfoque cuantitativo. Es de tipo cuasi experimental ya que tiene como objetivo diseñar y evaluar una propuesta de protocolo de comunicación para una arquitectura de IoT dando solución a problemas de Interoperabilidad y utilizando una infraestructura de cómputo de borde y niebla. Se utilizó el siguiente diseño para la investigación:

1. Identificación de problemática.
2. Revisión de literatura con respecto a la problemática.
3. Comparativa de alternativas de solución.
4. Elección de solución.
5. Análisis de requerimientos de solución.
6. Desarrollo de solución.
7. Experimentación y pruebas.
8. Ajustes en base a pruebas.

9. Prototipo funcional.

Esta investigación toma como base el trabajo denominado "Interfaz portable utilizando internet y tecnología orientada a aspectos, para la definición de tareas robotizadas basadas en visión computacional" presentado en [12].Este trabajo propuso desarrollar e implementar una arquitectura que habilita el establecimiento de comunicaciones entre diversos clientes para la liberación de tareas críticas mediante una interfaz gráfica usando Internet. Se investigaron los requerimientos y componentes funcionales necesarios en IoT [2] para realizar las adaptaciones necesarias y de esta forma crear una propuesta de protocolo de comunicación aplicado a la Arquitectura que a continuación se describe.

3.1.1 Servidor. Es el centro de control de la arquitectura. Tiene una dirección IP fija. Tiene la lista de usuarios y dispositivos conectados a él. Es el intermediario entre los dispositivos y los usuarios. Contiene dos entidades internas; el Servidor HTTP con el que los usuarios establecen comunicación y el Servidor WebSocket con el que los dispositivos establecen comunicación. Requerimientos: Computadora con JDK versión 1.8 o superior.

3.1.2 Dispositivos. Cuentan con una tarjeta controladora Raspberry PI 3A+ con sistema operativo Raspbian y con JDK 1.8 o superior. Cada tarjeta controladora cuenta con un dispositivo diferente conectado a ellas para obtener información del exterior (por ejemplo, sensores o cámaras).Se le asigna una dirección IP única para poder identificar cada objeto del mundo real. Contiene un agente inteligente que evalúa y clasifica la información que se publicará al servidor, este agente utiliza la librería JADE versión 4.5.Requerimientos:bibliotecas de funciones necesarias para sensores o cámara (Adafruit, OpenCV).

3.1.3 Usuarios. Los usuarios pueden conectarse al servidor mediante cualquier dispositivo con un explorador de internet. El usuario administrador puede consultar información, solicitar información y tomar decisiones del sistema. Puede ver la lista completa de usuarios y dispositivos conectados. Recibe notificaciones de eventos relevantes. Los usuarios visores solo pueden consultar y solicitar información del sistema, no pueden modificarla.

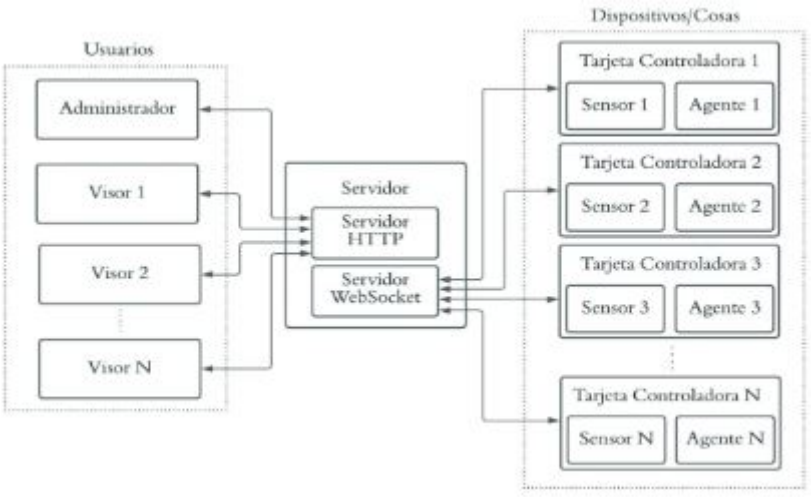

FIGURA 2.
Arquitectura propuesta enfocada a IoT

3.2 Protocolo de comunicación y sus características

A continuación, se describe cada una de las características del protocolo de comunicación desarrollado.

3.2.1 Interoperabilidad. Uno de los puntos más importantes para tener en cuenta al desarrollar el protocolo, es interoperabilidad, es decir, que la solución funcione con la mayor variedad de dispositivos y

sistemas operativos posibles. Por lo cual se eligió trabajar con el lenguaje de programación Java y su máquina virtual.

3.2.2 Manejo de dispositivos y comunicaciones. En IoT se puede llegar a tener una gran cantidad de dispositivos y por lo tanto un gran número de comunicaciones simultaneas, todos estos dispositivos pueden añadirse o retirarse dinámicamente sin que el comportamiento global del sistema se modifique.

Existen dos Interfaces de Programación de Aplicaciones (API,Application Programing Interface)para comunicaciones en IoT[2]:(1) API basada en REST (Representational State Transfer) (2) API basada en WebSocket. En este trabajo se consideraron ambas alternativas para la conexión de diferentes servicios; para establecer comunicación con los dispositivos o "cosas" se utiliza WebSocket y para establecer comunicación con los usuarios finales se utiliza REST, por tal razón, dentro de la entidad de servidor de la Figura 2 puede visualizarse un Servidor WebSocket que se usa para establecer un enlace de red bidireccional entre un "cliente" (Sensor) y un "servidor". En la misma Figura 2 también se visualiza el Servidor HTTP el cual se utiliza para administrar las conexiones con los usuarios finales.

Para activar el Servidor WebSocket y poder establecer conexión con los sensores, se creó un objeto de tipo ServerSocket y se enlazó al puerto 10578 como se muestra en la Figura 3.De la misma forma se crea un servidor HTTP enlazado al puerto 8080 destinado a responder solicitudes de los usuarios, es decir, brindar la información que los sensores están obteniendo del medio físico.

```
System.out.print("Inicializando SERVIDOR... \n");
//Se crea una nueva instancia de ServerSocket para recibir sensores
InetAddress addr = InetAddress.getByName(IP);
ss = new ServerSocket(10578, 0, addr);
System.out.print("Servidor SENSORES en el puerto " + 10578);
System.out.println("\t[OK]");

//Se crea una nueva instancia de ServerSocket para recibir usuarios
ServerSocket serverHttp = new ServerSocket(8080,0,addr);
```

FIGURA 3.
Establecimiento de servidor para sensores y servidor para usuarios

Se requiere una respuesta rápida a cada uno de los clientes conectados a los dos servidores antes mencionados. Es por ello, que para cada una de las conexiones entrantes se crea una instancia de ServidorHilo que hereda de la clase Thread (Figura 4),para que puedan trabajar independientemente una de otra, es decir, cada dispositivo o usuario, conectado a los Servidores podrá ejecutar acciones paralelamente para dar una respuesta rápida.

```
while (true) {
    Socket socketSensor;
    socketSensor = ss.accept();
    int PuertoLocal = socketSensor.getLocalPort();
    System.out.println("Nueva conexión entrante (SENSOR): " + socketSensor + "\n");
    ServidorHilo nCliente = new ServidorHilo(socketSensor, idSensor);
    sensores.add(nCliente);
    nCliente.start();
    idSensor++;
}
```

FIGURA 4.
Creación de instancias de Thread de nuevas conexiones

3.2.3 Estructura de mensajes. El mensaje consta de 4 partes:

- Cabecera:Define el tipo del mensaje (5 bytes).
- Longitud: Longitud de los datos (8 bytes).
- Datos: Información que se va a transmitir (N bytes).
- Checksum: Bytes al final del mensaje para validar integridad de la información (N bytes).

El primer bloque de tipos de mensaje (con código que comienza con el número 1 en la Tabla 2)se utilizan para establecer una conexión entre un nuevo dispositivo y el servidor. El último código de este bloque es para desconectase del servidor. El segundo bloque de tipos de mensaje (con código que empieza con el número 2 en la Tabla 2) corresponde a la suscripción a "tópicos" y publicación de información en dichos tópicos, la descripción a detalle de esta forma de comunicar la información se describe en la sub-subsección.

3.2.4 Metodología para IoT. El tercer bloque de tipos de mensaje (con código que comienza con el número 3 en la Tabla 2)tiene como objetivo el establecimiento de administradores. Finalmente, el cuarto bloque de tipos de mensaje (con código que comienza con el número 4 en la Tabla 2)se utiliza para determinar si la conexión con determinado cliente sigue activa.

TABLA 2.
Tipos de mensaje del protocolo de comunicación propuesto

Mensaje	Código
CONNECT	1A
CONNBACK	1B
ACKCONN	1C
DISCONNECT	1D
SUBS	2A
SUBSBACK	2B
ACKSUBS	2C
UNSUBS	2D
PUBLISH	2E
OFFERADM	3A
ACCEPT	3B
DECLINE	3C
ACKADM	3D
PING	4A
PONG	4B

3.2.5 Análisis de datos de entrada y procesamiento de mensajes. Existen varios métodos para implementar las tareas que tiene que realizar el cómputo de borde [10].En esta investigación se utiliza el Método Local considerando que en la Arquitectura anteriormente descrita se tienen tarjetas controladoras de bajo costo, pero con las capacidades de almacenamiento y procesamiento necesarios para este modelo. El modelo local plantea realizar el preprocesamiento necesario de la información obtenida del exterior antes de comunicarlo al Servidor. Se plantea que este preprocesamiento puede realizarse en un solo nodo, o también comunicándose con más nodos vecinos aplicando el concepto de comunicación de Maquina a Maquina (M2M, Machine-to-Machine)[13].Para realizar este preprocesamiento en el borde de la red en cada Raspberry se utilizan agentes de JADE. En [14] se describe que un agente es un componente especial de software que tiene autonomía, es decir, trabaja sin intervención humana, es social porque puede interactuar con otros agentes o con humanos. Es recreativo, porque recibe su entorno y responde de acuerdo con los cambios que suceden en él. Un agente puede aprender, adaptándose para encajar en su ambiente y a los deseos de sus usuarios.

En la Figura 5 se puede visualizar la conexión que se implementó de los contenedores de los Agentes de JADE. El conjunto de estos contenedores crea una Capa de Análisis horizontal a través de los dispositivos en la red, con este modelo, se propone algo diferente a lo que las Arquitecturas actuales de IoT ofrecen, ya que el análisis de los datos del exterior se realiza inmediatamente después de que se obtienen.

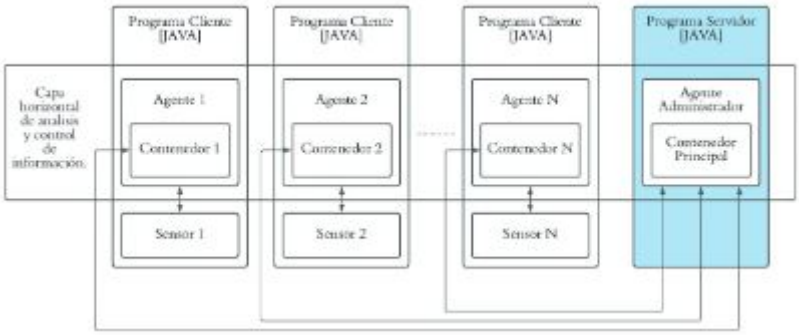

FIGURA 5.
Contenedores de Agentes en la Arquitectura

3.2.6 Seguridad. En [15] se exponen las vulnerabilidades para tener en cuenta al trabajar con IoT, es importante considerarlas ya que los dispositivos están expuestos a Internet y transmiten información privada e incluso controlan sistemas físicos. Para el protocolo de comunicación desarrollado se implementaron las medidas de seguridad para IoT que se describen en [1].Se usó Encriptación para evitar que los datos sean manipulados y de esta forma mantener la confidencialidad. Los mensajes que se envían son encriptados y desencriptados de extremo a extremo, para poder realizar este proceso se utiliza una llave por cada cliente, la llave se asigna en el proceso de crear una nueva conexión. Se utiliza Autenticación de los usuarios con la misma llave generada, y se solicita cada vez que el usuario ingresa al sistema.

4. Resultados

El protocolo de comunicación se implementó utilizando la Arquitectura descrita anteriormente en una casa utilizando múltiples sensores, una computadora personal como servidor y dispositivos móviles para usuarios finales. El proceso de comunicación y las tecnologías utilizadas se muestran en la Figura 6 y se describen a continuación cada uno de los pasos del proceso.

1. La información del exterior es capturada por los sensores utilizando sus respectivas bibliotecas de funciones (la mayoría están programadas para utilizarse con Python).

2. El programa en Python de los sensores se conecta mediante un socket al programa "Cliente" en Java dentro del mismo Raspberry enviándole los datos censados.

3. Dentro del programa de Cliente de Java existe una instancia de Agente de JADE para evaluar si la información es válida, necesita volver a censarse o es irrelevante.

4. En el programa "Cliente" construye la estructura del mensaje con la información que el agente considera valida, se encripta y se empaqueta dicha información.

5. El paquete de información es enviado al servidor de datos mediante sockets.

6. El Servidor al recibir el paquete no realiza ningún procesamiento adicional más que identificar el tópico del sensor del que proviene el mensaje y compartir la información con el servidor http.

7. En los dispositivos finales de los usuarios la información puede presentarse en una página web o en una aplicación móvil, las solicitudes de datos en ambas plataformas se realizan mediante REST.

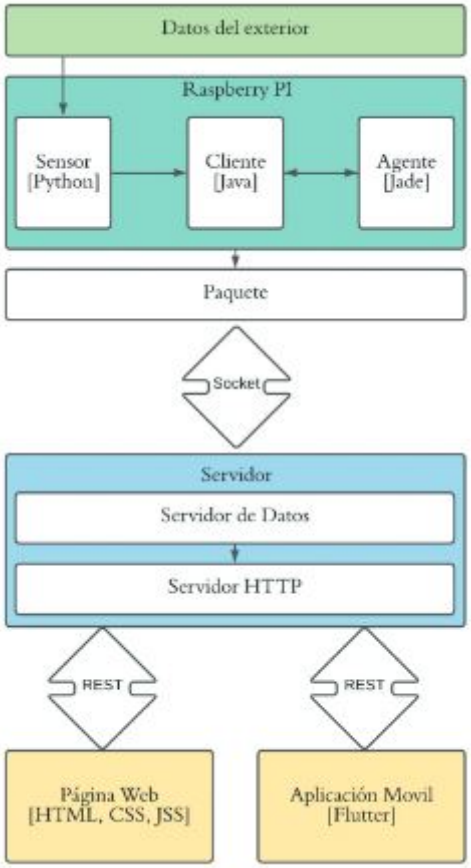

FIGURA 6.
Diagrama de comunicación

En el Servidor HTTP se implementó el manejo de URLs (Localizadores uniformes de recursos) para que los usuarios pudieran solicitar la información fácilmente desde uno de sus dispositivos móviles.La implementación se llevó a cabo en una red local,así que cuando un usuario accede al sitio Web de una URL habilitada en el Servidor HTTP mediante un navegador web o mediante la aplicación móvil, se envía una petición al Servidor HTTP y este responde al cliente enviando una página Web de HTML con la información solicitada.

Como se visualiza en las interfaces graficas (Figura 7 y Figura 8),el Servidor es capaz responder y mostrar eficientemente la información que se obtuvo desde los sensores utilizando el protocolo de comunicación desarrollado dentro del modelo de Arquitectura que también se plantea en este trabajo.

FIGURA 7.
Interfaz Web

FIGURA 8.
Interfaz Móvil

5. Discusión

La Arquitectura que se plantea en este trabajo pudiera compararse con la arquitectura de tres capas descrita en [5],sin embargo, es una alternativa más compleja que implementa modelos de cómputo de borde con el objetivo de comunicar y construir mensajes significativos de forma inteligente para brindar Servicios de Comunicación Consciente. La Arquitectura propuesta también difiere de la Arquitectura orientada a Servicios ya que "baja" el proceso de análisis de los datos de los sensores de la capa de negocio, a un nivel más cercano al origen de estos datos: la tarjeta procesadora de los sensores. Se identifica que IoT surge de la necesidad de lograr una comunicación eficiente entre cosas tal y como existe actualmente la comunicación entre humanos. Es aquí donde entra el termino M2M (machine to machine), el cual se define como la conexión y comunicación de dispositivos en red sin interacción humana [13].Es de suma importancia tomar esta premisa como base para el desarrollo de nuevas aplicaciones de IoT.

6. Conclusiones

Este trabajo ofrece una alternativa de protocolo de comunicación para brindar servicios en una Arquitectura enfocada en Internet de las Cosas, se desarrollaron e implementaron los mecanismos básicos para agregar dispositivos, recolectar información y además se dio el primer paso para brindar Servicios de Comunicación Consciente, ya que los Agentes Inteligentes dentro de la arquitectura eligen y analizan que información se va a comunicar desde el origen de los datos, algo que las arquitecturas actuales no implementan hasta capas más abstractas y centralizadas. Sin embargo, aún se puede mejorar la comunicación y la arquitectura de esta alternativa utilizando la racionalidad y el aprendizaje de los Agentes para lograr tener objetos inteligentes colaborando entre sí.

Agradecimientos

Se agradece la colaboración en esta investigación al estudiante de Ingeniería en Computación Daniel Omar Torres Carbajal y al estudiante de Ingeniería en Informática Alejandro Esaú Álvarez Ruiz,ambos de la Facultad de Ingeniería de la Universidad Autónoma de San Luis Potosí.

8. Referencias

[1] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things(IoT):A vision,architectural elements,and security issues," 2017.Doi:https://doi.org/10.1109/I-SMAC.2017.8058399

[2] R.M. Gomathi,G.H.S. Krishna,E.Brumancia, and Y. M.Dhas,"A Survey on IoT Technologies,Evolution and Architecture,"2018.Doi:https://doi.org/10.1109/ICCCSP.2018.8452820

[3] K.L. M. Ang and J. K. P. Seng, "Application Specific Internet of Things (ASIoTs): Taxonomy, Applications, Use Case and Future Directions," IEEE Access, 2019. Doi:https://doi.org/10.1109/ACCESS.2019.2907793

[4] B.TORĞUL,L. Şağbanşua, and F. B. Balo,"Internet of Things: A Survey," International Journal of Applied Mathematics, Electronics and Computers, 2016. Doi:https://doi.org/10.18100/ijamec.267197

[5] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things:Architecture, Enabling Technologies,Security and Privacy,and Applications," IEEE Internet of Things Journal, 2017.Doi:https://doi.org/10.1109/JIOT.2017.2683200

[6] I. Yaqoob et al.,"Internet of Things Architecture:Recent Advances, Taxonomy,Requirements,and Open Challenges," IEEE Wireless Communications, vol.24(3), pp. 10–16,Jun.2017.Doi:https://doi.org/10.1109/MWC.2017.1600421

[7] A. Čolaković and M. Hadžialić,"Internet of Things (IoT):A review of enabling technologies,challenges,and open research issues," Computer Networks,vol. 144, pp. 17–39, 2018. Doi: https://doi.org/10.1016/j.comnet.2018.07.017

[8] M. Bharti,R. Kumar, and S.Saxena, "Architectural Survey on Internet-of-Things,"2019.Doi:https://doi.org/10.1109/ICIIP47207.2019.8985897

[9] A. I. A. Ahmed et al.,"Service management for iot:Requirements,taxonomy, recent advances and open research challenges,"IEEE Access,vol.7,pp.155472–155488,2019.Doi: https://doi.org/10.1109/ACCESS.2019.2948027

[10] W.Yu et al.,"A Survey on the Edge Computing for the Internet of Things," IEEE Access, vol.6,pp. 6900–6919,2018.Doi: https://doi.org/10.1109/ACCESS.2017.2778504

[11] M. Noura,M. Atiquzzaman,and M. Gaedke,"Interoperability in Internet of Things:Taxonomies and Open Challenges,"Mobile Networks and Applications,2019.Doi:https://doi.org/10.1007/s11036-018-1089-9

[12] F. E. Martínez Pérez,"Interfaz portable utilizando Internet y tecnología Orientada a Aspectos,para la definición de tareas robotizadas basadas en visión computacional," M.S. thesis.Universidad Autónoma de San Luis Potosí, 2005

[13] V. Gazis,"A Survey of Standards for Machine-to-Machine and the Internet of Things,"IEEE Communications Surveys and Tutorials, 2017. Doi: https://doi.org/10.1109/COMST.2016.2592948

[14] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou,"Agent-oriented cooperative smart objects:From IoT system design to implementation," IEEE Transactions on Systems,Man, and Cybernetics:Systems, vol.48(11),pp.1949–1956,2018,doi: https://doi.org/10.1109/TSMC.2017.2780618

[15] I. Luis, F. Gélvez,P. D. Luz, and M. Santos, "Internet de las Cosas:una revisión de vulnerabilidades,amenazas y contramedidas,"Revista Ingenio,vol.17(1), pp. 36–44, 2020.Doi:https://doi.org/10.22463/2011642X.2370