

Tipo de artículo: Artículo original  
Temática: Inteligencia Artificial  
Recibido: 11/12/2017 | Aceptado: 22/01/2018

## **Alternativa *Open Source* en la implementación de un sistema IoT para la medición de la calidad del aire.**

### ***Open Source alternative in the implementation of an IoT system for the measurement of air quality.***

Alexei Ochoa Duarte<sup>1\*</sup>, Libia Denisse Cangrejo Aljure<sup>1</sup>, Tatiana Delgado<sup>2</sup>

<sup>1</sup> Universidad Nacional de Colombia. {agochoad, ldcangrejo}@unal.edu.co

<sup>2</sup> Universidad Tecnológica de la Habana. [tdelgado@ind.cujae.edu.cu](mailto:tdelgado@ind.cujae.edu.cu)

\*Autor para correspondencia: [agochoad@unal.edu.co](mailto:agochoad@unal.edu.co)

---

#### ***Resumen***

El avance tecnológico y científico actual, ha impulsado el desarrollo de sistemas que mejoren la calidad de vida de las personas, aportando bienestar a la comunidad mediante el suministro de información relevante y pertinente para la toma de decisiones. En el contexto tecnológico de Internet de las Cosas (IoT), estos sistemas suponen la medición y el monitoreo de diversas variables del entorno.

La heterogeneidad propia de los datos capturados y los instrumentos de medición utilizados, dificulta la interoperabilidad entre los diversos componentes de IoT. Tales problemas han generado interés en el desarrollo de métodos y herramientas que soporten la heterogeneidad de los datos de sensores, de las mediciones y de los dispositivos de medición. Existen herramientas privadas que han resuelto algunos de estos problemas de interoperabilidad, pero restringen a los desarrolladores de proyectos IoT a utilizar sensores de marcas específicas, limitando el uso generalizado en la comunidad. Adicionalmente, se requiere resolver el reto de integrar protocolos diversos en un mismo proyecto IoT.

Con el propósito de subsanar esas dificultades, se plantea una arquitectura basada en redes de sensores y software inspirados en la cultura libre, que permita la comunicación mediante protocolos diversos en un escenario de aplicación donde se monitorea la calidad del aire para informar a los usuarios, y que mediante la generación de alertas favorezca la toma de decisiones en su vida cotidiana, teniendo en cuenta los datos provenientes de los sensores.

**Palabras clave:** heterogeneidad, internet de las cosas, interoperabilidad, *open source*, sensores

### **Abstract**

*Nowadays, the scientific and technological advances have boosted the development of systems that improve people's life quality, contributing to the welfare of the community by providing relevant and pertinent information for decision-making. In the technological context of the Internet of Things, these systems demand measuring and monitoring several environmental variables.*

*The heterogeneity of the captured data and the measuring instruments used hinder the interoperability among the different components of the IoT. Such problems have raised an interest in the development of methods and tools that support the heterogeneity of the data from the sensors, the measurements and the measuring devices. There are private tools that have solved some of these interoperability problems but they force IoT projects' developers to use sensors from specific brands, limiting their generalized use in the community. Furthermore, it is required to solve the challenge of integrating different protocols in a same IoT project.*

*Under the purpose of overcoming these difficulties, an architecture based on a network of sensors and software inspired in the open-source culture is proposed, which allows communication using several protocols in a scenario in which air quality is monitored to inform users. Besides, by generating alerts, it may help making decisions on a daily basis, taking into account the data provided by the sensors.*

**Keywords:** *heterogeneity, internet of things, interoperability, open source, sensors*

---

## **Introducción**

En la actualidad existe un interés creciente, por medir y monitorear numerosas variables del entorno. Con el fin de proveer un mayor flujo de información, se hace necesaria la interacción de sensores de última generación con diversos principios de funcionamiento, para ofrecer mayores niveles de precisión, mayor eficiencia en el consumo de energía y menores costos y la integración de sus mediciones. (Akyildiz et al, 2002; Razzaque et al 2016)

Los procesos de integración e interoperabilidad de la información generada por esta clase de sensores tienen asociados importantes niveles de complejidad relacionados con la heterogeneidad de los dispositivos, no sólo en sus principios de funcionamiento, sino también en sus formatos, escalas y fuentes. (Perera et al, 2014). Hasta el momento, han surgido múltiples esfuerzos orientados a la integración de dichas mediciones asociadas a una gran diversidad de variables y protocolos. (Alaya et al, 2015; Asensio et al, 2014; Bellavista et al, 2013; Jazayeri et al, 2015; Miorandi et al, 2012; Sicari et al, 2016; Wang et al, 2015; Falocco et al, 2017; Jabbar et al, 2017; Sivarajah et al, 2017; Samara et al, 2016)

Aunque en la revisión realizada, se evidencia la utilización de estándares, también se revela su aplicación limitada al ámbito privado o en aplicaciones de dominio específicas.

Adicionalmente, se identifica la necesidad de desarrollar arquitecturas y sistemas, que sean desarrollados de manera comunitaria, favoreciendo la construcción colectiva del conocimiento. (Blanco, et al, 2017) Por ello, la propuesta presentada en este artículo enfatiza en el uso de herramientas de *hardware* y *software* libre, cuya filosofía de creación otorga libertades básicas al usuario, para utilizar la herramienta con cualquier propósito, estudiar su funcionamiento, con la posibilidad de modificarlo a conveniencia, y además, contar con la disponibilidad de copias para distribuir de manera gratuita, o bien con beneficios económicos.<sup>1</sup> El uso de herramientas que adoptan esta filosofía fortalece el trabajo colaborativo y facilita la solución de los problemas.

En este documento se presenta el desarrollo de un prototipo escalable de sistema de medición y monitoreo de algunas variables ambientales relacionadas con la calidad del aire. De esta manera, se puede contar con una serie de datos, que mediante un posterior tratamiento se convertirán en información de gran utilidad para la toma de decisiones.

La estructura del presente documento, contempla un marco teórico que da sustento al prototipo, seguido de la descripción de la arquitectura planeada. Posteriormente, se describen los componentes del sistema, la manera como se integran en el prototipo y los resultados obtenidos. Finalmente se plantea una discusión y las conclusiones obtenidas del desarrollo del prototipo.

## Marco teórico

Un sensor es un dispositivo capaz de detectar la medida de una magnitud, llamada variable de instrumentación, y convertirla en una señal eléctrica que puede ser procesada, almacenada o transmitida de acuerdo a la finalidad definida por el usuario. (Bröring et al., 2011)

Dependiendo de su utilización (Raghavendra et al, 2006) , los sensores pueden:

- Dar una lectura directamente en la unidad de interés
- Ser conectados a un instrumento indicador que se ocupe de leer la señal y traducirla a la unidad deseada
- Ser conectados a un instrumento que se ocupe de memorizar la señal para un siguiente procesamiento.

Los sensores se clasifican dependiendo de su utilización, de la tipología de señal de salida, o más comúnmente se clasifican dependiendo de la variable física que mide (Ligtenberg, 2009)

---

<sup>1</sup> Filosofía del software libre. Recuperada de: <https://www.gnu.org/philosophy/free-sw.es.html>

Con el avance de la electrónica, los sensores no solamente se ocupan de traducir cantidades físicas en visualizaciones más simples, sino que forman parte de gran cantidad de campos tecnológicos.

En la actualidad, se utilizan sistemas de medición, instrumentados con múltiples sensores interconectados, conocidos como redes de sensores, los cuales integran avances en tecnología electrónica, de comunicación y de computación, permitiendo utilizar redes interconectadas de dispositivos de medición, buscando obtener mediciones más precisas y distribuidas tanto espacial como temporalmente. (Cecílio et al., 2014; Akan et al., 2015a, 2015b, 2015c)

Según la Comisión Internacional Electrotécnica (IEC, 2014) las redes de sensores son “*Redes multicanal, autoorganizadas de nodos de sensores, con tecnología inalámbrica, que son usados para monitorear y controlar fenómenos físicos*”.

Según *National Instruments*, empresa que trabaja en el desarrollo de dispositivos electrónicos y *software* para instrumentación, es “*Una red inalámbrica que consiste en dispositivos distribuidos espaciados autónomos utilizando sensores para monitorear condiciones físicas o ambientales*”<sup>2</sup>.

Los nodos de sensores son el componente central de una red de sensores y generalmente se componen de: un módulo encargado de gestionar la energía para su funcionamiento, un sensor, un microcontrolador y un transmisor inalámbrico. (Yinbiao et al., 2014)

Los avances tecnológicos han hecho posible el despliegue masivo de pequeños dispositivos distribuidos, con características de bajo costo, bajo consumo de energía, y con capacidad de procesamiento local y comunicación inalámbrica, lo cual ha propiciado el desarrollo de un nuevo paradigma, llamado *Internet of Things (IoT)* (Atzori et al, 2010; Miorandi et al., 2012; Blanco et al., 2017), en el cual existe una numerosa cantidad de dispositivos y objetos físicos que se encuentran interconectados a Internet. (Rose Jaren et al., 2015)

Internet de las cosas se puede definir como una infraestructura de objetos, personas, sistemas y recursos de información que se interconectan mediante servicios inteligentes, lo cual les permite procesar datos de los mundos físico y virtual, con el fin de reaccionar de acuerdo a la información obtenida. (Atzori et al., 2010)

De esta manera, las redes de sensores se constituyen en una fuente de datos primaria para *IoT* (Partynski et al, 2013; Rezvan et al, 2015), dado que suministran una cantidad voluminosa de datos provenientes de sensores diversos,

---

<sup>2</sup> Definición de red inalámbrica de sensores, publicada por *National Instruments* en su sitio web: <http://www.ni.com/white-paper/7142/es/>

no sólo en sus principios de funcionamiento, sino también en las escalas de medición, la precisión con la cual realizan la medición, los formatos y unidades de medida, entre otras especificaciones.

Las mediciones provenientes de los sensores y redes de sensores de naturaleza heterogénea deben ser interoperables (Ko et al., 2011) no sólo entre sí, sino también con información adicional que las personas puedan proveer sobre las variables medidas (Bakillah et al., 2013).

## Arquitectura del sistema

Para el planteamiento de la arquitectura del sistema desarrollado, además de tener en cuenta los dispositivos a utilizar, se ha tomado como base la propuesta de arquitectura en capas para aplicaciones IoT (Cardozo, et al, 2016), que establece un diseño cuyos componentes base son: 1) sensores y actuadores, 2) puertas de enlace, 3) servidores de borde y de contexto, y 4) aplicaciones.

La arquitectura desarrollada en la Figura 1 presenta como componentes de *Hardware* los sensores, la tarjeta de desarrollo que permite integrar las mediciones y la interfaz de comunicaciones; como componentes de *Software* los diferentes programas, aplicaciones y APIs que integran los datos en Internet y permiten interactuar con otras aplicaciones.

De igual manera, teniendo como referencia la arquitectura base (Cardozo et al., 2016), se plantea la siguiente descripción de cada una de las capas que componen el sistema:

- A. Capa física y de acceso a la red: en esta capa se encuentran los dispositivos sensores y la tarjeta de desarrollo en la cual se integran las mediciones realizadas. Estos dispositivos se encargan de hacer las mediciones, suministrar la energía al sistema y ser el enlace con la interfaz de comunicaciones, mediante el programa cargado en la tarjeta.
- B. Capa de red: es la encargada de establecer la comunicación entre los datos medidos y el servidor Web de la siguiente capa. Consta de un módulo de conexión que facilita la transferencia de los datos a la red, un protocolo de comunicaciones y un lenguaje de intercambio que facilita la transferencia de datos.
- C. Capa de transporte: se encarga de interconectar las mediciones recibidas por el servidor *back-end*<sup>3</sup> con el resto de Internet. Consta de una plataforma en la que se implementa una base de datos, donde se almacenan los mensajes recibidos, y un servidor Web embebido en una plataforma de integración en el que se elabora una

---

<sup>3</sup> Parte de una aplicación web que se encarga de gestionar las bases de datos, hacer las validaciones necesarias, especificar las rutas de navegación y montar un servidor desde el que se atienden las solicitudes de los clientes.

*API REST*<sup>4</sup> capaz de relacionarse con la base de datos, permitiendo o negando el acceso y la manipulación de los datos recibidos en el lenguaje de intercambio utilizado. También, se encarga de implementar un servidor desde el que se atienden las solicitudes realizadas por la capa de aplicación.

- D. Capa de aplicación: en esta capa se encuentran los clientes front-end<sup>5</sup> que se comunican con el servidor de la capa de transporte, mostrando así, de manera cómoda para los usuarios, los resultados que se obtienen de las solicitudes realizadas al servidor.

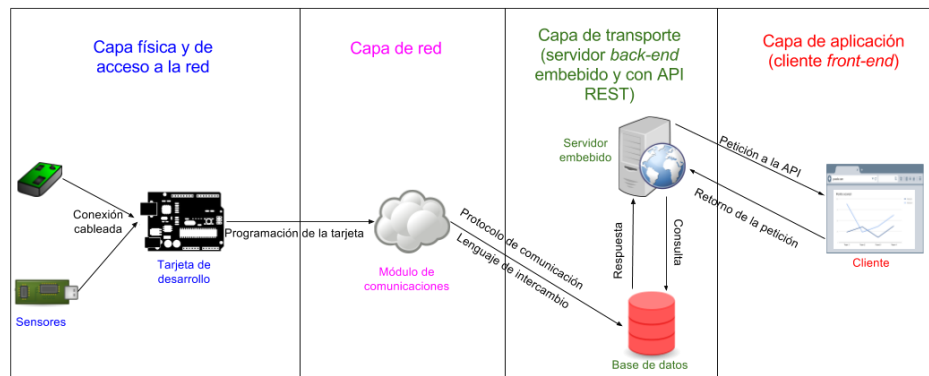


Figura 1. Capas e interacción entre ellas para la arquitectura de referencia

En el diagrama anterior se observan la estructura del sistema y la manera como fluyen los datos y la información en el sistema a través de cada una de las capas descritas con anterioridad.

## Componentes del sistema

### A. Elementos de Hardware

A continuación, se describen los dispositivos físicos que se usan para este sistema, clasificándolos como sensores, tarjetas de desarrollo, dispositivos de comunicación y plataforma de integración.







#### 1. Sensores

Los sensores a utilizar en el prototipo, se han seleccionado teniendo en cuenta la heterogeneidad propia de ellos en cuanto a sus principios de funcionamiento, variables de medida, señales de salida, entre otros aspectos. Una breve descripción de los sensores y sus características se muestra en la tabla 1.

<sup>4</sup> Enfoque de desarrollo de proyectos y servicios basados en la Web. Es un estándar eficiente para la creación de interfaces de desarrollo de aplicaciones API en Internet que se basa en el uso del protocolo HTTP para interactuar con diversidad de formatos de datos.

<sup>5</sup> Parte de una aplicación web encargada de implementar las vistas e interfaces con las que interactuará el usuario final. Para el desarrollo de esta capa se usan principalmente tecnologías propias de un navegador Web.

Tabla 1. Características de los sensores seleccionados

Sensor	VARIABLES de medida	Unidades de medida	Precisión	Principio de funcionamiento	Voltaje de funcionamiento	Señal de salida	Foto
<i>MQ7</i> <sup>6</sup>	Concentración CO	ppm	Depende de la resistencia de carga	Resistivo	PWM alternante entre 1.4V y 5V	Análoga	
<i>MQ135</i> <sup>7</sup>	Concentración CO <sub>2</sub> , NO <sub>x</sub> , NH <sub>3</sub>	ppm	Depende de la resistencia de carga	Resistivo	5V	Análoga	
<i>DHT11</i> <sup>8</sup>	Temperatura y humedad	°C y %	±2°C y ±5%	NTC Resistivo	5V	Digital calibrada	
<i>DHT22</i> <sup>9</sup>	Temperatura y humedad	°C y %	±0.5°C y ±2%	Capacitivo	5V	Digital calibrada	
<i>BMP180</i> <sup>10</sup>	Presión y temperatura	hPa y °C	± hPa y ±0.1°C	Piezo-resistivo	3.3V	Interfaz I2C	
<i>GP2Y1010AU0F</i> <sup>11</sup>	Densidad de polvo	mg/m <sup>3</sup>	±0.1 mg/m <sup>3</sup>	Óptico	5V	Análoga	

## 2. Tarjetas de desarrollo, módulos de comunicaciones y plataformas de integración

En esta sección se relaciona el uso de tecnologías basadas en la filosofía libre, con la intención de permitir un trabajo colaborativo entre usuarios que favorezca el proceso de diseño e implementación del prototipo planteado. Estas tecnologías se describen en la tabla 2.

Tabla 2. Características de los dispositivos utilizados

	Tarjeta de desarrollo	Módulo de comunicaciones	Plataforma de integración
--	-----------------------	--------------------------	---------------------------

<sup>6</sup> MQ7 Semiconductor sensor for carbon monoxide. Su URL es: <http://eph.ccs.miami.edu/precise/GasSensorSpecs/CO.pdf>

<sup>7</sup> MQ135 Gas sensor- Technical data. Su URL es: <https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/resources/SNS-MQ135.pdf>

<sup>8</sup> DHT11 Humidity & Temperature Sensor Datasheet. Su URL es: <http://www.micropik.com/PDF/dht11.pdf>

<sup>9</sup> DHT22 Humidity & Temperature Sensor Datasheet. Su URL es: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

<sup>10</sup> BMP 180 Digital pressure sensor. Datasheet. Su URL es: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>

<sup>11</sup> GP2Y1010AU0F Compact Optical Dust Sensor Datasheet. Su URL es: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>

Dispositivo seleccionado	Arduino <sup>12</sup>	ESP8266 <sup>13</sup>	Raspberry Pi <sup>14</sup>
<b>Características relevantes</b>	Esta tarjeta funciona como el nodo de la red de sensores que permite centralizar y procesar las mediciones realizadas por los sensores descritos anteriormente. Además, poseerá la capacidad de conectarse a internet para enviar y recibir información relevante para el sistema.	Una vez realizada la conexión con la red <i>Wifi</i> , el módulo es capaz de enviar información proveniente de los sensores hacia una dirección IP y puerto que sean establecidos.	Mini computador de placa reducida y de bajo costo, que se ha diseñado para facilitar el aprendizaje y la educación en las ciencias de la computación, creado por la <i>Raspberry Pi Foundation</i> .
<b>Interfaz de programación</b>	Arduino IDE, basado en lenguaje C.	Comandos AT <sup>15</sup> especiales para comunicación con módems e interfaces de telecomunicaciones.	Soporta sistema operativo Linux específico para la tarjeta, llamado <i>Raspbian</i> <sup>16</sup> .

De acuerdo a las tablas 1 y 2, se observa la heterogeneidad inherente a los sensores y dispositivos de *hardware* que son utilizados a lo largo del desarrollo, y los cuales se integran y se hacen interoperables a partir de la propuesta esbozada en este artículo.

### B. Software utilizado en el sistema

Con el fin de hacer interoperables las mediciones de los sensores descritos en el apartado anterior, se hace necesario el uso de un protocolo de comunicaciones que permite recibir datos de diversas fuentes simultáneamente.

Uno de los protocolos utilizados para el envío y recepción de datos provenientes de sensores en el marco de IoT es el MQTT (*Message Queue Telemetry Transport*) que consiste en un protocolo de mensajería de publicación/suscripción extremadamente ligero (Špeh et al, 2016), que permite la comunicación *Machine to machine* (M2M). En este protocolo existe un controlador central, llamado *broker*, que envía, filtra y prioriza las solicitudes que llegan de los nodos subscriptores. (Tantitharanukul et al., 2017)

Los mensajes enviados por los sensores o sistemas embebidos mediante este protocolo deben definir dos elementos que incluyen mensaje y *topic* para el MQTT *broker*. El mensaje, es una cadena de caracteres que contiene los datos a

<sup>12</sup> Compañía italiana, que funciona bajo filosofía libre. Produce tarjetas de desarrollo que utilizan microcontroladores y entornos de desarrollo (IDE).

<sup>13</sup> Módulo de comunicaciones que provee soporte Wifi y es compatible con las tarjetas de desarrollo *Arduino*.

<sup>14</sup> Página Web oficial de *Raspberry Pi Foundation*. <https://www.raspberrypi.org/>

<sup>15</sup> Lenguaje estándar abierto de comandos para configurar y parametrizar módems. Fue creado por la compañía *Hayes Communications*. La siguiente URL muestra un instructivo para su uso: <http://www.electronicaestudio.com/docs/ISTD-034.pdf>

<sup>16</sup> Distribución Linux, basada en Debian, específica para la Raspberry Pi. Se encuentra disponible en <https://www.raspberrypi.org/downloads/raspbian/>



compartir con los subscriptores al *topic*, mientras que este último es otra cadena que permite filtrar y decidir quiénes pueden recibir el mensaje, mediante suscripción. (Špeh et al., 2016)

De esta manera, se pueden integrar datos provenientes de diversas fuentes, que envían sus mensajes a los *topics* correctos, ayudando así a la solución de los problemas causados por la heterogeneidad de las fuentes de datos. Además, este protocolo es eficiente para gestionar datos a un bajo costo y se integra fácilmente con sistemas basados en *software* y *hardware* libre, teniendo una gran amplia gama de aplicaciones que van desde la domótica hasta las ciudades inteligentes. (Kodali & Soratkal, 2016; Kodali & Mahesh, 2016)

Con el fin de que la *Raspberry* tenga acceso a los datos, pueda manipularlos y visualizarlos, se hace necesaria la instalación de algunos programas que se muestran en la tabla 3.

Tabla 3. Características del *software* utilizado

	<b>Broker MQTT</b>	<b>Base de datos</b>	<b>Lenguaje de intercambio de datos</b>	<b>Lenguaje para el desarrollo Web</b>
<b>Herramienta</b>	<i>Mosquitto</i> <sup>17</sup>	<i>MongoDB</i> <sup>18</sup>	<i>JSON</i> <sup>19</sup>	<i>NodeJS</i> <sup>20</sup>
<b>Características relevantes</b>	Permite realizar los procesos de publicación / suscripción en la <i>Raspberry</i> . Se puede integrar con sensores de baja potencia o dispositivos móviles como teléfonos, mini computadores como la <i>Raspberry</i> o microcontroladores como el <i>Arduino</i>	Base de datos NoSQL (Not Only SQL) que permiten un enfoque hacia la gestión de datos y el diseño de base de datos de utilidad para conjuntos de datos distribuidos, como los provenientes de los sensores. Guarda las estructuras de datos en documentos similares a <i>JSON</i> <sup>21</sup> , facilitando la integración de los datos en aplicaciones de manera rápida. Organiza sus bases de datos en colecciones, y dentro de ellas se insertan documentos en los que se tiene un identificador y un valor asignado para esta variable, soportando así consultas dinámicas más	Formato de texto bastante ligero que es de fácil lectura y escritura tanto para los humanos como para las máquinas, siendo independiente del lenguaje que lo genere. De manera general un documento <i>JSON</i> se estructura mediante la utilización de una lista ordenada de parejas nombre/valor.	Los datos almacenados en la Base de datos, deben ser accedidos por herramientas que permitan su tratamiento, visualización y conexión con diversas aplicaciones. Entorno de ejecución basado en <i>JavaScript</i> , que se conecta con la base de datos y permite hacer el tratamiento y la implementación de servicios y aplicaciones web.

<sup>17</sup> Sitio web de *Mosquitto*. Su URL es: <https://mosquitto.org/>

<sup>18</sup> Página web de *MongoDB*. Su URL es: <https://www.mongodb.com/>

<sup>19</sup> Acrónimo de JavaScript-Object Notation.

<sup>20</sup> Página web de *NodeJS*. Su URL es: <https://nodejs.org/es/>. Este entorno de ejecución posee un ecosistema de librerías de código abierto muy amplio.

<sup>21</sup> Acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos.

		eficientes que en otros sistemas de bases de datos.		
--	--	---	--	--

### C. Interfaces de programación de aplicaciones (API)

Como *NodeJS* tiene una amplia variedad de librerías disponibles para programar, se establecen dos interfaces que permiten aumentar la interoperabilidad de los datos provenientes de los sensores y la generación de alertas de acuerdo a las mediciones. A continuación, la tabla 4, presenta la descripción de las principales interfaces utilizadas en el servidor.

Tabla 4. Características de las principales API utilizadas en el *back-end*

Nombre	Aporte al sistema
<i>Express</i> <sup>22</sup>	Infraestructura de aplicaciones <i>web</i> para <i>Node.js</i> que ofrece un conjunto de características para las aplicaciones <i>web</i> y móviles. Cuenta con gran variedad de métodos <i>HTTP</i> y un <i>middleware</i> que facilita la creación de aplicaciones.
<i>Socket.io</i> <sup>23</sup>	Librería de <i>NodeJS</i> que permite crear aplicaciones en tiempo real para casi todos los navegadores y dispositivos móviles, utilizando diferentes mecanismos y protocolos de transporte.
<i>JADE</i> <sup>24</sup>	Motor de plantillas de alto rendimiento, implementado con JavaScript para <i>nodeJS</i> y navegadores. Permite realizar documentos <i>HTML</i> para optimizar las rutas en las aplicaciones <i>web</i> .
<i>Handlebars</i> <sup>25</sup>	Motor de plantillas, parecido a <i>JADE</i> , que es utilizado para la creación de plantillas de tablas y gráficas.
<i>Chartjs</i> <sup>26</sup>	Generador de gráficos, basado en JavaScript que simplifica la labor de elaborar diagramas y figuras basadas en datos.

### D. Herramientas para la visualización

Las aplicaciones realizadas con *NodeJS* pueden ser accedidas mediante un navegador Web, accediendo a la dirección IP donde se encuentra alojado el servicio, esto sin importar si se trata de un PC, un portátil, una *Tablet*, un *Smartphone* u otra clase de dispositivo.

El diseño *front-end* del cliente se basa en la interactividad y la respuesta de las aplicaciones a las solicitudes del usuario final. Para esto, se utilizan algunas técnicas, lenguajes y API que se describen en la tabla 5.

Tabla 5. Características de las principales API utilizadas en el *front-end*

Nombre	Aporte al sistema
<i>AJAX</i> <sup>27</sup>	Técnica de desarrollo web que es utilizada para la creación de aplicaciones interactivas que son ejecutadas por el cliente, al tiempo que se mantiene una comunicación asíncrona con el servidor.
<i>JQuery</i> <sup>28</sup>	Librería amplia y versátil de <i>JavaScript</i> que funciona en gran variedad de navegadores y hace que las peticiones <i>AJAX</i> y la manipulación de documentos <i>HTML</i> se haga de manera más sencilla.

<sup>22</sup> Página web de *Express*. Su URL es: <http://expressjs.com/>

<sup>23</sup> Página web de *Socket.io*. Su URL es: <https://socket.io/>

<sup>24</sup> Página web de *JADE*. Su URL es <https://www.npmjs.com/package/jade>

<sup>25</sup> Página web de *Handlebars*. Su URL es <https://github.com/ericf/express-handlebars>

<sup>26</sup> Página web de *ChartJs*. Su URL es <http://www.chartjs.org/>

<sup>27</sup> Acrónimo de *Asynchronous JavaScript And XML*.

<sup>28</sup> Página web de *JADE*. Su URL es <https://www.npmjs.com/package/jade>

CSS <sup>29</sup>	Lenguaje que se utiliza para describir el estilo de un documento <i>HTML</i> . Con este lenguaje se diseña el estilo de los diferentes componentes de una página web.
HTML <sup>30</sup>	Lenguaje estándar a cargo del W3C <sup>31</sup> que define la estructura básica y el código necesario para definir el contenido de un sitio Web.

## Integración de los componentes de hardware y software en el prototipo

En la sección anterior se han definido diversas tecnologías que se integran en el desarrollo del prototipo de sistema IoT para la medición de calidad del aire.

A continuación, en la figura 2, se muestra la arquitectura solución utilizada para el prototipo. En esta figura se observan los diferentes componentes del sistema y se puede apreciar la manera en la que interactúan para el correcto funcionamiento del prototipo.

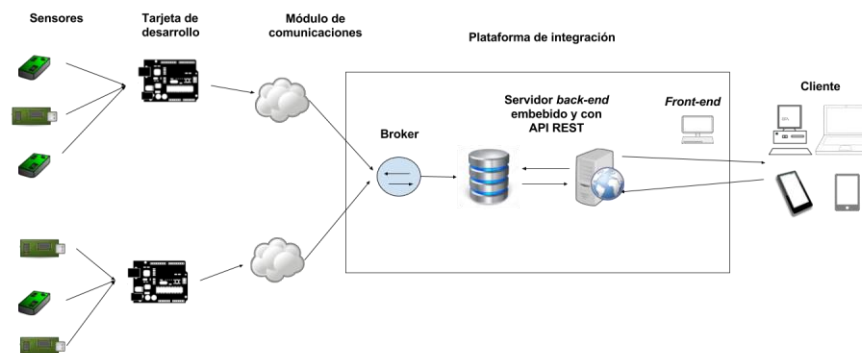


Figura 2. Arquitectura solución

## Resultados

El desarrollo del sistema IoT se basa en tres módulos que se encuentran programados en *NodeJS* y ejecutan aplicaciones web directamente en la *Raspberry pi*.

- A. *Módulo de registro de sensores*: Consiste en una aplicación que utiliza *NodeJS*, *Express*, *Socket.io* y otras API con el fin de realizar el modelo de la base de datos y hacer el registro de los sensores y dispositivos que se utilizan en el prototipo. Esta aplicación tiene una interfaz gráfica que facilita al usuario la realización del registro y actualización de los datos de sensores. De igual manera, se encarga de hacer las validaciones necesarias para el usuario

<sup>29</sup> Acrónimo de Hoja de estilos en Cascada. Es una tecnología utilizada junto con *JavaScript* y *HTML* para el desarrollo de aplicaciones Web.

<sup>30</sup> Acrónimo de Lenguaje de Marcado para Hipertextos. Se usa para crear la representación visual de una página Web.

<sup>31</sup> Consorcio internacional que define estándares y recomendaciones para el crecimiento de la *World Wide Web* a nivel mundial



Figura 3. Módulo de registro de sensores

- B. *Módulo de mediciones:* Es el encargado de la comunicación MQTT entre las tarjetas *Arduino* y la *Raspberry*. No cuenta con interfaz gráfica, pero implementa un sistema de alertas en caso de que las mediciones excedan valores determinados por el usuario.
- C. *Módulo de visualización:* Su tarea consiste en hacer peticiones a la base de datos, hacer algunas transformaciones de los mismos para organizarlos en un formato que mediante la comunicación con *ChartJS* proporcione al usuario la visualización del estado de las variables medidas por los sensores de manera gráfica. A continuación, se muestran dos de las gráficas obtenidas para las variables medidas por los sensores.

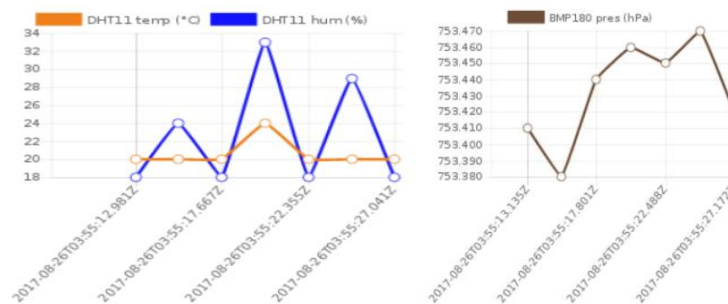


Figura 4. Módulo de visualización – mediciones de los sensores DHT11 y BMP180

Este módulo involucra la elaboración de gráficas en tiempo real en la generación de la interfaz de usuario.

## Discusión y conclusión

En este documento se presenta el desarrollo de un prototipo de sistema de medición y monitoreo de algunas de las variables ambientales relacionadas con la calidad del aire, a saber: temperatura, humedad, presión atmosférica, densidad de polvo y concentraciones de monóxido y dióxido de carbono, metano y óxidos de nitrógeno.

Los datos capturados por los sensores son integrados mediante el microcontrolador central de una tarjeta *Arduino*, la cual establece comunicación *Wifi* con un servidor Web que mediante el uso de un protocolo MQTT facilita la inserción

de los datos en una base de datos de MongoDB, con la cual se interactúa desde el servidor utilizando el lenguaje de intercambio JSON para hacer interoperables los datos. Además, el servidor implementa métodos para el tratamiento y visualización de datos por medio de un cliente basado en un navegador Web, lo cual hace que sean visibles desde diversidad de dispositivos.

Mediante el uso de las herramientas libres para el desarrollo del prototipo, se hace posible la integración, fusión e interoperabilidad de datos provenientes de sensores diversos, lo cual es un avance concreto en la interoperabilidad de la información proveniente de fuentes de datos heterogéneas.

Adicionalmente, el prototipo implementado, hace funcional la solución para nuevos sensores o sistemas de medición, mediante la implementación de pequeños cambios en la configuración del prototipo, lo cual permitirá el uso de diversos dispositivos interconectados, con la finalidad de realizar mediciones de las variables en diferentes puntos. De esta manera, se podrá contar con más datos, que mediante un posterior tratamiento se convertirán en información de gran utilidad para orientar los procesos de toma de decisiones en materia ambiental por parte de los usuarios, ya sean individuos o comunidades.

## Referencias

- Akan, O., Bellavista, P., Cao, J., Coulson, G., Dressler, F., Ferrari, D., ... Zomaya, A. Y. (2015). *Internet of Things. User-Centric IoT*. (R. Giaffreda, R.-L. Vieriu, E. Pashe, G. Bendersky, A. J. Jara, J. J. P. C. Rodrigues, ... B. Mandler, Eds.), *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Editorial Board.
- Akan, O., Bellavista, P., Cao, J., Coulson, G., Dressler, F., Ferrari, D., ... Zomaya, A. Y. (2015). *Mobile Computing, Applications, and Services*. (S. Sigg, P. Nurmi, & F. Salim, Eds.), *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Editorial Board.
- Akan, O., Bellavista, P., Cao, J., Coulson, G., Dressler, F., Ferrari, D., ... Zomaya, A. Y. (2015). *Sensor Systems and Software*. (E. Kanjo & D. Trossen, Eds.), *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Editorial Board. <https://doi.org/10.1007/978-3-642-32778-0>
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4), 393–422. [https://doi.org/10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)
- Alaya, M. Ben, Medjah, S., Monteil, T., & Drira, K. (2015). Toward semantic interoperability in oneM2M architecture. In *IEEE Communications Magazine* (Vol. 53, pp. 35–41). <https://doi.org/10.1109/MCOM.2015.7355582>

- Asensio, Á., Marco, Á., Blasco, R., & Casas, R. (2014). Protocol and architecture to bring things into internet of things. *International Journal of Distributed Sensor Networks*, 2014. <https://doi.org/10.1155/2014/158252>
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- Bakillah, M., Liang, S., Zipf, A., & Arsanjani, J. (2013). Semantic Interoperability of Sensor Data with Volunteered Geographic Information: A Unified Model. *ISPRS International Journal of Geo-Information*, 2, 766–796. <https://doi.org/10.3390/ijgi2030766>
- Bellavista, P., Cardone, G., Corradi, A., & Foschini, L. (2013). Convergence of MANET and WSN in IoT urban scenarios. *IEEE Sensors Journal*, 13(10), 3558–3567. <https://doi.org/10.1109/JSEN.2013.2272099>
- Blanco, T., Casas, R., Manchado-Pérez, E., Asensio, Á., & López-Pérez, J. M. (2017). From the islands of knowledge to a shared understanding: interdisciplinarity and technology literacy for innovation in smart electronic product design. *International Journal of Technology and Design Education*, 27(2), 329–362. <https://doi.org/10.1007/s10798-015-9347-7>
- Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., ... Lemmens, R. (2011). *New generation Sensor Web Enablement. Sensors (Basel, Switzerland)* (Vol. 11). <https://doi.org/10.3390/s110302652>
- Cardozo, A., Yamin, A., & Souza, R. (2016). An Architecture Proposal to Distributed Sensing in Internet of Things.
- Cecílio, J., & Furtado, P. (2014). Wireless Sensors in Heterogeneous Networked Systems. *Wireless Sensors in Heterogeneous Networked Systems*, 2, 39–59. [https://doi.org/10.1007/978-3-319-09280-5\\_4](https://doi.org/10.1007/978-3-319-09280-5_4)
- Falocco, S., Larsson, J., & Nandi, S. (2017). A (likely) X-ray jet from NGC6217 observed by XMM-Newton, 11(August), 1–11. <https://doi.org/10.1093/mnras/stx2168>
- IEC. (2014). Internet of Things : Wireless Sensor Networks Executive summary, 78.
- Jabbar, W. A., Ismail, M., Nordin, R., & Arif, S. (2017). *Power-efficient routing schemes for MANETs: a survey and open issues. Wireless Networks* (Vol. 23). Springer US. <https://doi.org/10.1007/s11276-016-1263-6>
- Jazayeri, M. A., Liang, S. H. L., & Huang, C. Y. (2015). Implementation and evaluation of four interoperable open standards for the internet of things. *Sensors (Switzerland)*, 15(9). <https://doi.org/10.3390/s150924343>
- Ko, J., Eriksson, J., Tsiftes, N., Dawson-Haggerty, S., Vasseur, J.-P. P., Durvy, M., ... Culler, D. (2011). Beyond

- Interoperability: Pushing the Performance of Sensornet IP Stacks. *Proceedings of the ACM Conference on Networked Embedded Sensor Systems, ACM SenSys 2011*, 1–11. <https://doi.org/10.1145/2070942.2070944>
- Kodali, R. K., & Mahesh, K. S. (2016). A low cost implementation of MQTT using ESP8266. *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 404–408.
- Kodali, R. K., & Soratkal, S. (2016). MQTT based Home Automation System Using ESP8266. *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*.
- Ligtenberg, A., & Kooistra, L. (2009). Sensing a changing world. *Sensors*. Retrieved from <http://www.mdpi.com/1424-8220/9/9/6819/htm>
- Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516. <https://doi.org/10.1016/j.adhoc.2012.02.016>
- Partynski, D., & Koo, S. G. M. (2013). Integration of Smart Sensor Networks into Internet of Things: Challenges and Applications. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 1162–1167. <https://doi.org/10.1109/GreenCom-iThings-CPSCOM.2013.202>
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context Aware Computing for The Internet of Things. *IEEE Communications Surveys & Tutorials*, *16*(1), 414–454. <https://doi.org/10.1109/SURV.2013.042313.00197> T4 - A Survey M4 - Citavi
- Raghavendra, C., Sivalingam, K., & Znati, T. (2006). *Wireless sensor networks*. Retrieved from <http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract>
- Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Cla, S. (2016). Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, *3*(1), 70–95. <https://doi.org/10.1109/JIOT.2015.2498900>
- Rezvan, M., & Barekatin, M. (2015). The Sensors Are Innovative in Internet of Things. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, *146*, 191–196. <https://doi.org/10.1007/978-3-319-18802-7>
- Rose Jaren, Eldridge Scott, C. L. (2015). The internet of things: an overview - Understanding the issues and challenges of a more connected world. *The Internet Society (ISOC)*, (October). <https://doi.org/10.5480/1536-5026-34.1.63>
- Samara, K., & Hosseini, H. (2016). Aware Diffusion: A Semi-Holistic Routing Protocol for Wireless Sensor Networks.



*Wireless Sensor Network*, 8(3), 37–49. <https://doi.org/10.4236/wsn.2016.83004>

- Sicari, S., Rizzardi, A., Miorandi, D., Cappiello, C., & Coen-Portisini, A. (2016). A secure and quality-aware prototypical architecture for the Internet of Things. *Information Systems*, 58. <https://doi.org/10.1016/j.is.2016.02.003>
- Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*, 70, 263–286. <https://doi.org/10.1016/j.jbusres.2016.08.001>
- Špeh, I., & Heđ, I. (2016). A Web - Based IoT Solution for Monitoring Data Using MQTT Protocol. *2016 International Conference on Smart Systems and Technologies (SST)*, 249–253. <https://doi.org/10.1109/SST.2016.7765668>
- Tantitharanukul, N., Osathanunkul, K., Hantrakul, K., Pramokchon, P., & Khoenkaw, P. (2017). MQTT-Topic naming criteria of open data for smart cities. *20th International Computer Science and Engineering Conference: Smart Ubiquitous Computing and Knowledge, ICSEC 2016*, 1–6. <https://doi.org/10.1109/ICSEC.2016.7859892>
- Wang, F., Hu, L., Zhou, J., Hu, J., & Zhao, K. (2015). A semantics-based approach to multi-source heterogeneous information fusion in the internet of things. *Soft Computing*, 21(8), 2005–2013. <https://doi.org/10.1007/s00500-015-1899-7>
- Yinbiao, S., Lee, K., Lanctot, P., Juanbin, F., Hao, H., Chow, B., ... Qui, W. (2014). Internet of Things: Wireless Sensor Networks. *Internation Electronic Commision*, (December), 1–78.