

INTEGRACIÓN DE UN SISTEMA INALÁMBRICO DE MUESTREO CON ESP8266 Y RASPBERRY PI

WIRELESS SENSOR SYSTEM INTEGRATION BASED ON THE INTERNET OF THINGS

Lorenzo Antonio García Tena

Universidad Autónoma de Ciudad Juárez, México
Lorenzo.garcia@uacj.mx

Osslan Osiris Vergara Villegas

Universidad Autónoma de Ciudad Juárez, México
overgara@uacj.mx

Vianey Guadalupe Cruz Sánchez

Universidad Autónoma de Ciudad Juárez, México
vianey.cruz@uacj.mx

Recepción: 28/octubre/2019

Aceptación: 23/noviembre/2019

Resumen

En el presente artículo, se muestra el desarrollo y los resultados obtenidos de un sistema inalámbrico para el muestreo de temperatura y humedad dentro del Laboratorio Nacional de Vivienda y Comunidades Sustentables ubicado en las instalaciones de la Universidad Autónoma de Ciudad Juárez. El sistema se compone de sensores DHT22, tablillas NodeMCU-ESP8266 y un Raspberry Pi 3 que funge como servidor. Además, por medio de una interfaz se ofrece un acceso al sistema que se encuentra conectado a una red de Internet, lo que permite el monitoreo en tiempo real y el análisis histórico de las variables medidas. El propósito del muestreo es medir las propiedades de aislamiento térmico de nuevos materiales propuestos para la construcción de viviendas. Para observar el desempeño del sistema construido se realizaron pruebas de precisión de las mediciones y de tiempos de acceso. Los resultados obtenidos demuestran que el sistema es robusto y ofrece mediciones en tiempo real con un error de 1 grado centígrado y con un tiempo máximo de acceso de 3.6 segundos.

Palabras Claves: Internet de las cosas, Linux, placa NodeMCU-ESP8266, Raspberry Pi, sensor de temperatura y humedad.

Abstract

In this paper, is presented the development and the results obtained of a wireless system for sensing temperature and humidity inside the National Laboratory of Housing and Sustainable Communities located in the facilities of the Autonomous University of Ciudad Juárez. The system is integrated using DHT22 sensors, a NodeMCU-ESP8266 development board and a Raspberry Pi 3 that acts as a server. Additionally, the system is accessible through an interface to the Internet network, which allows real-time monitoring and the historical analysis of measures variables. The main purpose of the sensing task is to measure the isolation properties of new proposed materials for housing development. To validate the performance of the built system, testing was performed on the data measured as well as access timing. The results obtained show the robust performance of the proposed system and how can it offer real-time data with a mean error of 1 Celsius degree, and with a maximum access time of 3.6 seconds.

Keywords: *Internet of things, Linux, NodeMCU-ESP8266 board, temperature and humidity sensor, Raspberry Pi.*

1. Introducción

Una de las tendencias tecnológicas más explotadas en años recientes, dentro del manejo de sistemas, es la que permite que la información de una vivienda o edificación esté disponible en cualquier momento independientemente del lugar o la ubicación geográfica [Rawat, 2014]. Lo anterior, se puede lograr gracias al Internet, a la existencia de potentes unidades de procesamiento y al uso de sensores que permiten crear una red para recolectar información de un ambiente físico [Abu, 2014], [Boukerche, 2019].

Es debido al llamado Internet de las cosas (IoT, por sus siglas en inglés) que los objetos de la vida diaria pueden ser aumentados y conectados con comunicación y habilidades de cómputo que pueden ser monitoreadas desde cualquier locación [Ruta, 2018]. El IoT es un paradigma de comunicaciones que permite el intercambio de información entre el mundo físico y el digital [Ray, 2018]. La idea principal se

enfoca en hacer al Internet cada vez más inmersivo y con mayor alcance [Horn, 2014], [Zanella, 2014].

Sin embargo, la posibilidad de tener acceso e interactuar con una variedad de sensores y sus correspondientes mediciones genera una gran cantidad de datos, los cuáles deben ser almacenados, procesados e interpretados para tomar decisiones inteligentes de manera autónoma [Skiadopoulos, 2019]. De hecho, es el manejo remoto de las grandes cantidades de datos (big data, por sus siglas en inglés) el núcleo y principal reto a resolver dentro de los servicios prestados por el IoT [Targio, 2016], [Wang, 2017].

El objetivo de construir una red inalámbrica de sensores (WSN, por sus siglas en inglés) es monitorear un determinado campo de interés y mantener un objetivo particular bajo estricta vigilancia. Además, se debe lidiar con las limitadas capacidades de memoria, procesamiento y batería que suelen tener los nodos de muestreo, además de su falta de flexibilidad [Sadegh, 2019]. Por lo tanto, para alcanzar el objetivo de un monitoreo efectivo, los diferentes sensores de la red deben cooperar en las tareas de procesamiento e intercambio de información.

En el presente artículo, se muestra el desarrollo e implementación de una red de sensores inalámbrica para la captura y procesamiento, a través de una interfaz de usuario, de la información de temperatura y humedad dentro del Laboratorio Nacional de Vivienda y Comunidades Sustentables ubicado en las instalaciones de la Universidad Autónoma de Ciudad Juárez (UACJ). El propósito del muestreo es medir las propiedades de aislamiento térmico de nuevos materiales propuestos para la construcción de viviendas confortables. A diferencia de otros trabajos presentados en la literatura, la implementación utiliza elementos de bajo costo y de código abierto los cuales son parametrizados según las necesidades específicas del usuario final. El resto del documento está organizado de la siguiente manera: en la Sección II, se presenta una descripción metodológica del sistema implementado, el flujo de información, la interfaz realizada para acceso a los muestreos y los aspectos de seguridad implementados. La sección III describe los experimentos realizados y los resultados obtenidos. En la sección IV, se presenta una discusión de los resultados obtenidos en términos de precisión y tiempo de acceso.

Finalmente, en la sección V se ofrecen las conclusiones del trabajo realizado, y se describen las oportunidades para trabajo a futuro.

2. Métodos

La formación del sistema implementado consta de 3 etapas definidas por dispositivos, desde la adquisición de la señal de muestreo, hasta la disponibilidad del dato en línea. Dichas etapas son:

- Captura de datos de temperatura y humedad por medio del sensor DHT22.
- Manejo de señal de muestreo en sistema embebido (NodeMCU) como dato dentro de una página HTML.
- Disposición de interfaz como acceso global para la obtención de datos de muestreo, en tiempo real y como historial en archivo de hoja de datos (csv).

La disposición de los datos en línea representa la necesidad de un esquema de seguridad apropiado, mismo que es explicado posteriormente ya que representa un ajuste de seguridad y no de funcionalidad del sistema implementado.

Sensor de Temperatura y Humedad DHT22

El DHT22 (AM2302) es un sensor digital para medir temperatura y humedad relativa de buen rendimiento y bajo costo. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos, no posee salida analógica [Mihai, 2016].

La señal que entrega el sensor es digital, lo cual permite un uso simplificado para su conexión con dispositivos embebidos como lo es el NodeMCU. En la figura 1 se muestra una imagen del sensor DHT22.



Figura 1 El sensor DHT22.

El bajo costo del sensor es una ventaja que permite, en gran medida, la reducción de costos en la implementación del sistema propuesto. Las alternativas de muestreo en el mercado son, en su mayoría, opciones integradas Hardware-Software, lo cual impide, típicamente, la flexibilidad de uso. Cada elemento de muestreo es integrado por un par de sensores DHT22, mismos que son conectados al siguiente elemento que representa la segunda etapa de la implementación.

Sistema Embebido NodeMCU (ESP8266)

El ESP8266 es un chip Wi-Fi de bajo costo que funciona mediante el protocolo TCP/IP. Incluye un microcontrolador (Tensilica Xtensa LX106) y el software necesario para la conexión 802.11. Además, la mayoría de los modelos dispone de entradas/salidas digitales de propósito general (GPIO), así como una entrada analógica. NodeMCU es una placa de desarrollo basada en el ESP12E el cuál es probablemente el módulo más popular que integra el SoC ESP8266 [Aziz, 2018]. En la figura 2 se muestra un ejemplo de la mencionada placa de desarrollo.



Figura 2 La tarjeta NodeMCU – ESP8266.

La gran ventaja del sistema propuesto, en el aspecto funcional, con respecto al resto de módulos básicos de ESP8266, es que su programación se hace totalmente transparente, al no requerir ningún cambio en sus pines para la programación, y disponer de conexión USB al igual que plataformas con base en Arduino.

Sin embargo, la ventaja principal de NodeMCU es para las aplicaciones que trabajaban con sistemas que requieren de Internet [Firmansyah, 2018]. Un módulo de Arduino Uno y un *shield* WiFi cuestan alrededor de 16 veces más que un esquema basado en ESP8266. NodeMCU, que realiza la misma función, se puede adquirir por un precio que ronda los 130 pesos mexicanos; eso deja al alcance de

muchos proyectos un chip que permite las mismas funciones de un Arduino: trabajar con sensores, motores, relevadores, pantallas touchscreen, etc.

Una diferencia importante es que Arduino se programa en C++ y NodeMCU en LUA, un lenguaje script que permite cargar scripts al NodeMCU para que corran en el inicio y mandarle comandos directamente mediante una consola o Telnet.

Raspberry Pi 3

Raspberry es un ordenador del tamaño de una tarjeta de crédito, consta de una placa base sobre la que se monta un procesador, un chip gráfico y una memoria RAM. Fue lanzado en 2006 por la Fundación Raspberry Pi con el objeto de estimular la enseñanza de informática en las escuelas de todo el mundo [Ferdoush, 2014]. El modelo específico empleado es Raspberry Pi 3 B+, mismo que es el más vendido históricamente de los modelos ofertados por el fabricante. Una imagen del Raspberry puede verse en la figura 3.



Figura 3 La Raspberry Pi 3 B+.

La Raspberry tiene un costo operativo de menos de 1000 pesos mexicanos, lo cual contribuye al ahorro en la implementación del sistema de manera significativa. Es una alternativa barata comparada con estructuras usadas para servidores de red comúnmente desarrollados [Vujovic, 2015].

La base de datos fue implementada con el lenguaje de consultas estructurado (SQL, por sus siglas en inglés), y para el servidor de red se utilizó Apache 2.0. La selección de Raspberry como plataforma para generar la base de datos y el servidor en red se basa, principalmente, en su costo y en la capacidad de correr el kernel Linux basado en Debian.

Interfaz Integrada por CSS, HTML y PHP

La interfaz gráfica de usuario (GUI, por sus siglas en inglés), para interactuar con la red de sensores y para el monitoreo se implementó como una página de Internet. El diseño de la GUI se basó en los lenguajes de hojas de estilo en cascada (CSS, por sus siglas en inglés), de marcado en hipertexto (HTML, por sus siglas en inglés), y de preprocesamiento de páginas en hipertexto (PHP, por sus siglas en inglés).

CSS es un lenguaje de diseño gráfico que permite definir la presentación de un determinado documento estructurado. Así como con HTML se define qué es cada elemento y dónde se sitúa, con CSS se define como se quiere presentar. Su principal utilidad es la de separar la presentación de la propia estructura del documento. De esa forma se pueden aplicar los mismos estilos a varios documentos similares o aplicar distintas presentaciones al mismo elemento, lo cual agrega valor y funcionalidad a una interfaz [Alawar, 2017].

Por otro lado, HTML es un lenguaje de marcado, sin estructuras lógicas, que permite organizar la información de forma que los navegadores puedan entenderla. HTML genera una estructura en forma de árbol que permite definir no sólo qué es cada elemento si no quién lo contiene, cuál va a ser el título de la página, cuál el menú de navegación, entre otros [García, 2017].

Finalmente, PHP es un lenguaje muy utilizado en el entorno del desarrollo web, y fue uno de los primeros lenguajes del servidor, es decir, al ser ejecutado en el servidor web directamente tiene acceso a bases de datos, conexiones en red y otras tareas antes de que se envíe la página final a la cual el cliente tendrá acceso. Además, PHP puede incorporar directamente un documento HTML sin la necesidad de tener que llamar a un archivo externo que procese los datos [Amanatidis, 2016].

Linux para Seguridad en el Sistema

Por su naturaleza, GNU/Linux es considerado un sistema operativo robusto al incorporar características comunes de UNIX relacionadas con la seguridad. A pesar de ello, el administrador de sistemas sigue jugando un papel fundamental en dicho aspecto, especialmente, cuando hay involucrados servidores que ofrecen servicios a miles de usuarios [Salah, 2013].

Los sistemas de tipo UNIX utilizan el llamado modelo de Control de Acceso Discrecional (DAC, por sus siglas en inglés), donde el usuario tiene completo control sobre los objetos que le pertenecen y los programas que ejecuta. En ese modelo, el programa ejecutado dispone de los mismos permisos que el usuario que lo está ejecutando. DAC implica que la seguridad del sistema depende de las aplicaciones que se están ejecutando y, por tanto, cuando se produce una vulnerabilidad afecta a todos los objetos a los que el usuario tiene acceso. Una alternativa es el Control de Acceso Obligatorio (MAC, por sus siglas en inglés), donde el administrador define una política de seguridad que los usuarios no pueden modificar. La política va más allá que el establecimiento de propietarios de archivos. MAC fija contextos donde se indica cuándo un objeto puede acceder a otro objeto, por lo que se puede mejorar el nivel de seguridad.

El sistema propuesto únicamente es accesible por segmentos de direcciones IP pertenecientes a las instituciones de educación superior dentro del grupo de Laboratorios Nacionales de Vivienda y Comunidades Sustentables del Consejo Nacional de Ciencia y Tecnología (CONACyT).

Metodología

El sistema emplea 2 sensores DHT22 para cada unidad de muestreo, esto es debido al requerimiento de tener un muestreo fuera y un muestreo dentro del material de construcción aislante a utilizar, como se muestra en la figura 4.

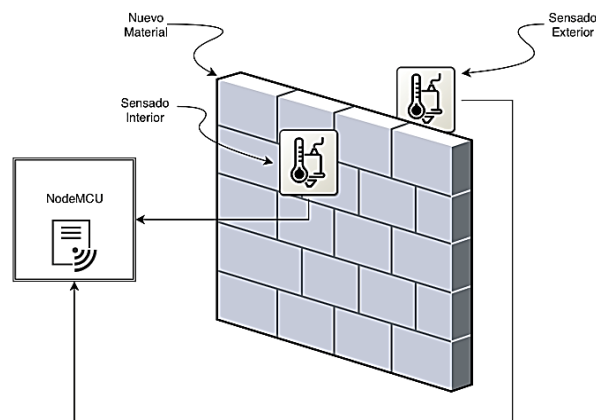


Figura 4 El sistema de muestreo inalámbrico propuesto.

Las señales de los sensores que incorporan las mediciones de temperatura y humedad son enviadas a la tablilla NodeMCU en formato digital, y los valores son usados como entrada para ser mostrados en medidores gráficos digitales.

Dentro de los requerimientos de uso de la plataforma, se tuvo la petición de poder incorporar un monitoreo en tiempo real, mismo que tiene como propósito ofrecer la facilidad de lectura para los usuarios del laboratorio, y con ello poder identificar lecturas erróneas o problemas que se pudieran estar presentado al momento, sin la necesidad de ingresar al sistema. Cada unidad de muestreo ofrece una carátula HTML que es utilizada como alimentación para los procesos que se corren dentro de la Raspberry Pi 3. El usuario puede acceder a la página y validar sus datos, lo cual le permite ingresar a la parte del monitoreo de sensores por unidad. En la interfaz se permite al usuario solicitar el historial de muestreo por medio de la base de datos SQL que administra el Raspberry Pi 3. En la figura 5 se puede apreciar el flujo del proceso de muestreo en el NodeMCU, y en la figura 6 se muestra el flujo de la base de datos y el servidor en el Raspberry Pi.

El sistema tiene implementado un proceso de respaldo de datos periódico dentro de la base de datos, por seguridad, la información específica de ese proceso no se explica en el presente artículo. El proceso establecido está sujeto a cambios requeridos según los resultados obtenidos del uso de la plataforma, la etapa inicial ha finalizado, sin embargo, la disponibilidad de modificar según necesidades es una etapa que requiere de un tiempo de uso de mínimo un año.

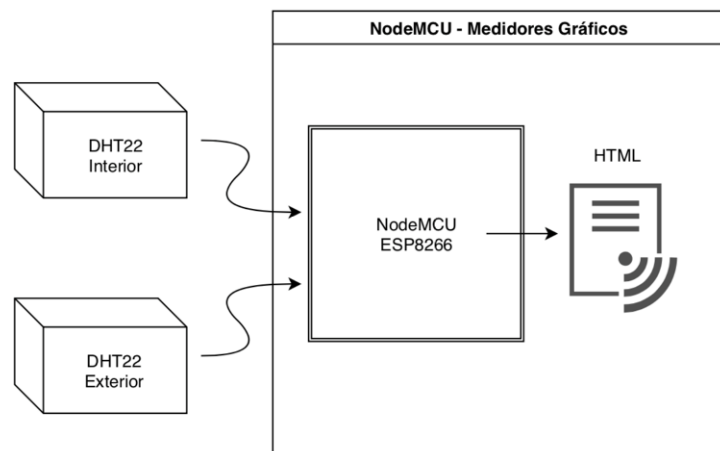


Figura 5 Flujo del proceso de muestreo.

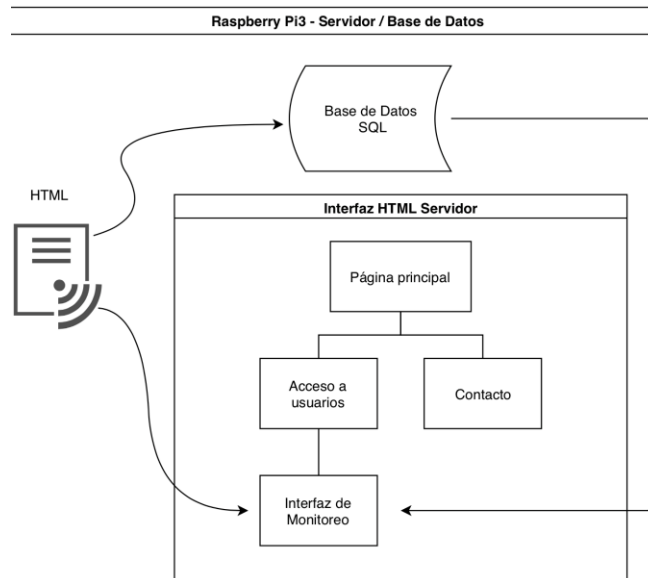


Figura 6 Diagrama de flujo para base de datos y servidor.

3. Resultados

El sistema se encuentra disponible en línea para su uso gracias a la red de la UACJ que permite un acceso directo con una IP global. Se realizaron pruebas iniciales para medir tiempos de respuesta del sistema. En la tabla 1, se pueden apreciar los tiempos de respuesta para 3 pruebas realizadas dentro de la red UACJ y 3 pruebas desde una red externa por medio de una red virtual privada (VPN).

Tabla 1 Tiempos de respuesta del sistema.

Prueba	Latencia (ms)	Página Principal (s)	Monitoreo (s)	Contacto (s)
1 - Red local	11	2.2	2.4	3.2
2 - Red local	19	1.7	2.2	3.3
3 - Red local	7	1.0	2.4	3.3
4 - Red externa	23	2.4	3.0	3.6
5 - Red externa	34	2.7	2.9	3.6
6 - Red externa	22	2.6	2.8	3.4

Los resultados obtenidos son favorables debido a que los tiempos de respuesta no afectan la utilidad general del sistema. Se midió como máximo un tiempo de respuesta de 3.6 segundos mismo que se justifica debido a la interacción del sistema con los servidores que dan soporte para el servicio de correo electrónico de la UACJ.

La exactitud de lecturas por parte de los sensores fue medida en comparativa con un termómetro infrarrojo FLUKE-62 MAX. En la tabla 2 se puede apreciar la comparativa de lecturas de temperatura.

Tabla 2 Validación de mediciones de temperatura.

PRUEBA	DHT22 (°C)	FLUKE-62 MAX (°C)	Error de lectura (°C)
1. Interior	18	19.1	1.1
2. Interior	16	17.3	1.3
3. Interior	24	22.1	1.9
4. Exterior	27	27.4	0.4
5. Exterior	26	25.2	0.8
6. Exterior	30	31.1	1.1
Promedio:			1.1

La figura 7 muestra la primera pantalla de la GUI, presentación de las funciones del laboratorio. La interfaz permite la validación como usuario actual de algún esquema de muestreo, y el acceso a una segunda pantalla donde se permite el monitoreo de los datos de temperatura y humedad por unidad de muestreo.



Figura 7 Interfaz principal HTML de la página del laboratorio.

En la figura 8 se muestra un ejemplo de la pantalla de acceso para usuarios. En dicha pantalla, el usuario puede iniciar la adquisición de datos por la base de datos, misma que incorpora una nueva lectura cada 10 minutos, a su vez, el usuario puede descargar los muestreos acumulados. Además, en la figura 9 se puede apreciar el monitoreo que ofrece la plataforma en línea por unidad de muestreo.

La asignación de usuarios se da una vez establecida la comunicación con los administradores del laboratorio, la acción puede ser iniciada por los usuarios en la primera pantalla de la interfaz en línea como se muestra en la figura 10.

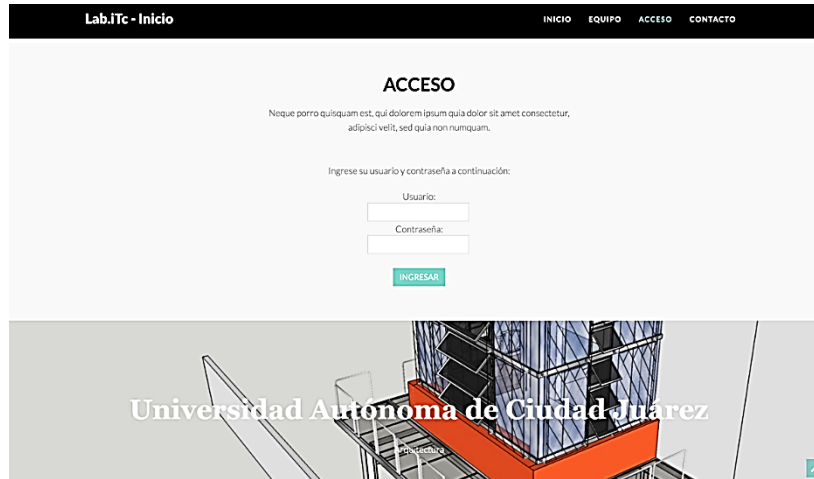


Figura 8 Interfaz HTML para acceso a monitoreo de sensores.

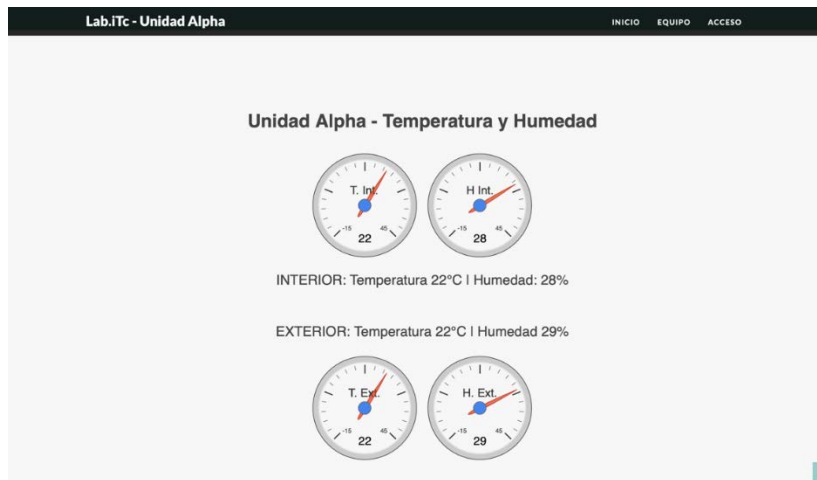


Figura 9 Lecturas de muestreo en Unidad Alpha.

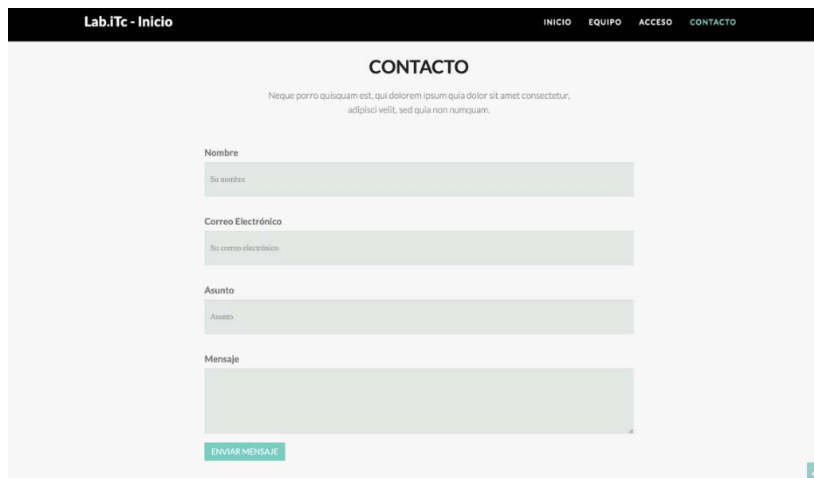


Figura 10 Interfaz HTML donde se establece contacto con el administrador.

4. Discusión

Para desarrollar el sistema de muestreo se invirtió un total de \$1,700.00 pesos, con dicha inversión se logró tener un total de 3 unidades de muestreo, recordando que cada unidad de muestreo ofrece 2 lecturas, una para interior y una para exterior. Si se compara la implementación contra lo que se había usado previamente, que es el sistema HOBO Temperature/Relative Humidity/Light/External Data Logger [Onset, 2019], se puede observar que el precio se redujo casi a la mitad.

El sistema usado previamente tiene un precio de \$3,000.00 pesos, y sirve para un solo muestreo, el cual no puede ser monitoreado en tiempo real y requiere de un software específico para lograr obtener los datos. El software tiene un costo de \$1,500 pesos [Onset, 2019]. Para lograr obtener la misma calidad de muestreo, el usuario debería invertir un total de alrededor de \$19,500 pesos. Cabe mencionar que dicha inversión seguiría sin ofrecer el monitoreo en tiempo real.

Los resultados presentados son una etapa inicial de pruebas, el uso continuo de los elementos dentro del proyecto como sistema, debe permitir identificar mejoras y nuevos requerimientos. El uso como búsqueda de la mejora continua es un elemento que implementaciones del IoT deben siempre buscar.

La contribución principal del presente trabajo es fructificar esfuerzos colaborativos entre distintas ramas de la ciencia. El área de IoT empieza a tomar forma cada vez más clara dentro de muchos sectores de la industria, sin embargo es fundamental buscar que dichas tecnologías no surjan únicamente bajo el concepto de marcas o plataformas, sino que puedan ser usadas, y mejoradas, por un usuario con conocimientos técnicos limitados, es ahí donde está el potencial de IoT y es por eso que es fundamental alimentar la colaboración entre el mayor número de disciplinas y la variedad de productos y tecnologías que ofrece actualmente IoT.

La actualización de los elementos por índole de seguridad es algo que conlleva razones suficientes para que se busque el mantenimiento periódico del sistema. Además, se requiere de una sinergia cíclica por parte de los usuarios de sistema y el equipo que integra los mismos. Es en esta labor donde el valor de los proyectos multidisciplinarios crea raíces que fortalecen el uso de la tecnología desarrollada para necesidades locales.

5. Conclusiones

En el artículo se mostró el desarrollo y los resultados obtenidos para el muestreo de temperatura y humedad dentro del Laboratorio Nacional de Vivienda y Comunidades Sustentables de la UACJ. El propósito general del proyecto fue alcanzado, haciendo uso de recursos limitados, no por falta de estos, sino por la capacidad de adquirir equipo barato con la propuesta hecha y realizada.

En la actualidad, se trabaja para generar otras implementaciones con el uso de IoT, para poder integrar soluciones flexibles en las manos de usuarios que normalmente frecuentan esquemas de Software y Hardware predeterminados. El ámbito multidisciplinario y el aprovechamiento del desarrollo de proyectos desde un origen de funcionalidad abre un mundo de posibilidades a futuro, eso nutre de una manera muy directa a cada miembro de los grupos multidisciplinarios. El ahorro monetario que pueden generar dichas mancuernas de trabajo entre disciplinas es significativo. El proyecto descrito en este documento es parte de un proyecto más grande y no se tuvieron las facilidades de diseñar todo el sistema con el tiempo debido, es por eso que hay elementos de IoT que se consideran para futuras mejoras dentro del laboratorio, como lo pueden ser Node-RED, MQTT, OpenHAB, entre otras.

6. Bibliografía y Referencias

- [1] Abu M., S. Lin, D. Niyato, H. Tan, Machine learning in wireless sensor networks: Algorithms, strategies, and applications, *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996-2018, 2014.
- [2] Alawar M., S. Abu, CSS-tutor: An intelligent tutoring system for CSS and HTML, *International Journal of Academic Research and Development*, vol. 2, no. 1, pp. 94-98, 2017.
- [3] Amanatidis T., A. Chatzigeorgiou, Studying the evolution of PHP web applications, *Information and Software Technology*, vol. 72, pp. 48-67, 2016.
- [4] Firmansyah R., A. Widodo, A. Romadhon, M. Hudha, P. Saputra, N. Lestari, The prototype of infant incubator monitoring system based on the Internet of things using NodeMCU ESP8266, *Journal of Physics*, vol. 1171, pp. 1-9, 2018.

- [5] Aziz D., Webserver based smart monitoring system using ESP8266 node MCU module, *International Journal of Scientific and Engineering Research*, vol. 9, no. 6, pp. 801-808, 2018.
- [6] Boukerche A., P. Sun, Connectivity and coverage based protocols for wireless sensor networks, *Ad Hoc Networks*, vol. 80, pp. 54-69, 2019.
- [7] Ferdoush S., X. Li, Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications, *Procedia Computer Science*, vol. 34, pp. 103-110, 2014.
- [8] García A., V. Fresno, R. Martínez, A. Zubiaga, Using fuzzy logic to leverage HTML markup for web page representation, *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 919-933, 2017.
- [9] Horn J., A Koohang, J. Paliszkievicz, The Internet of things: Review and theoretical framework, *Expert Systems with Applications*, vol. 133, pp. 97-108, 2019.
- [10] Mihai B., How to use the DHT22 sensor for measuring temperature and humidity with the Arduino board, *Acta Universitatis Cibiniensis – Technical Series*, vol. 68, pp. 22-25, 2016.
- [11] Onset, (Acceso en mayo 2019). Sistema HOB0 Temperature/Relative Humidity/Light/External Data Logger: <https://www.onsetcomp.com/products/data-loggers/u12-012>.
- [12] Onset, (mayo 2019). Software HOB0ware Pro: <https://www.onsetcomp.com/products/software/bhw-pro-dld>.
- [13] Rawat P., K. Deep, H. Chaouchi, J. Bonin, Wireless sensor networks: A survey on recent developments and potential synergies, *The Journal of Super Computing*, vol. 68, no. 1, pp. 1-48, 2014.
- [14] Ray P., A survey on Internet of things architectures, *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291-319, 2018.
- [15] Ruta M., F. Scioscia, G. Loseto, F. Gramegna, S. Ieva, A. Pinto, E. Di Scioscio, Social Internet of things for domotics: A knowledge-based approach over LDP-CoAP, *Semantic Web*, vol. 9, no. 6, pp. 781-802, 2018.

- [16] Sadegh M., H. Barati, Dynamic key management algorithms in wireless sensor networks: A survey, *Computer Communications*, vol. 134, pp. 52-69, 2019.
- [17] Salah K., J. Alcaraz, J. Bernal, J. Marín, S. Zeadally, Analyzing the security of Windows 7 and Linux for cloud computing, *Computers & Security*, vol. 34, pp. 113-122, 2013.
- [18] Skiadopoulos K., A. Tshipis, K. Giannakis, G. Koufoudakis, E. Christopoulou, K. Oikonomou, G. Kormentzas, I. Stavrakakis, Synchronization of data measurements in wireless sensor networks for IoT applications, *Ad Hoc Networks*, vol. 89, pp. 47-57, 2019.
- [19] Targio I., V. Chang, N. Badrul, K. Adewole, I. Yaqoob, A. Gani, E. Ahmed, H. Chiroma, The role of big data in smart city, *International Journal of Information Management*, vol. 36, no. 5, pp. 748-758, 2016.
- [20] Vujovic V., M. Maksimovic, Raspberry Pi as a sensor web node for home automation, *Computers & Electrical Engineering*, vol. 44, pp. 153-171, 2015.
- [21] Wang K., Y. Wang, X. Hu, Y. Sun, D. Deng, A. Vinel, Y. Zhang, Wireless big data computing in smart grid, *IEEE Wireless Communications*, vol. 24, no. 2, pp. 58-64, 2017.
- [22] Zanella A., N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, 2014.