

Diseño y desarrollo de un sistema de monitoreo remoto implementando Internet de las cosas

Y. Aguilar Molina¹, S.A. Sosa Ramírez¹, J.E. Galindo Morales¹,
A.N. Morales Reyes², R. Gallardo Sánchez², I.V. Villaseñor García³

¹ Universidad de Guadalajara,
México

² Instituto Tecnológico de Toluca,
México

³ Universidad Michoacana de San Nicolás de Hidalgo,
México

yehoshua@valles.udg.mx,
{sergio.sosa0939, josue.galindo4506}@alumnos.udg.mx,
{amoralesr1, rgallardos}@toluca.tecnm.mx

Resumen: La constante transformación de la industria, nuevas tendencias tecnológicas y el desarrollo de nuevos procesos de manufactura suponen nuevas necesidades que mejoran la calidad de los procesos y transforman por completo, la interacción humana con las máquinas. El *Internet de las cosas* se presenta como solución a estas áreas de oportunidad, dando como resultado un concepto llamado *Industria 4.0*. Actualmente existe un interés creciente por medir y monitorear numerosas variables del entorno. Con el fin de proveer un mayor flujo de información haciendo uso del Internet de las cosas (IoT, por sus siglas en inglés), el cual está cambiando la forma en que las personas se organizan, interactúan y participan en diversas facetas de la sociedad. *Internet of things* se considera una integración perfecta de usuarios y dispositivos para converger el ámbito físico con entornos virtuales derivados de humanos. El presente artículo presenta un sistema de monitoreo remoto implementado, con base en una Raspberry Pi 3B+, un servidor web y sensores de temperatura, humedad y presión atmosférica. Para recopilar y presentar datos medidos, se desarrolló un sitio web, donde el usuario accede para observar los datos recopilados.

Palabras clave: Monitoreo remoto, IoT.

Design and Development of a Remote Monitoring System Implementing the Internet of Things

The constant transformation of the industry, new technological trends and the development of new manufacturing processes pose new needs that improve the quality of processes and completely transform human interaction with machines. The Internet of Things is presented as a solution to these areas of opportunity, resulting in a concept called Industry 4.0. Currently there is a growing interest in

measuring and monitoring numerous variables in the environment. In order to provide a greater flow of information by making use of the Internet of Things (IoT), which is changing the way in which people organize, interact and participate in various facets of society. Internet of things is considered a seamless integration of users and devices to converge the physical realm with virtual environments derived from humans. This article presents a remote monitoring system implemented, based on a Raspberry Pi 3B +, a web server and temperature, humidity and atmospheric pressure sensors. To collect and present measured data, a website was developed, where the user accesses to observe the collected data.

Keywords: Remote Monitoring, IoT.

1. Introducción

El Internet de las cosas (IoT), denominado también “Internet de los objetos”, representan un parteaguas entre el hoy y ayer. Tomando ventaja de las capacidades que nos otorga el Internet (como herramienta de comunicación por excelencia), el Internet de las cosas es el máximo exponente del mismo; un concepto con un extenso margen de aplicación y un incuestionable impacto en los procesos auténticos modernos. Mismo que se encuentra presente en sectores como la Industria 4.0, domótica, medicina, agricultura, ganadería, hotelería, entretenimiento, solo por mencionar algunos. “La Industria 4.0 hace referencia a una nueva fase en la revolución industrial caracterizada por la automatización, la digitalización de los procesos y el uso de las tecnologías de la electrónica y de la información en la manufactura. El concepto fue acuñado en Alemania en 2011, dentro del contexto de una política económica gubernamental basada en estrategias de alta-tecnología (C. Cortés, J. Landeta and J. Chacón, 2020). La medición de los cambios meteorológicos ha sido siempre indispensable, la obtención y conocimiento de valores como temperatura, humedad, viento, presión atmosférica, entre otros; en un lugar específico representa a ventaja para el desarrollo de las actividades comerciales y de producción. La elaboración de este sistema de monitorio ambiental permitirá la medición de datos confiables para realizar una validación, manipulación y análisis de la información recolectada y está servir como base para futuras predicciones meteorológicas.

En la literatura puede encontrarse la propuesta de Palacios, P., Córdova, A., 2018, quien propone un sistema de IoT basado en Raspberry Pi, un sensor de temperatura, una base de datos (cloud DataBase) enfocada a las casas inteligentes con actuadores para interactuar con el sistema medido. El problema con esta propuesta es que limita al usuario a usar una aplicación de Android para hacer uso de las funciones; mientras que nuestra propuesta permite al usuario acceder desde cualquier dispositivo conectado a internet. Además, el sistema propuesto en este artículo es modular, lo que permite agregar más estaciones de monitoreo ambiental, sensores y actuadores; así como otorgar permisos de administrador a ciertos usuarios registrados.

Además, también se encuentra la propuesta de Danita, M., Blessy, M., Nithila, S., Namrata, S., Paul, J., 2018, en donde un sistema de control de condiciones (intensidad lumínica, humedad, etc.) se implementa en un invernadero. En dicha propuesta no se

otorga al usuario la capacidad de interactuar ni modificar el estado de los actuadores, lo cual puede ser preciso en caos muy particulares.

El desarrollo de este sistema de monitoreo remoto precisó la implementación de la metodología cascada, donde se realiza un estudio de los requerimientos técnicos para dar pie al inicio del proyecto. Definiendo el alcance del proyecto según los recursos disponibles. Durante el análisis se desarrolló un servidor web basado en Apache2, MariaDB, PHP 7.3, MySQL y PHPMyAdmin; basándonos bajo la filosofía LAMP (Linux, Apache, MySQL, PHP), los cuales son recursos completamente Open Sources. También se diseñó la base de datos y sitio web con conexión a la estación de monitoreo con sensores y microcontroladores. Para el diseño del sitio web se realizaron archivos .PHP, .HTML y .CSS donde se visualiza la información de la base de datos. Asimismo, el sitio web fue realizado siguiendo un diseño web responsivo. Para el diseño electrónico se utilizó la tarjeta Raspberry Pi 3B+, a los que se le conectan los sensores de: temperatura, humedad y presión atmosférica. Mismos que fueron programados y controlados con la Raspberry Pi y el lenguaje de programación Python; el cual utiliza para la lectura e inserción de datos en la página web.

2. Materiales y equipos

Para el desarrollo del proyecto, se utilizaron elementos de software y hardware para la implementación del sistema de internet de las cosas, en los cuales se implementa el desarrollo de su arquitectura.

2.1. Descripción del hardware utilizado

El hardware utilizado en el presente artículo incluye una tarjeta Raspberry Pi, una tarjeta Arduino Uno, un circuito integrado ATMEGA328P, los sensores BME280, YL-69, ML8511 y el receptor Neo-6, los cuales se describen a continuación:

Raspberry Pi: Es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo, el cual estimula la enseñanza de informática. El modelo original se convirtió en más popular de lo que se esperaba, hasta incluso vendiéndose fuera del mercado objetivo para usos como robótica.

Arduino UNO: Es una placa de microcontrolador basada en ATmega328P. Tiene 14 pines de entrada / salida digital (de los cuales 6 se pueden usar como salidas PWM), 6 entradas analógicas, un resonador cerámico de 16 MHz (CSTCE16M0V53-R0), una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio.

ATMEGA328P: Es un circuito integrado de alto rendimiento basado en un microcontrolador RISC, combinando 32 KB (ISP) flash una memoria con la capacidad de leer-mientras-escribe, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interfaz serial orientada a byte de 2 cables, SPI puerto serial, 6-canales 10-bit Conversor A/D (canales en TQFP y QFN/MLF packages), temporizador "watchdog" programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software.

Sensor BME280: Es un sensor de humedad especialmente desarrollado para aplicaciones móviles y wearables donde el tamaño y el bajo consumo de energía son parámetros clave de diseño. La unidad combina sensores de alta linealidad y precisión y es perfectamente factible para un bajo consumo de corriente, estabilidad a largo plazo y alta robustez de EMC. (C. A. Merlo, M. Antonio, R. Silva, C. Y. Sosa y M. Marciano, 2014).

Sensor YL-69: El sensor de humedad del suelo, se usa generalmente para detectar la humedad en la tierra o arena o cualquier sustrato permeable a la humedad y que se pueda enterrar. Por lo tanto, es perfecto para construir un sistema de riego automático o para controlar la humedad de tus plantas en un sistema “indoor”. El sensor está conformado por dos piezas o placas y un cable que une: la placa electrónica YL-38 (a la izquierda) y la sonda con dos patas, que detecta el contenido de agua YL-69 (a la derecha).

Sensor ML8511: Es un sensor UV, adecuado para adquirir intensidad UV en interiores o exteriores. El ML8511 está equipado con un amplificador interno, que convierte la fotocorriente en voltaje dependiendo de la intensidad de los rayos UV. Esta característica única ofrece una interfaz sencilla para circuitos externos como ADC. En el modo de apagado, típico corriente de espera es 0.1 A permitiendo una mayor duración de la batería.

Receptor Neo-6: Familia de receptores GPS independientes con el u-blox 6 de alto rendimiento motor de posicionamiento. Estos receptores flexibles y rentables ofrecen numerosas opciones de conectividad en una miniatura” (U-blox, 2020). Su arquitectura compacta y sus opciones de alimentación y memoria hacen que los módulos NEO-6 ideal para dispositivos móviles.

2.2. Descripción del software utilizado

Para la implementación de este artículo se utilizaron los lenguajes de programación PHP y Python, además de HTML, SQL y JavaScript; así como Fritzing y Proteus para simulación electrónica.

PHP: Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. (PHP, 2020).

Python: Lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. (Martelli, 2007).

HTML: Es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web. Otras tecnologías además de

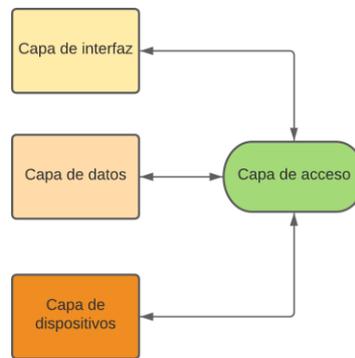


Fig. 1. Diagrama de capas.

HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). (HTML, 2020).

Fritzing: Esta plataforma “brinda apoyo a diseñadores y artistas para el desarrollo de dispositivos electrónicos mediante la simulación de los mismos” (Knöring, A., Wettach, R., Cohen, J., 2009), facilitando la conexión entre los mismo y permitiendo corregir errores y observar el comportamiento de nuestro circuito.

3. Arquitectura de un sistema de IoT

El éxito del IoT depende en gran medida de una arquitectura que permita que sea una tecnología dinámica y segura, por ello, la arquitectura debe cumplir ciertos requerimientos para que esta tecnología sea viable. A gran escala, debe permitir que la tecnología sea distribuida, donde los objetos puedan interactuar entre ellos, escalable, eficiente y segura (Hernández, 2020). La tecnología distribuida, implica que la información adquirida pueda provenir de diferentes lugares y, a su vez, procesada por máquinas o servidores diferentes. La arquitectura debe ser capaz de mostrar todos los componentes como un único sistema a los ojos de los usuarios y desarrolladores.

La interacción entre objetos conectados, hace referencia a que, muchas veces, los dispositivos no se pueden comunicar entre ellos; en el IoT, el intercambio de datos debe ser entre cualquier objeto y de forma bidireccional. Se requiere que entre los diferentes dispositivos exista interacción. Creando un proceso de estandarización de protocolos de comunicación.

La arquitectura escalable necesita de una arquitectura que permita la escalabilidad, es decir, que de si hoy se conectan 10,000 y mañana 1,000,000 el sistema funcione exactamente igual. Dando lugar a que el sistema o el entorno pueda crecer continuamente. La eficiencia energética consiste en tener una autonomía máxima, no se puede depender completamente de baterías de corta duración y, ya que están diseñadas para no apagarse, que su consumo no implique un aumento considerable del consumo de energía. Asimismo, una de las ideas fundamentales es integrarlo con energías renovables e inagotables.

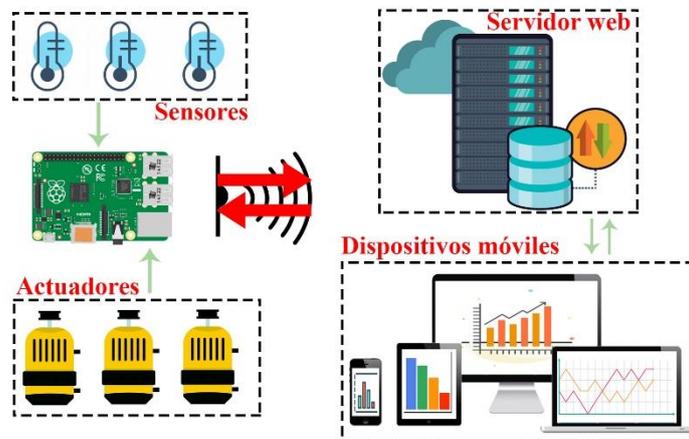


Fig. 2. Diagrama de funcionamiento.

Por otro lado, no se puede construir una arquitectura totalmente impenetrable. La seguridad en el IoT es algo que se debe tener en cuenta desde el principio, ya que no solo están en peligro nuestros datos, inclusive podrían estar vulnerable nuestra integridad física.

La arquitectura del sistema IoT utilizada, está conformado por la parte lógica y la física; que, a su vez, se clasifican en cuatro capas: capa de interfaz (página web), capa de datos (base de datos), capa de acceso (protocolos de comunicación) y la capa de dispositivos que integra los controladores, actuadores y sensores, ver Figura 1.

La capa de datos se encarga del almacenamiento, captura y consulta de los datos recolectados por la capa de dispositivos (sensores y actuadores). En la captura, los datos son obtenidos desde la capa acceso, la cual recibe los datos de los sensores y actuadores para, posteriormente, mostrarlos en la capa de interfaz y de ahí ser capturados y enviados vía WiFi a la base de datos para su almacenamiento y consulta por la página web en la capa interfaz.

La capa de acceso se encarga de transmitir la información entre la capa de interfaz, capa de datos y capa de dispositivos. Es, básicamente, el conjunto de protocolos que permiten la comunicación en nuestro sistema y que este dé un funcionamiento correcto. Asimismo, se encarga en enviar las acciones requeridas por el usuario al microcontrolador para que los actuadores cumplan la acción de usuario sobre el sistema.

La capa de interfaz se encarga de brindar al usuario la interacción con todo el sistema, permitiéndole consultar los registros almacenados en la base de datos. De igual manera, se permite a este recibir, en tiempo real, los datos enviados por los sensores y actuadores desde los microcontroladores y, por último, le otorga al usuario control sobre los actuadores.

La capa de dispositivos, se encarga de enviar datos o recibir órdenes desde y hacia el o los microcontroladores, para que estos lean los datos de los sensores y ordenen a los actuadores cumplir su respectiva función.

Como se puede ver en la Figura 2., se presentan los componentes divididos separados en “Hardware” y “Software” intercomunicados entre sí. Por el lado del hardware

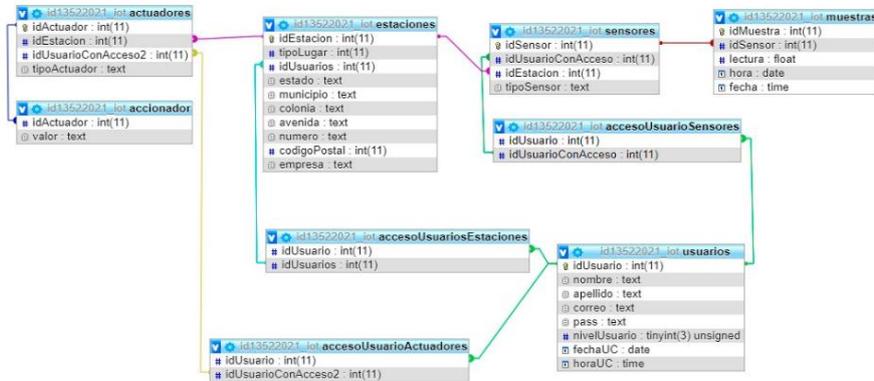


Fig. 3. Modelo de base de datos relacional.

contamos con una Raspberry Pi 3B+ funcionando como microcontrolador de los sensores y actuadores conectados a la misma. Por el lado del software tenemos un servidor web con la base de datos y la interfaz web con la que el usuario podrá interactuar con el sistema. Conectados ambos por medio de internet.

4. Desarrollo de un prototipo de estación para monitoreo remoto

La naturaleza multidisciplinaria de este sistema de monitoreo remoto y la complejidad de éste precisa de cuatro pilares fundamentales para el proceso de elaboración y desarrollo del mismo. Dichos pilares que son:

- Diseño de base de datos.
- Diseño del sitio web.
- Diseño electrónico.
- Diseño mecánico.

4.1. Diseño de la base de datos

Modelo entidad-relación: Es un tipo de diagrama de flujo que ilustra cómo las “entidades”, como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Los diagramas WE se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación.

Modelo relacional: Utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único. Se escogió el modelo relacional para el diseño de la base de datos.

Las tablas pertenecientes a la base de datos diseñada, así como su relación se muestra en la Figura 3.

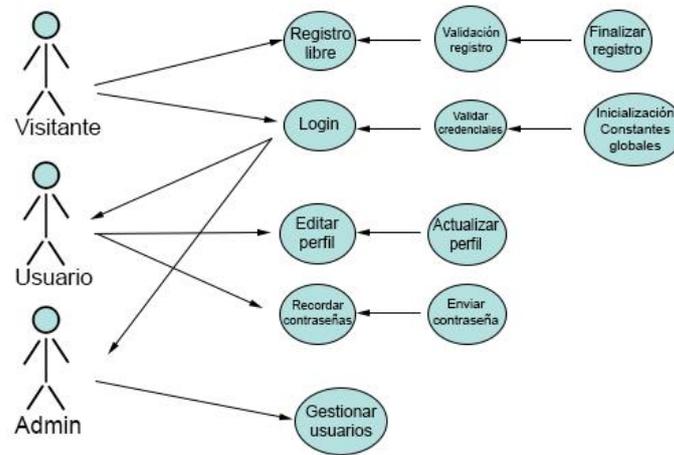


Fig. 4. Diagrama de jerarquías de usuarios.



Fig. 5. Interfaz web.

El modelo relacional presentado corresponde al modelo desarrollado para este proyecto. Como se puede observar, se presentan diversas tablas, donde se almacena información específica como en la tabla “usuarios”, “sensores” y “estaciones”; asimismo se muestra las tablas de control que tienen la finalidad de facilitar la interacción entre dos tablas para así generar una relación entre las tablas y almacenar la información referente al acceso de cada usuario, de esta forma, es posible controlar que puede y que no puede verse y/o modificar un usuario en particular, manteniendo así el control en la interfaz.

Inicio de Sesion

Usuario:

Contraseña:

No soy un robot 
reCAPTCHA
Privacidad - Condiciones

Fig. 6. Inicio de sesión.

Vista... Período... Recargar

Tablas Últimas 24 horas

Buscar

Fecha	Temperatura	Humedad
2020-08-07 11:00:00	24	65
2020-08-06 11:00:00	55	12
2020-08-05 10:58:10	12	1
2020-08-04 10:58:10	12	1
2020-08-03 11:08:10	12	11
2020-08-02 11:08:10	12	11
2020-08-01 10:58:10	28	10
2020-07-31 10:58:10	28	10
2020-07-30 11:00:58	22	15
2020-07-29 11:00:10	19.1	22
2020-07-28 11:00:01	21.5	33
2020-07-27 11:00:10	23	33

Fig. 7. Tablas de datos.

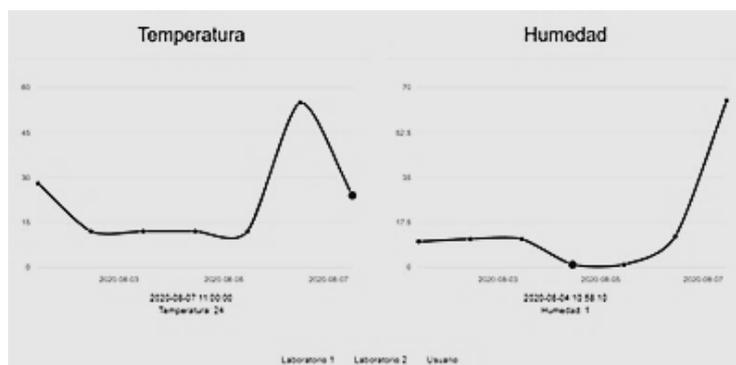


Fig. 8. Gráficas de datos.

Esta base de datos se encarga de almacenar todos los datos tanto de usuarios como de los sensores, actuadores, historiales de muestreo y de accionamiento; posteriormente, como se puede apreciar en el siguiente segmento, esta base de datos interactúa con una página web ya montada en un dominio.

Adicional a las bases de datos diseñadas y desarrolladas, es necesario también generar las jerarquías de usuarios, además de los métodos que son posibles realizar en cada una de su asignación.

4.2. Diseño del sitio web

Para el diseño de la página web se trabajó con los lenguajes de programación: HTML, PHP y JavaScript. Inicialmente se realizaron pruebas en servidor local dentro de la Raspberry Pi y, posteriormente, se trasladó a un dominio web que cuenta con su propio servidor. La navegación del sitio web fue diseñada para ser intuitiva y fácil de usar por el usuario. Contando con cuatro botones que redireccionan al servicio o información requerida: Inicio (para regresar a la página principal), Servicios, Galería y Contacto; así mismo se presentan datos generales. Ver Figura 4. El sitio web desarrollado puede consultarse en la siguiente url: <http://iot-udg.000webhostapp.com/>.

Para iniciar sesión se utiliza un formulario (Figura 5). Al iniciar sesión se personaliza a cada usuario, además, permite otorgar privilegios de administrador a ciertas cuentas. Esto se logra mediante una consulta a la base de datos mediante una “query” (consulta) que se crea a partir del ingreso del formulario hecho por el usuario. Esto mediante el método POST para así proteger la información del usuario, esto implica mayor seguridad en el manejo de información. Una vez iniciada la sesión, al usuario se le otorgan permisos y acceso a las distintas secciones del sitio web.

En la navegación como usuario registrado y con privilegios, es posible acceder a la página principal de este proyecto la cual es la encargada de mostrar los datos que se obtiene del hardware.

Como podemos apreciar en la Figura 6 y 7, se agrega un indicador, el cual presenta el espacio físico que se está monitoreando. En este caso tenemos el laboratorio 1. Asimismo, se presenta un par de menús desplegables encargados de controlar el tipo de vista y el periodo a mostrar. El tipo de vista se refiere a las dos posibles maneras de recuperar los datos de las tablas de muestras en la base de datos que hace referencia al historial de muestras realizados, para poder verlas como gráficas o tablas; mientras que el menú de periodo corresponde a un filtro que permite recuperar los datos de las últimas 24 horas, los últimos 7 días o el último mes.

4.3. Diseño electrónico

Haciendo uso del software de simulación Fritzing y Proteus; se presenta la conexión para cada uno de los sensores y el GPS.

En términos generales, el software Fritzing se utilizó con el propósito de ilustrar, de una manera sencilla, las conexiones entre los componentes del sistema; así como conseguir una simulación más rápida para el prototipado del mismo. Los diagramas hechos en Fritzing resultan ser más fáciles de entender e interpretar no solo para los expertos, sino para los usuarios en general.

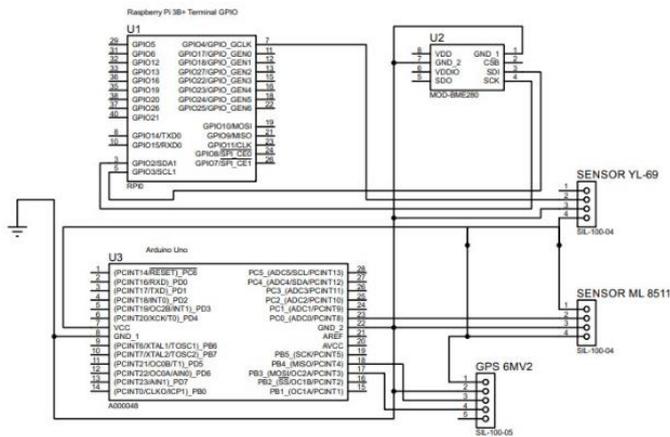


Fig. 9. Diagrama Proteus.

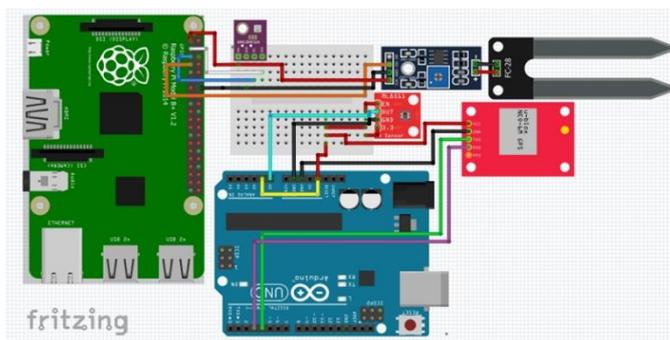


Fig. 10. Diagrama Fritzing.

Por su parte, Proteus nos ofrece un entorno más profesional y detallado del proyecto. Para el diseño, Proteus nos ofrece un entorno más profesional y detallado del proyecto. Para el diseño, se utilizaron un total de 9 piezas independientes y 3 preensambles para lograr el ensamble completo de esta estación. Cada elemento del sistema cuenta con una carcasa externa para protección contra el ambiente, por lo que tanto los sensores como la propia Raspberry Pi estarán protegidos.

5. Resultados

Esta estación de monitoreo remoto desarrollada es un proyecto multidisciplinar que cumple con la función de mostrar información en tiempo real de datos (inicialmente) atmosféricos recolectados y procesados por una tarjeta Raspberry Pi 3B+ con ayuda de un Arduino UNO. Se diseñó una página web montada en su propio dominio con características responsivas, misma que permite la interacción con el usuario y el acceso a la información recolectada por los sensores. Dicho sitio web también permite otorgar privilegios de administrador a usuarios específicos y seleccionar el laboratorio (estación

de monitoreo) al que se quiera acceder; permitiéndonos medir variables en más de un solo lugar. Toda la información proveniente del exterior es capaz de ser organizada y proceda gracias a una base de datos relacional. Por lo que los alcances de este proyecto son bastos.

6. Conclusiones

Debido a la naturaleza multidisciplinaria de este proyecto y al auge de estas nuevas tecnologías de intercomunicación, los alcances de este se amplían a muchos más sectores de la economía (como la ganadería y agricultura), sectores en donde se puede mejorar la calidad de la producción y mejorar la experiencia de los trabajadores y clientes. Sin embargo, debe considerarse el estudio posterior con el fin de visualizar el diseño, desarrollo del comportamiento y tendencias del mercado. Las herramientas y tecnologías de hardware y software “Open Source” utilizados en este sistema de monitoreo remoto demostraron su alta capacidad, tanto en alcance como en trabajos previos realizados. Estas herramientas muestran se alta capacidad, tanto en alcance como en trabajos previos realizados. Estas herramientas muestran un alto potencial de escalabilidad y crecimiento futuro en trabajos de IoT e Industria 4.0.

Referencias

1. Morteo, F., Bocalandro, N.: Un enfoque práctico de SQL. Ediciones cooperativas (2004)
2. Knowlton, J.: Python. Fernández-Vélez, M.J. (ed.), Anaya Multimedia-Anaya Interactiva (2009)
3. Marteli, A.: Python. Guía de referencia. Gorjón-Salvador, B. (ed.) Anaya Multimedia-Anaya Interactiva (2007)
4. Merlo, C.A., Antonio, M., Silva, R., Sosa, C.Y., Marciano, M.: Sistemas mecánicos subactuados pendulares, Boletín UPIITA, 42(30) (2014)
5. Block, D.J.: Mechanical design and control of the pendubot. University of Illinois (1996)
6. Andrade-Santamaría, R.G., Fuentes-Castillo, R.C.: Análisis, diseño y construcción del Pendubot. Escuela Politécnica Nacional de Ecuador (2000)
7. Cortés, C., Landeta, J., Chacón, J.: Implicaciones y perspectivas futuras. El entorno de la industria 4.0. Dialnet (2020)
8. Html: Lenguaje de etiquetas de hipertexto. Documentación web de MDN (2020)
9. PHP: ¿Qué puede hacer PHP? - Manual. Php.net (2020)
10. Hernández, L.: ¿soy?, Arquitectura IoT, prototipando los dispositivos del futuro. Programar fácil con Arduino (2020)
11. BBC - dot.Rory: A 15 pound computer to inspire young programmers. Bbc.co.uk (2020)
12. Palacios, P., Córdova, A.: Approximation and Temperature Control System via an Actuator and a Cloud: An Application Base don the IoT for Smart Houses. In: International Conference on eDemocracy and eGovernment (ICEDEG), pp. 241–245 (2018)

13. Danita, M., Blessy, M., Nithila, S., Namrata, S., Paul, J.: IoT based Automated Greenhouse Monitoring System. In: Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1933–1937 (2018)
14. Knöring, A., Wettach, R., Cohen, J.: Fritzing – A tool for advancing electronic prototyping for designers. In: Proceedings of the Third International Conference on Tangible and Embedded Interaction (TEI'09) (2009)
15. Silberschatz, A., Korth, H., Sudarshan, S.: Fundamentos de bases de datos. McGraw-Hill/Interamericana de España, S.A.U. (2002)