Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Diseño e implementación de un dispositivo de Internet de las Cosas para el monitoreo del medio ambiente vía Wi-Fi

Design and implementation of a device for monitoring the environment connected to the Internet via Wi-Fi

Domínguez-García, R.O.[a], Aguilar-Molina, Y.[a], Rumbo-Morales, J.Y.[a], Sorcia-Vázquez, F.[a], Ortiz-Torres, G.[a], González-Dueñas, M.[b]

[a] Departamento de Ciencias Computacionales e Ingenierías; Centro Universitario de los Valles de la Universidad de Guadalajara; C.P. 46600, Ameca, Jalisco.
[b] Departamento de Ciencias Sociales y Humanidades; Centro Universitario de los Valles de la Universidad de Guadalajara; C.P. 46600, Ameca, Jalisco.
odomi@academicos.udg.mx; yehoshua.aguilar@academicos.udg.mx; jesse.rumbo@academicos.udg.mx; felipe.sorcia@academicos.udg.mx; gerardo.ortiztorres@academicos.udg.mx; miriam.gduenas@academicos.udg.mx

Innovación tecnológica: Herramientas tecnológicas para monitorear el medio ambiente.

Área de aplicación industrial: Edificios, Invernaderos, Aulas, Naves industriales.

Recibido: 07 mayo 2020
Aceptado: 19 enero 2021

Abstract

The environment needs more attention due to the phenomena happening with climate change, so this topic is of great importance today. The monitoring of the environment, using the technologies that involve the Internet of Things (IoT), generates a large amount of information that is obtained through wireless sensor technology, allowing the design of support systems for low-cost decisions.

This work presents a monitoring system that provides information on environmental conditions at the local level, through the implementation of a device for low-cost environmental monitoring and rapid implementation. Using wireless sensor network technology which is composed of a Raspberry Pi_3B + card using Wi-Fi and an Arduino Nano card, the first under Raspbian v14.3, using Python 3.6, and the second using C / C ++ for Arduino.

The contribution of this research is the use and combination of both systems, which significantly increase the monitoring, control, storage and data transmission capabilities; each card collects the

information from the different digital or analog sensors, respectively, saving it in the Raspberry Pi SQLite database and subsequently transmitting it through an Application Programming Interface, to the remote database programmed in MySql, where it is permanently stored with the aim of being able to generate graphic web or mobile device interfaces for monitoring and controlling the IoT device.

Keywords: IoT, Raspberry Pi, Arduino.

Resumen

El medio ambiente necesita mayor atención debido a los fenómenos que se presentan con el cambio climático, por lo que este tema es de gran importancia. El monitoreo del medio ambiente aprovechando las tecnologías que involucran el Internet de las Cosas, generan gran cantidad de información que se obtienen mediante la tecnología de sensores inalámbricos, lo que permite diseñar sistemas de soporte a las decisiones de bajo costo.

Este trabajo presenta un sistema de monitoreo que brinda información sobre las condiciones ambientales a nivel local, a través de la implementación de un dispositivo para el monitoreo ambiental de bajo costo y rápida implementación. Usando la tecnología de redes de sensores inalámbricos el cual está compuesto de una tarjeta Raspberry Pi_3B+ usando Wi-Fi y una tarjeta Arduino Nano, la primera bajo Raspbian v14.3, usando Python 3.6, y la segunda usando C/C++ para Arduino.

La aportación de esta investigación es la utilización y combinación de ambos sistemas, el cual incrementan de manera significativa las capacidades de monitoreo, control, almacenamiento y transmisión de datos; cada tarjeta recaba la información de los diferentes sensores de tipo digital o analógico respectivamente, guardándola en la base de datos SQLite de la Raspberry Pi y posteriormente transmitiéndola a través de una Interface de Programación de Aplicaciones, a la base de datos remota programada en MySql, donde se almacena de forma permanente con el objetivo de poder generar interfaces gráficas web o de dispositivos móviles para el monitoreo y control del dispositivo IoT.

Palabras Clave: IoT, Raspberry Pi, Arduino.

1. Introducción

El Internet de las cosas (IoT, por sus siglas en inglés) habilitada por las tecnologías de Red de Sensores Inalámbricos (WSN, por sus siglas en inglés) abarca muchas áreas de la vida moderna. Este ofrece la capacidad de medir, inferir y comprender indicadores ambientales, desde ecologías y recursos naturales para entornos urbanos y rurales, hasta instalaciones tecnológicas en las grandes industrias [1].

La proliferación de estos dispositivos en una red de comunicación-activa impulsa el Internet de la Cosas, en el que los sensores y los actuadores se combinan perfectamente con el entorno que los rodea, generando información que se comparte entre plataformas [1]. Así mismo, la información compartida implica el manejo y

manipulación de grandes cantidades de datos que deben ser almacenados en bases de datos, procesados y presentados de forma fluida, eficiente y de fácil interpretación [1]. Las palabras “Internet de las cosas” (IC) significan una red mundial interconectada basada en sensores, comunicación, redes, y tecnologías de procesamiento de información, las cuales son la nueva versión de las Tecnologías de la Información y Comunicación (TICs) [2]. Con base a esto, diversas aplicaciones son planteadas a partir de los protocolos de comunicación entre redes y la medición de variables físicas, como ejemplo de aplicaciones se encuentra el monitoreo de las condiciones ambientales para diversos usos y la toma de decisiones [3], [4], [5].

Actualmente los sistemas que monitorean las variables del medio ambiente, tales como: temperatura, humedad relativa, luz ultravioleta, dióxido de carbono, velocidad del viento, dirección del viento, precipitación pluvial, luminosidad, entre otras; son aplicadas para monitorear estas variables en la agricultura protegida o en invernaderos [3], recolectar información climatológica-ambiental de microclimas, para la toma de decisiones en sistemas de riego e informar a la población de condiciones del clima tales como: viento, calidad del aire en interiores [4], o calidad del aire en aulas [5].

El estudio de los sistemas de monitoreo ambiental conjuntamente con el IC, es un campo que presenta problemas que deben tomarse en cuenta para solucionarlos.

Diversos trabajos publicados combinan las ventajas de las plataformas utilizadas para el desarrollo del sistema de monitoreo ambiental, propuesto en este artículo. En la literatura se encontraron algunas implementaciones que combinan estas dos plataformas es decir Raspberry pi y Arduino nano, para la implementación de sistemas de monitoreo ambiental [6], [7], [8] y [9].

Se han propuesto sistemas para monitoreo del ambiente como se describe en [6], el cual trata sobre el avance tecnológico en el monitoreo del medio ambiente y toma ventaja de las características de la observación del entorno actual. [6] Entre ellos, un sistema que involucra un Arduino, Raspberry Pi 3, XBee y su protocolo Zigbee y Adafruit IO[1] con buena eficiencia y logrando un bajo consumo de energía. Otro sistema de monitoreo está descrito en [7], donde se propone un sistema de supervisión que brinda información sobre las condiciones ambientales en un nivel local, utilizando sensores inalámbricos conectados a una Raspberry Pi, un Arduino Nano, XBee con su protocolo Zigbee, formando una red de sensores inalámbricos (WSN). La recopilación de los datos obtenidos por estos sensores se realiza en la Raspberry Pi y los resultados se visualizan en una interfaz gráfica de usuario (IGU). Un ejemplo más de aplicación de esta tecnología está en [8], que presenta un sistema para el monitoreo de una granja de cangrejos de río utilizando sensores inalámbricos para medir la temperatura y los niveles de agua. Esta aplicación proporciona información sobre la granja tales como temperatura adecuada, fecha en que se debe cambiar el agua, tiempos de alimentación y cantidad de alimentos [8]. En [9] se presenta un sistema de sensores de bajo costo, eficiente en energía y confiable mediante el uso de una arquitectura de clúster simplificado, que involucra un Arduino, y una Raspberry Pi quienes monitorean las condiciones de temperatura, humedad, intensidad de luz, nivel de concentración de gas GLP en una

[1] Adafruit Industries es una empresa de hardware de código abierto con sede en la ciudad de Nueva York. Fue fundada por Limor Fried en 2005.

habitación y nivel de agua (nivel de inundación).
Este trabajo tiene como objetivo el diseño y la implementación de un dispositivo para el monitoreo del medio ambiente basado en la interconexión de una tarjeta electrónica Raspberry Pi 3B+ (Pi) y una Arduino Nano, aplicando el Internet de las Cosas, mediante un sistema local de monitoreo ambiental, para aplicaciones en domótica e inmótica, de bajo costo, uso inmediato y confiable.

2. Materiales y Equipos

En este trabajo se utilizan conjuntamente dispositivos de hardware y de software. A continuación, se describen cada uno de ellos.

A. *El hardware*

Raspberry Pi 3B+ (Pi) utilizada. La Pi es una computadora de un tamaño no mayor al de una tarjeta de crédito, pero con cualidades muy significativas que la hacen un buen candidato para desarrollo de sistemas [10], la cual se muestra en la Figura 1. El Modelo 3B+ tiene un procesador Broadcom BCM2837B0 de cuatro núcleos a 1.4 GHz. Además, tiene conectividad inalámbrica de banda dual de 2.4 GHz y 5 GHz y Gigabit Ethernet a través de USB 2.0, que admite hasta 300Mbps. El conector de 40 pines de Entradas y Salidas de Propósito General (GPIO, por sus siglas en inglés) y su espaciado de 0.1" en la Pi le permite contar con 27 GPIO, UART, I2C, SPI, así como a fuentes de 3.3 y 5 V. La Raspberry Pi cuenta con 1 GB de memoria RAM y la memoria requerida para el almacenamiento es proporcionada por una tarjeta SD [11].



Figura 1. Raspberry Pi 3B+ [10].

Sensor DHT22. El DHT22. Es un sensor digital de temperatura y humedad relativa de bajo costo. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, utilizando una señal digital en el pin de datos [12], tal como se observa en la Figura 2.

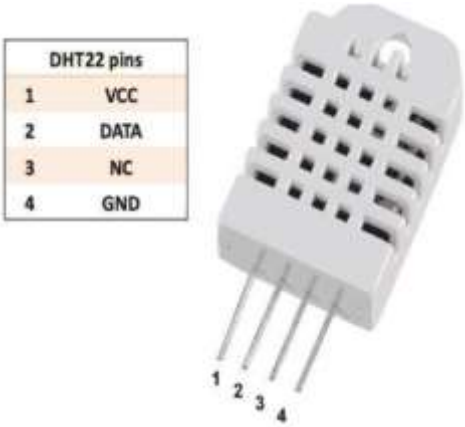


Figura 2. Sensor DHT22.

Sensor MG-81. El MG811. Es un sensor de dióxido de carbono (CO_2) basado en el principio de funcionamiento de células de electrolito sólido. Este suministra un voltaje inversamente proporcional a la concentración de CO_2 del ambiente. La tensión de salida se genera por el contacto de las moléculas de dióxido de carbono con un dispositivo de alta temperatura en el interior del sensor La salida del módulo es de tipo analógico [13], tal como se observa en la Figura 3.



Figura 3. Sensor dióxido de carbono MG-811.

Sensor UVM30A. Sensor para detectar el índice de intensidad ultravioleta (UV). La radiación electromagnética que emite este sensor, tiene longitudes de onda de: 200 ~ 370 nm, más cortas que las longitudes de onda del espectro visible que es de entre 0.4 y 0.7 micrones. Este sensor se muestra en la Figura 4, cuya salida del módulo, es de tipo analógico [14].



Figura 4. Sensor de Ultravioleta UVM30A.

Sensor YL-69 y módulo YL-38. Detecta la humedad del suelo. Su funcionamiento se basa en 2 electrodos que emiten corriente eléctrica a través del suelo, cuya resistencia entre los electrodos depende del contenido de humedad del suelo, para un suelo muy húmedo se tiene una resistencia muy baja (cortocircuito) y para un suelo muy seco la resistencia será muy alta [15]. Este sensor se observa en la Figura 5.

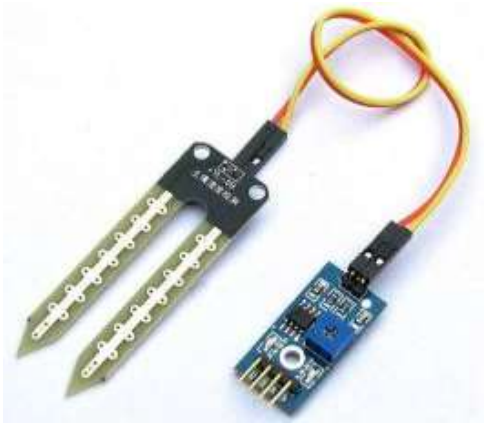


Figura 5. Sensor de humedad del suelo YL-69 y módulo YL-38.

Módulo sensor de luz digital BH1750. Este sensor permite realizar mediciones de flujo lumínico a través del bus I2C. Posee un conversor interno de 16-bits, tiene una salida digital y su desempeño es mejor al de un Foto-Resistor, ya que no es necesario realizar conversiones de voltaje para obtener datos interpretables [16], (Figura 6).

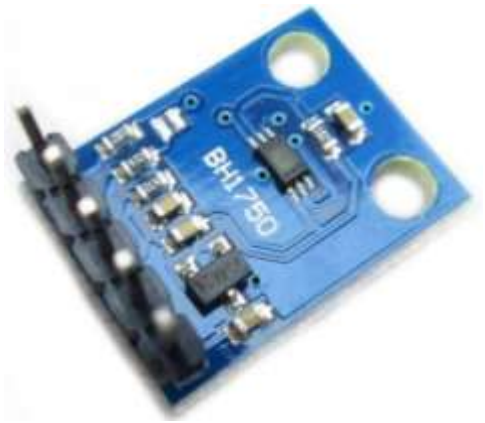


Figura 6. Sensor de Luz digital BH1750.

Tarjeta Arduino Nano (Nano). Es una pequeña tarjeta basada en el microcontrolador (µC) ATmega328, opera a corriente directa de 5V y memoria flash 32 KB de las cuales 2 KB son usados para el bootloader, SRAM de 2 KB, EEPROM 1 KB, velocidad de reloj 16 Mhz, 8 entradas analógicas, 22 Entradas/Salidas digitales, 6 de las cuales son PWM [17], (Figura 7). Los

datos recopilados por los sensores conectados a esta tarjeta se transmiten a la estación base a través de una comunicación serial vía el cable USB Mini-B [18].



Figura 7. Tarjeta Arduino Nano.

B. *El software*

Para construir el sistema propuesto se utilizó Raspbian [19] como el sistema operativo (SO) principal. Para realizar la programación de la Nano y la Pi se utilizaron los lenguajes de programación Python y C/C++ [20], [21]. Para la gestión de la base de datos, se utilizó SQLite [22].

3. Metodología

Para desarrollar el nodo WSN, se utilizaron una serie de componentes de hardware y software, además del diseño de su arquitectura. La estructura del sistema propuesto se muestra en la Figura 8. Este sistema tiene como base el procesamiento realizado por la tarjeta Raspberry PI, bajo el SO Raspbian, así como SQLite para lograr el almacenamiento de los datos y la comunicación Wi-Fi a través de una puerta de enlace (*Gateway)* que hace posible él acceso al exterior. Los puertos generales de entradas y salidas (GPIO's, por sus siglas en inglés) son programados mediante lenguaje Python. La tarjeta Nano tiene como principal función la expansión de puertos de E/S digitales y una nueva fuente de puertos de E/S de tipo analógico. La arquitectura del sistema que se muestra en la Figura 8, es la combinación de Raspberry Pi, Arduino Nano y sus sensores para fines de monitoreo. En esta arquitectura, la integración de la plataforma de hardware y software se obtiene combinando el programa en Python (NodoWSN.py) que integra la base de datos bajo SQLite, la lectura de sensores digítales, la lectura de los sensores analógicos vía la Nano a través del puerto USB y la comunicación al Internet vía Wi-Fi. Mientras que el programa "SensoresAnalogicos.ino" en C/C++ para Arduino, adquiere los datos detectados por los sensores analógicos, y son transmitidos a la Pi en intervalos de 6 segundos.

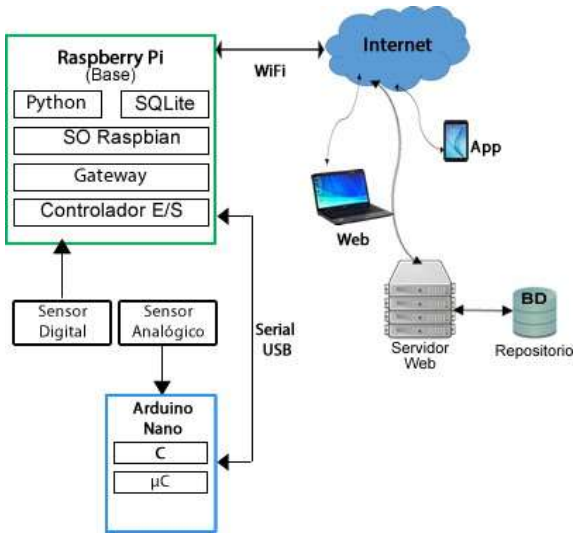


Figura 8. Arquitectura general del sistema utilizado.

La plataforma de Internet permite que el usuario acceda de forma remota al sistema o simplemente supervisarlo vía el comando de Linux *ssh*, accediendo de forma segura al sistema. Todo esto se logra debido al sistema operativo que está instalado en la Raspberry Pi, Raspbian.

A. Diseñando el nodo

Se desarrolló el nodo WSN, utilizando las plataformas de desarrollo Raspberry Pi y Arduino Nano, mediante Wi-Fi. El diagrama de bloques de la estación base y los componentes del sistema se muestran en la Figura 9.

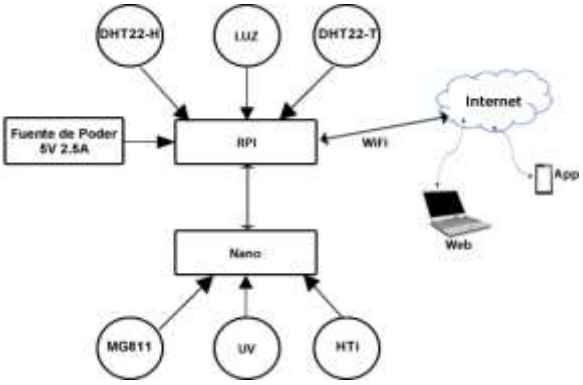


Figura 9. Esquema general del diseño del Nodo.

Los sensores con salida digital están conectados a la estación base (Pi), y los sensores con salida analógica están conectados a la tarjeta Nano. El sensor digital DHT22 monitorea la temperatura ambiental y la humedad relativa, mientras que el sensor digital de luz BH1750 monitorea el nivel de iluminación. El sensor MG811 monitorea los niveles de CO_2, en tanto que el sensor UVM30A determina la intensidad de los rayos UV y el Sensor YL-69 determina el nivel de humedad del suelo. Estos últimos son sensores analógicos.

4. Configuraciones Experimentales y Resultados

Para el desarrollo de esta aplicación, se configuró la Pi 3B+ bajo el SO Raspbian. El uso del software NOOBS es la forma más fácil de instalar Raspbian en la memoria SD [23]. Teniendo el dispositivo Pi listo para ser utilizado, se conecto el DHT22 al GPIO (17) y el sensor BH1750 al GPIO (02) y (03), A continuación, vía la Nano se conectó el MG811 al puerto A0, el sensor UVM30A al puerto analógico A2 y por último el sensor YL-69 al puerto A3, como se observa en la Figura 10.

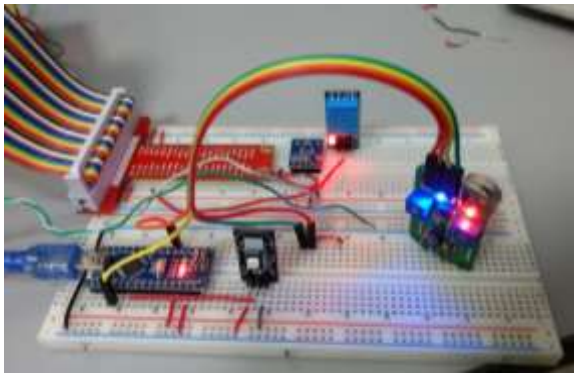


Figura 10. Sensores vía Pi y Nano.

El nodo WSN como se observa en la Figura 11, donde la Pi es la estación base de donde se alimenta todo el sistema y se trasmite la información a la red y la Nano aportando sus puertos analógicos y su bajo consumo.

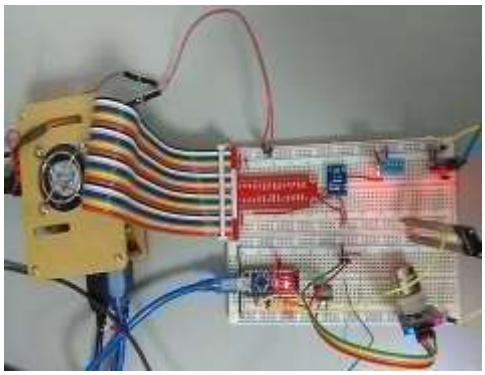


Figura 11. Prototipo del nodo WSN.

Además, se implementaron los algoritmos para leer los sensores antes mencionados (DHT22, BH1750, MG811, UVM30A, YL-69), se inició con los sensores analógicos. En el IDE de Arduino se creó el programa "SensoresAnalogicos.ino", ver anexo A, para leer los sensores, UVM30A de Radiación Ultravioletas (UV), cuya longitud de onda va: 200~370 nm, muestreado en once partes el índice UV. MG811 de CO_2, que mide la cantidad de CO_2 en partes por millón, de 400 a 10000 ppm; y el de humedad del suelo YL-69, en un rango de 1023 a 0, que inicia en desconectado o fuera del suelo, a seco, húmedo y completamente en el agua. La acción del sensor MG811 se puede observar considerando la gráfica de

operación del sensor, la cual se observa en la Figura 12.

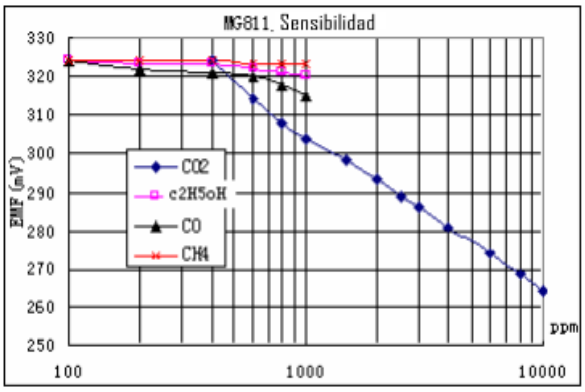


Figura 12. Lectura del Sensor MG811-EMF, bajo diferentes concentraciones de gases [24].

Se realizaron pruebas para el acondicionamiento de los y sensores de temperatura, humedad, UV y Lumens. Las lecturas obtenidas por los sensores del sistema (DHT22, BH1750, UVM30A) se muestran Figura 13, los cuales fueron comparados con las lecturas de un Luxómetro Digital LUX LT-YK10LX, un medidor portátil de humedad/temperatura. OMEGA RH820 y un medidor de rayos UV modelo 850009, ver Figura 14 y Tablas 1 y 2.

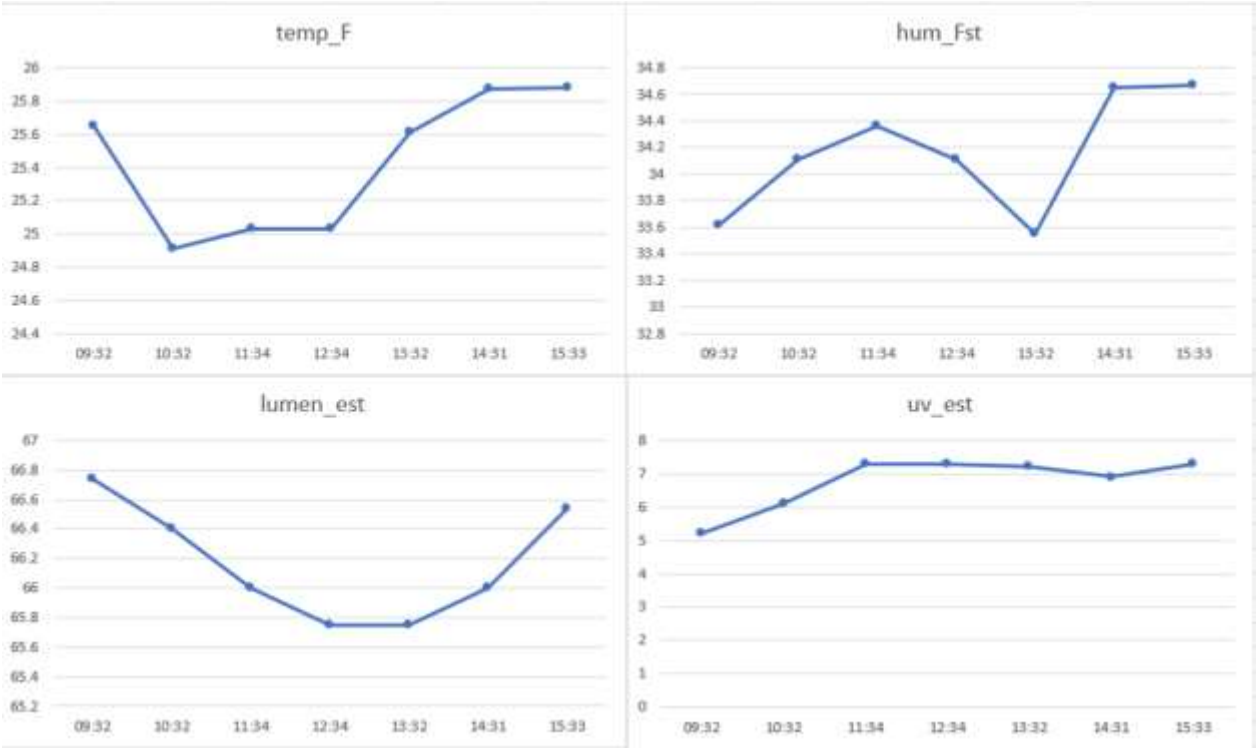


Figura 13. Gráficas de datos que corresponden a variables obtenidas por sensores del sistema.

Tabla 1. Valores de cinco variables obtenidos por los sensores de la estación en diferentes horarios de toma.

Temp_est	Hum_est	CO_2_est	lumen_est	uv_est	Fecha_toma	Hora_toma
25.65	33.62	440	66.74	5.2	11/01/2020	9:32
24.91	34.11	415	66.4	6.1	11/01/2020	10:32
25.03	34.36	423	66	7.3	11/01/2020	11:34
25.03	34.11	430	65.75	7.3	11/01/2020	12:34
25.61	33.55	433	65.75	7.2	11/01/2020	13:32
25.87	34.65	433	66	6.9	11/01/2020	14:31
25.88	34.67	440	66.54	7.3	11/01/2020	15:33

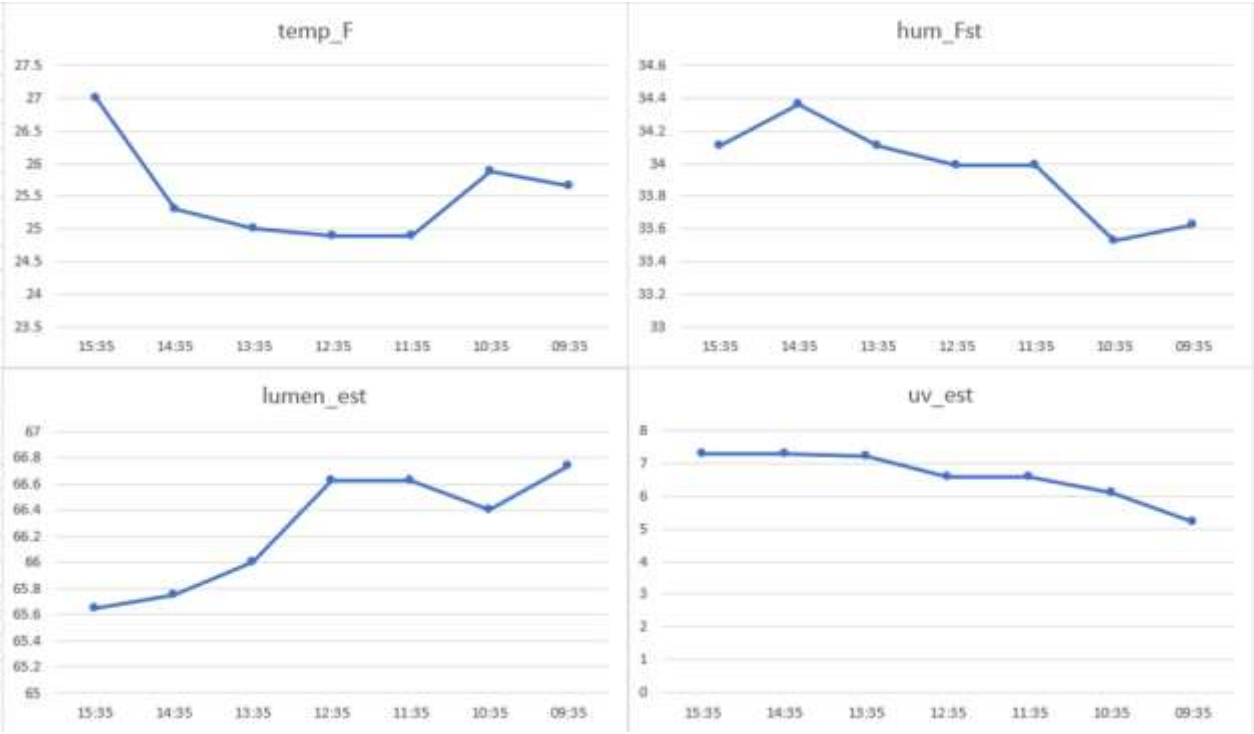


Figura 14. Gráficas de los datos obtenidos por Fluke 91.

Tabla 2. Valores de cinco variables obtenidos por medidor industrial. En diferentes horarios de toma.

Temp_F	Hum_F	CO_2_F	Lumen_F	UV_F	Fecha_toma	Hora_toma
27.0	34.11	3.18	65.65	7.3	11/01/2020	15:35
25.3	34.36	3.18	65.75	7.3	11/01/2020	14:35
25.01	34.11	3.08	66	7.2	11/01/2020	13:35
24.9	33.99	3	66.63	6.6	11/01/2020	12:35
24.9	33.99	3	66.63	6.6	11/01/2020	11:35
25.87	33.53	3.08	66.4	6.1	11/01/2020	10:35
25.65	33.62	3.03	66.74	5.2	11/01/2020	09:35

Esquema del flujo de datos hacia las bases de datos (BD)

En la Figura 15 se muestra la ruta que siguen los datos colectados por los sensores para ser almacenados primeramente en la BD local (SQLite); y de forma más permanente en la base de datos remota ejecutando (MySQL).

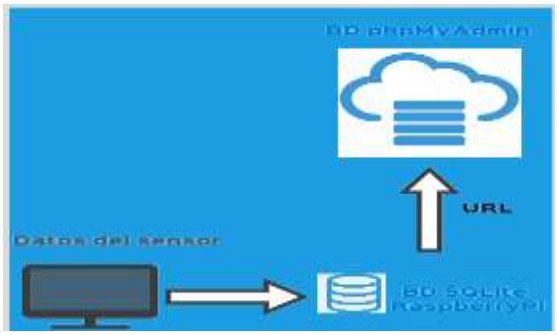


Figura 15. Conexión de las BD.

La BD *nodo*, compuesta de una tabla llamada *sensoresDA*, integrada por los

registros *Fecha_nod, HT_nod, CO2_nod, Luz_nod, UV_nod, TR_nod y HR_nod,* cuyo esquema se observa en la Figura 14. El código de creación de la BD *nodo* se muestra en el Anexo B. La cual almacena los datos captados por los sensores en la Pi.

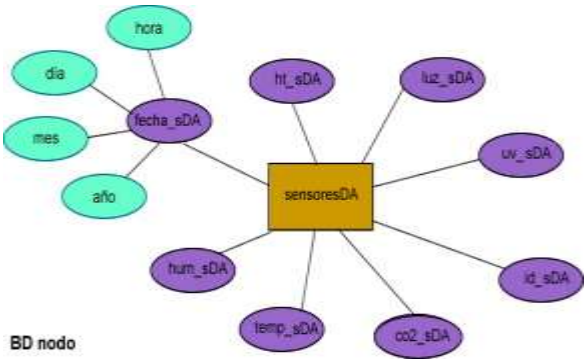


Figura 14. Esquema de la base de datos.

La lectura de los sensores de Luz o lúmenes, Humedad relativa y Temperatura ambiental, se pueden observar más abajo en la Figura 15, los datos son los esperados con un margen de error de +/- 2% comparados con dispositivos de fábrica.

Para el envío e inserción de los valores recolectados por los sensores a la BD remoto (*invernaderos)*, se codificó un algoritmo (ver el Anexo C), bajo los siguientes requerimientos: a) primero se registra la lectura de las diferentes variables de forma secuencial, un segundo por cada uno y estos valores son almacenados en la base de datos local (*nodo)* ejecutando SQLite, b) los datos son enviados a la BD remota (*invernaderos)*. Cada vez que se realiza una inserción a la base de datos se registra la hora y fecha en que ocurre cada suceso. Esto a través del programa *NodoWSN.py* que es el algoritmo que corre en la Pi y es el programa que lee los sensores digitales, se comunica con la Nano para obtener la lectura de los sensores analógicos, despliega los resultados en la consola, y además guarda los datos en la base local SQLite: "nodo.sqlite", como se observa en la Figura 15.

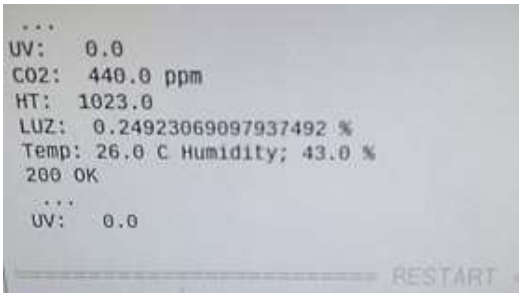


Figura 15. Datos desplegados en la consola de Python.

Los datos son enviados a la base de datos remota "*invernadero"* a través de una API desarrollada en MySQL, en donde los resultados son guardados en forma definitiva para su futuro procesamiento y uso. La base de datos "*invernadero*" está constituida por cinco tablas: *estación, estacióng, migrations, password_resets y users*. La tabla *estación* almacena los datos de identificación de la estación, *estacióng* almacena los registros de los diferentes sensores con los que se equipó la estación, *migrations* es la responsable de llevar un registro histórico de los cambios realizados a la base de datos "*invernadero*", *password_resets* es la responsable de brindar un mecanismo de seguridad para el cambio de contraseña para el usuario administrador y *users* es la tabla que almacena los usuarios registrados en la estación.

Resultados

La interconexión de dos plataformas electrónicas para el desarrollo de sistemas como la raspberry Pi 3b+ y la Arduino Nano conformando una nueva plataforma, vislumbran lo siguiente: la fácil integración de una serie de sensores digitales y/o analógicos de forma práctica, logrando integrar un dispositivo capaz de detectar diferentes variables del entorno, sin necesidad de circuitos extras evita tener un host para la base de datos local y agregar hardware para la transmisión de los datos.

Una ventaja del sistema que integra Pi y Nano es que el sistema es dinámico, ya que puede ser modificado fácilmente, porque se puede eliminar o agregar nuevos sensores directamente, modificando y/o agregando código nuevo. A su vez, se logran un mayor número de puertos de propósito general (GPIO) al sumar ambas tarjetas. La disponibilidad de una base de datos local hace a esta plataforma poderosa en el almacenamiento de datos. Así como también en la transmisión de la información vía Wi-Fi. Este tipo de prototipado (unión de Pi + Nano) permiten el desarrollo de dispositivos de una forma más rápida y económica.

Los resultados son guardados en la base de datos remota "invernadero" implementada en MySQL para su futuro procesamiento y uso. Se puede observar en la BD los datos que se almacenan: (temperatura, humedad, CO_2, humedad del suelo, UV), acompañada de sus respectivas fechas de lectura.

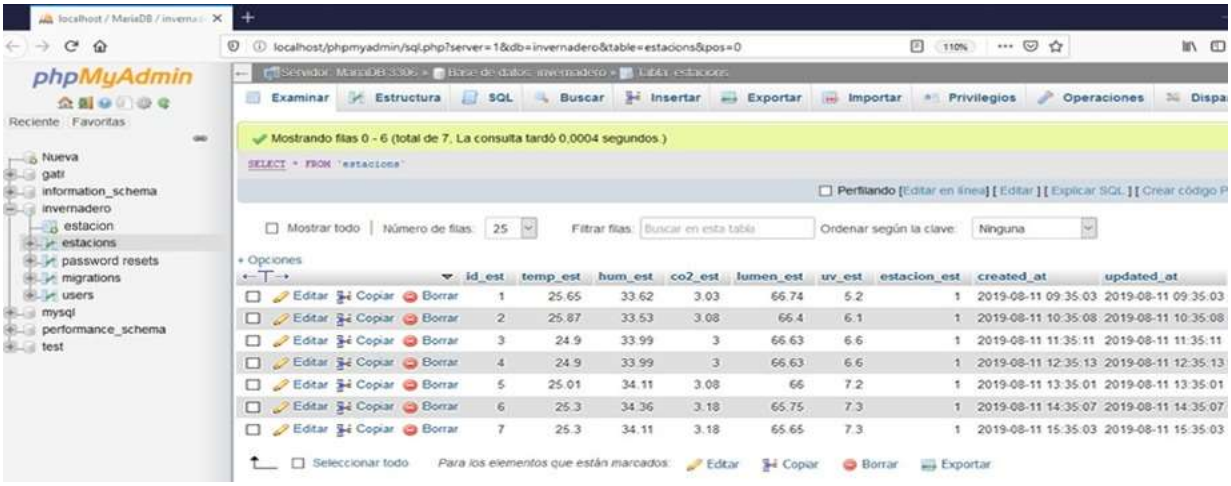


Figura 16. Datos almacenados en la base de datos remota.

5. Conclusiones

El desarrollo de sistemas es más rápido, comparado con sistemas basados en microcontroladores donde se tiene que implementar toda o gran parte de los circuitos necesarios para su comunicación con el mundo externo como un PIC 16F877A, 18F4550. Esto permite que se puedan desarrollar nuevos dispositivos de manera fácil, rápida y económica con base en el esquema de integración Pi+Nano.

El presente prototipo con características de un sistema dinámico, es fácil de modificar, y puede ser ampliado y disminuido. La unión de ambas plataformas crea una nueva plataforma más potente y práctica, ya que ambas se compensan. La Nano proporciona puertos Analógicos que la Pi no presenta, ampliando el número de puertos de la nueva plataforma, que es más pequeña y de menor consumo, pero carece de un sistema operativo y de poca memoria ram. La Pi proporciona un ambiente de desarrollo que incluye una base de datos (SQlite) facilitando el almacenamiento de los datos proporcionados por los sensores; además, cuenta con una potente red de transmisión Ethernet, Wireless y bluetooth, facilitando otras alternativas de transmisión de datos, se incrementan de manera significativa las capacidades con el esquema (Pi+Nano). Los GPIO de ambas tarjetas trabajan con diferentes niveles de voltaje 3.3VCD para Pi y 5VCD para Nano, permiten diferentes configuraciones.

Agradecimientos

Los autores desean agradecer al Centro Universitario de los Valles de la Universidad de Guadalajara, en específico a la Maestría en Ingeniería de Software, por el financiamiento y facilidad en el uso de sus laboratorios.

6. Referencias

[1] J. Gubbi, R. Buyyab, S. Marusic y M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", *Future Generation Computer Systems*, vol 29, p. 1645–1660, Septiembre 2013, doi: 10.1016/ j.future.2013.01.010.

[2] S. Li, L. Da Xu y S. Zhao, "The Internet of Things: a Survey", *Springer Science+Business Media* New York, Published online, Abril 2014, doi: 10.1007/s10796-014-9492-7.

[3] M. Ibrahim, A. Elgamri, S. Babiker y A. Mohamed, "Internet of Things Based Smart Environmental Monitoring Using the Raspberry-Pi Computer", in *Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*. IEEE, 2015. doi: 10.1109/ICDIPC.2015.7323023.

[4] V. Rao y K. V. Prema, "Internet-of-Things Based Smart Temperature Monitoring System", in *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2018, IEEE. doi: 10.1109/RTEICT42901.2018.9012652.

[5] S. N. Ramos, S. Arruda y D. Samara, "Concentration of Carbon Dioxide in UFCG Classroom, Artificially Climatized*",* in *Congresso Técnico Científico da Engenharia e da Agronomia – CONTECC'2018*, 21 a 24 de agosto de 2018 – Maceió-AL, Brazil.

[6] S. R. Patil y Sriyanka, "Smart Environmental Monitoring Through Internet of Things (IoT) Using RaspberryPi 3", in *International Conference on Current Trends in Computer, Electrical, Electronics and Communication*, 2017, DOI:10.1109/CTCEEC.2017.8455056.

[7] A. D. Deshmukh y U. B. Shinde, "A Low-Cost Environment Monitoring System Using Raspberry Pi and Arduino with Zigbee", pp. 1-6, IEEE Xplore, 2016, doi: 10.1109/INVENTIVE.2016.7830096.

[8] C. Vongchumyen, P. Jiarawongsakul, P. Sombutdee, K. Rueangpoon y K. Tongprasert, "Semi-Automatic Crayfish Farm System", in *International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, Phuket, 2018, pp. 1-4. doi: 10.1109/ICEAST.2018.8434485.

[9] A. Setiawan, S. J. Sahat, L. B. Subekti, Y. Bandung and K. Mutijarsa, "Design and Implementation of IoT Sensor System for Home Power Manager and Environment Condition

Monitor", in *International Symposium on Electronics and Smart Devices (ISESD)*, Bandung, 2018, pp. 1-6, doi: 10.1109/ISESD.2018.8605464.

[10] G. Halfacree, *THE OFFICIAL Raspberry Pi, Beginner's Guide How to use your new computer. Raspberry Pi Trading Ltd*, 2018, Station Road, Cambridge.

[11] DATASHEET, "Raspberry Pi_3 Model B+", 2019. [En línea]. Disponible: https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM_3p0.pdf. [Accedido: 27-oct-2019].

[12] Electronilab, "Sensor DHT22", 2020. [En línea]. Disponible: https://electronilab.co/tienda/sensor-de-temperatura-y-humedad-dht11/. [Accedido: 27-ene-2020].

[13] A. de Paz Menéndez, "Desarrollo de un sistema de sensores para la detección de sustancias peligrosas", Proyecto fin de carrera, Universidad Autónoma de Madrid, Escuela Politécnica Superior, 2015.

[14] Rambal, automatización y robótica, "Sensor Ultravioleta (UV) UVM30A", 2020. [En línea]. Disponible: https://rambal.com/color-luz-forma/192-sensor-ultravioleta-uv-uvm30a.html. [Accedido: 28-ene-2020].

[15] Electrónicos CALDAS, "YL69", 2020. [En línea], Disponible: https://www.electronicoscaldas.com/sensores-de-humedad-lluvia-inundacion/461-sensor-de-humedad-en-suelo-yl-69.html. [Accedido: 29-ene-2020].

[16] NayLamo Mechatronics, "Sensor de Luz digital BH1750", 2020. [En línea]. Disponible: https://naylampmechatronics.com/sensores-luz-y-sonido/76-modulo-sensor-de-luz-digital-bh1750.html. [Accedido: 29-ene-2020].

[17] Arduino, "Arduino Nano", 2019. [En línea]. Disponible: https://store.arduino.cc/usa/arduino-nano. [Accedido: 27-nov-2019].

[18] A. Kurniawan, *Arduino Nano A Hands-On Guide for Beginner*. PE Press, 2019.

[19] W. Harrington, *Learning Raspbian, Get up and Running with Raspbian and Make the Most out of your Rasperry Pi.* Packt Publishing, Open Source, Birmingham, 2015.

[20] C. R. Severance, *Python for Everybody, Exploring Data Using Python 3.* Editorial: Elliott Hauser, 2015.

[21] S. Monk. *Programming Arduino, Getting Started with Sketches,* 2nd. ed. McGraw Hill Professional, 2016, ISBN-13:978-1259641633.

[22] J. A. Kreibich, *Using SQLite*. Publisher: O'Reilly Media, 2010, ISBN: 9780596521189.

[23] NOOBS, "Offline and network install", 2019. [En línea]. Disponible: https://www.raspberrypi.org/downloads/noobs/. [Accedido: 27-nov-2019].

[24] ALLDATASHEET.COM, "MG811 Datasheet (PDF)", 2019. [En línea]. Disponible: http://www.alldatasheet.com/datasheet-

pdf/pdf/172004/ETC2/MG811.html. [Accedido: 27-nov-2019].

[25] Sandbox Electronics, “MG-811 CO2 Sensor Module”, 2019. [En línea]. Disponible: https://sandboxelectronics.com/?p=147. /. [Accedido: 27-nov-2019].

Anexo A. Código parcial en C/C++ de Arduino, lectura de los sensores analógicos [25]

// SensoresAnalogicos.ino

```
//#define DC_GAIN            (8.5) //define the DC gain of amplifier
// Measure model.
#define LOG_400_VOLTAGE (0.30) // Voltage when the concentration of CO2 is 400ppm.
#define LOG_1000_VOLTAGE (0.25) // Voltage when the concentration of CO2 is 1000ppm.
float co2_log_slope = (LOG_1000_VOLTAGE - LOG_400_VOLTAGE) / (LOG_1000 - LOG_400);
float v400 = LOG_400_VOLTAGE - co2_log_slope * LOG_400;
float uv, valor, med,volts;
int percentage;

int PHt = A3;
int UV = A2;
int SensorHT = 0;
int SensorUV = 0;
void setup() {
  Serial.begin(9600);
  pinMode(MG811_DIGITAL_PIN, INPUT);
  digitalWrite(MG811_DIGITAL_PIN, HIGH);
  pinMode(VENT_PIN, OUTPUT);
  digitalWrite(VENT_PIN, HIGH);
}
void loop() {
  float volts = MG811ReportRaw();
  MG822ReportPPM(volts);
  MG822ReportDigital();
  Serial.print("\n");
  SensorHT = analogRead(PHt);
  fun_htierra(SensorHT);
SensorUV = analogRead(UV);
Fun_UV(SensorUV);
  delay(2000);}
```

Anexo B. Creación de la DB local.

```
conn= sqlite3.connect ('nodo.sqlite')
cur= conn.cursor()
cur.execute('DROP TABLE IF EXISTS sensoresDA)
cur.execute('''CREATE TABLE sensoresDA(
            Id INTEGER PRIMARY KEY AUTOICREMENT, Fecha DATETIME,
             TemperaturaReal FLOAT, HumedadRela FLOAT, Humedad_Tierra FLOAT,
            CO2 FLOAT, Luz FLOAT, UV FLOAT)''')
conn.commit()
```

Anexo C. Código en Python para la lectura de los sensores e inserción de los valores captados por los sensores a la base de datos remota.

```
# NodoWSN.py

import serial
from time import sleep
from dhtxx import DHT22
dht22 =DHT22(4)
def convertToNumber(data):
   return ((data[1] + (256 * data[0])) / 1.2)
 def readLight(addr=DEVICE):
 return convertToNumber(data)

while True:
   #Elemento 1
   txt = arduino.readline()
   txt = float(txt)
   print ("UV:  ",txt)
   sleep(1)
   #Elemento 2
   txt2 = arduino.readline()
   txt2 = float(txt2)
   print ("CO2: ",txt2, "ppm")
   sleep(1)
   #Elemento 3
   ht = arduino.readline()
   ht1 = float(ht)
   print("HT: ", ht1)
   sleep(1)
   #Elemento 4
   lu= float(readLight())
   luz= (lu/65535)*100
   print ("LUZ: ",luz, "%")
   sleep(1)
   #Elemento 5
   r = dht11.get_result(max_tries=10) #intentos a realizar para obtener resultados validos
   if type(r[0]) == 'NoneType' or type(r[1]) == 'NoneType':
      r[0] = -1
      r[1] = -1
   print('Temp: {0:0.1f} C Humidity; {1:0.1f} %'.format(r[0], r[1]))
   sleep(1)
   fecha= datetime.datetime.now()
   valores=[(fecha),(r[0]),(r[1]),(txt2),(luz),(txt)]
   cur.execute("INSERT INTO sensoresDA
```

```
VALUES (NULL,?,?,?,?,?,?)",valores)
   conn.commit()
      #Almacenamiento remoto
   try:
      r = requests.post ("http://10.225.41.152/invernadero/api/estacion/?", {
         'hum':format(r[1],'.2f'),
         'temp':str(r[0]),
         "c02":str(txt2),
         "lumen":format(luz,'.2f'),
         "uv":str(txt),
         "estacion":2
         } )
      print (r.status_code, r.reason)
      print (r.text[:500],'...')
   except:
      print("Error al enviar los datos al servidor remoto")
   conn.close()
arduino.close()
```