TELEMÁTICA
REVISTA DIGITAL de las Tecnologías de la Información y las Telecomunicaciones
Revista Telemática. Vol. 17. No. 1, enero-abril, 2018, p.42- 55 ISSN 1729-3804

Monitoreo ambiental en tiempo real acrítico mediante red inalámbrica de sensores

Heykel Hernández Hernández[1], Ramsés Zaldívar Estrada[1], Ismael González Metauten[1], Guillermo Ayran Torres Lores[2]

[1]e-mail: mailto:heykelh33@gmail.com

[1]Instituto de Geofísica y Astronomía, Calle 212, No. 2906 e/ 29 y 31, La Lisa, La Habana, [2]First Due, 107 7th St, Garden City, New York

RESUMEN

Este artículo presenta un prototipo para la monitorización ambiental en tiempo real acrítico en zonas urbanas e industriales, utilizando una red inalámbrica de sensores y compuesto por sistemas embebidos de plataformas de hardware y software abierto. El hardware consiste en Arduino, Raspberry Pi, módulos de Xbee y sensores de temperatura y humedad. El software se desarrolla sobre GNU/Linux, usando Python, C++ y frameworks como Django y AngularJS. Los datos son almacenados en una base de datos MySQL y disponible en tiempo real acrítico a través de la aplicación web empleando la tecnología WebSocket, accesible mediante WiFi. Para comprobar la funcionalidad del sistema, se realiza la monitorización del empleo de la CPU/RAM en la estación base y un muestreo de la temperatura y la humedad de un local del Instituto de Geofísica y Astronomía. Los resultados obtenidos se comparan con los de una estación meteorológica. Las pruebas realizadas señalan que el prototipo podría ser de gran utilidad para la monitorización de variables ambientales en Cuba.

PALABRAS CLAVE: Red Inalámbrica de Sensores, WebSocket, Sistemas Embebidos, Monitoreo Ambiental en Tiempo Real Acrítico.

ACRITICAL REAL-TIME ENVIRONMENTAL MONITORING THROUGH WIRELESS SENSOR NETWORK

ABSTRACT

The current paper presents a prototype for acritical real-time environmental monitoring in urban and industrial areas, using a wireless sensor network and comprised of embedded systems of open hardware and software platforms. The hardware consists of Arduino, Raspberry Pi, Xbee modules and temperature and humidity sensors. The software is developed on GNU/Linux, using Python, C ++ and frameworks such as Django and AngularJS. Data is stored in a MySQL database, reachable in real-time through the web application using WebSocket technology, accessible via WiFi. To check the functionality of the system, it is performed a CPU/RAM monitoring in the base station and a sampling of the

temperature and humidity of a local at the Geophysics and Astronomy Institute. Results obtained are compared with those of a weather station. Tests carried out indicate that the prototype could be very useful for the monitoring of environmental variables in Cuba.

KEYWORDS: Wireless Network of Sensors, WebSocket, Embedded Systems, acritical real-time environmental monitoring.

INTRODUCCIÓN

La protección del medio ambiente ha sido uno de los principales objetivos de Cuba desde hace muchos años. La actual política ambiental cubana se cumple mediante una gestión integral de los instrumentos de la gestión, que constituyen las herramientas para enfrentar los problemas ambientales [1]. Una de estas herramientas lo constituye el "Sistema de Información Ambiental", cuyo objetivo esencial es garantizar la información requerida para el conocimiento, evaluación y toma de decisiones relativas al medio ambiente a decisores y sociedad en general. Para ello, el sistema se subdivide en los "Indicadores de Medio Ambiente para el Desarrollo Sostenible" y el "Sistema Nacional de Monitoreo Ambiental", este último, encargado de monitorizar y evaluar el comportamiento de un conjunto de variables susceptibles de ser observadas y evaluadas, en función de ofrecer una visión comparativa del estado del entorno ya sea a nivel nacional o territorial en un período de tiempo determinado [2]. Además, la monitorización aporta información que resulta relevante para el funcionamiento de otros instrumentos de la gestión ambiental como son: evaluación de impacto ambiental, sistema de inspección ambiental estatal, la investigación científica y la innovación tecnológica [1], [3].

En materia ambiental, las variables presentan un comportamiento dinámico, territorial y complejo. Esto hace necesario su medición in-situ por un período necesariamente prolongado de tiempo, para poder reflejar verídicamente su comportamiento. Sin embargo, actualmente existen dificultades para cumplir el planteamiento anterior, ya que el equipamiento es altamente costoso, existen carencias de recursos y los métodos de medición existentes están basados en la transmisión de la información por cable, con la dificultad de su emplazamiento [2]. Con mucha frecuencia se aprecia que los valores de las variables en los estudios han sido medidas durante períodos cortos, reportadas mediante registradores de datos portátiles o estaciones de monitorización lejanas al lugar de estudio. En estos casos, no se refleja adecuadamente el comportamiento real del medio ambiente [4], [5].

Para ello, se requiere del desarrollo de sistemas capaces de monitorizar, registrar y controlar de manera precisa las variables, con altas tasas de muestreos en el lugar que se está analizando [6], lo cual hoy en día ya es posible gracias al desarrollo de la comunicación inalámbrica, tecnología de sensores y sistemas embebidos. Las redes inalámbricas de sensores (RIS), están siendo usadas para la monitorización ambiental y han demostrado ser más viables y económicas que los métodos tradicionales [6]-[10]. La RIS se basan en dispositivos de bajo costo, capacidad reducida de procesos, memoria limitada y bajo consumo de energía [5]. Estos pueden ser alimentados por batería o energía solar fotovoltaica y distribuidos en un área en número de cientos o miles, a los cuales se conectan alámbricamente sensores

homogéneos o heterogéneos y módulos de comunicación inalámbrica (nodos de sensores). Estos sensores son capaces de obtener información del medio observado en el área de cobertura de la red, procesarla, y comunicarla a través de enlaces inalámbricos hasta un nodo central de coordinación (estación base). El nodo central es encargado de procesar la información, almacenarla y mostrar los resultados a los decisores para poder evaluar así, el estado del entorno [11].

Existen varios ejemplos de aplicaciones medioambientales de las RIS, como la vigilancia de la calidad del aire [12], [13], detección de derrames en los sistemas fluviales, calidad del agua en cuerpos de agua para uso recreativos y detección de fuego en bosques, [14], [15]. El empleo de las RIS tiene tres ventajas significativas: (1) No es necesario la instalación de redes de cables, la red inalámbrica sólo se despliega una vez por lo que el impacto creado por el hombre es mínimo; (2) es posible una mayor densidad de puntos de muestreos y los datos tiene una alta precisión; (3) los nodos de sensores permiten el monitoreo remoto sin supervisión [15]. Con ello, las RIS pueden medir en tiempo real una gran variedad de variables asociadas a zonas urbanas e industriales, sin detrimento de la calidad de la información y publicarlas en sitios web en tiempo real, aliviando así al ser humano del trabajo duro de la recolección, manejo y transporte de muestras para su análisis [16].

Las aplicaciones de monitorización que han tenido un rápido crecimiento son, el agrícola, de hábitat, de invernaderos, climático y de bosques. Schubert et al., [9], construyeron un sistema de monitorización agrícola en tiempo real de cultivos protegidos. El monitoreo del hábitat es una de las partes esenciales del control ambiental, para asegurar que las especies sean autónomas y evitar cualquier perturbación ecológica. En este sentido, S. Bing desarrolla un sistema de monitoreo en tiempo real de un hábitat no protegido [17]. R. Gao [18] desarrolla un sistema de monitorización de efecto invernadero usando la plataforma TinyOS para monitorear parámetros ambientales como temperatura, radiación y humedad. Por otro lado J. Lloret, et al. [19] propone un sistema de detección de incendios en bosques, que usa una red de área local inalámbrica (WLAN) junto con tecnología de nodo de sensores con cámaras basadas en IP, para detectar la presencia de incendios y proteger de esta manera los bosques, tan importantes para el equilibrio ecológico.

El trabajo presenta un prototipo para la monitorización ambiental en tiempo real acrítico en zonas urbanas e industriales, utilizando una red inalámbrica de sensores y plataformas de hardware y software abiertos. Los datos obtenidos son almacenados en una base de datos MySQL y disponible a través de la aplicación web en tiempo real acrítico empleando la tecnología WebSocket, accesible mediante WiFi. Para comprobar la funcionalidad del sistema se realiza la monitorización de la CPU/Memoria en la estación base y un muestreo de la temperatura y humedad relativa de un local del Instituto de Geofísica y Astronomía. Los resultados obtenidos se comparan con los de una estación meteorológica (utilizada como referencia). Las pruebas realizadas señalan que el prototipo podría ser de gran utilidad para el monitoreo de variables ambientales en Cuba.

DESCRIPCIÓN DEL SISTEMA

Arquitectura general del sistema

El sistema consiste en una RIS con topología de estrella, formados por varios nodos de sensores y una estación base que se comunican de manera inalámbrica. Los nodos de sensores están compuestos por un módulo de adquisición de datos (sensores), un módulo de procesamiento de datos (microcontroladores) y un módulo de comunicación inalámbrica. Los datos capturados por los sensores

son procesados y transmitidos por los nodos de sensores a la estación base la cual los procesa, almacena y los visualiza a través de una página web mediante una red cableada o WiFi.

La construcción de la RIS requiere el desarrollo e integración de varios componentes de hardware y software. La Figura 1 muestra la arquitectura general del sistema para el monitoreo ambiental que se ha desarrollado.

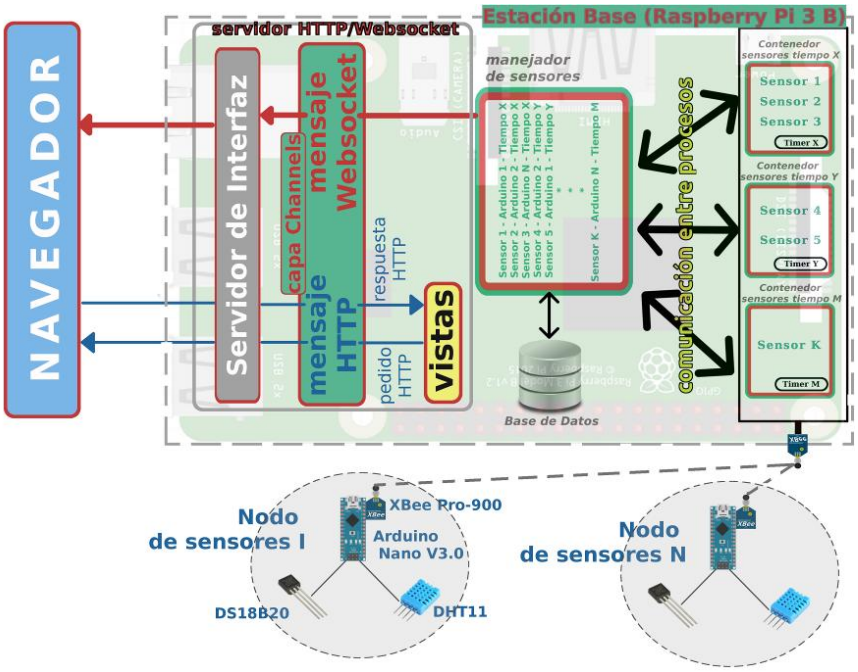


Figura 1: Arquitectura del sistema.

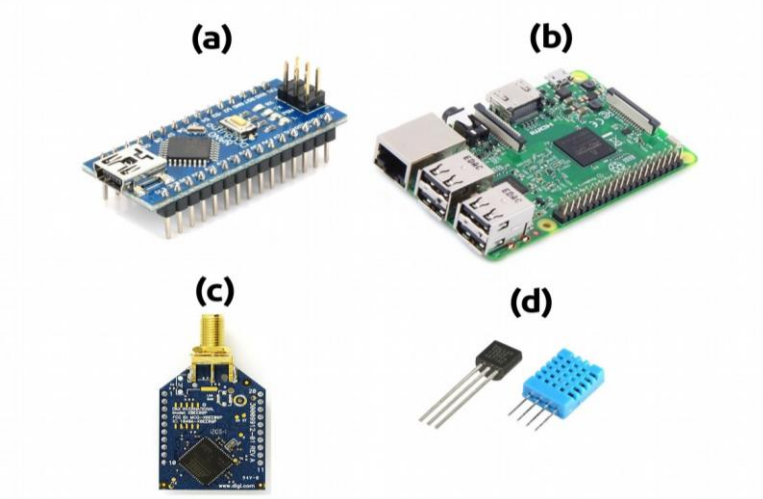
Desarrollo del hardware

Para la construcción de los nodos de sensores se utilizan módulos Arduino Nano V3.0 como microcontrolador, módulos Xbee-Pro 900 como tranceptor RF y sensores de temperatura one Wire DS18B20 y humedad DHT11. El Arduino Nano V3.0 es una plataforma de desarrollo de microcontroladores de placa única y hardware abierto, ampliamente usada por su fácil uso (Figura 2a). Se basa en un microprocesador Atmel Atmega328P de 8 bits y velocidad de reloj de 16MHz. Tiene 8 entradas analógicas y 14 pines de E/S digitales por lo que es posible conectar varios sensores a una sola placa del Arduino, 32 KB de memoria Flash de los cuales 2 KB es usado para el bootloader, 2 KB de SRAM y 1 KB de EEPROM.

Para la comunicación inalámbrica se utilizan módulos XBee-Pro 900 (Figura 2c), que operan en la banda ISM (Industrial, Scientific and Medical) de 900 MHz y que implementan el protocolo ZigBee [20], basados en el estándar IEEE 802.15.4, aplicado a redes inalámbricas de área personal con bajas tasas de transmisión de datos. El rango de comunicación interior del módulo es de 140 m, mientras que el

alcance exterior es aproximadamente 3 km bajo ciertas condiciones (la línea de visión entre los dispositivos no debe tener obstáculos y emplearse una antena de alta ganancia). Presenta bajos consumos de energía y velocidades de datos de hasta 156 kbps. Con estos dispositivos es posible construir desde una red simple en estrella hasta una compleja de malla [16], [20].

En este trabajo se han utilizado dos sensores con el fin de ilustrar el funcionamiento de la RIS (Figura 2d). Se utiliza un sensor de temperatura DS18B20 y un sensor de humedad DHT11. El DS18B20 es un termómetro digital de 9 a 12 bits de resolución que permite la conexión mediante un solo cable de datos con un microcontrolador. Este sensor se alimenta directamente de la línea de datos, eliminando la necesidad de una fuente de alimentación externa, tiene un rango de operación de -55 ºC a +125 ºC, sensibilidad de ±0.5 ºC en un rango de -10 ºC - 85 ºC [21]. El DHT11 es un sensor básico de humedad empleado en proyectos ambientales y cuenta con un sensor de capacidad para la humedad relativa [21], registra valores entre 20 y 80 % con una precisión de 5 %. Los componentes de hardware utilizados en el sistema se muestran en la Figura. 2.



(a) Arduino Nano v3.0; (b) Raspberry Pi 3; (c) Xbee-Pro 900; (d) sensor DS18B20 (negro), sensor DHT11 (azul)

Figura 2: Componentes de la Red Inalámbrica de Sensores

La estación base es una Raspberry Pi 3 modelo B V1.2 (RPi) configurada para trabajar de manera desatendida o headless. La RPi (Figura 2b) es un dispositivo embebido de placa reducida (SBC) del tamaño de una tarjeta de crédito (85 x 56 x 17 mm), de bajo costo y altas velocidades de procesamiento. Incluye un microprocesador (CPU) ARM Cortex-A53 de 64 bits y cuatro núcleos a 1,2 GHz, incluye conexión Wi-Fi de 802.11n y Bluetooth 4.1, coprocesador multimedia de doble núcleo Videocore IV®, Memoria LPDDR2 de 1 GB, 1x Puerto Ethernet 10/100, 1x Conector de vídeo/audio HDMI, 1x Conector 3.5mm audio/video compuesto, 4x Puertos USB 2.0, 40x Pines GPIO, Conector de pantalla DSI y ranura de tarjeta microSD [21].

El dispositivo es compatible con varios sistemas operativos, incluida una distribución de GNU/Linux basada en Debian (Raspbian Jessie 2016-05-27-raspbian-jessie-lite empleada en este proyecto). La RPi se puede conectar a una red de área local a través de un cable Ethernet o un adaptador WiFi USB, y acceder a ella a través del inicio de sesión pudiéndose configurar como una computadora de escritorio, o trabajar en remoto mediante el protocolo Secure Shell (SSH).

Lo más destacable de la RPi, además de su tamaño y precio, es su bajo consumo de energía, la posibilidad de uso de lenguajes de programación de alto nivel como Python y el conector GPIO de 40 pines configurables, exceptuando los de alimentación y tierra. Cuenta con pines para la conexión serie como UART, al cual se ha acoplado un dispositivo XBee-Pro 900, como módulo de comunicación inalámbrica con los nodos de sensores. También cuenta con puerto I2C y SPI y el resto de los pines son de entrada/salida, lo cual permite la comunicación con dispositivos externos y el accionamiento sobre diferentes sistemas [21], [22]. Existen varios ejemplos de aplicaciones científicas que emplean sensores y la RPi como unidad de procesamiento [10].

Desarrollo del software

El lenguaje de programación utilizado en cada nodo de sensores es C++. Este software es el encargado de manejar los datos de muestreo de los sensores de una manera bien definida y la comunicación, mientras que en la estación base se implementa una aplicación de manejador de nodos de sensores para la administración de los mismos. Este software se utiliza para controlar el comportamiento de los nodos distribuidos y capturar la información enviada en tiempo real acrítico o soft, la cual es una versión mucho menos restrictiva de un sistema en tiempo real hard y donde no se garantiza que la tarea de solicitud/captura de la información se complete en un plazo de tiempo determinado, como para ser considerado un error. No obstante, el sistema operativo intenta completar la tarea mediante la asignación de máxima prioridad durante su ejecución, lo cual es muy usual en sistemas de adquisición de datos.

Para el almacenamiento y administración de los datos se emplea un sistema de administración de bases de datos relacionales (RDBMS) MySQL. La visualización de los datos es a través de una aplicación web desarrollada con un servidor web HTTP Nginx, el cual es ligero, de alto rendimiento, de código abierto, de menor consumo de memoria RAM, multiplataforma y se integra mejor con el framework de desarrollo web utilizado. El server-side se desarrolla con el framework Django incorporando el paquete Channels, a fin de usar WebSocket, convirtiendo la aplicación web en tiempo real, mientras que la interfaz del lado del cliente se implementa usando HTML, CSS y bibliotecas de JavaScript como jQuery, AngularJS, D3.js y Bootstrap. Los clientes/administradores pueden interactuar con la aplicación web, a través de un dispositivo conectado a una red de área local (LAN) o desde cualquier terminal conectado a través de WiFi a la estación base.

Manejador de nodos de sensores

Para el manejo de la red inalámbrica de sensores se desarrolla un Script programado en Python 2.7 en la estación base. Python es un lenguaje de programación libre, que permite un desarrollo de software ágil, tiene una comunidad extensa, en continuo crecimiento y actualización. Además posee bibliotecas de cálculo de elevada potencia. La capa manejador de sensores es programada siguiendo el paradigma de

programación orientado a objetos (POO) y es la encargada del manejo de la comunicación multiproceso con los nodos de sensores conectados, manejar los intervalos de muestreo de todas las variables medidas por el sistema, almacenar los datos recolectados de los nodos y notificar en tiempo real acrítico a la interfaz web de la entrada de nuevos elementos.

La ejecución del Script instancia un proceso padre denominado manejador de sensores, que genera tantos procesos hijos como intervalos de medición se hayan establecidos para la captura del valor de cada sensor. Cada proceso hijo, denominado contenedor sensores de tiempo, se encarga de comunicarse con el nodo de sensores cuyo sensor tiene establecido su intervalo de medición y éste le transfiere el valor obtenido al proceso padre (Figura 3). Para la comunicación entre procesos se utiliza el paquete de Python Pyro4. Pyro es una biblioteca que permite la comunicación entre objetos a través de la red. Con ello, se separa la responsabilidad de manejo de la información de la visualización de la misma.

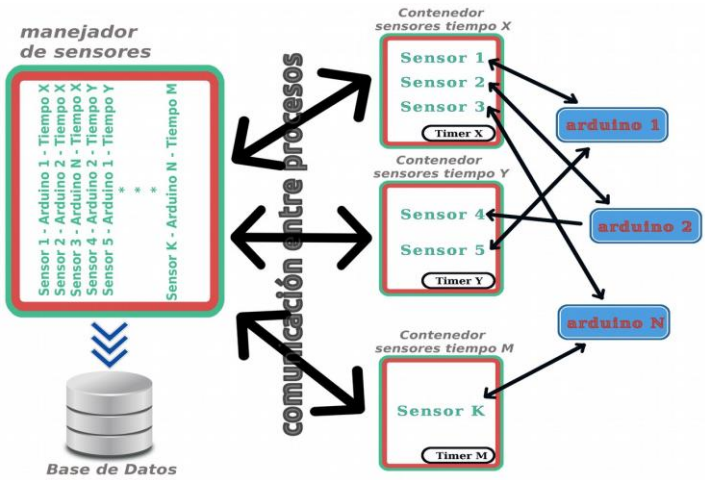


Figura 3: Arquitectura de la capa manejador de nodos de sensores.

Interfaz web

Para el desarrollo de la interfaz web se utiliza la plataforma de desarrollo Django y bibliotecas de JavaScript basadas en software libre. La Figura 4 muestra la página web desarrollada para la visualización de las mediciones en tiempo real.

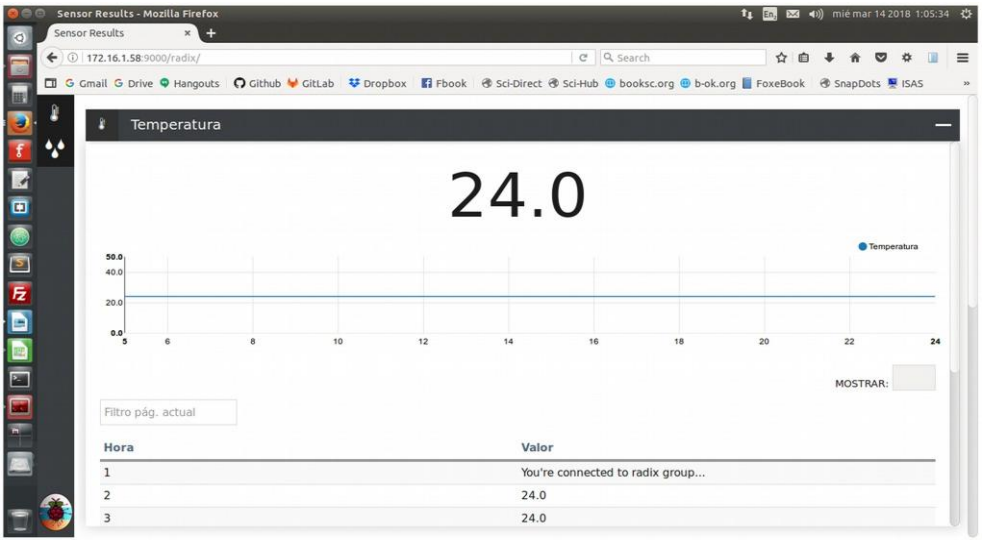


Figura 4: Página web del sistema para la visualización de las mediciones.

Django es un miembro importante de una nueva generación de frameworks web de código abierto y escrito en Python. Cuenta con una gran biblioteca de aplicaciones, por lo que permite incrementar la velocidad de desarrollo. Se basa en el patrón de diseño Modelo-Vista-Plantilla, basado a su vez en el Modelo-Vista-Controlador (MVC). Este define una forma de desarrollar el software, en la que el acceso a los datos (el modelo) está separado del pedido lógico de asignación de ruta (el controlador) y de la interfaz de usuario (la vista). Esto permite que los componentes tengan un loosely coupled entre sí, significando que cada pieza de la aplicación web tenga un único propósito clave, el cual puede ser modificado independientemente, sin afectar las otras piezas.

Un eslabón importante en el desarrollo de la aplicación web, lo constituye la visualización de los datos en tiempo real. Esto se logra con la implementación de WebSocket. Esta tecnología es un protocolo de comunicación que provee canales de comunicación bidireccional (Full-Duplex) sobre una conexión TCP. Esto permite la comunicación en dos sentidos distinto, a diferencia de lo que ocurre en el clásico modelo pedido/respuesta, siendo la interacción entre un navegador y un servidor web con gastos generales más bajos, a la vez que facilita la transferencia de datos en tiempo real desde y hacia el servidor [23], [24]. Para la implementación de WebSocket en Django se utiliza el paquete Channels [25]. Channels reemplaza en Django la base de pedidos y respuestas, por mensajes que son enviados a través de canales (Figura 1). Los clientes se suscriben a los canales y son notificados cuando sucede un evento.

El front-end se desarrolla utilizando código HTML, CSS y bibliotecas de JavaScript como jQuery, AngularJS, D3.js y Bootstrap. jQuery ofrece una serie de funcionalidades que permiten simplificar la manera de interactuar con los documentos HTML, manipular el Document Object Model(DOM) y manejar eventos. AngularJS es un framework de código abierto, basado en el patrón de diseño Modelo-Vista-Controlador (MVC), para el desarrollo de lo que se conoce como “Single Page Applications” en el front-end. Este framework permite la implementación de módulos que faciliten la adición de nuevas variables y sensores sin necesidad de adicionar código extra, permitiendo que sea una aplicación reusable y mantenible. D3.js es una biblioteca para producir gráficos dinámicos e interactivos en navegadores web a partir de datos. Bootstrap, es un framework cuya particularidad es la de

implementar la técnica de diseño responsive design o diseño adaptativo, con lo cual el sitio web se adapta automáticamente al dispositivo desde donde se visualiza la página web.

RESULTADOS EXPERIMENTALES

Monitorización de la CPU/RAM en la estación base

Para comprobar el desempeño del prototipo (Figura 5), se realiza un seguimiento del consumo de la CPU/RAM de los Scripts asociados al sistema en la estación base, durante 5 días ininterrumpidos, adquiriéndose valores de temperatura y humedad cada 5 segundos, a modo de estresar el sistema para detectar posibles fallos como falta de capacidad de procesamiento o memoria.

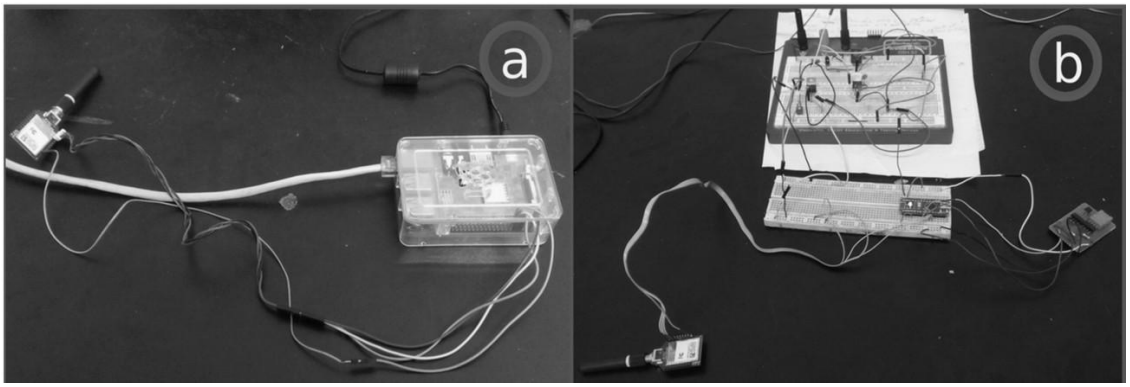


Figura 5: Funcionamiento de los dispositivos del sistema. (a)- Estación base Raspberry Pi 3 B y módulo de comunicación Xbee-Pro 900; (b)- Nodo de sensores con módulos Arduino Nano v3.0, sensores DS18B20 y DHT11 y módulo de comunicación Xbee-Pro 900.

Durante todo el tiempo que el sistema estuvo en funcionamiento, los valores de consumo de la CPU en cada núcleo fueron inferiores al 10 % mientras que el consumo de la memoria oscilaba entre 150 y 250 MB de 1GB disponible. El proceso manejador de sensores consume un 42/45 % de la CPU/RAM consumida, el servidor web 30/15 % de la CPU/RAM, y la capa de comunicación Pyro4 consume 7/4 % con respecto al total de CPU/RAM consumida por todo el sistema operativo (20 %/250 MB de CPU/RAM).

Muestreo de la temperatura y humedad de un local del Instituto de Geofísica y Astronomía

Se realiza un muestreo de las variables temperatura y humedad relativa en un local del centro de cálculo del Instituto de Geofísica y Astronomía (IGA). Los resultados obtenidos se comparan estadísticamente con los valores obtenidos por una estación meteorológica marca “La Crosse Technology WS-2310 Weather Pro Center” modelo WS-2310-11 [26], utilizada en la estación de observación radioastronómica de La Habana, ubicada en el IGA. Esta se empleada en este estudio como dispositivo de referencia. Las mediciones fueron realizadas con intervalo de medición de 1 hora, durante el periodo comprendido entre el 12/3/18 y 13/4/18, para un total de 153 mediciones realizadas durante 22 días de muestreo.

Las muestras obtenidas se analizan mediante estadística descriptiva, empleando estadígrafos de tendencia central y dispersión e inferencial para conocer si existen diferencias significativas entre los dos dispositivos (prototipo y referencia). Se emplea el software R para el análisis estadístico. La prueba de normalidad usando Shapiro-Wilk, para la temperatura y humedad tanto para el prototipo como la referencia, arroja que para las dos variables los valores de p-value fueron bajos o cercanos al nivel alfa (0,05), entonces se puede concluir que es rechazada la hipótesis nula, existiendo evidencias que los datos no se distribuyen normalmente para un intervalo de confianza de un 95 % (Tabla 1), de ahí la necesidad de utilizar métodos no paramétricos. Para ello, se realiza la prueba de los rangos con signo de Wilcoxon, para determinar si existen diferencias de las muestras de temperatura y humedad relativa entre los dispositivos.

Tabla 1: Resultados de la prueba de normalidad por Shapiro-Wilk

	Prototipo	Referencia
Temperatura	W = 0.94385, p-*value* = 8.624e-06	W = 0.95691, p-*value* = 0.0001091
Humedad relativa	W = 0.98285, p-*value* = 0.05414	W = 0.98312, p-*value* = 0.05815

El comportamiento de las variables temperatura y humedad relativa del sistema (prototipo) y la estación WS-2310-11 (referencia) en el local, se muestran en la Figura 6. Las dos variables indican un comportamiento similar en los dos sistemas. Los valores obtenidos de temperatura de referencia se encuentran dentro del valor del error del sensor DS18B20 igual a ± 0.5 oC, con excepción de 4 valores, donde el valor de temperatura en la referencia fue inferior al valor obtenido en el prototipo menos el error asociado al sensor. En el caso de la humedad relativa todos los valores obtenidos en la referencia se encuentran dentro del intervalo asociado al error suministrado por el fabricante e igual a ± 5 %

Los estadígrafos de tendencia central, indican similitud entre los dispositivos. El valor medio de temperatura en el prototipo fue de 27,69 oC, mientras que en la referencia fue de 27,70 oC, la mediana en los dos fue de 28 oC, mientras la moda fue 28,7 oC para el prototipo y 28,4 oC para la referencia. Por otro lado, la media para la humedad relativa en el prototipo fue de 50,87 % mientras que la referencia fue de 50,68 %, la mediana para los dos sistemas fue de 51,0 % y la moda de 50,0 %.

Los estadígrafos de dispersión se emplean para estudiar la variabilidad de las muestras. El rango de temperatura fue de 5,8 y 5,9 oC para el prototipo y referencia respectivamente, la desviación estándar de 1,50 oC para ambos casos, mientras que la varianza fue de 2,25 y 2,26 para el prototipo y referencia respectivamente. Lo anterior indica poca dispersión en la variable temperatura. La pendiente de un ajuste lineal para la temperatura del prototipo con respecto a la referencia fue de 0,992, lo que indica una buena correspondencia entre los dos dispositivos para esta variable (Figura 7a).

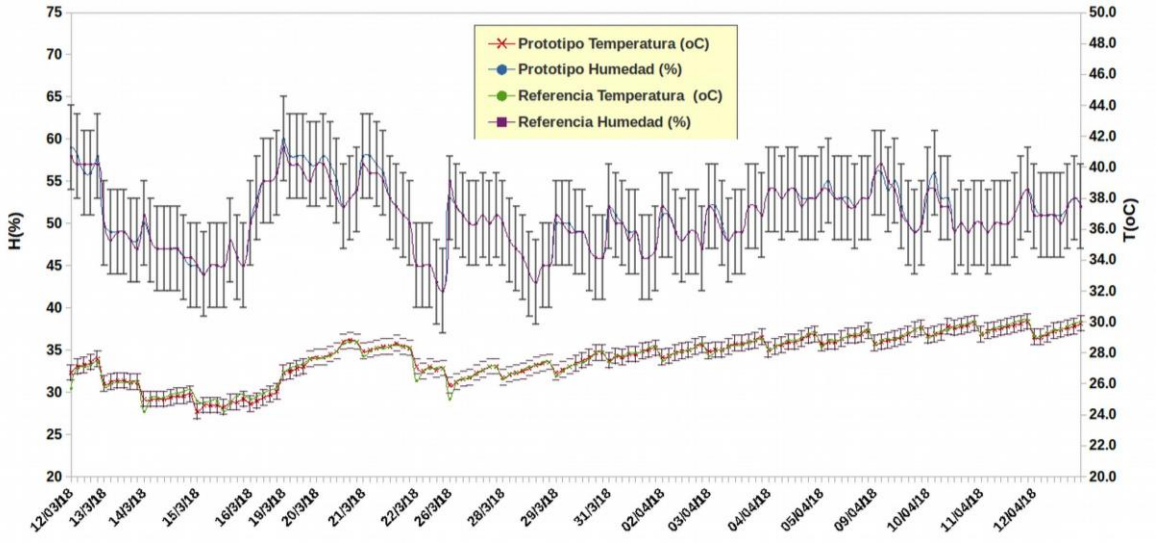


Figura 6: Comportamiento de la temperatura (T) y la humedad relativa (H) en el local

Para la humedad relativa, el rango fue de 18,0 % para el prototipo y 17,0 % para la referencia, la desviación estándar 3,85 y 3,65 % para el prototipo y referencia respectivamente, mientras que la varianza fue de 14,79 para el prototipo y de 13,31 para la referencia. Lo anterior indica poca dispersión en la variable humedad relativa. La pendiente de un ajuste lineal para la humedad relativa del prototipo con respecto a la referencia fue de 0,937, lo que indica una buena correspondencia entre los dos dispositivos para esta variable (Figura 7b).

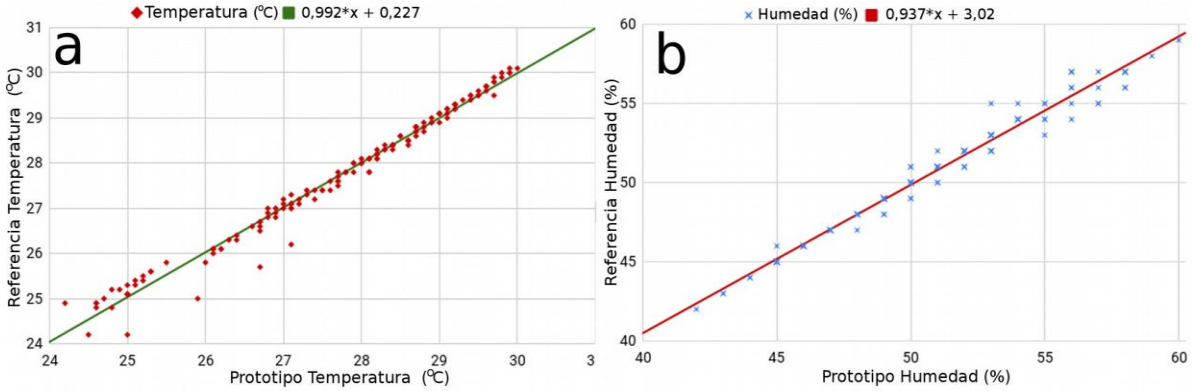


Figura 7: Dispersión de la temperatura y la humedad relativa. (a) Dispersión de la temperatura del prototipo con respecto a la referencia; (b) Dispersión de la humedad relativa del prototipo con respecto a la referencia.

Para comprobar la existencia de diferencias estadísticamente significativas entre los dos dispositivos, se realiza la prueba de los rangos con signo de Wilcoxon. Los resultados de la prueba de Wilcoxon se muestran en la Tabla 2.

Tabla 2: Resultados de la prueba de Wilcoxon para la temperatura y la humedad relativa

Temperatura	W = 11642, p-*value* = 0.9356
Humedad relativa	W = 11966, p-*value* = 0.7345

Dado que para las dos variables hay un p-value elevado y superiores al nivel alfa (0,05), se puede concluir que no se puede rechazar la hipótesis nula. O sea, tanto para la temperatura como para la humedad relativa del prototipo, con relación a la temperatura y humedad relativa en el dispositivo de referencia, no hay diferencias estadísticamente significativas, para un intervalo de confianza de un 95%.

CONCLUSIONES

El prototipo para la monitorización ambiental empleó una red inalámbrica de sensores con topología estrella, hardware basado en Arduino Nano v3.0, Raspberry Pi 3B, sensores de temperatura DS18B20 y humedad DHT11, módulos de comunicación inalámbrica XBee-Pro 900 y software en los nodos de sensores desarrollado en C++ y Python 2.7 en la estación base para la capa manejador de nodos de sensores e interfaz web en Django con el paquete Channels para implementar WebSocket. El prototipo demostró capacidad para medir en tiempo real acrítico variables ambientales de manera precisa y con altas tasas de muestreo, pudiendo dar solución al problema existente con las pocas mediciones realizadas en los estudios ambientales actuales, que no reflejan el real comportamiento del medio ambiente, por tener éste un carácter muy dinámico.

El desempeño del prototipo a través del seguimiento del consumo de la CPU/RAM de los Scripts, señalaron que el consumo de la CPU fue inferior al 10 % por núcleo, la RAM osciló entre 150 y 250 MB de 1 GB disponible y no existieron fallas por caídas de algún Scripts. Las pruebas estadísticas realizadas a las muestras obtenidas del muestreo de la temperatura y la humedad relativa en el prototipo y en la estación meteorológica empleada como referencia, indicaron poca dispersión. La pendiente de un ajuste lineal para las dos variables fue de 0,992 y 0,937, por lo que existe buena correspondencia entre los valores del dispositivo con respecto a la referencia. La prueba de los rangos con signo de Wilcoxon arroja valores de p-value superiores al nivel alfa (0,05), lo que permite concluir que tanto la temperatura como la humedad relativa en el prototipo, con relación a la referencia, no presenta diferencias estadísticamente significativas para un intervalo de confianza de un 95 %.

REFERENCIAS

1 “Estrategia ambiental nacional 2011/2015,” *CITMA*, pp 1-22, 2010.

2 R. Pérez de los Reyes, et al. “El sistema nacional de monitoreo ambiental, una herramienta de apoyo a la gestión ambiental a los estudios territoriales de riesgos y a las investigaciones sobre tendencias del medio ambiente cubano,” en *VI Convención Internacional de Medio Ambiente y Desarrollo*, 2007.

3 B. Sloan, B. Morrison, P. Beaupré-Moreau, and A. Mathieu, “Report of the Commissioner of the Environment and Sustainable Development. Chapter 5—A Study of Environmental Monitoring,” Office of the Auditor General of Canada (OAGC), Canada, 2011.

4 S. Reis, “Integrating modelling and smart sensors for environmental and human health, ” *Environ. Model. Softw.*, vol 74, pp. 238–246, 2015.

5 M. Mohamed, W. Wu, M. Moniri, “Data reduction methods for wireless smart sensors in monitoring water distribution systems,” *Procedia Eng.*, vol. 70, pp. 1166–1172, 2014.

6 Ferdoush, Sheikh Mohammad. “A Low-Cost Wireless Sensor Network System Using Raspberry Pi and Arduino for Environmental Monitoring Applications”. Major Professor: Xinrong Li. Thesis for Master of Science (Electrical Engineering), University of North Texas, Texas, 2014.

7 Z. Yuan, and Q. Lan, “Design of Water Quality Monitoring System based on WSN and ZigBee,” *Journal of Residuals Science & Technology*, vol. 13, pp. 1-13, 2016.

8 V. S. Tomar, and V. Bhatia, “Low Cost and Power Software Defined Radio Using Raspberry Pi for Disaster Effected Regions,” *Procedia Comput. Sci.*, vol. 58, pp. 401–407, 2015.

9 R. Schubert, G. Tatiana, G. Carlos, “A system for the monitoring and predicting of data in precision agriculture in a rose greenhouse based on wireless sensor networks,” *Procedia Computer Science*, vol. 121, pp. 306–313, 2017.

10 S. Ferdoush and X. Li, “Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications,” *Procedia Comput. Sci.*, vol. 34, pp. 103–110, 2014.

11 L. Li, et al., *Wireless Sensor Network*, Tsinghua: Tsinghua Univ. Press, 2005.

12 A. Sherin, L. Xinrong, “A Cost-Effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications,” *Procedia Computer Science,* vol. 34, pp. 165 – 171, 2014.

13 B. Oluleke, X. Liang, T. Llewellyn, “An indoor environment monitoring system using low-cost sensor network,” *Energy Procedia*, vol. 141, pp. 660-666, 2017.

14 M. F. Othman and K. Shazali, “Wireless sensor network applications: A study in environment monitoring system,” *Procedia Eng.*, vol. 41, pp. 1204–1210, 2012.

15 S. Daudi, and F. Y. Shi, “Water Quality Monitoring and Control for Aquaculture Based on Wireless Sensor Networks,” *Journal of networks*, vol. 9, pp. 840-849, 2014.

16 C. Sushabhan, K. Piyush, S. Rajesh, Anita, “ZigBee and Bluetooth Network based Sensory Data Acquisition System,” *Procedia Computer Science*, vol. 48, pp. 367 – 372, 2015.

17 S. Bing, S. Victor, Z. Dean, D. Soulin, J. Jianming, “A wireless sensor network-based monitoring system for freshwater fishpond aquaculture,” *Biosystem Engineering,* vol. 172, pp. 57–66, 2018.

18 R. Gao, “A Wireless Sensor Network Environment Monitoring System Based on TinyOS,” in in *Proceedings of International Conference on Electronics and Optoelectronics (ICEOE 2011)*, 2011, pp. 497–501.

19 J. Lloret, et al., “A wireless sensor network deployment for rural and forest fire detection and verification,” *Sensors*, vol. 9, no. 11, pp. 8722–8747, 2009.

20 D. I. Inc., “XBee-PRO® 900/DigiMesh™ 900 OEM RF Modules”: Digi International Inc (D. I. Inc.), Minnetonka, Dec. 2008.

21 M. Schmidt, *Raspberry Pi. A Quick Start Guide*, Dallas: The Pragmatic Programmers, LLC*.,* 2012.

22 D. Wolfram, *Learn Raspberry Pi programming with Python*, California: Apress, 2014.

23 K. Ma and R. Sun, “Introducing WebSocket-Based Real-Time Monitoring System for Remote Intelligent Buildings,” *Int. J. Distrib. Sens. Networks*, vol. 2013, 2013.

24 J. Lengstorf and P. Leggetter, *Realtime Web Apps with HTML5 WebSocket, PHP and Jquery*, California: Apress, 2013.

25 “Django Channels — Channels 2.1.2 documentation,” 2017. [Online]. Available:https://channels.readthedocs.io/en/latest/. [Accessed: 10-Jul-2018].

26 “WS-2310 Weather Pro Center,” 2017. [Online]. Available: https://www.lacrossetechnology.com/2310-weather-center/. [Accessed: 09-Jul-2018].