

## Sistema de monitoreo de una cámara de germinación hidropónica con IoT basado en Raspberry Pi.

Rubén R. Urquijo, Marcelo J. Marinelli

Instituto de Biotecnología de Misiones “María Ebe Reca”. Campus U.Na.M.  
Ruta 12 Km 7,5. Miguel Lanús, Posadas, Misiones, Argentina.  
{chinourquijo, marcelomarinelli}@gmail.com

**Resumen.** Se presenta el diseño de un sistema de monitoreo y telemetría social utilizando tecnología de Internet de las Cosas (IoT, *Internet of Things*) para una cámara de germinación de cultivos hidropónicos. Este sistema, que fue desarrollado utilizando software NODE-RED y hardware Raspberry Pi, toma los valores de las mediciones de humedad y temperatura dentro de la cámara.

El objetivo es implementar sobre un sistema de cultivos hidropónicos, funcionalidades de IoT. Con esto, se pretende sacar provecho del auge que está teniendo dicha tecnología en distintos ámbitos, y aplicarlo a la cámara de germinación hidropónica. Los resultados muestran la implementación del sistema de monitoreo informando, mediante el envío de correos electrónicos y la publicación en Twitter, de los valores en tiempo real de las variables de humedad y temperatura, permitiendo recibir el estado del sistema desde cualquier dispositivo con conexión a internet.

**Keywords:** Hidroponía, Internet de las Cosas, Web Services, E-Mail, Twitter, Raspberry Pi.

### 1 Introducción

La cámara de germinación para cultivos hidropónicos consiste en un recipiente plástico con una solución de nutrientes sobre la cual flota una placa de poliestireno expandido con perforaciones rectangulares en las cuales se deposita un cubo de espuma fenólica con la semilla. La evolución de las tecnologías de Internet tienden a integrar servicios de la web, redes sociales, redes de sensores que permiten obtener información de las variables ambientales o de cualquier otro tipo y además, controlar actuadores, cámaras, etc.; la integración de estos servicios y dispositivos dan como resultado la IoT [2] [3]. Cuando se trabaja con un conjunto de sensores inalámbricos distribuidos espacialmente e interconectados por distintos concentradores, se denomina WSN (*Wireless sensor network*) utilizado para monitoreo ambiental, sistemas interconectados de energía, etc. [4].

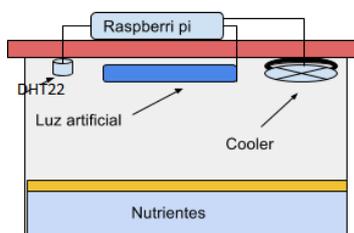
A la hora de desarrollar aplicaciones del tipo IoT, deberemos tener en cuenta que se deben considerar la adquisición de datos proveniente de sensores, tomar decisiones para realizar acciones sobre los actuadores, cámaras o cualquier otro dispositivo y, en

tiempo real, interactuar con servicios en línea. Si se tuviera que programar una aplicación IoT de este tipo, sería necesario conocer varios lenguajes y protocolos, así como desarrollar APIs específicas y poder vincularlas con todos los componentes del sistema [5]. Por esto, surgen varias aplicaciones que integran los componentes que permiten desarrollar aplicaciones IoT. Estas se componen de nodos con funcionalidades específicas y se interconectan por conexiones que intercambian flujo de datos.

Para alojar cubos de espuma fenólica (Figura 1), en estos orificios se siembran las semillas para su germinación y desarrollo hasta su trasplante a otro sistema hidropónico que puede ser de raíz flotante o NFT (*Nutrient Film Technique*) [1]. La cámara posee iluminación artificial mediante paneles LED o luces de bajo consumo y un sistema de ventilación forzada cuyo flujo de aire se controla mediante un control PWM (*pulse-width modulation*), en su interior posee un sensor de humedad y temperatura DHT22 (Figura 2). Para mejorar el proceso de desarrollo de las plántulas, se diseñó e implementó un sistema embebido, que consta de un control difuso que, en función de los valores de las variables de humedad y temperatura, dentro de la cámara, proveen una salida de control de potencia del sistema de ventilación forzada [1]. Sobre este sistema embebido, se implementa un sistema de control basado en tecnología IoT (*Internet of Things*).



**Figura 1.** Placa de poliestireno expandido con los cubos de espuma fenólica en los cuales se desarrollados las plántulas.



**Figura 2.** Diagrama de la cámara de germinación.

## 2 Hardware del sistema

En el presente trabajo se desarrolló un sistema de control de una cámara de germinación con un computador de placa simple o SBC (*Single board computer*) Raspberry Pi (<https://www.raspberrypi.org>), que tiene la característica de ser un dispositivo de bajo coste y soporta distintos sistemas operativos GNU/Linux ARM (Debian, Fedora, Arch Linux), RISC OS2, Windows 10 IOT CORE, entre otros.

Es de tamaño reducido (85mm de longitud por 56mm de ancho), entre sus características posee una salida HDMI, una placa de red tipo Ethernet, una placa de red inalámbrica y 4 puertos USB con lo que, si se conecta un monitor, teclado y mouse, se convierte en una computadora personal. Tiene una CPU ARM1176JZF-S (armv6k) a 700 MHz, GPU Broadcom VideoCore IV3, memoria RAM de 512 MBytes, capacidad de almacenamiento por medio de Tarjeta SD o SDHC (Figura 3).

Debido a su pequeño tamaño, se puede montar sobre la cámara de germinación y con sus puertos de propósito general GPIO (*General Purpose Input/Output*), es posible controlar los sensores y actuadores para desarrollar un sistema de control de la misma.



Figura 3. Raspberry Pi 2 modelo B.

### 2.1 Puertos GPIO

Los puertos GPIO son, como lo indica su nombre, puertos de entrada/salida de propósito general, es decir, un conjunto de conexiones que se pueden utilizar como entradas o salidas para distintos usos. Estos puertos representan la interfaz entre la Raspberry Pi y el medio exterior [6].

En total, posee 17 puertos configurables como entrada o salida. Por defecto, están todos configurados como entradas, excepto los GPIO 14 y 15 que operan como salidas (Figura 4).

En este caso, se utilizó el puerto GPIO 5 como entrada digital para la obtención de los datos que provienen del sensor de humedad y temperatura.



Figura 4. Salidas GPIO de Raspberry Pi.

## 2.2 Sensor DHT22

Para poder obtener los valores de las variables de humedad relativa y temperatura, se han utilizado sensores DHT22 (Figura 5 y Tabla 1) que miden estos parámetros. Estos sensores poseen un rango de operación de 0 a 100% HR y de -40 a 80°C y contando con una precisión de humedad de 2% RH y de temperatura de 0,5%. Por otra parte, posee la característica de poder trabajar con protocolo serial, enviando los datos únicamente por el pin de datos (Figura 5).

Tabla 1. Pines de Salida de DHT22

Pin	Función
1	5 v
2	Datos
3	NC
4	GND



Figura 5. Sensor DHT22

De acuerdo con la Tabla 1, el pin 1 corresponde a la alimentación, el pin 4 a masa (GND) y el pin 2 tiene una resistencia de *pull up* de 10 K por donde son enviados los datos de humedad y temperatura, en forma serial hacia el puerto GPIO correspondiente de la Raspberry Pi (Figura 6).

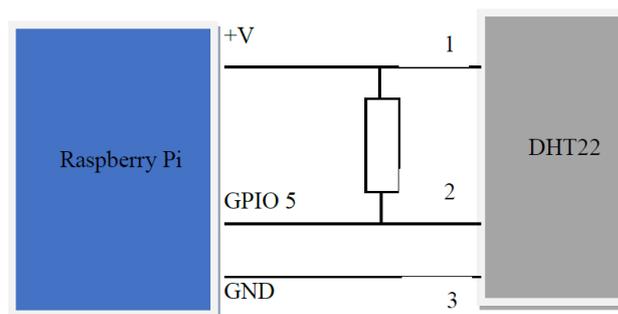


Figura 6. Conexiones del sensor DHT22 con el Puerto GPIO.

### 2.3 Control del sistema de ventilación forzada

La salida PWM del puerto GPIO (GPIO 17) se conecta a un amplificador que nos permite controlar los niveles de potencia del sistema de ventilación que posee la cámara hidropónica. Ésta salida entrega un nivel de potencia calculado por un controlador difuso, diseñado con la herramienta FIS de MATLAB y codificado en *JavaScript*, en función de las variables de entrada y de la base de conocimiento del mismo.

## 3 NODE-RED

Es un editor basado en web para la generación de aplicaciones IoT. Es de código abierto, creado por el equipo de *IBM Emerging Technology*. Posee una interfaz amigable que, mediante la función de arrastrar y soltar, permite colocar en el área de trabajo, nodos que representan APIs de web, servicios en línea o dispositivos de hardware. Cada nodo puede ser de uno de los siguientes tipos: nodo de inyección o nodo de función. Los nodos de inyección producen un mensaje sin tener la necesidad de una entrada y lanzan el mensaje al nodo siguiente conectado a éste. Los nodos de función poseen una entrada y se realiza algún trabajo en él. Con una gran cantidad de nodos disponibles para elegir a la hora de diseñar aplicaciones IoT, NODE-RED hace que conectar dispositivos de hardware, APIs y servicios en línea entre sí, sea más fácil que nunca.

La unión de los nodos, representado mediante una línea continua, indica el flujo de datos que se transfieren entre nodos [7]; también es posible programar funciones en *JavaScript*, de ésta manera se pueden incorporar nodos que tengan dentro funcionalidades específicas definidas y programadas con el lenguaje antes mencionado.

El motor en tiempo real se basa en Node.js, que es un entorno para ejecutar *JavaScript*, realizado con el motor V8 de Chrome. Dado que Node.js trabaja con arquitecturas basadas en eventos no bloqueantes que se ejecutan del lado del servidor, esto hace que consuma pocos recursos de hardware, lo que lo hace ideal para poder correr en placas Arduino ([www.arduino.cc](http://www.arduino.cc)) o Raspberry Pi [8].

### 3.1 Diseño de la aplicación IoT

Para el diseño de la aplicación IoT se han utilizado los nodos de inyección para mantener ejecutado el sistema y capturar los datos de las variables. El diseño de la aplicación puede verse en la Figura 7.

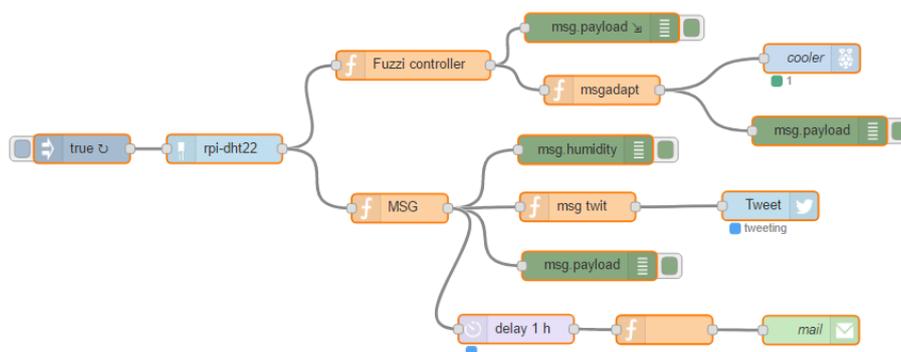


Figura 7. Diagrama de nodos del sistema de monitoreo de la cámara de germinación.

Se exporta de los repositorios de nodos, los específicos para la utilización de Raspberry Pi, es por ello que se escoge el que captura los datos leídos por el sensor DHT22. Mediante la utilización de los nodos de función, se desagregan los datos obtenidos y se transforman en variables de tipo texto, cuyo valor son enviados a los siguientes nodos para poder preparar los mensajes que serán enviados vía mail, a través de Gmail, y vía Twitter.

En la cuenta de correo de Gmail [9], hay que configurar previamente para permitir que los mails sean enviados por la aplicación IoT. Para ello, se debe habilitar en la configuración de la cuenta de Google, en la sección de “Aplicaciones con acceso a la cuenta” y marcar como activo la opción de “Permitir el acceso de aplicaciones menos seguras” (Figura 8). Con esto, nos aseguramos que los mails enviados por la aplicación IoT a la cuenta de Gmail, serán recepcionados en la bandeja de Recibidos.

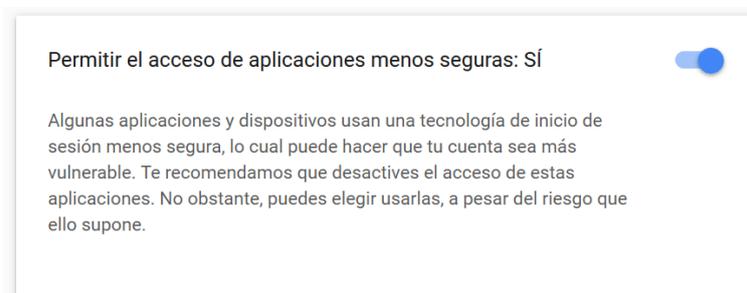


Figura 8. Configuración de seguridad para la cuenta de correo Gmail.

### 3.2 Validación de la aplicación

Para validar los datos obtenidos mediante los sensores, se procede a ejecutar varias lecturas de los mismos en el lugar de la cámara. Los mensajes que envía la aplicación, con los valores obtenidos, hacia la red social Twitter (Figura 9) están configurados para hacerlo cada 10 minutos, y hacia el correo de Gmail cada 60 minutos (Figura 10 y 11).

Estos mensajes son configurables y permiten obtener la información a distancia en un dispositivo móvil, para estar al corriente de los valores de las variables que nos interesa saber en la cámara de germinación y, en caso de algún valor anómalo, poder tomar alguna acción correctiva.

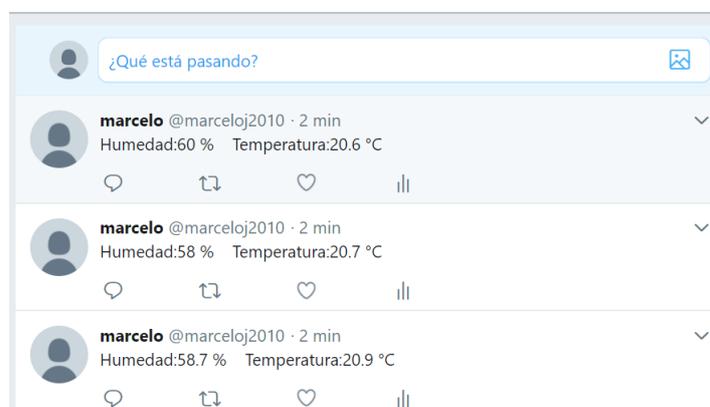


Figura 9. Mensajes recibidos en Twitter.

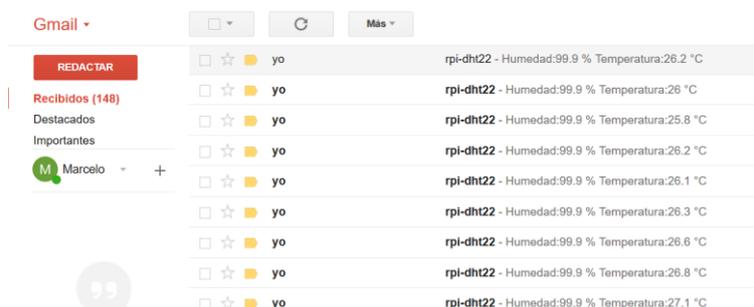


Figura 10. Bandeja de Recibidos de Gmail.



Figura 11. Contenido del mensaje con los valores de las variables.

## 4 Conclusiones y Trabajos Futuros

En esta época altamente desarrollada donde, directa o indirectamente, todo depende de la computación y la tecnología de la información, Raspberry Pi demuestra ser una plataforma potente, económica y eficiente para poder implementar soluciones informáticas en distintas situaciones. Sumado a esto, la facilidad con que se pueden crear aplicaciones IoT utilizando NODE-RED, se pueden crear sistemas de monitoreo, como el que se muestra en este documento, para distintos ámbitos.

La telemetría social permite informar de los estados de las variables de la cámara por mail y por Twitter, de esta manera se tiene actualizado el estado de las variables que intervienen y poder tomar acciones correctivas, en caso de ser necesario.

En la aplicación presentada, está constantemente ejecutándose la misma para la lectura de los datos de las variables. Como mejora, se propone que las personas interactúen con la aplicación IoT, solicitando el estado de las variables intervinientes. Es decir, que las personas puedan consultar enviando un mail o tweet a la aplicación y ésta, deberá armar la respuesta para enviársela nuevamente a la persona que lo solicitó.

Otra mejora a implementar en la aplicación presentada, es el de la generación automática de alarmas, cuando se detecten valores anormales que pongan en peligro el

cultivo. Se consideran valores anormales a aquellos que se encuentren por debajo del umbral establecidos previamente en el controlador difuso.

Actualmente, los datos obtenidos por la lectura de las variables son expuestos vía mail y Twitter, no quedando almacenados en ninguna base de datos, para consultas de históricos de mediciones. Por lo cual, se tratará de implementar dentro de la misma aplicación, el uso de un motor de base de datos, por ej. MySQL o Cassandra para el almacenamiento de series temporales de datos que provienen de la aplicación IoT. Esto, además, nos permitirá poder representar gráficamente el histórico de lecturas de cada una de las variables que intervienen.

## 5 Referencias

1. Marinelli, Marcelo J, Otegui, Mónica B, Zapata, Pedro D, & Acosta, Nelson H. (2015). Control Difuso de una Cámara de Germinación para Hidroponía. *Revista de Ciencia y Tecnología*, (24), 42-47. Recuperado en 25 de abril de 2017, de [http://www.scielo.org.ar/scielo.php?script=sci\\_arttext&pid=S1851-75872015000200007&lng=es&tlng=es](http://www.scielo.org.ar/scielo.php?script=sci_arttext&pid=S1851-75872015000200007&lng=es&tlng=es).
2. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.
3. Kelly, S. D. T., Suryadevara, N. K., & Mukhopadhyay, S. C. (2013). Towards the implementation of IoT for environmental condition monitoring in homes. *IEEE Sensors Journal*, 13(10), 3846-3853.
4. Mainetti, L., Patrono, L., & Vilei, A. (2011, September). Evolution of wireless sensor networks towards the internet of things: A survey. In *Software, Telecommunications and Computer Networks (SoftCOM)*, 2011. 19th International Conference on (pp. 1-6). IEEE.
5. Blackstock, M., & Lea, R. (2014, October). Toward a distributed data flow platform for the web of things (distributed node-red). In *Proceedings of the 5th International Workshop on Web of Things* (pp. 34-39). ACM.
6. Jain, S., Vaibhav, A., & Goyal, L. (2014, February). Raspberry Pi based interactive home automation system through E-mail. In *Optimization, Reliability, and Information Technology (ICROIT)*, 2014 International Conference on (pp. 277-280). IEEE.
7. Kleinfeld, R., Steglich, S., Radziwonowicz, L., & Doukas, C. (2014, October). glue. things: a Mashup Platform for wiring the Internet of Things with the Internet of Services. In *Proceedings of the 5th International Workshop on Web of Things* (pp. 16-21). ACM.
8. Salihbegovic, A., Eterovic, T., Kaljic, E., & Ribic, S. (2015, May). Design of a domain specific language and IDE for Internet of things applications. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015 38th International Convention on (pp. 996-1001). IEEE.

9. M. Marinelli & R. Urquijo, “Sistema de Control de una cámara de germinación hidropónica con IoT”. Libro de Actas del Congreso Argentino de Ciencias de la Computación (CACIC) 2017, (pp. 1101-1107) ISBN 978-950-34-1539-9.