

Sistema de control y telemetría de datos mediante una aplicación móvil en Android basado en IoT para el monitoreo de datos

Data control and telemetry system through an Android mobile application based on IoT for data monitoring

Angel Alejandro RODRIGUEZ Aya [1](#); John Alejandro FIGUEREDO Luna [2](#); Juan Alejandro CHICA García [3](#)

Recibido: 02/02/2018 • Aprobado: 03/03/2018

Contenido

- [1. Introducción](#)
 - [2. Hardware](#)
 - [3. Software y plataformas online](#)
 - [4. Desarrollo del sistema de control y telemetría de datos](#)
 - [5. Resultados](#)
 - [6. Conclusiones](#)
- [Referencias bibliográficas](#)

RESUMEN:

El presente artículo muestra el desarrollo de una aplicación móvil implementada en Appinventor para el monitoreo y control de sistemas eléctricos de mediana y baja potencia mediante el principio del internet de las cosas (IoT), el hardware se basa en una tarjeta WeMos D1 mini, la cual posee un microcontrolador ESP8266 y tiene la tarea de medir, procesar, controlar y enviar las variables de medición a la nube mediante la plataforma IoT thinger.io. El resultado obtenido es una herramienta de hardware y software de bajo costo (menor a 20 dólares) para controlar equipos electrónicos a través de internet, brindando oportunidad de mejora y de bajo costo en la transferencia tecnológica a pequeñas y medianas empresas interesadas en aplicar este tipo de tecnologías a sus procesos.

Palabras clave: IoT, Appinventor, Telemetría de datos

ABSTRACT:

This article shows the development of a mobile application implemented in Appinventor for the monitoring and control of medium and low power electrical systems through the Internet of Things (IoT) principle, the hardware is based on a WeMos D1 mini card which It has an ESP8266 microcontroller and has the task of measuring, processing, controlling and sending the measurement variables to the cloud through the IoT thinger.io platform. The result obtained is a tool of hardware and software of low cost (minor to 20 dollars) to control electronic equipment through Internet, offering opportunity of improvement and of low cost in the technological transfer to small and medium companies interested in applying this type of technologies to their processes.

Keywords: IoT, Appinventor, Data telemetry

1. Introducción

Desde la invención del transistor, los desarrollos electrónicos han crecido de una manera exponencial, actualmente estos sistemas se basan en la implementación de sistemas embebidos que sean capaces de controlar y monitorear un sistema, llámese máquina, proceso o equipo electrónico para un determinado fin. Los primeros procesos de automatización se realizaron mediante microcontroladores los cuales cumplían un fin específico dentro del equipo a controlar, pero el problema radicaba que estos sistemas eran aislados o en otros casos se implementaban con algún protocolo de comunicación para transmitir la información a un computador o equipo cercano de visualización para monitorear el sistema, este tipo de aplicaciones hasta antes del año 2000 se caracterizaban por ser costosas y era necesario tener un conocimiento pleno de su hardware para que se programara eficientemente.

En el año 2005 Banzi (Severance, 2014), uno de los fundadores de lo que hoy se conoce como Arduino, analizo esta problemática y gracias a ello se propuso a realizar una tarjeta de desarrollo económica, de fácil acceso para las personas y que no fuera necesario tener muchos conocimientos en electrónica para realizar desarrollos sobre ella. El impacto fue tal, que hoy en día existen una gran variedad de tarjetas embebidas compatibles con el lenguaje de programación de Arduino y se le han añadido diferentes funcionalidades tales como conexión a internet a través de Wifi integrado o conexiones Ethernet, logrando aplicar el IoT de manera sencilla y económica.

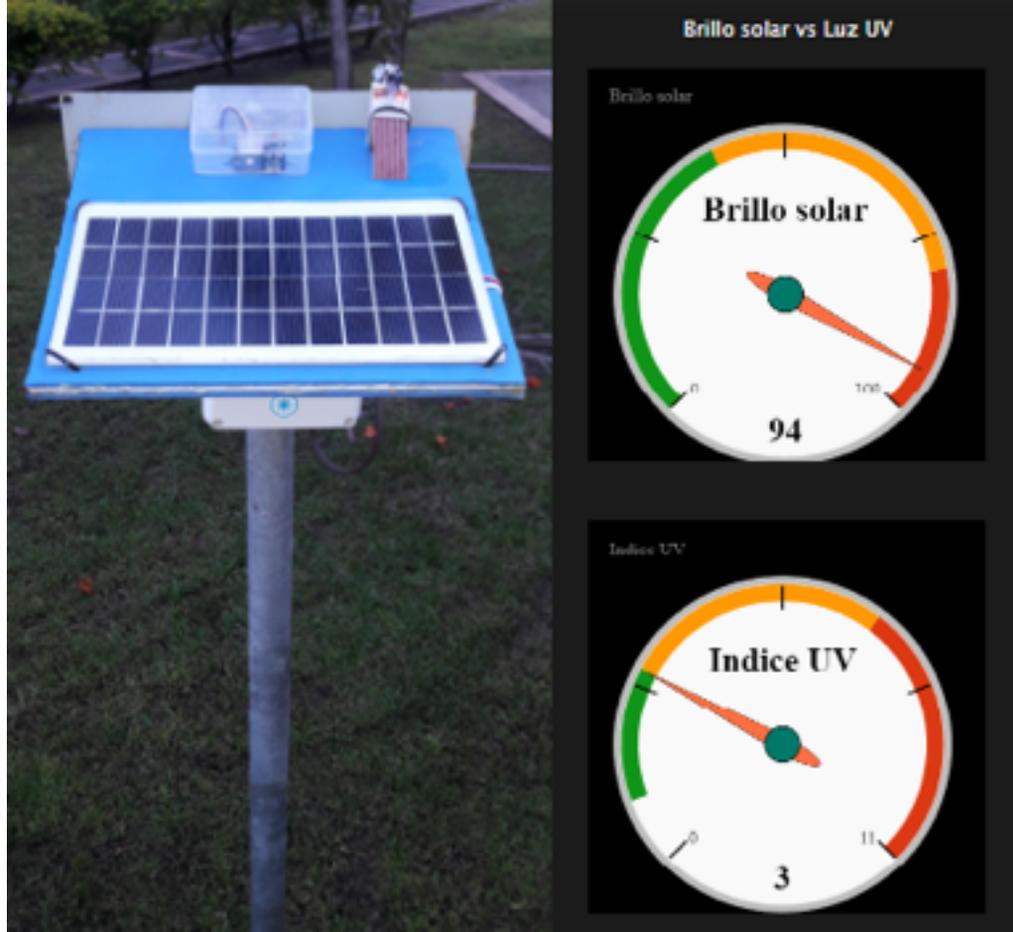
En el mercado actual existe un gran número de tarjetas embebidas para aplicar IoT, como, por ejemplo: la tarjeta Photon de la empresa Particle, tarjetas embebidas ADABOX, Arduino MKRFOX1200, HUZDAH con ESP8266, WiPy, WeMos D1 mini, entre otras; algunas con plataformas propias en la nube para la gestión de los datos, procesamiento, visualización y otras que pueden utilizarse con plataformas externas, además de ser compatibles con diferentes marcas existentes en el mercado.

En la industria del IoT existen diversas plataformas web que ofrecen servicios encaminados para esta labor, algunas de ellas son: Particle, Adafruit, Arduino Cloud, Cayenne, Thinger.io, aREST, ThingSpeak, Artik Cloud entre otras, cada una de ellas ofrece diferentes funcionalidades, ya sea de forma gratuita con algunas limitaciones, o de pago con altas funcionalidades para el manejo de gran cantidad de dispositivos e información.

En el año 2016 los autores de este artículo implementaron una estación meteorológica en la Universidad Nacional Abierta y a Distancia del CEAD de Acacias, para medir las variables de temperatura, humedad relativa, índice UV, sensor de lluvia, brillo solar y potencia solar, la cual se desarrolló con la tarjeta Photon IoT y su plataforma, además se presentaron los datos mediante una página web en la cual se mostraban las variables cada 10 minutos y su comportamiento en el tiempo real gracias al historial y base de datos implementada.

Figura 1

Estación Meteorología en la UNAD CEAD Acacias



Fuente: Extraído de (Serna, 2016)

Hoy en día y de acuerdo a la experiencia en la práctica de varias tarjetas y plataformas para IoT, se utilizó una de las tarjetas más económicas del mercado y se optó por una aplicación para IoT de forma gratuita con el fin de disminuir costos, pero sin bajar las prestaciones del potencial de aplicaciones de proyectos en IoT, es por esta razón que se ha implementado la Tarjeta WeMos D1 mini basada en el microcontrolador ESP8266 como tarjeta embebida para este desarrollo y la plataforma Thingier.io, la cual provee un servicio gratuito con grandes prestaciones en comparación a otras plataformas, además, permite hacer los llamados a los datos de forma externa mediante protocolos de comunicación JSON a través de las funciones GET y POST.

La plataforma seleccionada para el desarrollo de la aplicación móvil para Android es Appinventor, esta plataforma desarrollada por el MIT y Google, se caracteriza por ser de fácil manejo y la programación se hace mediante bloques predefinidos, además es gratuita para los usuarios que tienen una cuenta de correo con Google, permitiendo desarrollar aplicaciones en pocos minutos y con pocos conocimientos de programación.

Este artículo tiene como finalidad dar a conocer el desarrollo de bajo costo de un sistema de medición y control de dispositivos electrónicos a través de herramientas IoT, con el fin de que se pueda exponer diferentes alternativas de diseño y aplicación de las herramientas comerciales existentes y que se pueden aplicar en diferentes campos de acuerdo al modelo de utilidad.

2. Hardware

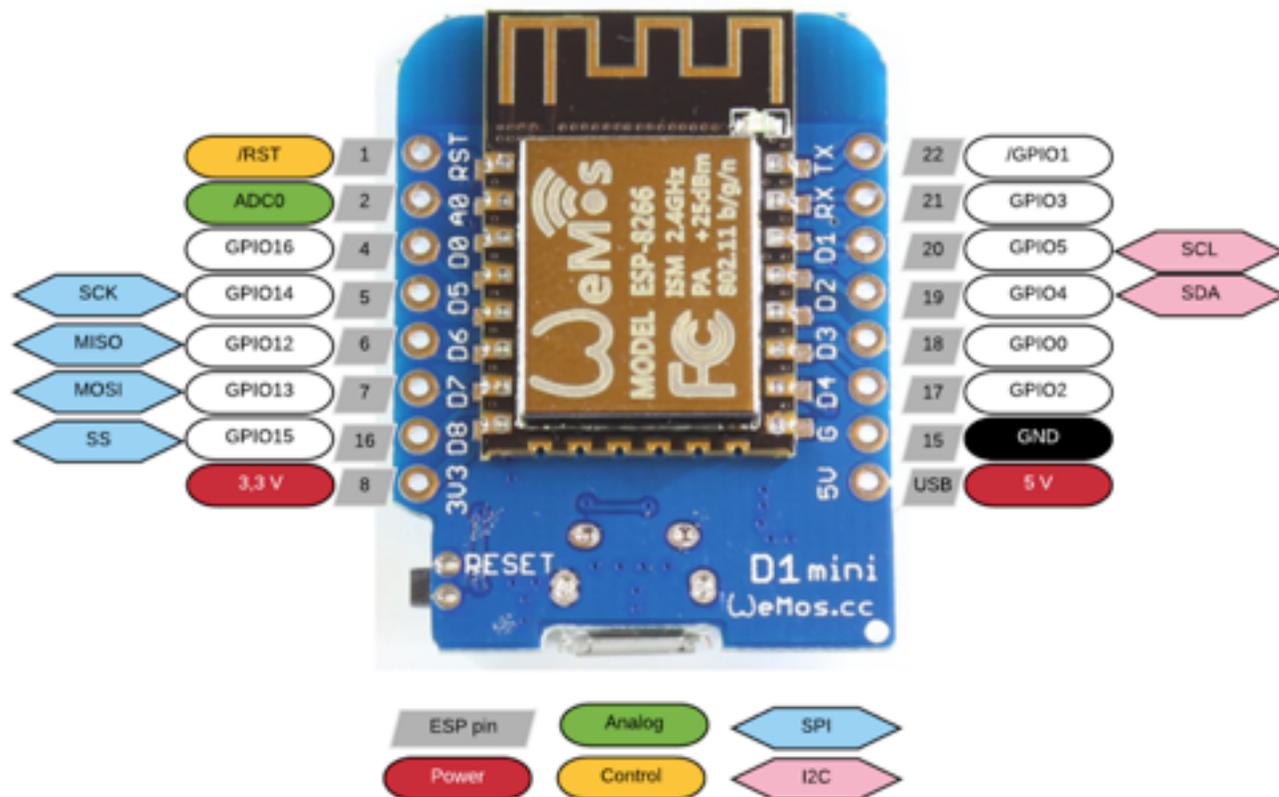
A continuación, se relaciona el hardware necesario para implementar el sistema de control y telemetría de datos el cual será controlado mediante la Appinventor a través de internet.

2.1. WeMos D1 Mini

Esta es una tarjeta embebida basada en el microcontrolador ESP8266, la cual está encaminada a realizar aplicaciones de IoT, por otra parte, esta tarjeta posee comunicación Wifi para hacer el envío de información a plataformas en la nube, además, posee un reloj con una frecuencia de trabajo de 80Mhz con 4Mb de Flash, 11 puertos digitales de I/O permitiendo la comunicación con el hardware exterior, ya sea a través de interrupciones, comunicación I²C, PWM 1-Wire, adicionalmente, posee una entrada analógica, para finalizar, esta tarjeta tiene la particularidad de ser compatible con el IDE de Arduino (Arduino, 2010),

adicionado algunos compiladores y librerías para su programación, haciendo que esta tarjeta sea ideal para aplicaciones de bajo costo y de fácil programación debido a las características mencionadas anteriormente.

Figura 2
Datasheet WeMos D1 Mini



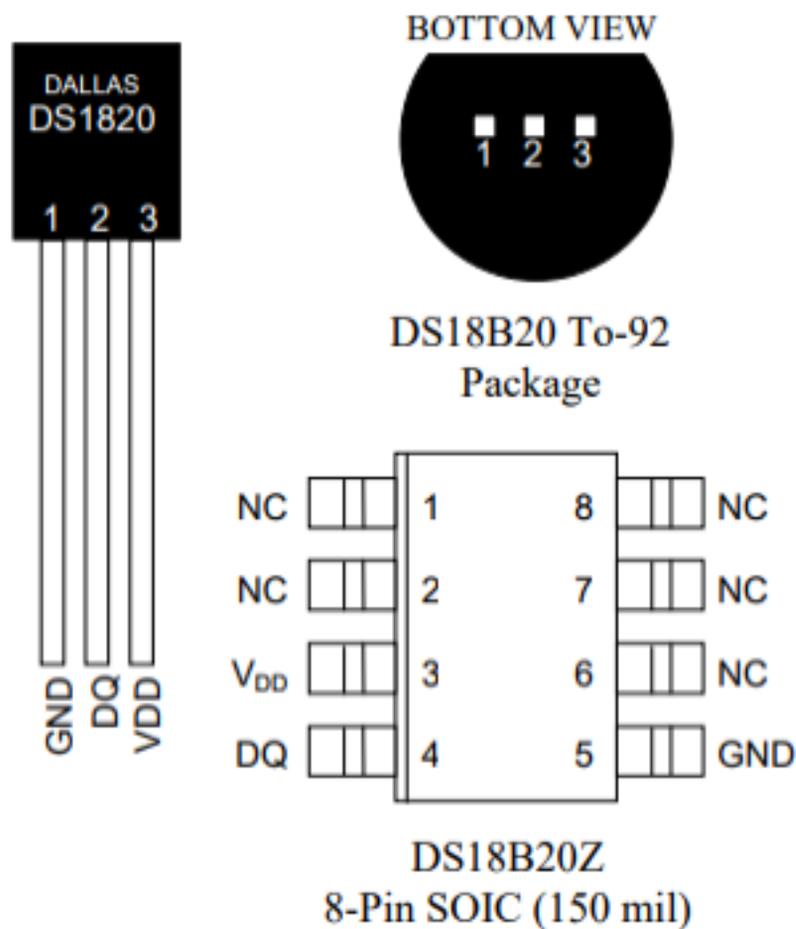
Fuente: Extraído de (Escapequotes, 2016)

2.2. DS 18B20

Este dispositivo es un sensor de temperatura que permite la conexión por medio de una sola línea de datos (1 Wire), el cual tiene la posibilidad de acceder a la información del sensor por una sola línea digital de datos, adicionalmente, este sensor de temperatura tiene un rango de operación de -50 a 125°C con una precisión de ± 0.5 °C y una resolución de 9 y 12 bits (maxim integrated, 2015), permitiendo una precisión aceptable en la lectura de temperaturas en un espacio o ambiente climatizado, lo cual permite hacer una aplicación a menor costo en diversos proyectos.

Figura 3
Encapsulado y distribución de pines DS18B20

PIN ASSIGNMENT

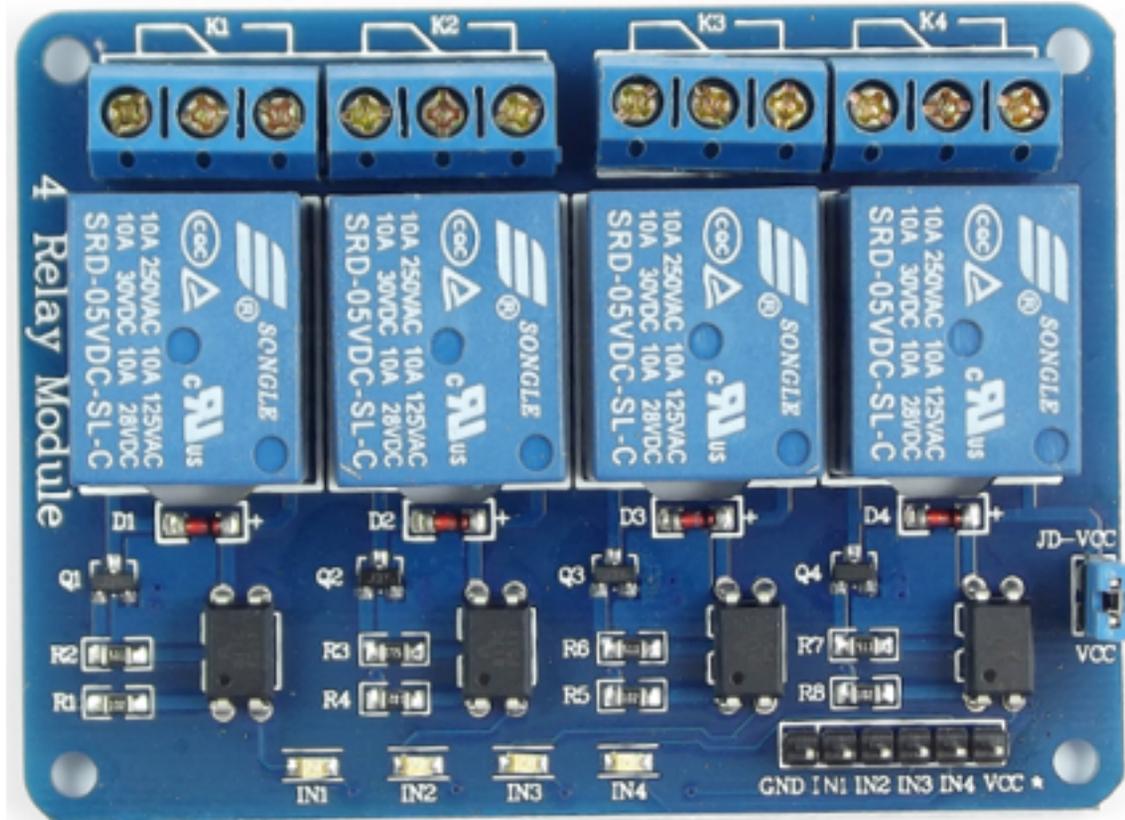


Fuente: Extraído de (Dallas Semiconductor, 2012)

2.3. Módulo de Relés

Este módulo tiene como finalidad conmutar cargas de potencia, este dispositivo funciona de manera electromecánica, el cual consiste en una bobina que produce un campo electromagnético, el cual hace mover una lámina conductora para abrir o cerrar el paso de corriente entre sus terminales, normalmente es utilizado para conmutar corrientes del orden de mA en DC a corrientes del orden de amperios, ya sea en DC o AC (Baichtal, 2013), este es un módulo de 4 relés y tiene la particularidad de activar cada canal con una corriente de 15 a 20mA, además, puede ser controlado directamente con circuitos lógicos, microcontroladores y tarjetas embebidas para la activación de diferentes circuitos, tales como, lámparas, ventiladores, calefactores o cualquier electrodoméstico conectado a él, para este caso se utilizara para controlar cargas AC que controlaran el encendido y apagado de diferentes dispositivos electrónicos.

Figura 4
Módulo de relés de 4 canales



Fuente: Extraído de (Naylamp Mechatronics, 2016)

3. Software y plataformas online

En el siguiente apartado se listarán las diferentes plataformas implementadas para controlar y comunicar el hardware y la aplicación móvil a través de internet

3.1. Arduino IDE

Es una plataforma de código abierto de hardware y software encaminada a controlar las tarjetas embebidas desarrolladas por esta marca (Arduino, 2010), esta fue desarrollada en un principio para programar sistemas embebidos de una manera fácil, como la reconocida tarjeta Arduino, esta programación se realiza por medio de Sketches de fácil programación y con la ayuda de librerías propias o de terceros para controlar el hardware, a medida que pasa el tiempo, diferentes desarrolladores han incluido diversas tarjetas de terceros para ser programadas bajo esta plataforma, puesto que este lenguaje se está convirtiendo en un referente y estándar para la programación de la mayoría de tarjetas embebidas que existen actualmente en el mercado.

3.2. Thinger.io

Es una plataforma online de código abierto, la cual ofrece una infraestructura en la nube escalable para conectar una gran variedad de hardware IoT, esta plataforma ofrece la programación y administración de los dispositivos mediante una consola de administración propia o permite la integración de la información con otras plataformas mediante comandos API REST, además, permite hacer la conexión de los dispositivos en pocos minutos y sin preocuparse por una infraestructura en la nube requerida, adicionalmente, proporciona diferentes tipos de cuentas para los usuarios de acuerdo a las necesidades de hardware y transferencia de información, como por ejemplo, su plan gratuito permite la conexión de hasta 2 Dispositivos, 4 Dashboards, 4 Endpoints, 4 Buckets, con llamados de escritura del Bucket 1 cada 60s y llamados de Endpoints 1 cada 10 segundos (Thinger.io, 2017), permitiendo un almacenamiento de la información y retención de los datos hasta por 1 año, convirtiéndola en una herramienta útil para la implementación de sistemas IoT de baja transferencia de datos dirigida a hogares, universidades y/o pequeñas industrias.

3.3. Appinventor

Esta herramienta es un entorno de programación intuitiva y visual que permite a una gran

variedad de público (incluso niños) crear aplicaciones móviles para Smartphone o Tablets en poco tiempo, gracias a la implementación de programación por bloques, la cual facilita la creación de aplicaciones complejas en mucho menos tiempo, esta iniciativa es liderada por el MIT (Massachusetts Institute of Technology) con el fin de capacitar al desarrollo de software a todo tipo de personas cambiando el modelo de consumo de tecnología a la creación de tecnología (Massachusetts Institute of Technology, 2012), por otra parte, cabe resaltar que esta plataforma es de fácil acceso, solo es necesario tener un correo de Gmail para poder acceder y programar, actualmente se puede desarrollar aplicaciones para gestionar diferentes funcionalidades de los Smartphone tales como GPS, Bluetooth, acelerómetro, Mensajes de texto, Wifi, etc. Por el cual se puede desarrollar una gran variedad de aplicaciones, incluso aplicaciones IoT, además, esta plataforma cuenta con cerca de 6 millones de usuarios registrados

4. Desarrollo del sistema de control y telemetría de datos

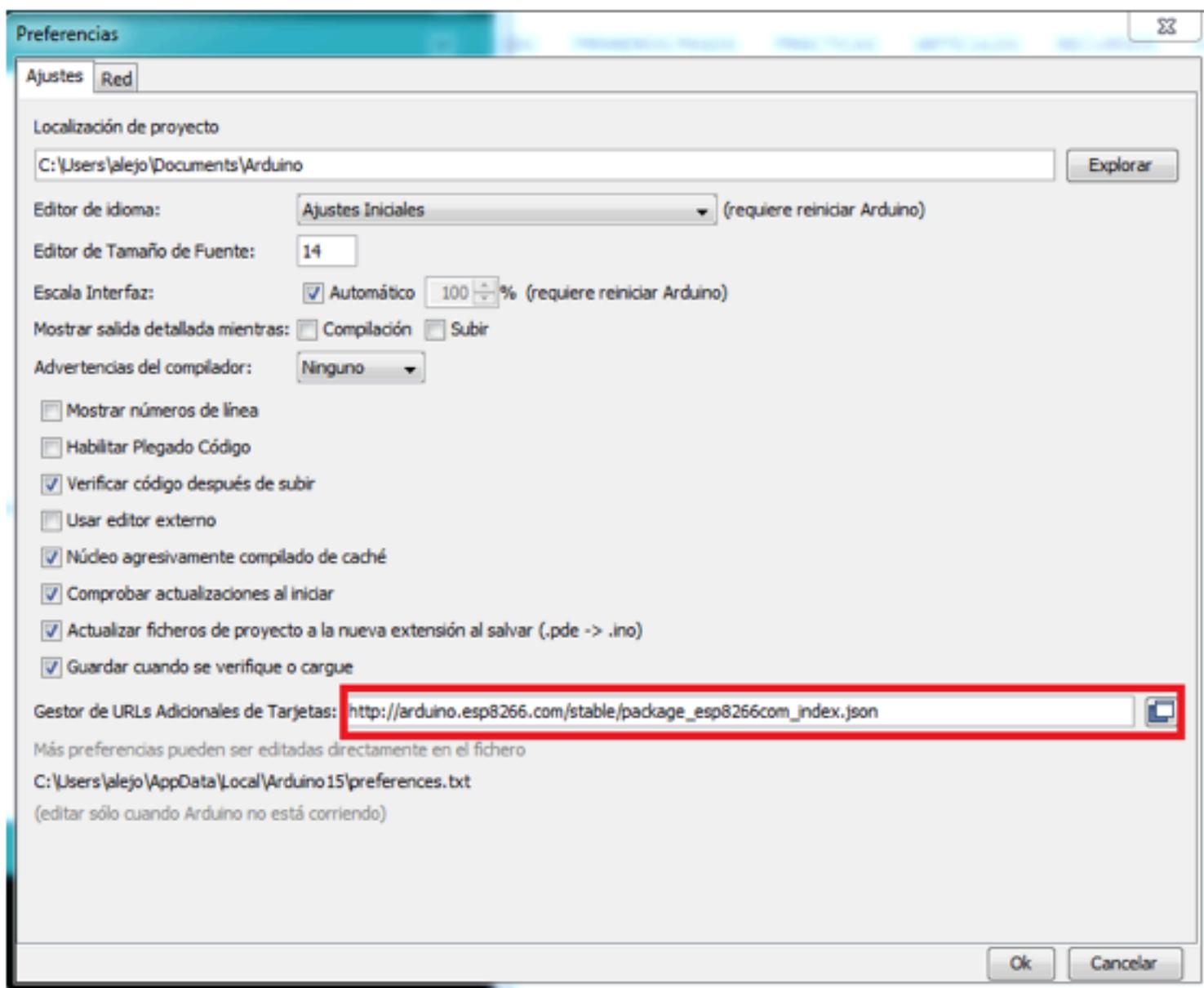
El sistema a desarrollar pretende medir la temperatura ambiente de un espacio cerrado, adicionalmente, controlar el encendido y apagado de 4 dispositivos electrónicos, controlados mediante el módulo de 4 relés, a través de una aplicación móvil de fácil manejo para el usuario a través de Appinventor, en el cual se pueda verificar el valor de la temperatura y el estado (encendido/apagado) de cada dispositivo electrónico conectado a la tarjeta WeMos D1 Mini. La plataforma Thinger.io tendrá la tarea de servir como interfaz de comunicación en la nube entre el hardware y la aplicación móvil con la particularidad de que puedan ser controlados desde cualquier parte del mundo.

4.1. Preparación del IDE de Arduino

Para realizar la programación de la tarjeta WeMos D1 mini es necesario incluir en el IDE de Arduino los compiladores de la tarjeta, lo anterior debido a que la tarjeta WeMos D1 mini no es nativa para el IDE de Arduino, para ello se debe ingresar a las preferencias del IDE e incluir en el gestor de URLs adicionales de tarjeta la siguiente dirección http://arduino.esp8266.com/stable/package_esp8266com_index.json, posteriormente se debe ingresar a las herramientas y abrir el gestor de tarjetas y buscar la tarjeta ESP8266 e instalar el gestor de las tarjetas basadas en este microcontrolador de las cuales se encuentra la WeMos D1 mini.

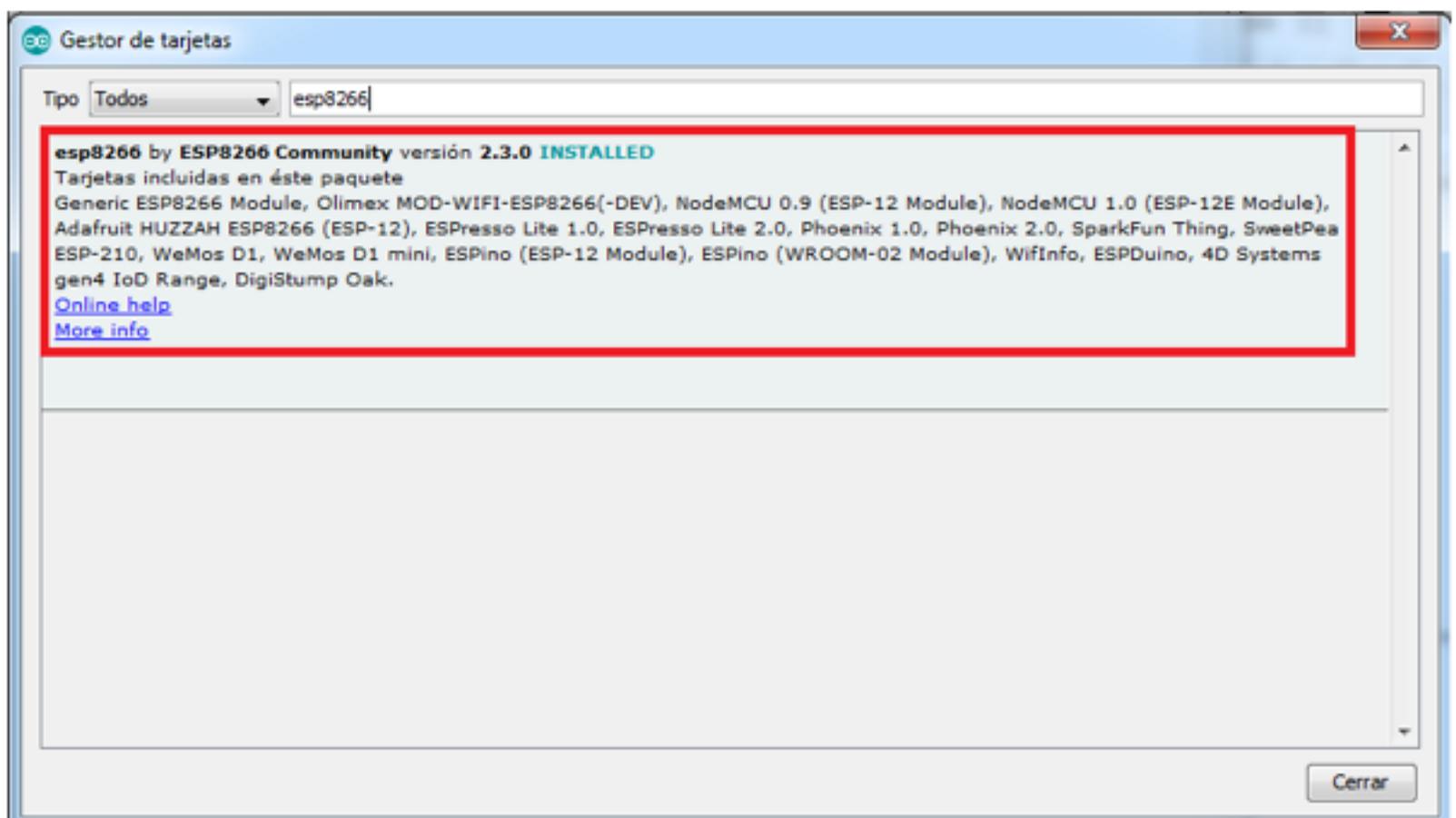
Figura 5

Gestor de URLs adicionales de tarjeta en IDE Arduino



Fuente: Los autores

Figura 6
Gestor de Tarjetas en Arduino IDE, instalación del ESP8266



Fuente: Los autores

Adicionalmente, es necesario instalar las librerías de Thingier.io en el IDE de Arduino, lo

anterior con el fin de poder comunicar las variables de lectura y escritura que serán publicadas en la plataforma Thinger.io, esta operación se realiza de la forma tradicional como se incluye cualquier librería utilizando el gestor de librerías proporcionado por el IDE de Arduino.

4.2. Creación y configuración de cuenta en Thinger.io

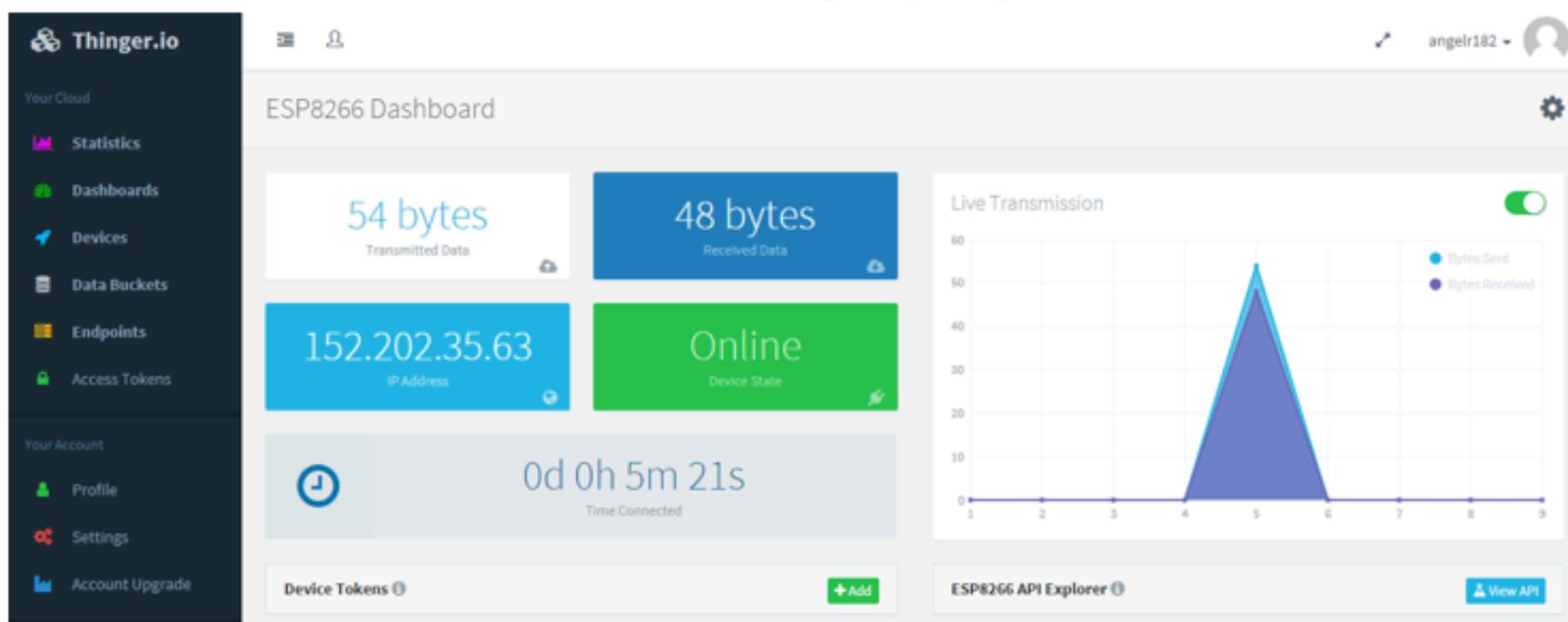
La creación de la cuenta en la plataforma IoT Thinger.io se crea con una cuenta de correo existente y con una información básica, al principio todos los usuarios inician con una cuenta gratuita con las limitaciones mencionadas anteriormente, pero con la posibilidad de actualizar a planes de pago con mejores prestaciones en el servicio, para este caso se trabajará con la cuenta gratuita, la cual posee las funcionalidades óptimas para la realización de este proyecto.

Como primera medida, se hace necesario la configuración del dispositivo que se va a conectar a la cuenta que se ha creado, para ello se debe ingresar a la opción "devices" en donde aparecerá la posibilidad de crear un nuevo dispositivo, en este apartado es necesario asignarle un nombre, una descripción y una credencial de acceso al dispositivo; cabe resaltar que esta información consignada en este dispositivo debe estar incluida en el Sketch que se desarrolla para el WeMos D1 mini.

Posteriormente, luego de hacer esta configuración, si las credenciales de acceso tanto en la tarjeta WeMos D1 mini y en Thinger.io coincide debe mostrar que el dispositivo está conectado y podremos acceder a View API, en donde se visualizara las variables previamente creadas en la Tarjeta WeMos D1 mini y se podrán leer o escribir.

Figura 7

Verificación de la conexión entre Thinger.io y la tarjeta WeMos D1 mini

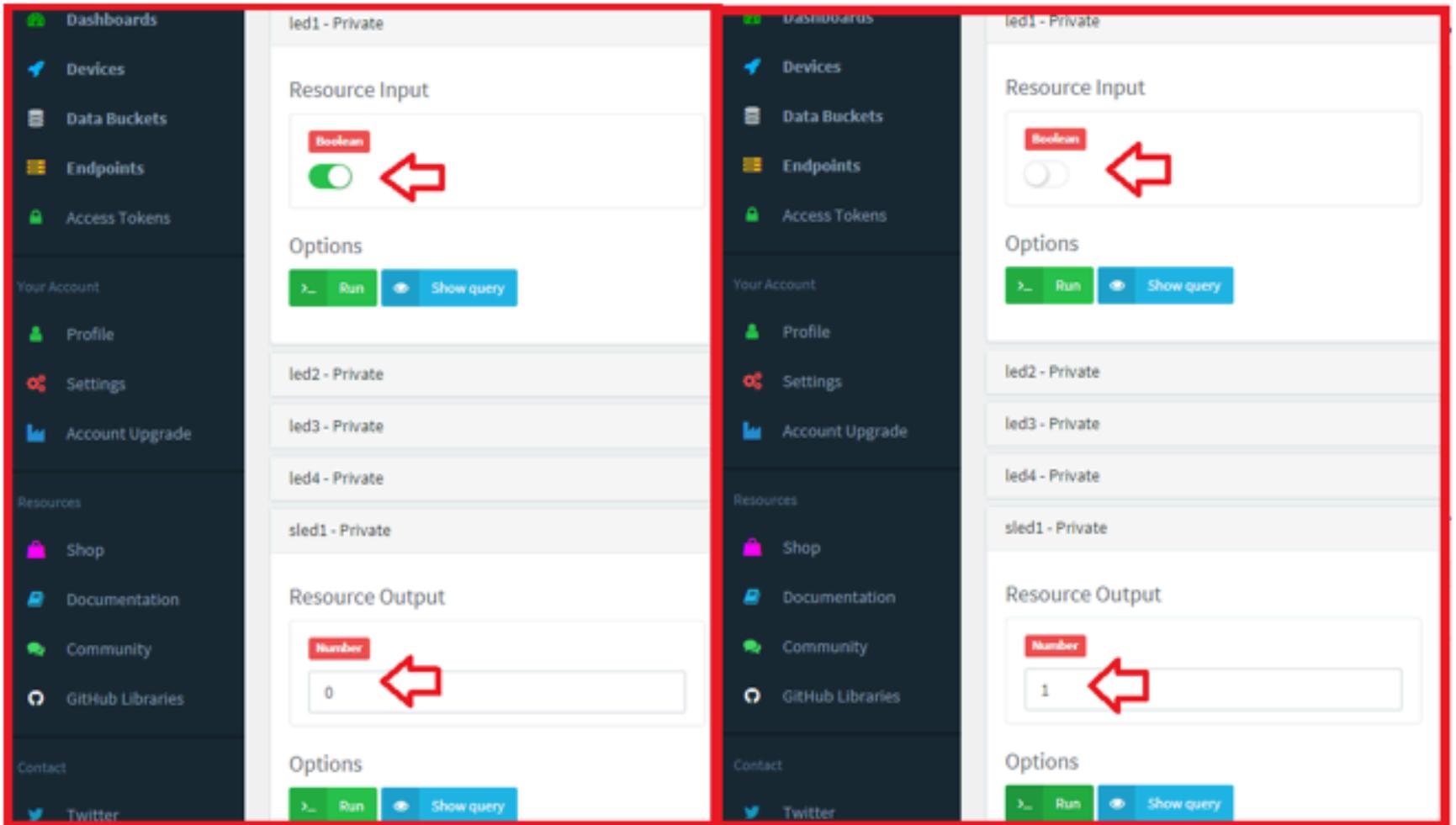


Fuente: Los autores

Para finalizar, se puede comprobar la lectura y escritura por medio de la plataforma Thinger.io como se muestra en la figura 8, aunque la idea principal del proyecto no es el uso de esta plataforma para el monitoreo y control de las variables, sino un puente de comunicación entre la tarjeta WeMos D1 mini y Appinventor, para esta labor se implementará la comunicación por medio del API REST (Thinger.io, 2015) que ofrece la plataforma y los protocolos de comunicación en JSON con los llamados de GET y POST.

Figura 8

Verificación de las variables a través de Thinger.io



Fuente: Los autores

4.3. Lectura y escritura de datos desde Thinger.io a Appinventor

El desarrollo de la aplicación en Appinventor tiene como función principal hacer los llamados de GET y POST al API REST de Thinger.io, con el fin de leer y escribir las variables y enviar la información a la tarjeta WeMos D1 mini, para este caso, se consulta la documentación proporcionada por Thinger.io para realizar la función de GET y POST, un ejemplo claro de hacer una petición de GET y POST se muestra a continuación:

Figura 9
Llamado GET en Appinventor a Thinger.io



Fuente: Los autores

Figura 10
Llamado POST en Appinventor a Thinger.io

```

when Button3 .Click
do
  set POST .Url to "https://api.thinger.io/v2/users/angelr182/device..."
  set POST .RequestHeaders to
    make a list
    make a list "Content-Type"
    "application/json; charset=UTF-8"
    make a list "Authorization"
    "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqd2UiOiJ1b3VudG91IiwiaWF0IjoiMTY1MjM0NTY3In0.eyJqd2UiOiJ1b3VudG91IiwiaWF0IjoiMTY1MjM0NTY3In0."
    make a list "Accept"
    "application/json, text/plain, */*"
  set global led to "sled2"
  call POST .PostText
  text {"in":false}

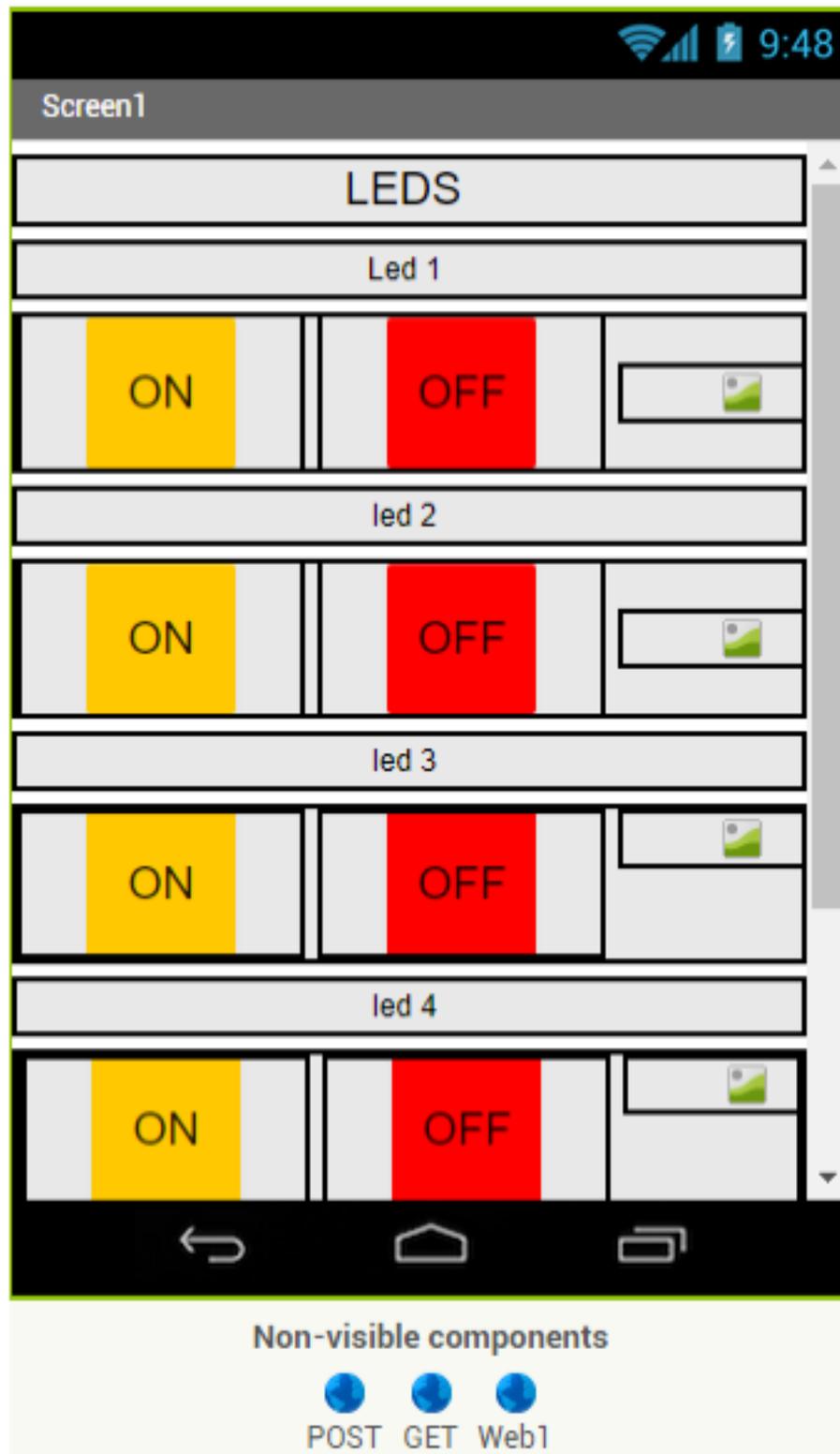
```

Fuente: Los autores

Después de tener la comunicación de las variables entre Appinventor, Thinger.io y a su vez a la tarjeta WeMos D1 mini, se procede a realizar la programación en bloques de la información que se publicara en la pantalla de la aplicación, la cual consta de 4 interruptores con la opción de ON y OFF, cada una con la verificación del envío de información por medio de una imagen y por último la visualización de la temperatura ambiente medida.

Figura 11

Diseño del Screen de la App para el envío y recepción de la información



La presentación final de la aplicación móvil desarrollada en Appinventor permite interactuar de una forma intuitiva la acción de cada uno de los botones e indicadores de estado, como se puede apreciar en la Figura 12.

Figura 12

Aplicación móvil terminada del sistema de control y monitoreo de variables



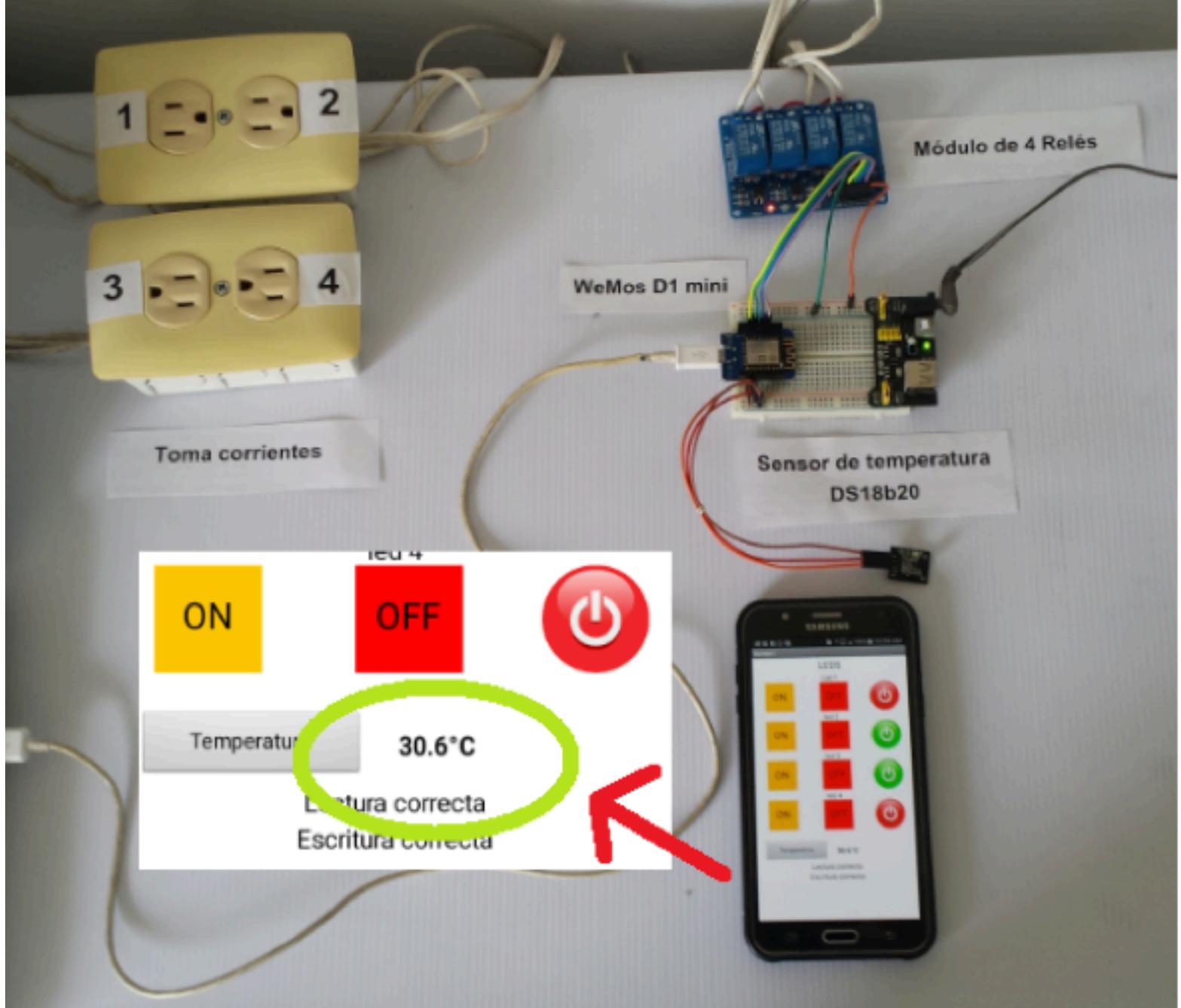
Fuente: Los autores

5. Resultados

El desarrollo de este proyecto logró generar una herramienta IoT de bajo costo de implementación (menor a 20 dólares), en el cual se puede apreciar en la figura 13 el funcionamiento de la aplicación móvil desarrollada en Appinventor, la comunicación en la nube con Thinger.io y la lectura y escritura de datos en la tarjeta WeMos D1 mini con sus respectivos actuadores, en aras de realizar una telemetría y control de datos en diferentes áreas, ya sean de orden académico aplicadas a prácticas de laboratorio o de carácter industrial a baja escala encaminadas a la aplicación de algún proceso productivo.

Figura 13

Pruebas de laboratorio del sistema de control de telemetría de datos



Fuente: Los autores

La herramienta Thinger.io denota una facilidad de aplicación en diferentes entornos, para estudiantes o profesionales en las áreas de tecnología, ya sean con las opciones que brinda la misma aplicación o con aplicaciones de terceros haciendo comunicación por medio de REST API y así leer cualquier transductor y controlar cualquier actuador.

El sistema de telemetría de datos se le realizó un seguimiento y pruebas con el fin de verificar la funcionalidad y la disponibilidad del servicio, estas pruebas consistían en el envío de información y lectura de datos durante 1 hora al día por un mes: los resultados obtenidos mostraron una disponibilidad del servicio del 96%, arrojando errores de comunicación entre las aplicaciones cercanas al 4%.

Tabla 1
Mediciones de disponibilidad del servicio

Descripción	Lectura	Escritura
Envío de datos por día	30	30
Perdida promedio de datos	1,05	1,26
% perdida de datos	3,50%	4,20%

Fuente: Los autores

6. Conclusiones

El sistema de control y telemetría de datos, permitió la investigación de diversas plataformas con el fin de realizar aplicaciones IoT, puesto que no todas las plataformas ofrecen cuentas gratuitas o con las funcionalidades mínimas para realizar un envío continuo de datos, las cuales pueden ser limitantes para el desarrollo de los proyectos de esta índole, por lo cual se logró comprobar que la plataforma Thinger.io ofrece actualmente una buena tasa de transferencia de datos por minuto, además, permite pensar en crear aplicaciones de baja escala para industrias pequeñas que cuenten con la necesidad de implementar lo que hoy se conoce como IoT de una manera rentable.

Las tarjetas embebidas basadas en el microcontrolador ESP8266 actualmente están abaratando los costos del hardware aplicado hacia el IoT, debido a que este microcontrolador es desarrollado y distribuido por industrias chinas, además, este incluye el protocolo comunicación inalámbrica Wifi dentro del microcontrolador, lo anterior teniendo en cuenta que muchas tarjetas embebidas traen por separado este módulo, situación que permite que las tarjetas basadas en el ESP8266 en algunos casos sean 4 veces más económicas que algunas marcas reconocidas como por ejemplo Particle, WiPy, Pycom y Arduino.

La interconectividad del hardware a la nube, permite a las industrias tener un monitoreo más eficiente de la información derivada de cada operación objeto de su planta de producción, permitiendo no solo a parte operativa conocer el estado actual de un proceso sino también de los supervisores, con el fin de hacer un seguimiento de su cadena de valor.

El desarrollo de este proyecto, abre las puertas de la imaginación de los investigadores puesto que permite visibilizar un potencial latente de diseños en automatización, monitoreo y control de diversos proyectos, tales como agricultura de precisión, medición y predicción del clima, robótica aplicada, automatización y control, ciudades inteligentes, monitoreo de vehículos y confort en el hogar, teniendo grandes áreas de aplicación, además, se podría llegar a concluir que las aplicaciones son ilimitadas.

Referencias bibliográficas

Arduino. (15 de Junio de 2010). *Arduino*. Recuperado el 11 de Enero de 2018, de <https://www.arduino.cc/en/Guide/Introduction>

Baichtal, J. (6 de Septiembre de 2013). *Makezine*. Recuperado el 11 de Enero de 2018, de <https://makezine.com/2013/09/06/component-of-the-month-relays/>

Barbon, G. Margolis, M. Palumbo, F. Raimondi, F. Weldin, N. (2016). Taking Arduino to the Internet of Things: The ASIP programming model. *Computer Communications*, Volumes 89–90, 1 September 2016, Pages 128-140.

Christos Stergiou, Kostas E. Psannis, Byung-Gyu Kim, Brij Gupta. (2018). Secure integration of IoT and Cloud Computing. *Future Generation Computer Systems*, Volume 78, Part 3, January 2018, Pages 964-975.

Dallas Semiconductor. (10 de Agosto de 2012). *Sparkfun*. Obtenido de Programmable Resolution 1-Wire Digital Thermometer: <https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf>

Escapequotes. (19 de Febrero de 2016). *EscapeQuotes*. Obtenido de WeMos D1 mini pins and diagram: <https://escapequotes.net/esp8266-wemos-d1-mini-pins-and-diagram/>

Letham, B. et al. (2016). Environmental monitoring system based on an Open Source Platform and the Internet of Things for a building energy retrofit. *Automation in Construction*, Volume 87, March 2018, Pages 201-214.

Massachusetts Institute of Technology. (10 de Octubre de 2012). *Appinventor*. (Massachusetts Institute of Technology) Recuperado el 11 de Enero de 2018, de <http://appinventor.mit.edu/explore/about-us.html>

Maxim integrated. (10 de Agosto de 2015). *Datasheets maxim integrated*. Recuperado el 10 de enero de 2018, de <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

Mussab Alaa, A.A. et al. (2017). A review of smart home applications based on Internet of Things. *Journal of Network and Computer Applications*, Volume 97, 1 November 2017, Pages 48-65.

Naylamp Mechatronics. (20 de Octubre de 2016). *Naylamp Mechatronics*. Obtenido de Drivers: <http://www.naylampmechatronics.com/drivers/152-modulo-relay-4-canales-5vdc.html>

Pritoni, M. Ford, R. Karlin, B. Sanguinetti, A. Categories and functionality of smart home technology for energy management. *Data in Brief*, Volume 16, February 2018, Pages 71-74.

Scarpa, M. Ravagnin, R. Schibuola, L. Tambani, C. (2017). Development and testing of a platform aimed at pervasive monitoring of indoor environment and building energy. *Energy Procedia*, Volume 126, September 2017, Pages 282-288.

Serna, E. (2016). Selección e implementación de un prototipo de estación meteorológica aplicando IoT y herramientas Google. En *Desarrollo e Innovación en Ingeniería* (págs. 341-352). Medellín: Instituto Antioqueño de Investigación.

Severance, C. (2014). Massimo Banzi: Building Arduino. *Computing Conversations*, 14(1), 11-12.

Teixeira Monteiro, I. (2017). Signifying software engineering to computational thinking learners with AgentSheets and PoliFacets. *Journal of Visual Languages & Computing*, Volume 40, June 2017, Pages 91-112.

Thinger.io. (10 de Febrero de 2015). *Thinger.io - Server API*. Recuperado el 12 de Enero de 2018, de <http://docs.thinger.io/api/>

Thinger.io. (20 de Febrero de 2017). *Thinger.io*. Recuperado el 11 de Enero de 2018, de <https://thinger.io/>

WEMOS. (20 de Octubre de 2017). *WEMOS*. Recuperado el 10 de Enero de 2018, de <https://www.wemos.cc/>

Xiaoqu H. et al. (may 2017). Integration of Organic Rankine Cycle with Lignite Flue Gas Pre-drying for Waste Heat and Water Recovery from Dryer Exhaust Gas: Thermodynamic and Economic Analysis. *Energy Procedia*, Volume 105, May 2017, Pages 1614-1621.

Xiong, F. (2015). Wireless temperature sensor network based on DS18B20, CC2420, MCU AT89S52. *2015 IEEE International Conference on Communication Software and Networks (ICCSN)*, (págs. 294-298). Chengdu.

1. Ingeniero Electrónico, Magister en Desarrollo Sostenible y Medio ambiente. Universidad Nacional Abierta y a Distancia – UNAD. Escuela de Ciencias Básicas Tecnología e Ingeniería. Docente del programa de Ingeniería Electrónica. Correo: angel.rodriguez@unad.edu.co

2. Ingeniero de Sistemas, Magister en Educación. Universidad Nacional Abierta y a Distancia – UNAD. Escuela de Ciencias Básicas Tecnología e Ingeniería. Líder de Escuela ECBTI. Correo: John.figueroa@unad.edu.co

3. Ingeniero Electrónico, Especialista en Alta Gerencia. Corporación Autónoma de Nariño. Departamento de Investigación. Líder de Investigación. Correo: d.investigacion@auravillavicencio.edu.co

Revista ESPACIOS. ISSN 0798 1015
Vol. 39 (Nº 22) Año 2018

[Índice]

[En caso de encontrar algún error en este website favor enviar email a webmaster]

©2018. revistaESPACIOS.com • Derechos Reservados