3

Control de Factores de Riesgos mediante un Sistema de Telemetría

Hugo Gaibor Estupiñán / Iván Silva Feraud

Resumen

Cuando ocurre un siniestro dentro de un centro de cómputo, el tiempo de respuesta que el administrador debe enfrentar para tomar decisiones es crítico. Este trabajo se enfoca en diseñar un sistema de telemetría, para monitorear en tiempo real los factores de riesgo de un centro de cómputo utilizando herramientas de código abierto. Los datos de los sensores son enviados a un servidor, el cual contiene un gestor de alertas automatizado. Para probar la eficacia del sistema, se realizó una simulación a través de la comparación de lecturas de temperatura y de pruebas de transmisión de datos por la red. Los resultados obtenidos demuestran que hay diferencia entre la lectura del sensor y la de un termómetro externo y que el envío de paquetes es efectivo al enviarlos a través de la red.

Palabras clave:

Arduino, telemetría, código abierto, sensores, centro de cómputo.

Abstract

Time response is critical for technology managers to make decisions when a computing center is struck by disaster. This papers purports to use the tools of open code telemetry systems to design a telemetry system to monitor in real time the risk factors of a computing center. Sensor data are sent to a server which has an automatic alert system. In order to verify the system's efficiency, temperature readings and data transmission through the net were tested by simulation. Results show differences in the sensor's reading and an outside thermometer. Sending packages through the net proved effective.

Keywords:

Arduino, telemetry, open source, sensors, computer center.

Introducción

En el Ecuador, cada vez son mayores las instituciones que hacen uso de las tecnologías de la información y comunicación TIC's. En la parte educativa, el gobierno está llegando a gran cantidad de escuelas públicas y comunidades rurales (Ministerio de Educación, 2012). Tanto las instituciones privadas (Mosquera de Calderón, 2011), como públicas están colocando un mayor número de aulas y localidades con la capacidad de acceder a recursos educacionales en Internet.

Sin embargo, pese a la notable inversión en equipos de computación y TIC's, en la actualidad muchas empresas no consideran la inversión en sistemas que realicen monitoreo y ayuden a proteger sus recursos tecnológicos. Es común observar las instalaciones de los centros de cómputo, o servidores conectados directamente sin protección contra picos de voltajes, sin resguardo físico de los equipos e incluso con los equipos dispuestos a nivel del suelo; sin tomar en cuenta factores ambientales como el polvo, agua u otros riesgos potenciales de estas prácticas (Peñarreta, 2013).

Esto puede deberse a que son pocas las soluciones en torno a sistemas de monitoreo que tengan un costo módico. Comúnmente no permiten flexibilidad de implementaciones y no son fácilmente integrables con sensores y sistemas de monitoreo existentes en una instalación. Su precio desalienta a los administradores a invertir en algo que no consideran imprescindible para el funcionamiento de su empresa. Esto se ve agravado por una cultura general en el país, en donde no existen hábitos de gestión proactiva para mitigar fallos y se actúa cuando se presenta un daño o desperfecto (Izquierdo, 2013).

Por estas razones, se consideró la necesidad de crear un sistema modelo utilizando tecnologías de código abierto, que sea flexible, escalable y que tenga un precio asequible para instituciones pequeñas, medianas y grandes. Tanto los componentes de hardware, como de software que constituirán este modelo, estarán basados en tecnologías de código abierto.

Una de las ventajas de utilizar este tipo de tecnologías reside en que los elementos cuentan con el respaldo de grandes comunidades de entusiastas, los cuales aportan con ideas, comunican errores e interactúan para ayudar a mejorar constantemente el código de un programa, o de un proyecto (Golden, 2005).

Otra ventaja reside en que se pueden integrar diversas soluciones y de esta manera extender la funcionalidad de este modelo. Dado que las tecnologías de código abierto no tiene restricciones de lectura, o modificaciones a su código fuente, se logrará realizar alteraciones que le podría permitir a este modelo, comunicarse con sensores o aplicaciones de terceros, incluso hasta de código cerrado sin mayores problemas (González, Seone, & Robles, 2015).

Este modelo permitirá monitorear diversas variables en un área determinada, o en los equipos sensibles de una empresa, ya sea por el valor que representan estos equipos, por la importancia de la información que puede estar contenida en ellos, o porque se quiere dar un mayor nivel de seguridad a sus instalaciones.

Adicionalmente, dada la apertura de este modelo, su utilización podrá abarcar diversas áreas aparte de su implementación en instalaciones de TICs y áreas de una empresa: tales como la agricultura, el hogar, diversos tipos de comercios, entre otros.

Fundamentación Teórica

Se entenderá como factores físicos de riesgo a todos los elementos que puedan afectar a un área sensible de una organización, ya sea a nivel de la construcción, equipos de TICs, o a otros equipos que sean considerados críticos para las operaciones de una empresa (Bosworth, Kabay, & Whyne, 2012). Los sensores utilizados para monito-

rear los factores físicos de riesgo son diseñados específicamente para detectar cada tipo de amenaza. Es por esto que se deben considerar todas las amenazas que pueden representar en un riesgo potencial a lo largo de la vida de una instalación (Cowan & Gaskins, 2011).

Seymour Bosworth, M. Kabay y Eric Whyne, en su libro Computer Security Handbook Set (2012) enumeran los 7 mayores factores de riesgo que pueden afectar físicamente a una instalación de TICs:

- Temperaturas extremas.- Sean estas de frío o calor, entre ellas pueden incluirse desastres por incendios, erupciones volcánicas, lava, entre otros.
- Gases.- En esta categoría constan elementos compuestos de partículas suspendidas en el aire, gases tóxicos, corrosivos, gases de químicos inflamables, gases de centrales de aires acondicionados, humo, humedad, gases de uso militar, entre otros.
- Líquidos.- De los cuales se destacan el agua proveniente de inundaciones, lluvias, roturas de tuberías; productos químicos de limpieza, bebidas que pueden derramarse por accidente, aceites, corrosivos, entre otros
- Movimientos.- Estos sucesos afectan el funcionamiento del equipo. Pueden ir desde una vibración que afecte a un disco duro, hasta un terremoto o deslave que comprometa todo el centro de IT
- Anomalías de energía.- Entre ellas están las variaciones de voltaje, apagones, inducción magnética, electricidad estática, pulsos electromagnéticos, interferencias de radiofrecuencia o microondas, radiación y cualquier otra forma de energía que afecte, o dañe la circuitería interna de un equipo de IT sensible a este tipo de interferencias.

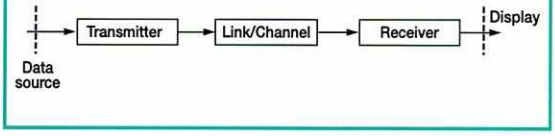
Por otra parte, un sistema puede ser definido como un conjunto de componentes con funciones particulares, que trabajarán en conjunto para cumplir con una tarea específica (Perdikaris, 1996).

En su libro Instrumentation And Process Control, Janardan Prasad, M Jayaswal y Priye (2009) desglosan el término telemetría en dos partes: tele, que significa distancia, transmisión y meter, que significa medición. A partir del origen etimológico de la palabra se puede concluir, que la telemetría es la medición remota de algún factor y la transmisión de estos datos a un lugar distante.

Frank Carden, Russell Jedlicka & Robert Henry establecen el propósito de un sistema de telemetría en su libro Telemetry Systems Engineering (2002), como un sistema para recolectar datos de un lugar que es remoto o inalcanzable para una persona, ya sea por condiciones de peligrosidad o dificultad de acceso presencial. Estas recolecciones de datos pueden ser el desempeño de un sistema, el monitoreo de un animal en su entorno, el reactor de una planta nuclear, o cualquier cosa que pueda ser medida y cuantificada. Cuando un sistema de telemetría tiene componentes que permiten controlar remotamente aparte de monitorear, toma el nombre de sistema SCADA[1].

El principio de telemetría se fundamentará en que los componentes de medición de este modelo, enviarán las lecturas obtenidas un componente de recopilación, el cual estará en una ubicación distante, a través de un medio de transmisión (Carden, Jedlicka, & Henry, 2002).

Figura 1.
Diagrama de bloque de un sistema de telemetría.



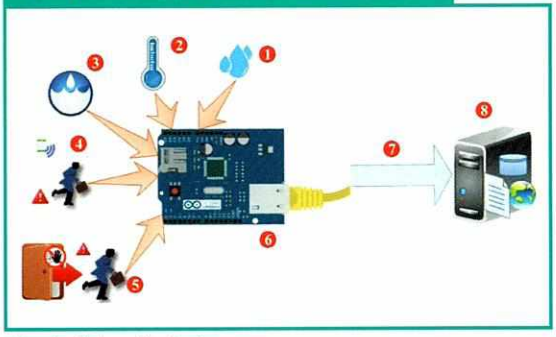
Fuente: *Patranabis, D. (1999). Telemetry Principles. S/C: Tata McGraw-Hill Education.*

[1] SCADA: Supervisory control and data acquisition System

El sistema de telemetría propuesto permitirá el despliegue de distintos tipos de sensores. El caso de aplicación en particular le permitirá medir cambios en los factores físicos de riesgo en un centro de cómputo.

Los componentes que conformarán el modelo sistema de telemetría se muestran en la siguiente figura:

Figura 2.
Componentes del sistema de telemetría.



Fuente: *Elaboración Propia*

1. Sensor de humedad relativa
2. Sensor de temperatura
3. Sensor de filtraciones e inundaciones
4. Sensor de movimiento
5. Sensor de contacto
6. Nodo de procesamiento de mediciones y comunicación intermedia
7. Petición HTTP[2] con la trama de datos y mediciones
8. Servidor de recopilación de datos y gestión de alertas

En primera instancia, los sensores captarán las variables del entorno y las trasmitirán al Nodo de procesamiento de mediciones en forma de señales digitales, mediante modulación de ancho de pulso, PWM en inglés, o en señales analógicas mediante variaciones en el voltaje.

Dependiendo del área que se desee monitorear, se podrán colocar sensores en áreas estratégicas para obtener las mediciones esenciales; o se podrá realizar un despliegue extenso de sensores, obteniendo así información granular sobre el entorno monitoreado y poder destinar esfuerzos, o medidas correctivas de manera dirigida y eficiente para mantener niveles recomendados de seguridad en torno a los factores físicos de riesgo.

El bajo costo de estos sensores y el hecho de que todo sea basado en código abierto, podría permitir desplegar sensores de una manera menos prohibitiva que al usar soluciones cerradas y con elevados costos por unidad de medición.

Los sensores que captarán las mediciones de los factores físicos de riesgo estarán conectados a una placa electrónica que procesará estas lecturas, el nodo de procesamiento de mediciones y comunicación intermedia, que será denominado en adelante de manera simplificada como nodo.

Dependiendo de la cantidad de sensores que se desee implementar y de las distancias que los separen entre sí, se podrán desplegar varios nodos a donde se podrán conectar múltiples sensores.

La plataforma más influyente de esta tendencia en la actualidad es el conjunto de tarjetas fabricadas por Arduino, cuyo nombre fue colocado en honor al Rey Arduino de la Italia del 1002. Este nombre surgió curiosamente en una reunión de sus creadores en un bar que posee el mismo nombre de este rey (Kushner, 2011). Esta plataforma fue diseñada en el 2005, por el equipo de Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mellis. El objetivo era desarrollar una plataforma que publicara abiertamente los detalles de su arquitectura, diagramas electrónicos y esquemáticos de todas sus placas, de manera similar a los programas de código abierto, una tendencia que se ha mantenido durante algunos años (Barrett, 2012).

[2] HTTP: HyperText Transfer Protocol. Protocolo de la capa de aplicación del modelo OSI, montado sobre TCP/IP. Creado para la distribución de archivos de hipermedia. Ampliamente usado para responder a peticiones de páginas web en Internet desde 1990. (Fielding, et al., 1999)

Los nodos estarán basados en la plataforma Arduino. Se cargará un programa al microcontrolador de la placa, con la lógica necesaria para procesar las mediciones de los sensores y transmitirlas al servidor de recopilación.

Otro de los beneficios de implementar un sistema basado en una plataforma abierta como Arduino, reside en que se podrá modificar la lógica del sistema para permitir la interconexión de los nodos con múltiples tipos de sensores, o incluso con sistemas SCADA u otros sistemas de medición de terceros. Esto le dará al modelo presentado flexibilidad e interoperabilidad a la hora de hacer implementaciones en caso de que existieran sistemas previamente instalados.

A partir de entrevistas a expertos (Quintana, 2013) y de las recomendaciones de estandarización en torno a las instalaciones de TI (TIA, 2005) (BICSI, 2011), se pudo determinar que los entornos que se verían beneficiados del uso de este modelo de sistema poseen por lo menos de conectividad a través de una red LAN.

Es por ello que en este modelo, se utilizará una comunicación a través de una red de cableado Ethernet, realizando peticiones HTTP desde el nodo para enviar las mediciones hacia el servidor de recopilación. Esta comunicación estará basada en el protocolo TCP/IP, el cual se ha convertido en el estándar por omisión para las comunicaciones globales (Gokhale, 2004).

Para dotar a la placa Arduino de conectividad LAN, se le conectará una placa Ethernet, o shield como es denominado por Arduino. Esta será la responsable de permitir la conectividad y la comunicación del nodo en la red.

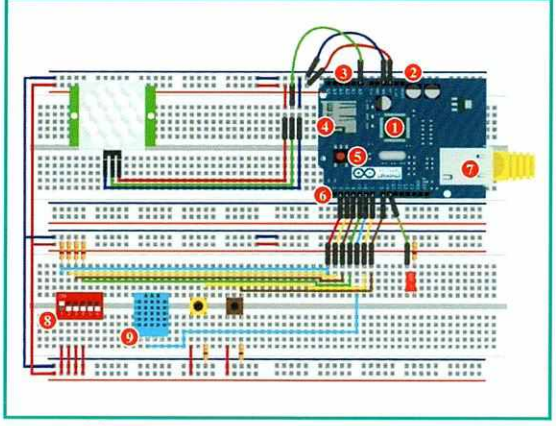
Cabe destacar que si existieran instalaciones satélites, como nodos de comunicaciones o equipos a la intemperie, la flexibilidad de tecnología en la que se basa este sistema de monitoreo permitiría su modificación para adaptarse a una gran variedad de entornos y condiciones de operación como puede ser observado en el sitio web del fabricante (Arduino, 2015), al utilizar módulos adicionales para extender su funcionalidad. Existen shields que permiten conectividad a través de Bluetooth, redes inalámbricas, enlaces de radio frecuencia, enlaces infrarrojos e incluso a través de redes GSM de telefonía.

Metodología

Este estudio propone la implementación de un sistema de telemetria utilizando herramientas de código abierto. Para el desarrollo del sistema se ha dividido en fases que son las siguientes: armado y programación del hardware, programación del servidor para la recolección de datos y finalmente se probará en conjunto el hardware y software en el sistema de telemetria.

La primera parte para el desarrollo del sistema de telemetria fue el armado y configuración del hardware, el cual consiste en conectar la placa (arduino) con el sensor de temperatura, que es el que se va a utilizar en la experimentación; como muestra la Figura 3, se puede observar el diseño del sistema de telemetría.

Figura 3.
Construcción del Nodo de procesamiento de mediciones y comunicación intermedia.



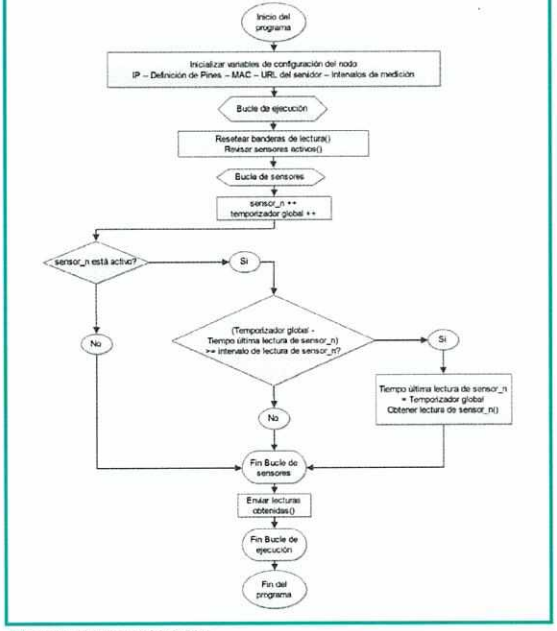
Fuente: *Elaboración Propia*

1. Placa Arduino UNO con el shield Ethernet montado sobre la misma.
2. Salidas de voltaje de 3.3V y 5V
3. E/S analógicas
4. Lector de tarjetas micro SD incorporado al shield Ethernet
5. Pulsador de reset
6. E/S Digitales
7. Puerto de conexión RJ45 del shield Ethernet
8. Banco de interruptores para controlar activación de lecturas de sensores
9. Sensor de temperatura / humedad relativa DHT11

Para obtener los valores de lectura de los sensores se desarrolló la lógica de la plataforma Arduino en lenguaje C/C++. Se escogió C/C++ porque puede integrarse de manera sencilla con las librerías de sensores de terceros, como en el caso del DHT11, el sensor de temperatura y humedad relativa. Para enviar los valores registrados por el sensor se crea una trama la cual es enviada a un servidor mendiante una petición http.

En la imagen a continuación se puede observar el diagrama de flujo del programa del Arduino:

Figura 4.
Diagrama de flujo del programa cargado en el nodo de procesamiento de mediciones y comunicación intermedia



Fuente: *Elaboración Propia*

De manera simplificada, este programa tendrá un bucle que recorrerá la cantidad de sensores que hayan sido configurados previamente en código. Todas las variables que manejan la información relacionada a los sensores estarán dispuestas en arreglos para que el bucle pueda recorrerlos.

La siguiente fase es el desarrollo del servidor para la recopilación y gestión de alertas. Este servidor se encargará de guardar la información enviada por el Arduino, lectura de los sensores y en caso de que un valor sobrepase el umbral setado, enviará un correo electronico (alerta).

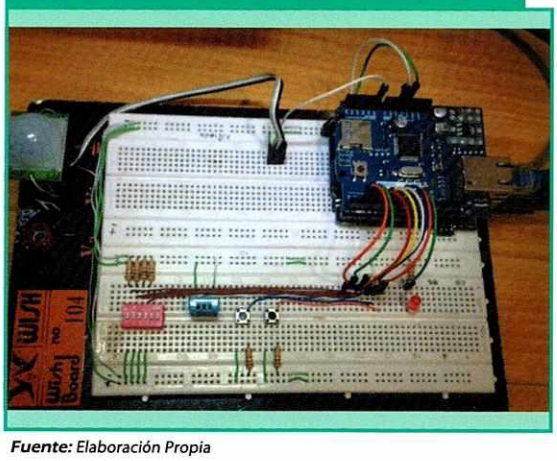
Para el desarrollo del servidor se utilizó herramientas de código abierto. Para el sistema operarativo se instaló la última versión del sistema operativo CentOS, el cual está basado en la misma arquitectura de RedHat Linux y es usado por grandes titanes tecnológicos como Facebook (Kerner, 2012).

El servidor tiene instalado el servicio de Apache, que recibe los datos de las lecturas provenientes de los nodos. Los valores de las lecturas se procesan por scripts programados en PHP y son invocados al realizar las peticiones HTTP al servidor web. Estos scripts son parte de la URL a la cual los nodos envian sus datos.

Para la interfaz de administración, se utilizó un entorno de desarrollo llamado Xataface. Este permitió crear un sitio web que con una interfaz gráfica de usuario para administrar una base de datos de manera automática partiendo de la estructura de datos de una base MySql (Xataface, 2015; González, Seone, & Robles, 2015).

Para finalizar el desarrollo se realizaron pruebas de integración entre el hardward (protoboard) y el software (Servidor) para validar el funcionamiento del sistema de telemetría realizado como se puede observar en la Figura 5.

Figura 5.
Modelo funcional de nodo de procesamiento de mediciones y comunicación intermedia



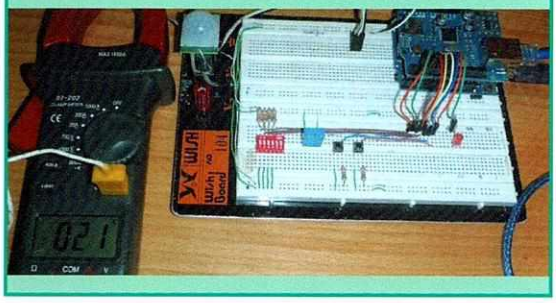
Fuente: *Elaboración Propia*

Para simular la generación de una alerta, se colocó el nodo de procesamiento de mediciones y comunicación intermedia en un entorno con una temperatura similar a la que se puede encontrar en un centro de cómputo. El cuarto fue enfriado mediante un aire acondicionado estándar hasta aproximadamente 20 grados centígrados.

Adicionalmente se colocó, al lado del sensor de temperatura y humedad relativa del nodo (ver figura 6), un termómetro externo basado en un termopar tipo K[3] para comparar las lecturas de temperatura registradas.

En la siguiente imagen se puede observar al nodo de procesamiento de mediciones y comunicación intermedia tomando las mediciones de temperatura y humedad relativa. Al lado se encuentra el termómetro de termopar tomando lecturas de manera similar.

Figura 6.
Lecturas de temperatura captadas por el termómetro dispuesto al lado del sensor de temperatura y humedad relativa.



Fuente: *Elaboración Propia*

Conjuntamente a la información mostrada por el medidor de temperatura externo, se observó la información de las lecturas de temperatura registradas por el sensor del nodo, a través del monitor de depuración del entorno de desarrollo de Arduino. Este monitor se conectó al nodo mediante una comunicación serial con la computadora.

Se confirmó que la lectura de temperatura en el monitor de depuración coincide con la temperatura reportada por el termómetro externo. También se comprobó el proceso de comunicación hacia el servidor de recopilación de datos y gestión de alertas mediante la petición HTTP. Como la lectura no excede el rango de valores configurados como seguros, el servidor de recopilación de datos y gestión de alertas no procederá con ninguna acción de envío de correos.

Luego se procedió a simular un aumento de temperatura en el ambiente, bajo el cual opera el sensor. Se introdujo un flujo de aire caliente, el cual en situaciones reales podría representar a equipos de cómputo con recalentamiento, un recalentamiento en el cableado eléctrico, o incluso el preámbulo a un posible incendio. Como instrumento para generar el aumento de temperatura, se utilizó un secador de pelo casero.

Posteriormente se pudo observar un aumento de temperatura respecto a las mediciones anteriores. De manera similar a la prueba anterior, estas mediciones coinciden

[3] Termopar tipo K: También se conoce como termocupla, es un transductor conformado por la unión de dos materiales. El termopar tipo K se compone de un cable de níquel-cromo y otro de níquel-aluminio. Posee una buena resistencia al óxido y un rango de temperatura de operación desde -200 °C a +1372 °C; posee una sensibilidad de 41μV/°C aproximadamente (Pallás, 2003).

con las lecturas capturadas por el termómetro externo.

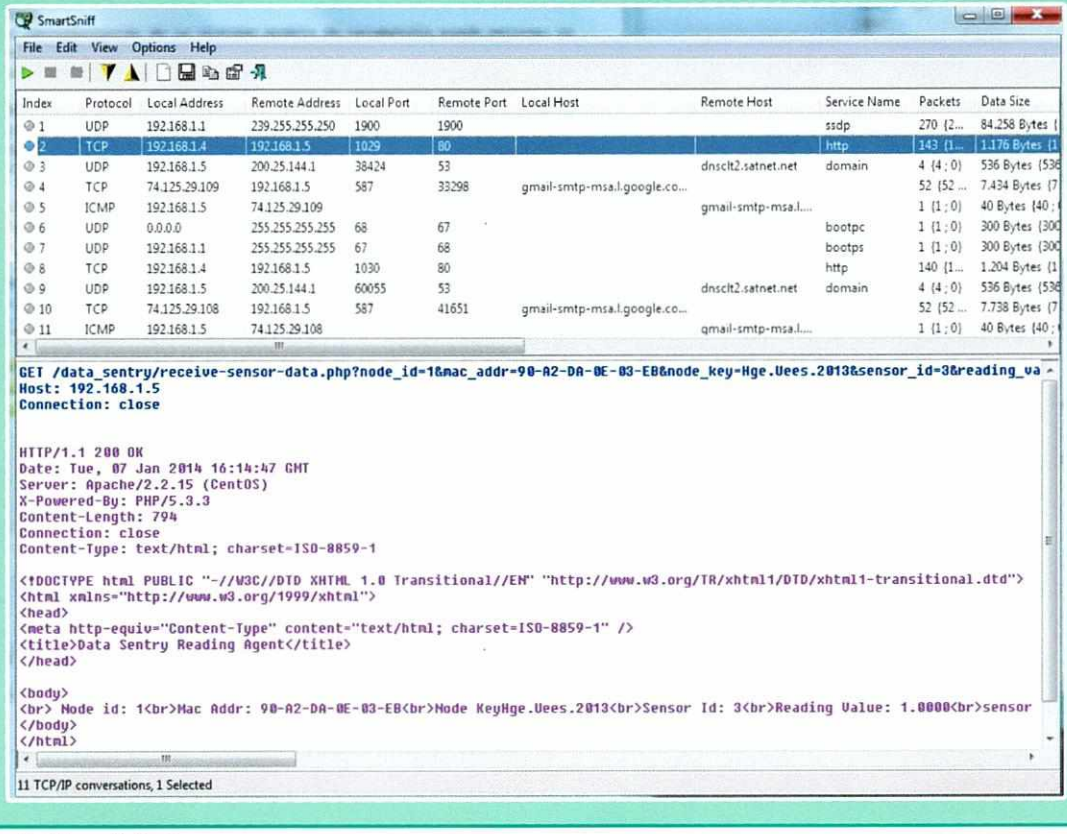
Dado que los valores capturados excedieron el rango seguro de temperatura que fue configurado para este sensor, al llegar la petición HTTP con los valores de estas lecturas al servidor de recopilación de datos y gestión de alertas, este realizó las acciones configuradas para este sensor y envió un correo de alerta.

Análisis de resultados

Para demostrar la comunicación a través de la red LAN entre el nodo de procesamiento de mediciones y el servidor de recopilación de datos y gestión de alertas, se utilizó un programa de captura de paquetes de red. Se denominará a este servidor de manera simplificada a partir de ahora, bajo el nombre de servidor de recopilación.

El programa utilizado fue SmartSniff, mismo que organiza la información de los paquetes, capturada de manera que se pueda observar el flujo de la comunicación. En la imagen siguiente, se puede mirar en azul el contenido de un paquete con una petición HTTP originado desde la dirección 192.168.1.4, la cual pertenece al nodo y la respuesta del servidor a esta petición mostrada en violeta.

Figura 7.
Interfaz del programa SmartSniff y Contenido de paquete enviado desde la dirección IP del nodo y respuesta del servidor a esta petición



Fuente: *Elaboración Propia*

Adicionalmente, se capturaron paquetes que muestran el proceso de conexión hacia gmail, por parte del servidor de recopilación. Esto sucede luego que el nodo envió una medición que generó una alerta, por lo que el servidor de recopilación ejecutará la acción

de enviar un correo electrónico a los usuarios configurados mediante el administrador web del sistema de telemetría.

Una vez determinada la IP del servidor de correos, se observó en la captura un conjunto de paquetes que contenían la comunicación, entre el servidor de Google smtp.gmail.com, con la IP 74.125.29.109 y el servidor de recopilación, 192.168.1.5.

Cabe mencionar que aunque el programa muestra el flujo de paquetes entre la dirección IP de Gmail y la dirección local del servidor web, existen mecanismos de comunicación intermedia y encapsulamiento de paquetes como en cualquier red de telecomunicaciones.

Conclusiones

Con el modelo desarrollado, se demostró la factibilidad de crear un sistema para monitorear factores físicos de riesgo, de bajo costo, utilizando tecnologías de código abierto. Se dotó a este modelo con la capacidad de alertar a las personas pertinentes sobre incidencias con un factor de riesgo cuando ocurra.

El modelo propuesto en esta investigación puede servir como base para otras áreas como seguridad, agricultura, medio ambiente, entre otras, dado que se le puede agregar mas sensores como: humedad, proximidad, movimiento, etc., con mucha facilidad.

Los sistemas de telemetría basados en plataformas de código abierto tienen un potencial que no ha sido explotado todavía. Esta tendencia tomará más fuerza gracias al surgimiento de nuevas placas electrónicas abiertas, a comunidades de aficionados a la robótica y a una creciente cultura DIY, o hágalo usted mismo en español.

Referencias bibliográficas

Arduino. (2015). Obtenido de Arduino: https://www.arduino.cc/

Barrett, S. (2012). *Arduino Microcontroller: Processing for Everyone!* Morgan & Claypool Publishers.

BICSI. (marzo del 2011). *Bicsi advancing the information and commnications tecnhonology community.* Obtenido de http://www.bicsi.org/uploadedfiles/BICSI_002_Sample.pdf

Bosworth, S., Kabay, M., & Whyne, E. (2012). *Computer Security Handbook, Set.* John Wiley & Sons.

Carden, F., Jedlicka, R., & Henry, R. (2002). *Telemetry Systems Engineering.* Artech House.

Cowan, C., & Gaskins, C. (2011). *Sales tools: APC Media.* Recuperado el 15 de enero del 2014, de APC Media: http://www.apcmedia.com/salestools/JMON-5ZLP8M/JMON-5ZLP8M_R3_EN.pdf

Fielding, R., Gettys, J., Mogul, J. C., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (junio de 1999). *Part of Hypertext Transfer Protocol -- HTTP/1.1.* Obtenido de http://www.w3.org/Protocols/rfc2616/rfc2616-sec1.html#sec1.4

Fresnel Tech. (2014). *Fresnel Lenses.* Recuperado el 2 de febrero del 2014, de http://www.fresneltech.com/pdf/FresnelLenses.pdf

Gokhale, A. (2004). *Introduction to Telecommunications.* Normal: Cengage Learning.

Golden, B. (2005). *Succeeding with Open Source.* Reading: Addison-Wesley Professional.

González, J., Seone, J., & Robles, G. (2015). *Materials: FTA.* Recuperado el 12 de enero del 2014, de Free Tecnology Academy: http://ftacademy.org/files/materials/fta-m1-intro_to_FS-v1.pdf

Izquierdo, A. (13 de noviembre del 2013). Opinión referente a seguridades en Centros de Cómputo en Guayaquil. (H. Gaibor, Entrevistador)

Kerner, S. M. (10 de septiembre del 2012). *Facebook Linux, What Distro is it?* Obtenido de Internet News: http://www.internetnews.com/blog/skerner/facebook-linux-CentOS.html

Kushner, D. (26 de octubre del 2011). *The Making of Arduino*. Recuperado el 30 de septiembre del 2013, de IEEE Spectrum: http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino

Ministerio de Educación. (2012). *Unidades Educativas del Milenio*. Obtenido de Ministerio de Educación: http://www.educacion.gob.ec/caracteristicas-de-las-uem.html

Mosquera de Calderón, S. (20 de noviembre del 2011). Educación digital. *La Revista (Edición Digital)*, http://www.larevista.ec/orientacion/orientacion/educacion-digital.

Pallás, R. (2003). *Sensores y acondicionadores de señal.* Barcelona: Marcombo.

Patranabis, D. (1999). *Telemetry Principles.* McGraw-Hill Education.

Peñarreta, N. (30 de octubre del 2013). Opinión referente a seguridades en Centros de Cómputo en Guayaquil. (H. Gaibor, Entrevistador)

Perdikaris, G. A. (1996). *Computer Controlled Systems: Theory and Applications.* Netherlands: Kluwer Academic Publishers.

Prasad, J., Jayaswal, M., & Priye, V. (2009). *Instrumentation And Process Control.* Bangalore: I. K. International Pvt Ltd.

Quintana, M. (2 de noviembre del 2013). Opinión referente a seguridades en Centros de Cómputo en Guayaquil. (H. Gaibor, Entrevistador)

TIA. (2005). *TIA-942 -Telecommunications Infrastructure Standard for Data Centers.*

Xataface. (2015). *About Xataface.* Recuperado el 2 de febrero del 2014, de Xataface web site: http://xataface.com/wiki/about

Hugo Gaibor Estupiñán

Ingeniero en Sistemas. Facultad de Sistemas, Telecomunicaciones y Electrónica de la Universidad Espíritu Santo - Ecuador

E-mail: hgaibor@uees.edu.ec

Ivan Silva Feraud

Ingeniero en Ciencias Computacionales. Máster en Ciencias Tecnológicas de la Computación. Docente tiempo completo de la Universidad Espíritu Santo - Ecuador.

E-mail: ivansilva@uees.edu.ec