

# Internet del Futuro: Middleware WSN-IP-WWW como Soporte de Ambientes Inteligentes

Benítez Juan<sup>1</sup>, Gloza Gonzalo<sup>3</sup>, Sosa Eduardo<sup>2</sup>, and Godoy Diego<sup>1</sup>

<sup>1</sup> Centro de Investigaciones en Tecnologías de la Información y Comunicaciones (CITIC). Universidad Gastón Dachary Av López y Planes 6519, 3300 Posadas

`juan.benitez, diego.godoy@citic.ugd.edu.ar`

<sup>2</sup> Secretaría de Investigación y Posgrado (SECIP). Facultad de Ciencias Exactas, Químicas y Naturales, Universidad Nacional de Misiones

Félix de Azara 1552, 3300 Posadas

`es@fceqyn.unam.edu.ar`

<sup>3</sup> `ggloza@gmail.com`

**Resumen** La Internet del Futuro y dentro de ella los Ambiente inteligentes abarcan tecnologías sumamente complejas, las cuales necesitan de interacción de manera totalmente transparente del usuario con las redes de sistemas computacionales. En las WSN encontramos una de esas tecnologías ya que una de sus principales características es la reducción del tamaño de los dispositivos o nodos sensores, para así lograr una disminución de los costos y poder desplegarlos en cualquier lugar con facilidad de forma invisible. Debido a esto, cada nodo tiende a tener únicamente el hardware estrictamente necesario para realizar de manera efectiva la tarea que se le ha encomendado. Sin embargo existen situaciones en que la conectividad que poseen los nodos sensores de una WSN (data céntricas) no son compatibles con las redes existentes tradicionales como Ethernet e IPV4 (nodo céntricas), por lo que se requiere un elemento capaz de realizar las interacciones necesarias de interconexión entre estas redes, para así lograr una homogeneización de los datos que fluyen por ellas y hacerlos accesibles por ejemplo a través de internet. En este trabajo se presenta un prototipo de dispositivo capaz de interactuar como middleware WSN-IP-WWW para la integración de redes de sensores inalámbricos con la Web.

**Keywords:** Nodos Sensores, Internet, WSN, Middleware.

## 1. Introducción

El ámbito de las redes de sensores y actuadores inalámbricos está en sus primeras etapas, con lo cual muchos aspectos se encuentran actualmente en fase de desarrollo temprano e investigación. Estos desarrollos son el campo aplicativo ideal de soluciones middleware.

La Internet del Futuro y dentro de ella los Ambiente inteligentes abarcan tecnologías sumamente complejas, las cuales necesitan de interacción de manera

totalmente transparente del usuario con los las redes de sistemas computacionales [7,12]. Esto favorece la integración de infraestructuras fijas, a nuevas tecnologías de “conectividad móvil”. Una Red de Sensores Inalámbricos se conforma de dispositivos integrados, incluyendo microprocesadores, radio, y sensores/actuadores varios. Así forman sistemas distribuidos proveyendo acceso ubicuo a la información, permitiéndonos participar activamente en la creación de “ambientes inteligentes” [11]. El alcance y diversidad de las aplicaciones de WSN, los requerimientos, diseños y plataformas son virtualmente ilimitados[1]. Las WSN representan la expansión de Internet actual al mundo físico real.

La realidad indica que los principales inconvenientes encontrados por las WSN tienen que ver con las diferentes metodologías de programación, como también con la heterogeneidad existente[6] tanto en hardware como en sistemas operativos.

Si la tendencia actual no declinara, los datos capturados por las WSN influenciarían el flujo de control de los procesos en tiempo real, e incluso podrían disparar nuevos procesos. Para lograr este nivel de interacción, las WSNs deben relacionarse, con el resto de las redes existentes como las redes Ethernet, las cuales hoy en día son las más utilizadas para el acceso a la WWW.

El desarrollo alcanzado por tecnologías de fabricación de circuitos integrados, tales como microcontroladores y dispositivos de lógica programable (FPGA), han aumentado el uso de las aplicaciones con sistemas embebidos[5]. Las mismas se han convertido en la solución a gran número de problemas que anteriormente requerían de una electrónica mucho más compleja, de mayor tamaño y mayor costo como por ejemplo una PC.

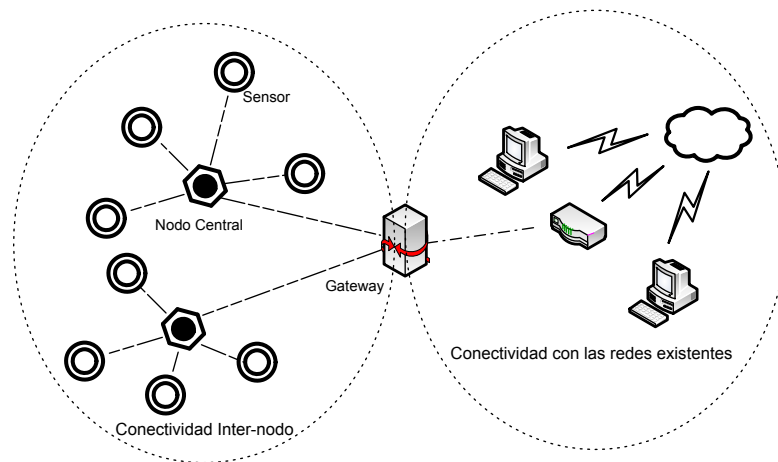
El middleware aquí implementado tiene como objetivo permitir la interacción entre las WSN y las redes IP. Considerando que la tendencia de los nodos de una WSN es la miniaturización, se debe aceptar la necesidad de limitar el hardware que puede llegar a contener cada nodo sensor, tanto por razones estéticas, como necesidad de bajos consumos de energía. De esta manera cada nodo contará únicamente con el hardware estrictamente necesario para poder realizar de manera efectiva la tarea que se le ha encomendado.

Por lo anterior consideraremos la conectividad a) entre nodos y b) con otras redes ya existentes (Ver Figura 1).

El despliegue de una red de sensores en un determinado escenario tiende a ser un proceso continuo, por ejemplo para reemplazar los nodos con baterías o nodos defectuosos que han sido objeto de las inclemencias ambientales.

Debido a su elevado número, los nodos deben operar sin supervisión después de la implementación. Esta tarea puede ser coordinada por una entidad externa conectada a la red de sensores, tales como un usuario con un PDA, un avión o algún dispositivo en Internet. Son concebibles asimismo redes aisladas, que están programadas para cumplir una tarea de detección determinada, cuyo resultado actuador controla nodos que son también parte de la red.

La necesidad de conectividad entre los nodos permite transferir información entre nodos vecinos, como así también nos permite realizar un encaminamien-



**Figura 1.** Conectividad en redes WSN

to multi-salto de los paquetes de información, reduciendo así, la potencia de transmisión de cada nodo y con ello el consumo energético.

Sin embargo estos datos brindados por la WSN son poco útiles a nivel de usuario y aplicaciones, si no podemos tener un acceso de forma sencilla a ellos, para lo que es necesaria la conectividad de las WSN hacia las redes existentes, tanto locales como Metropolitanas. Considerando las capacidades limitadas de hardware de los nodos, no es común encontrar nodos sensores, capaces de realizar los dos tipos de conectividad, inter-nodo y extra-nodo. Dentro de ellos encontramos a algunos con capacidad de manejo de paquetes IP[10].

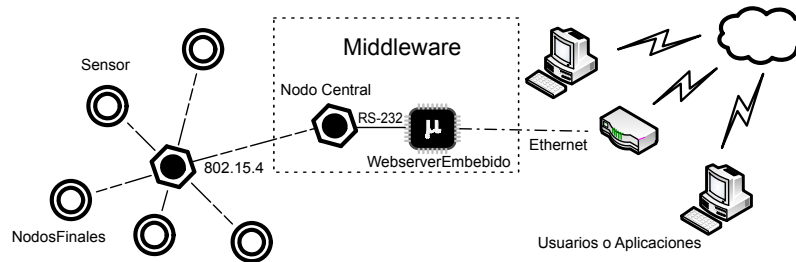
6LoWPAN[4] brinda un estándar a las WSN con fragmentación y compresión de las cabeceras de los paquetes IPv6 para ser transmitidos por 802.15.4. Esto no les permite la conexión directa a las redes externas a la WSN, lo que hace necesaria la utilización de una puerta de enlace capaz de direccionar los paquetes IPv6 de la WSN hacia el exterior. Si bien Existen variadas aplicaciones que brindan la posibilidad de publicar y manejar los datos de una WSN, requieren en todos los casos de una infraestructura de hardware ajena a la WSN para el envío de datos[8], concluyendo con la necesidad de la utilización de una estación de trabajo como ser una PC para poder publicar la información de la WSN.

Para ciertas aplicaciones de las WSN, se torna necesario el concepto de contar con un sistema autónomo remoto para recolectar datos y transmitirlos de manera transparente al usuario, como por ejemplo el escenario de una WSN ubicada para el control de radiaciones peligrosas para un ser humano.

## 2. Arquitectura Propuesta

Para el presente trabajo se ha planteado un prototipo de middleware WSN-IP-WWW para así proveer una solución a la necesidad de contar con una estación

de trabajo para que la WSN pueda publicar los datos recogidos a través de una página web. En la Figura 2, se puede apreciar la arquitectura propuesta, donde se detalla la integración del sistema middleware entre las redes WSN y las demás redes a través del protocolo Ethernet.



**Figura 2.** Arquitectura Middleware Propuesta

Podemos considerar al sistema en cuatro partes fundamentales:

- **Nodo central (Sumidero):** es un nodo perteneciente a la WSN, el cual es responsable de capturar todas las comunicaciones de la red, enviadas o consultadas por un usuario o aplicación. Este nodo central recoge la información proveniente de los nodos de la WSN. El direccionamiento de los paquetes se realiza por medio de las direcciones físicas de cada dispositivo (MAC). Con dicha información de direccionamiento se determina desde que sección de la WSN fue recibido el paquete de datos, paquete que procesado por el middleware se encargara de extraer la información útil y de empaquetarla dentro de una estructura de datos nueva para ser enviada hacia el servidor http.
- **Servidor Web:** es el encargado de lograr la interconexión entre la red WSN y la red IP mediante la implementación de una pila de protocolos TCP/IP. Incorpora en sí mismo un servidor web embebido que permite visualizar el estado de la red WSN mediante un navegador. Es capaz de generar contenidos dinámicos a partir de la incorporación de llamadas a funciones en lenguaje C desde el código de las páginas web. La comunicación se ha establecido por medio de una conexión serial, protocolo RS-232, mediante tramas propiamente definidas
- **Nodos finales:** son nodos dentro de la red WSN que se comunican con protocolo IEEE 802.15.4. Están equipados con distintos sensores (temperatura, humedad y presión). Cada nodo se comunica directa y únicamente al nodo central. Estos nodos realizan una lectura periódica de sus variables a medir y envían dicha información hacia el nodo sumidero.
- **Usuarios y/o Aplicaciones:** son los consumidores de toda la información que entrega el middleware.

### 3. Especificaciones e Implementación

#### 3.1. Red WSN

El nodo gateway, como así también los nodos finales de la WSN se han implementado utilizando la plataforma iSense[3](Figura 3), la cual cuenta con un procesador JennicJN5139, un sistema radial de 2,4 GHz compatible con normas IEEE 802.15.4, con ancho de banda estimado de 250kbit/s, sistema de criptografía por hardware AES, capacidad de cálculo de tiempo de vuelo (ToF) para cálculo de distancias entre nodos, 192kB de ROM, 96kB of RAM, como una variada posibilidad para la utilización de periféricos analógicos y digitales.



**Figura 3.** Nodo WSN implementado en el proyecto

El escenario elegido ha sido aquel donde se determinaran valores de la vida real como temperatura, humedad relativa y presión atmosférica.

Además de la plataforma iSense, cada nodo sensor incorpora un módulo “ambiental” capaz de realizar las mediciones de las variables antes mencionadas.

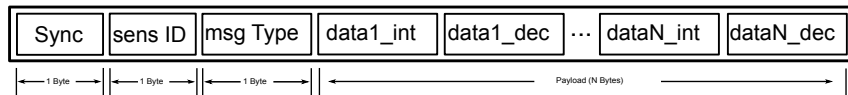
Un atributo importante de este módulo es el reducido consumo de energía en estado de reposo (inferior a 1uA), prolongando así la vida útil del equipo. Las características se observan en el Cuadro 1.

#### 3.2. Comunicación RS-232

La comunicación entre el nodo sumidero y el servidor web se establece por protocolo RS-232, proveyendo un enlace bidireccional a ambos dispositivos para el intercambio de datos. El nodo sumidero con su función de concentrador identifica la información recibida por medio del enlace 802.15.4 y genera un paquete de datos con el siguiente formato(Figura 4).

**Cuadro 1.** Características sensores iSense

Características Nodo Ambiental iSense	
Rango Utilización (T/Hr/P)	-20°C a 70°C / 0 a 100 % / 10 a 1100 mbar
Precisión (T/Hr/P)	1°C / 3 % / 1,5 mbar
Resolución (T/Hr/P)	0,1°C / 0.1 % / 0.1 mbar
Consumo normal	800uA
Consumo stand-by	0.5uA

**Figura 4.** Paquete de datos generado

- Sync: secuencia de inicio de trama. Es una secuencia de 8 bits la cual indica al webserver el inicio de una trama de datos.
- sens ID: representa a los últimos 8 bits de la dirección Mac del sensor que envió la información para ser representada en el web server.
- msgType: representa al tipo de mensaje que se envía, donde puede ser mensaje de información normal, mensaje con prioridad y alarma.
- Los siguientes campos representan la carga útil de la trama a ocupar por las determinaciones de cada nodo.

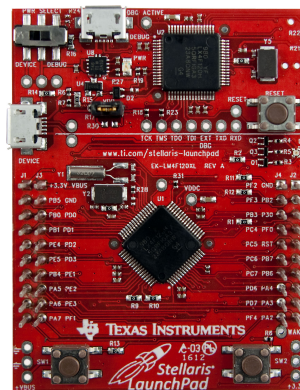
El nodo sumidero trabaja bajo demanda, lo que quiere decir que, únicamente cuando llegue un paquete procedente de la WSN se enviarán los datos hacia el web server. Esto se implementa a través de interrupciones en el despachador del sistema operativo. Este manejador de recepción monitorea el enlace inalámbrico continuamente en búsqueda de un paquete de datos destinado a la dirección MAC de su hardware `os().id()`; o a la dirección de broadcast del sistema. Una vez que encuentra dicha coincidencia, llama a una interrupción que llamará al método de recepción para que procese dicho paquete de datos.

### 3.3. Conectividad IP

Para la implementación del servidor web se utilizó una placa de desarrollos Stellaris Launchpad de Texas Instrument[9].

La interfaz está conformada por un micro controlador ARM Cortex M4 LM4F120H5QR con punto flotante, capacidad de trabajo de hasta 80MHz, 256Kb de memoria Flash, 32Kb de SDRAM, 8 transceptores asíncronos universales(UART), 4 buses de intercomunicación en serie I2C, 4 interfaces seriales periféricas, 27 Timers, interfaz USB para ICDI (In CircuitDebug Interface), Micro B USB para debug, Micro B USB para conexión con aplicaciones, 2 botones para aplicaciones de usuario, botón de reset y RGB LED (Figura 5)

El acceso desde/hacia redes TCP/IP es realizado mediante un servidor web embebido en dicha plataforma. La conectividad TCP/IP se logró implementando



**Figura 5.** TI Stellaris Launchpad

la librería uIP, la cual es una aplicación opensource en lenguaje C para microcontroladores de la pila de protocolos TCP/IP, cuya principal particularidad es la minimización de la memoria utilizada. Ésta soporta los protocolos TCP, UDP, IP, ICMP y ARP

La librería uIP provee una interfaz de programación de aplicación (API) que define como las aplicaciones interactúan con la pila TCP/IP. Dicha API se denomina “protosockets”, la cual es muy similar a la API de sockets Berkeley[2] pero sin la sobrecarga de los entornos multitareas.

Ciertas modificaciones se realizaron para portar la stack uIP de manera adecuada a la arquitectura ARM, para evitar incompatibilidades en cuanto a los tipos de datos y temporizadores requeridos en la implementación.

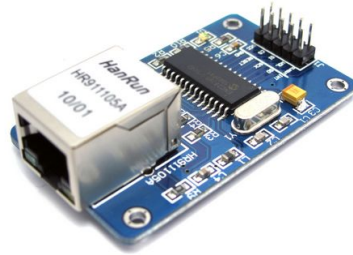
Mientras que la librería uIP proporciona conectividad TCP/IP, las funcionalidades de la capa de enlace de datos se implementó mediante el controlador Ethernet ENC28J60 de Microchip (Ver Figura 6), añadido como periférico en modo esclavo a la plataforma Stellaris mediante un bus SPI en modo 0,0.

Un diagrama temporal del funcionamiento del bus junto con sus señales se muestra en la Figura 7.

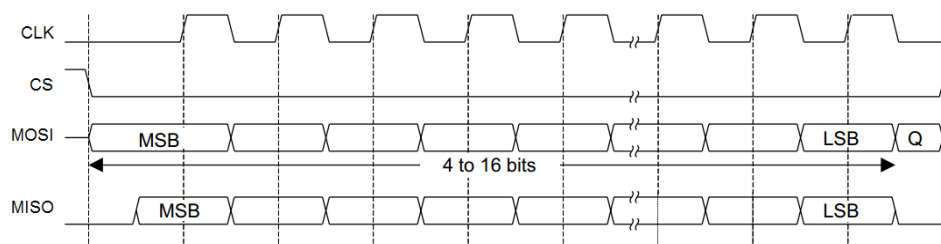
El bus SPI incluye una línea de reloj CLK de sincronización, dato entrante (MOSI), dato saliente (MISO) y un pin de chip select (CS), que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse.

### 3.4. Servidor Web

Esta herramienta permite consultar la información de la red de sensores desde la red IP. Para ello se ha implementado un servidor web embebido capaz de generar contenido dinámico a partir de la incorporación de llamadas a funciones en lenguaje C desde el código de las páginas.



**Figura 6.** Controlador ENC28J60



**Figura 7.** Diagrama temporal del bus SPI

Cuando se recibe información de la red WSN por medio del enlace RS-232, esta se clasifica y se almacena en un arreglo de estructuras que definen la información requerida por el usuario, como ser el ID del sensor que envió la información, los valores de temperatura, humedad, presión, el estado de batería de los mismos, como así también banderas de alarmas de las variables censadas. Toda esta información, ya almacenada, se genera de forma dinámica en la página web solicitada.

La inserción dinámica de contenido en una página solicitada del servidor web es posible mediante el conjunto de directivas SSI (Server Sides Includes). SSI hace uso de etiquetas especiales que informan al servidor que necesitan ser reemplazadas por otro contenido. El servidor es capaz de procesar solicitudes de páginas html, css y shtml, con capacidades de representar imágenes en formato jpg, gif y png.

Los contenidos están asociados a un tipo de encabezado html, a partir del cual el cliente (navegador) es capaz de procesar la información recibida. Las páginas web, así como otro tipo de información, son almacenadas en la memoria ROM del procesador ARM.

Las etiquetas para llamadas SSI tienen el siguiente formato:

```
<!--#execcgi=\NombreFuncion"-->
```

Ingresando la dirección IP del servidor web en un navegador realiza una petición de un recurso conteniendo etiquetas SSI, el servidor necesita realizar



una copia de la página HTML en la memoria RAM hacia el buffer de salida, con el objetivo de poder agregar la información de las variables enviadas por la red WSN en forma de contenido dinámico y generar el código HTML a ser enviado. Mientras esto sucede, el servidor determina si existen directivas SSI, para determinar la función C necesaria para procesar dicha directiva, recibiendo como argumento un puntero del actual buffer de salida. La función entonces puede insertar dinámicamente el contenido correspondiente por medio del envío de caracteres ASCII al buffer.

Además de directivas SSI, el servidor es capaz de procesar scripts CGI, permitiendo utilizar formularios y botones HTML. Además del formato HTML, el middleware puede generar contenido en formatos interoperables como ser RSS y JSON, esto permite dar soporte a posibles aplicaciones como ser de ambientes inteligentes, que consuman dichos formatos, realizando las consultas correspondientes.

#### 4. Pruebas y Resultados Obtenidos

En la verificación formal del sistema, primero se procedió a la verificación de la pila TCP/IP implementada en conjunto con la aplicación del webserver corriendo en la plataforma Stellaris Launchpad. En el siguiente cuadro se describe el escenario en el cual se realizó las distintas pruebas de conectividad.

**Cuadro 2.** Configuración de Red

Red	192.168.1.0/255
IP Servidor	192.168.1.110
IP Cliente	192.168.1.103
Máscara de Red	255.255.255.0
Gateway	192.168.1.1
Velocidad de Vínculo Negociada	10 Mb/s

Una forma simple de testear el estado de comunicación entre hosts en una red IP es la utilidad Ping, la cual emplea paquetes ICMP (capa de red) de solicitud y respuesta. Mediante esta utilidad se puede diagnosticar el estado, velocidad y calidad de una red determinada.

Los resultados obtenidos mediante esta prueba se muestran a continuación.

```
root@kali:~/Documents/stellaris/enc28j60_uip# ping 192.168.1.110
PING 192.168.1.110 (192.168.1.110) 56(84) bytes of data:
64 bytes from 192.168.1.110: icmp_req=1 ttl=64 time=2.30 ms
64 bytes from 192.168.1.110: icmp_req=2 ttl=64 time=2.29 ms
64 bytes from 192.168.1.110: icmp_req=3 ttl=64 time=2.74 ms
```

Este tipo de prueba es una garantía razonable de que las capas inferiores de la stack funcionan correctamente, donde además se puede observar de los resultados anteriores que la misma responde de manera estable dado los valores casi constantes de los tiempos de respuesta.

Si bien la prueba anterior demuestra el funcionamiento de la pila de protocolos de capas inferiores, para determinar el correcto funcionamiento de la stack, incluido protocolos de nivel superior como TCP o UDP, se empleó la herramienta Iperf.

Iperf crea flujos de datos TCP y UDP entre el host y el servidor. Además permite al usuario ajustar varios parámetros que pueden ser usados para hacer pruebas en una red, o para optimizar y ajustar la red. Iperf puede funcionar como cliente o como servidor y puede medir el rendimiento entre los dos extremos de la comunicación, unidireccional o bidireccionalmente.

Típicamente la salida de Iperf contiene un informe con marcas de tiempo con la cantidad de datos transmitidos y el rendimiento medido.

En este caso, el servidor fue configurado en un host dentro la red IP utilizada para realizar las pruebas.

Mediante el mismo host se estableció la comunicación entre el servidor y el servidor web embebido, intercambiando un flujo de datos TCP sobre el puerto 80, pudiendo además determinar el throughput entre el host y el servidor. Los resultados obtenidos en este test se muestran a continuación:

```
root@kali:~/Documents/stellaris/enc28j60_uip# iperf -c 192.168.1.110 -i 1 -t 10 -p 80 -w 64k
-----
Client connecting to 192.168.1.110, TCP port 80
TCP window size: 128 KByte (WARNING: requested 64.0 KByte)
-----
[ 3] local 192.168.1.103 port 45972 connected with 192.168.1.110 port 80
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec   245 KBytes    2.01 Mbits/sec
[ 3] 1.0- 2.0 sec   244 KBytes    2.00 Mbits/sec
[ 3] 2.0- 3.0 sec   244 KBytes    2.00 Mbits/sec
```

El banco de prueba para este sistema consistió en una red de diez nodos WSN, los cuales fueron configurados con sus respectivos módulos para medir variables como temperatura, humedad y presión de forma periódica cada 30 minutos, enviando dichos datos al nodo central, conectado por RS-232 a la plataforma Stellaris Launchpad. Estos valores fueron monitoreados desde una computadora conectada en la misma red a la que pertenece el servidor http, mediante un navegador web según se observa en la captura de pantalla indicada en la Figura 8.

Además se simularon condiciones de alarma, de manera de determinar si el sistema es capaz de responder a las mismas, mostrando dicha condición en el navegador web.

En cuanto a la pruebas realizadas como soporte a los ambientes inteligentes, se han tomado como escenario dos proyectos.

El primero de ellos es un proyecto denominado Prototipo para monitoreo eficiente de iluminación basado en WSN utilizando HTML5 para ambientes inteligentes, cuyo objetivo general es el diseño de un prototipo de software y adaptación de hardware para monitoreo de iluminación basado en WSN utilizando HTML5.

En este sentido tomando como escenario las instalaciones de una Institución Educativa, las WSN serían útiles para realizar la captura de los datos relacionados a la iluminación dentro de las aulas donde se imparten las clases y el

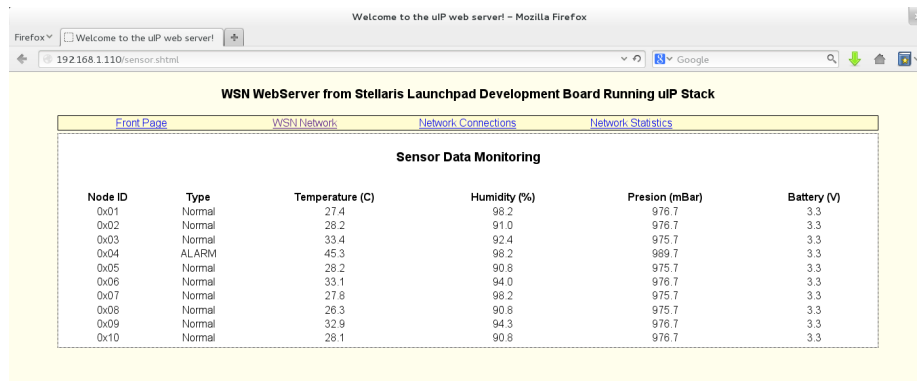


Figura 8. Monitoreo de Sensores via WWW

correspondiente tránsito que se genera en estas durante una jornada completa. Si bien existen soluciones que utilizan WSNs para el monitoreo de valores la posibilidad de contar con una gráfica sobre un plano de planta de un edificio de cómo están ocurriendo los fenómenos físicos basado en una solución que sea multiplataforma utilizando HTML 5 la última tecnología WEB y aprovechando los datos arrojados por el middleware propuesto en este trabajo sería de gran utilidad para en un proceso posterior, actuar en consecuencia.

El segundo proyecto, tiene como objetivo implementar una WSN, para la monitorización ambiental en procesos industriales en los establecimientos elaboradores de té negro (Secaderos) en el litoral. En ambos casos el intercambio de la solución por defecto la cual es la utilización de un nodo sensor sink, conectado a una pc a través de una interfaz USB por el middleware propuesto en este trabajo, se ha realizado de forma transparente para las aplicaciones de software para ambientes inteligentes que se mencionaron anteriormente.

## 5. Conclusiones y Trabajos Futuros

En este trabajo se presentó un modelo de interoperabilidad middleware WSN-IP-WWW para la Integración de Redes IP y la Web con Redes de Sensores Inalámbricos para su aplicación en Ambientes Inteligentes con el fin de generar agentes con capacidades para percibir el mundo, tomar decisiones y actuar sobre el entorno. La arquitectura de hardware y software propuesta permite que se cumpla satisfactoriamente el objetivo propuesto. Además con la posibilidad que brinda el middleware de publicar la información a través de formatos interoperables, ayuda a la futura integración con otras aplicaciones y en otros ambientes.

Los principales desafíos a enfrentar, dentro de una solución que debe ser resuelta de una manera completamente escalable; se refieren a aspectos tales como el número de nodos sensores, ancho de banda exiguo, administración de recursos y las cuestiones relacionadas con la gestión de la energía. Para que un sistema

sea aplicable en la vida real, el sistema subyacente debe ser capaz de lidiar eficazmente con la comunicación distribuida no fiable y dinámica, limitaciones de potencia de los dispositivos inalámbricos, el fallo de los dispositivos de hardware en entornos hostiles y la asignación de las tareas de procesamiento remoto distribuidos a lo largo de la red inalámbrica.

Es imprescindible avanzar en el estudio de algoritmos para sistemas con gran cantidad de nodos, conjuntamente con la optimización de comunicación entre pares, puertas de enlace y sumidero.

En base a los resultados y al prototipo establecido, la experiencia ha demostrado la importancia de la investigación relacionada con WSN en nuestro país convirtiendo a la tecnología WSN en una herramienta indispensable de Internet del futuro e Internet de las Cosas.

## Referencias

1. Arampatzis T., Lygeros J. y Manesis S., A Survey of Applications of Wireless Sensors and Wireless Sensor Networks, de Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation (2005).
2. Berkeley sockets, University of Princeton, (2013). [En línea]. Available: <http://goo.gl/Z4GwL>.
3. Coalesenses GmbH, Bridging the Gap Between Virtuality and Reality, [En línea]. Available: <http://www.coalesenses.com>. [Último acceso: sep 2011]
4. Internet Engineering Task Force (IETF), IPv6 over Low power WPAN (6lowpan), (2013). [En línea]. Available: <http://goo.gl/eMyk1>
5. Mayer-Lindenberg F., High-Level FPGA Programming through Mapping Process Networks to FPGA Resources, de International Conference on Reconfigurable Computing and FPGAs, (2009).
6. Mottola L. y Pietro P., Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art, ACM Computing Surveys, vol. 43, n°4, (2010)
7. Sosa E., Tesis Doctoral: Contribuciones al Establecimiento de una Red Global de Sensores Inalámbricos Interconectados. Universidad de La Plata (2011).
8. Sosa E., Fischer S. y Díaz F., Twitter, Soporte de una Red de Sensores Inalámbricos, de XII Reunión de Otoño de Potencia, Electrónica y Computación, Colima, Mexico, (2010).
9. Texas Instruments, Stellaris LM4F120 LaunchPad Evaluation Kit, (2013). [En línea]. Available: <http://www.ti.com/tool/ek-lm4f120x1>.
10. Teubler T., Hail M. y Hellbruck H., Transparent Integration of Non-IP WSN into IP Based Networks, de 8th International Conference on Distributed Computing in Sensor Systems, (2012).
11. Tilak S., Abu-Ghazaleh N. and Heinzelman W. , A Taxonomy of Wireless Micro-Sensor Network Models. ACM Mobile Computing and Communications Review. Vol. 6, pag. 28-36 (2002).
12. Tselentis G., Galis A., Gavras A., Krco S., Lotz V., Simperl E., Stiller B. y Zahariadis T. - Towards the Future Internet - Emerging Trends from European Research, (2010) ed., vol. 1, Hamburg.