Ambientes Inteligentes: Middleware de Soporte para la Captura, Almacenamiento y Publicación de Datos de una Redes de Sensores Inalámbricos.

Diego A. Godoy[1] , Eduardo O. Sosa[1,2], Raúl Neis[1], Rebeca Díaz Redondo[3]

[1] Universidad Gastón Dachary, Centro de Investigación en Tecnología de la Información y Comunicaciones, Departamento de Ingeniería y Ciencias de la Producción.
[2] Secretaría de Investigación y Posgrado (SECIP). Facultad de Ciencias Exactas, Químicas y Naturales, UNaM
[3]Departamento de Ingeniería Telemática. E. E. Telecomunicación. Universidad de Vigo
{diegodoy, eduardo.sosa, rneis}@citic.ugd.edu.ar,
rebeca@det.uvigo.es

Resumen. En este trabajo se presenta un middleware para la captura, almacenamiento y publicación de datos obtenidos de redes de sensores inalámbricos para su utilización en aplicaciones de Ambientes Inteligentes e Internet de las Cosas. Se ha desarrollado un framework para abstraer las tareas de captura, almacenamiento y Publicación. Se han llevado a cabo pruebas de la utilización del framework en cuanto a su funcionalidad y versatilidad de consultas de los distintos datos generados por la red de sensores inalámbricos.

Palabras Claves: Ambientes Inteligentes, Redes de Sensores Inalámbricas, Fameworks.

1 Introducción

Las consecuencias de la ley de Moore se reflejan cada vez más en la vida cotidiana. El aumento de la capacidad de cómputo, la miniaturización de los componentes y el abaratamiento de los dispositivos hacen posible cada vez más la visión de la Internet de las Cosas (IoT, Internet of Things) [**1**] y de la Inteligencia Ambiental [**2**], en la que los objetos cotidianos contarán con sensores conectados a Internet.

El desarrollo de esta visión promete grandes oportunidades de aprovechamiento económico e industrial, y a la vez representa un enorme desafío tecnológico. Como parte del desarrollo de las tecnologías de la IoT, las Redes de Sensores Inalámbricos (WSN, Wireless Sensor Network) [**3**] son un área de álgido estudio y desarrollo [**4**]. Hoy en día es factible pensar en WSNs con miles de nodos, recolectando datos en infinidad de entornos, para todo tipo de fines y a un costo relativamente razonable.

Debido a que la Web es una de las tecnologías más utilizadas de Internet, resulta natural que las Redes de Sensores se integren a la red de redes utilizando esa tecnología.

La visión más pura de la IoT exige que cada nodo de la red esté conectado directamente a Internet, recolectando y proveyendo información de acuerdo a su fin

específico. Sin embargo, esta arquitectura directa resulta poco viable actualmente, debido a que requiere que cada nodo ejecute un servidor web que se conecte a Internet. La capacidad de cómputo, de energía, y de almacenamiento disponibles en los sensores es muy limitada como para permitir esto. Además, debido a la reducida capacidad de almacenamiento, no siempre resulta viable almacenar las mediciones históricas realizadas por el sensor.

A causa de tales restricciones, en una arquitectura típica de WSN actual, los nodos se comunican entre sí mediante algún protocolo interno (probablemente optimizado para dispositivos con capacidades limitadas) y los datos obtenidos por estos se propagan hasta un nodo 'colector' o sink node. Este último se conecta indirectamente a Internet a través de un Gateway.

El Gateway, que puede ser una computadora corriente o un hardware específico, realiza las labores de almacenamiento y publicación en Internet. Además puede ejecutar aquellas tareas de agregación, filtrado y corrección de datos que resultan computacionalmente costosas de llevar a cabo en los sensores.

Esta arquitectura indirecta presenta sus propios problemas. Por un lado, dada la diversidad de tipos de sensores, el Gateway debe lidiar con la conexión física propia de cada WSN (RS232, USB, IEEE 802.3), y con la interpretación de los datos recibidos desde la red, esto es, la manera en la que deben parsearse los datos obtenidos. Paralelamente, existe un gran abanico de formatos de publicación de datos en la Web (SOAP, RPC, REST, RSS, XML, JSON, CSV), que representan un desafío para la flexibilidad del sistema. En cuanto al almacenamiento, la diversidad de escenarios existentes dificulta la reutilización de código en este sentido debido a la multitud de variables que pueden censarse en una WSN. Cualquier implementación particular requiere de modificaciones considerables ante el cambio de alguna de las variables sobredichas, y resulta difícil la reutilización de código.

Actualmente existen soluciones específicas para resolver el problema del almacenamiento y publicación, como se verá en los trabajos relacionados (Apartado 2), pero ninguna plantea una solución abstracta del proceso completo, desde la captura de los datos de la WSN hasta su publicación en Internet. Es por ello que en este trabajo se presenta el diseño e implementar de un prototipo de Middleware en forma de framework para la captura, el almacenamiento y la publicación de datos de Redes de Sensores Inalámbricos, orientado a la visión de la IoT.

El trabajo está estructurado de la siguiente manera: En el apartado 2 se presentan trabajos e implementaciones de middlewares relacionados, en el apartado 3 se describen las metodologías y tecnologías utilizadas y en apartado 4 se presenta el diseño del middleware propuesto para la captura, almacenamiento y publicación. Seguidamente en el apartado 5 se muestran las pruebas realizadas sobre el middleware y finalmente en apartado 6 se presentan las conclusiones y trabajos futuros.

2 Trabajos Relacionados

Twitter, Soporte de una Red de Sensores Inalámbricos: Es un trabajo en el cual se utiliza la red social Twitter para visualizar las publicaciones realizadas por una

WSN. De esta manera, con solo subscribirse a la cuenta que realiza la publicación, se pueden visualizar desde cualquier lugar en el mundo donde se tenga acceso a Internet, tanto de una PC o notebook como de un SmartPhone, los mensajes que emiten los sensores [**5**].

Web Messaging for Open and Scalable Distributed Sensing Applications: Propone pautas para la construcción de una API para publicar datos de WSNs basada en REST. Realiza un estudio del estado del arte de las APIs más conocidas y propone una solución que aprovecha las ventajas de cada una. El trabajo está totalmente centrado en la interfaz de publicación, y no aborda aspectos internos del servidor, ni la obtención de datos desde la WSN [**6**].

Thingspeak: Es una aplicación en línea que permite publicar colecciones de datos obtenidos de sensores (no es necesario que el origen sea una WSN). Brinda una API que permite la publicación de los datos, así como herramientas para la confección de gráficos y la posibilidad de desarrollar plugins para extender la funcionalidad del sitio. La API de publicación y consulta de datos está basada en REST, y permite recibir datos en formato XML, JSON o CSV [**7**].

Cosm: La aplicación web Cosm (antes Pachube) permite la conexión entre dispositivos y aplicaciones, proveyendo un servicio de almacenamiento de datos en tiempo real, y una API de publicación y consulta. Una característica que destaca de la empresa es que ofrece hardware para el censado y la publicación de datos, por lo que los tiempos de implementación pueden reducirse en gran medida si se implementan utilizando esta solución. A diferencia de Thingspeak, la aplicación web de Cosm no es de código abierto. A la fecha de consulta el uso del servicio es gratuito, pero los términos de uso establecen que la empresa se reserva el derecho de establecer tarifas para el uso del servicio o de nuevas características [**8**].

3 Materiales y Métodos

En primera instancia se realizó la búsqueda de información relacionada a los términos claves de WSN a fin de profundizar el conocimiento del funcionamiento de los sensores que se utilizarán, haciéndose mayor hincapié en aspectos relacionados con la comunicación de los sensores con el Gateway, para los sensores que se utilizaron y algunas otras variantes populares de hardware. Paralelamente se estudiaron conceptos más detallados relacionados a IoT, especialmente los relacionados a las necesidades de obtención de datos en tal contexto, utilizando como referencia la arquitectura propuesta por la ITU [**9**].

En una segunda etapa, se definieron los requerimientos que debe cumplir el diseño del middleware, tanto en términos de captura como de almacenamiento y publicación de datos. Esto dará complimiento al tercer objetivo del trabajo.

Luego, se realizó el diseño del middleware, utilizando diagramas de UML como herramientas de modelado y patrones de diseño [**10**]. El diseño abstrae aspectos comunes a todos los escenarios de IoT, procurando una estructura flexible y desacoplada.

Posteriormente se procederá a desarrollar un prototipo que implemente el diseño del middleware realizado en la etapa anterior. Como se mencionó en el alcance, el prototipo utilizará sensores de marca ISense, fabricados por la empresa Coalesenses [**11**], programados en C++. Para el desarrollo se utilizarán como lenguajes C y PHP y para la persistencia se utilizarán los motores MySQL y PosgreSQL. Todo el prototipo se ejecutará en entornos GNU/Linux. Se prevé la programación de dos alternativas de bases de datos, para poder demostrar la flexibilidad del diseño en ese sentido, y con el mismo fin, se programarán algunos formatos de publicación.

El prototipo toma los datos relevados por la WSN a través del nodo sink, conectado a una computadora de escritorio (Gateway) mediante una conexión USB. Luego se realiza un parseo y el posterior almacenamiento en una base de datos para finalmente publicarlos en la red local del Campus de la UGD y en Internet.

Finalmente, se montaron dos redes con los sensores cada una con una con 10 nodos, que generan datos de temperatura ambiente, humedad relativa y presión atmosférica, ubicadas en puntos geográficos distintos, una en el Campus de la Universidad Gastón Dachary y otra, en la Facultad de Ciencias Exactas de la Universidad Nacional de Misiones. Se pretende dar la posibilidad de publicar datos tanto en redes locales como en Internet. También se pueden variar los formatos de publicación, las variables medidas (temperatura, humedad y presión) y los motores de base de datos a fin de demostrar la flexibilidad del diseño. Debido a que los sensores con los que se cuenta no poseen sensores de geolocalización, estos datos serán simulados, a fin de poder confeccionar un mapa para la visualización de los datos.

4 Diseño del Middleware de Captura, Publicación y Almacenamiento

El diseño cuenta con cuatro módulos. El módulo Sensor se encargará de la obtención de datos desde la WSN. El módulo Parser encargado de realizar el trabajo de interpretación de los mensajes obtenidos de la WSN y de construir, con la información obtenida, un objeto de datos que abstrae la estructura de datos del dominio, estos módulos se utilizan en la fase Captura. El Módulo de Almacenamiento permitirá la persistencia y consulta de los datos. Por último, el módulo Publicador realizará las labores de publicación en la Web.

Cada módulo cuenta con una o más clases abstractas que generalicen el comportamiento común en cada etapa, y diversas clases que implementen las cuestiones específicas de cada escenario (tipo de sensor, formato de publicación, base de datos, etc.). El esquema propuesto se presenta en la Figura 1, ejemplificando algunas implementaciones concretas.

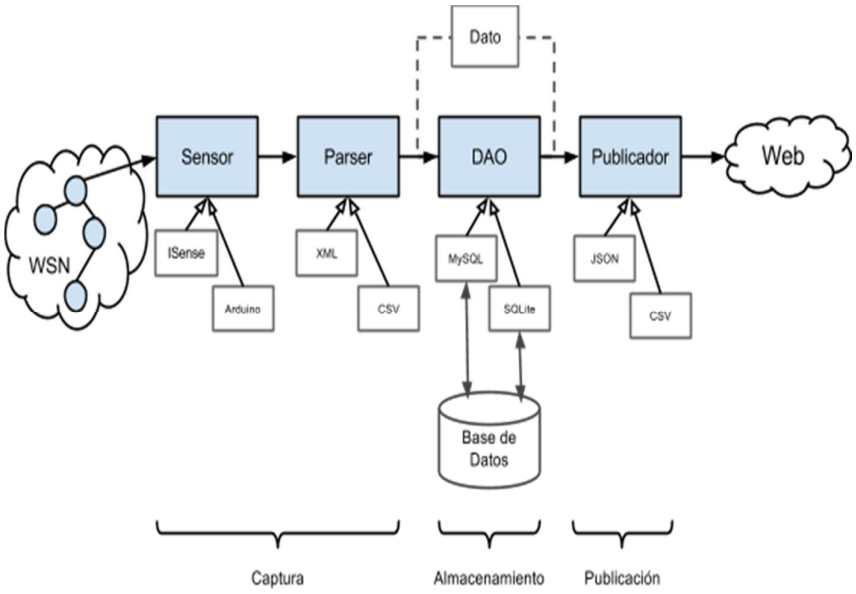

Figura 1 - Esquema propuesto de la solución (elaboración propia)

Captura y Parseo: La captura incluye la abstracción de manera en que se obtienen los datos desde la WSN para luego realizar el parseo de las cadenas y las convertirlas a una representación interna. En la Figura 2 se pueden ver las abstracciones tanto de los nodos de la WSN (Sensor) como del Parser.

Los datos que se deben provenientes de la WSN que se deben parsear tienen el siguiente formato:

id:0x1c34,t:WM,temp1:26.0,rh:57.6,pre:1006.0,temp2:25.3

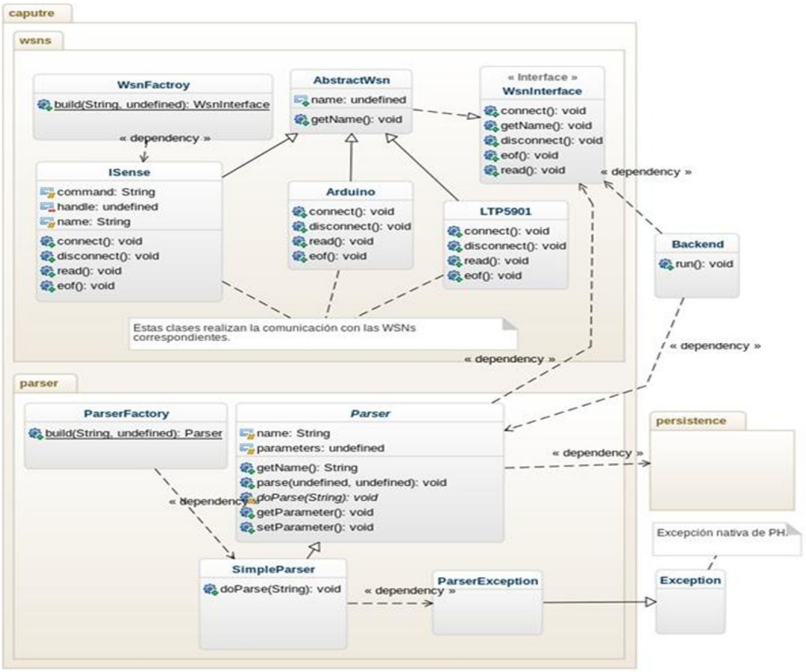

Figura 2 – Clases que intervienen en la Caputura de Datos

Los pasos a realizar se representan en el Diagrama de Secuencia de la Figura 3.

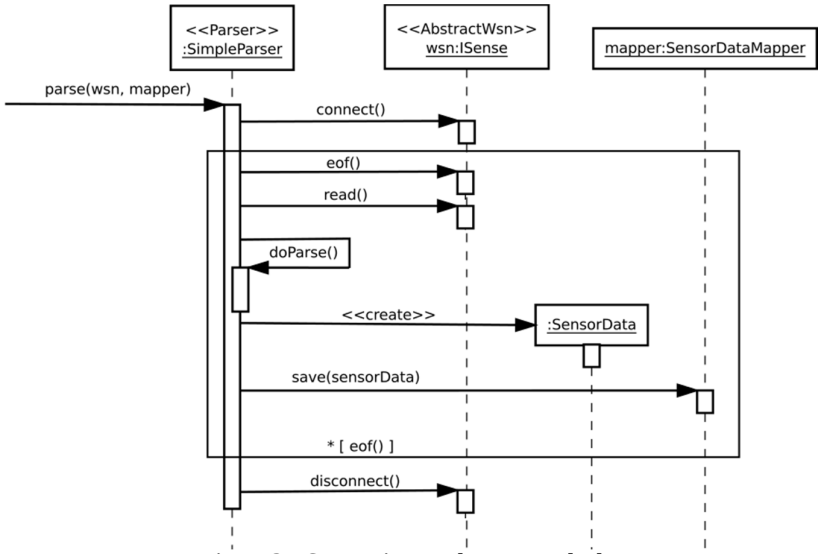

Figura 3 – Secuencia para la captura de datos.

Almacenamiento: El almacenamiento abstrae de la forma de almacenamiento en tablas físicas e independiza el al resto de middleware, ORM o motor de base de datos utilizado. Las clases de este módulo se presentan en la figura 4

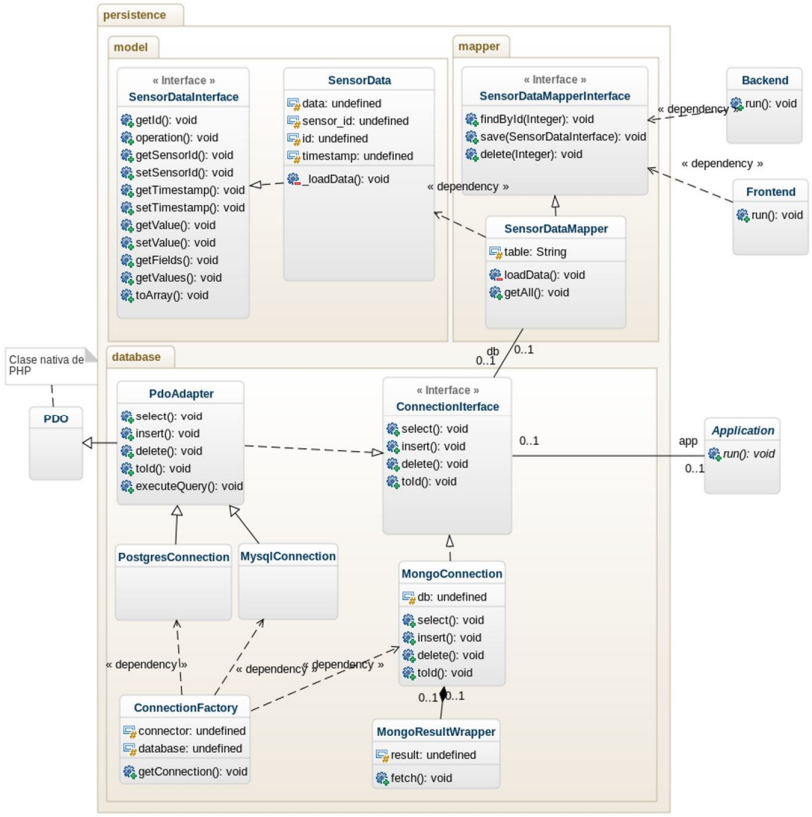

Figura 4 – Clases que intervienen en el Almacenamiento.

Para la representación interna se ha definido formato siguiente formato. Es decir a partir de la trama capturada desde los nodos, en figura 5 se puede ver la clase que abstrae los objetos SensorData.

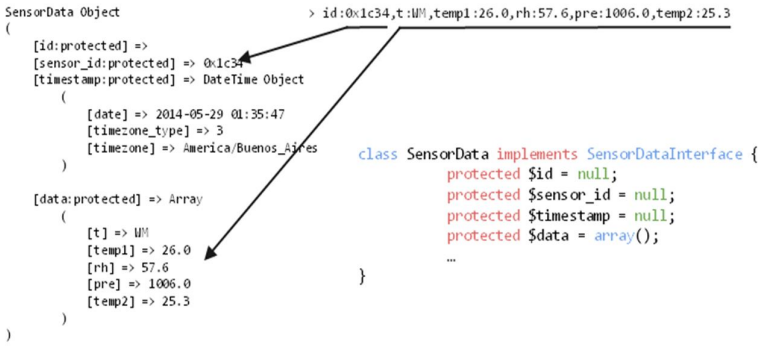

Figura 5 –Cadena obtenida de desde los sensores, objetos Json y clase SensorData.

Publicación: La publicación sobre HTTP está basada en REST Representational state transfer y permite múltiples formatos como interoperables como JSON y XML y otros formatos tradicionales. Como HTML y CSV. En la figura 6 se pueden ver las clases que permiten estas abstracciones y también clases que permiten dar formatos y realizar filtros para las consultas a los datos de capturados.

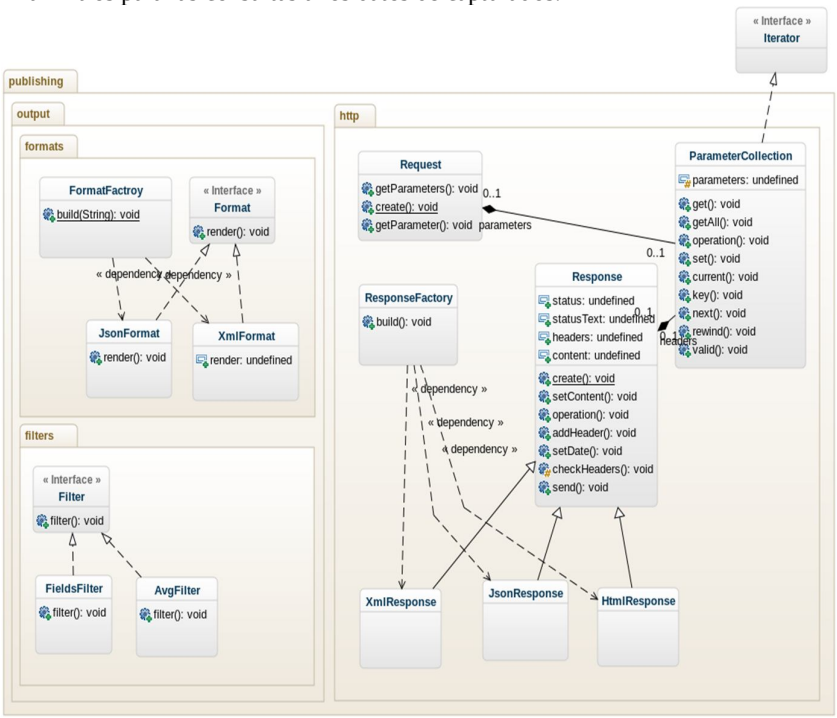

Figura 6 – Clases que intervienen en la Publicación.

5 Pruebas

En las pruebas se muestran salidas en distintos formatos de las consultas a los datos capturados por el Middleware. En la figura 7 se obtiene la salida de una consulta GET solicitando los datos de temperatura y marca de tiempo promedio entre dos fechas dadas, en formato Json.

GET
http://localhost/tesis/app/web/datos.json?fields=temp1,timestamp&from=2014-05-29%2001:35&to=2014-05-29%2001:40&avg=temp1&avg-interval=minute

```
{
  "results":[
    [
      {
        "timestamp":"2014-05-29 01:35:00",
        "temp1":"26.2",
        "avgFrom":"2014-05-29 01:35:47",
        "avgTo":"2014-05-29 01:35:58"
      },
      …
    ]
  ]
}
```

Figura 7 – Captura del resultado de un GET en formato Json.

En la figura 8 se obtiene la salida de una consulta GET solicitando los datos de temperatura y marca de tiempo en formato XML.

GET
http://localhost/tesis/app/web/datos.xml?limit=5&fields=timestamp,temp1

```
<?xml version="1.0" encoding="UTF-8"?>
<results>
  <result>
    <timestamp>2014-05-29 01:35:47</timestamp>
    <temp1>26.0</temp1>
  </result>
  …
</results>
```

Figura 8 - Captura del resultado de un GET en formato XML.

6 Conclusiones

En el presente trabajo se presentan los avances en el desarrollo de un middleware, que abstrae a los desarrolladores de aplicaciones orientadas a Ambientes Inteligentes e Internet de las Cosas de las peculiaridades de las distintas redes de sensores inalámbricas, formatos de datos, motores de gestión de persistencia bases datos y formatos de intercambio de datos interoperables, que pueden ser utilizados por aplicaciones de capas superiores constituyendo una alternativa de soporte similares a

las estudiadas en [**12**] aunque multiplataforma. Esta abstracción se presenta en formato de framework, el cual se puede extender según la necesidad de las aplicaciones. El framework permite capturar, almacenar y posteriormente publicar datos en formatos interoperables, que pueden ser utilizados por aplicaciones de capas superiores.Este middleware ha sido utilizado en otros proyectos de aplicaciones Mashup para la determinación de nivel de llegado de Contenedores de residuos [**13**] y de Mashup de Monitorización de temperatura en secaderos de Té [**14**].

7 Bibliografía

[1] K Ashton, ""That 'Internet of Things' Thing"," 2009 (rev. 2011).

[2] Ahola J., Ambient Intelligence, 2001.

[3] W. Dargie and C. Poellabauer, *"Fundamentals of Wireless Sensor Networks - Theory and Practice*. Reino Unido: Wiley, West Sussex, 2010.

[4] Eduardo O. Sosa, Contribuciones al establecimiento de una red global de Sensores Inalámbricos.Tesis Doctoral, Junio 17, 2011.

[5] E. O. Sosa et al., "Twitter, Soporte de una Red de Sensores Inalámbricos," , Morelia, Mexico, 2010.

[6] V. Trifa, D. Guinard, V. Davidovski, A. Kamilaris, and I. y Delchov, "Web Messaging for Open and Scalable Distributed Sensing Application," , 2010.

[7] ThingSpeak. (2013, Aug.) Internet of things – ThingSpeak. https://www.thingspeak.com

[8] Cosm. (8, Aug.) Cosm - Internet of Things Platform Connecting Devices and Apps for Real-Time Control and Data Storage. http://www.cosm.com/

[9] International Telecommunication Union ITU, "Ubiquitous Sensor Networks (USN)," vol. 4, 2008.

[10] G. Erich, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software.*: Addison-Wesley, 1995.

[11] Coalesense. (2013) iSense Wireless Sensor Network Software. http://www.coalesenses.com/index.php?page=isense-software

[12] Diego Alberto Godoy, "Plataformas para la creación de mashups sensibles al contexto en entornos de inteligencia ambiental," Universidad Nacional de la Plata, La plata, Trabajo Final de Especialidad 2013.

[13] Eduardo Omar Sosa et al., "Internet del futuro y ciudades inteligentes," in *XV Workshop de Investigadores en Ciencias de la Computación*, Paraná, 2013.

[14] Paola Quiñones, Diego Alberto Godoy, and Eduardo O. Sosa, "Redes Inalámbricas De Sensores: Una Experiencia En La Industria Del Té," in *42° Jornadas Argentina de Informática* , Córdoba, Argentina, 2013.