

Manejo e integración de bases de datos en redes de sensores inalámbricas

Database management and integration in wireless sensor networks

Diana C. Restrepo P., Ing, Demetrio A. Ovalle C., PhD y Alcides de J. Montoya C., MSc.
GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial, Escuela de Sistemas Universidad Nacional de Colombia – Sede Medellín
{dcrestr2, dovalle, amontoya}@unalmed.edu.co

Recibido para revisión 24 de enero de 2009, aceptado 20 de mayo de 2009, versión final 28 de mayo de 2009

Resumen—Una Red de Sensores Inalámbrica o WSN (por su acrónimo en inglés Wireless Sensor Networks), está definida como un conjunto de nodos los cuales constan de memoria, micro-controladores, sensores y radio incorporados, con capacidades limitadas. Lo anterior permite el monitoreo de variables físicas y ambientales tales como temperatura, luz, humedad, etc. requeridas para una aplicación específica. Debido a la importancia de las mediciones tomadas por estos dispositivos el poder disponer de una base de datos asociada a dicha red es indispensable, sin embargo, puede ocasionar desafíos como lo es el procesamiento de consultas. Adicionalmente, es necesario capturar, procesar y agregar datos de forma confiable, escalable y a tiempo que no interfiera con el desempeño de la red.

El objetivo de este artículo es a partir del modelo de arquitectura para el diseño de una base de datos espacial y dinámica (representación en tiempo real de la topología de la WSN), conocido como SensorDB, proponer un asistente de configuración que integra las herramientas de administración TinyDB y SwissQM con el fin de dar flexibilidad y apoyo al administrador de bases de datos.

Palabras Clave—Sistemas Distribuidos, Redes, Bases de Datos, WSN (Wireless Sensor Networks), TinyOS, TinyDB, SwissQM.

Abstract—A Wireless Sensor Network (WSN) is defined as a set of nodes having incorporated memory, micro-controllers, sensors, and radio with limited capacities. Those characteristics permit the monitoring of physical and environmental variables such as temperature, humidity, luminosity, etc. which are required for a specific application. Because the importance of measures taken by those devices, it is indispensable be able to have a database associated to the WSN, but this can bring some challenges like for example the query processing. Additionally, it is necessary to capture, process, and aggregate data in a reliable, scalable, and puntual way that does not interfere with the network's performance. The aim of this article is starting from the architectural model for designing a spatial and dynamic database (real time representation of network topology), well known as SensorDB, to propose a configuration assistant that integrate the management tools of TinyDB and SwissQM, in order

to give flexibility and support to a database administrator.

Keywords—Distributed Systems, Networks, Databases, WSN (Wireless Sensor Networks), TinyOS, TinyDB, SwissQM.

I. INTRODUCCIÓN

En los últimos años, las redes de sensores han sido la solución para aplicaciones como el monitoreo del medio ambiente, el cual es de crucial importancia para la comunidad científica y la sociedad.

Miles de kilómetros cuadrados pueden ser supervisados a través de los años, puede monitorearse el crecimiento de la vegetación, la calidad del aire, derrames de aceite y de forma ensamblar y además, la estación central se convirtió, de esta forma, en punto de falla único para la red. Por lo tanto, supliendo las limitaciones de estas redes cableadas, surgieron las redes de sensores distribuidas e inalámbricas.

Desde su aparición, muchos esfuerzos en investigación tratan de resolver las limitaciones de hardware y software asociadas con su manejo y administración. Pues al igual que una LAN tradicional una Red de Sensores Inalámbrica ó WSN (por su acrónimo en inglés Wireless Sensor Networks), es un conjunto de nodos que se comunican entre sí para intercambiar información relevante de su aplicación. Pero a diferencia de una red común, cada estación posee memoria, procesador, radio y energía limitados. En complemento, a través de una WSN se transportan mediciones físicas, en lugar de voz, datos y servicios como es el caso de una LAN.

Debido a lo anterior, consideraciones adicionales se deben tener en cuenta para la organización de la información. Pues en

este caso, se realiza de forma continua y en cualquier momento debe estar disponible para un usuario en particular, como lo es su administrador ó para diferentes aplicaciones que puedan requerir dicha información. Puede ser un sistema de control de variables ambientales, un sistema de rastreo y/o monitoreo de habitats, etc. [1].

El resto del artículo está organizado de la siguiente forma: inicialmente, en la sección 2, se presentará un marco teórico acerca de las redes de sensores inalámbricas y las bases de datos, su arquitectura y protocolos respecto a un modelo de capas. Luego, en la sección 3, un estado del arte de las aplicaciones actualmente desarrolladas, seguido en la sección 4, del manejo de datos para una WSN y una evaluación de desempeño de las herramientas para su administración (secciones 4 y 5) TinyDB y SwissQM, haciendo énfasis en las ventajas y debilidades de las mismas. Adicionalmente, en la sección 6, se propone un asistente de configuración que integra dichas herramientas. Finalmente, se presentan las conclusiones y los trabajos futuros.

II. MARCO TEÓRICO

A continuación se presentará una breve definición de las Redes de Sensores Inalámbricas; cómo están conformadas, en qué consisten y cómo es su esquema general. Luego se explica el funcionamiento de una base de datos de una WSN. La cual es representada mediante un modelo de capas propuesto por [2].

Adicionalmente, se presenta una descripción de las herramientas necesarias para el manejo de información en una WSN. Como principal requerimiento se encuentra el sistema operativo TinyOS, el cual está diseñado específicamente para este tipo de redes. Seguido de las aplicaciones de administración de bases de datos, TinyDB y SwissQM.

A. Redes de Sensores Inalámbricas (WSN)

Las WSN esencialmente consisten en una red inalámbrica de dispositivos autónomos distribuidos, los cuales hacen uso de sensores para medir temperatura, sonido, vibración, presión y luz, y de esta forma realizan el monitoreo de variables físicas y ambientales en cualquier ubicación. En la Fig. 1, se puede apreciar un esquema general de lo que es una WSN.

Teniendo en cuenta desde cada nodo particular hasta un sistema completo, la arquitectura se distingue por las características presentadas a continuación: un dispositivo pequeño con sensores, memoria, procesador y radio incorporado, con capacidades limitadas. Cada nodo toma medidas físicas de su entorno y las trasmite vía radio. Luego, existe un proceso de sincronización y comunicación que es controlado por los protocolos descritos en [1]. A continuación, una estación base ó gateway, como es mostrado en la Fig. 1, captura la información de toda la red y la entrega al usuario. Finalmente, el usuario toma decisiones con referente a la

información obtenida y realiza tareas adicionales como programación de nodos particulares o una agrupación de nodos, adquisición de medidas críticas o atípicas, nuevas aplicaciones para funciones específicas, entre otras.

coordinada crearse una vista completa de estos ambientes naturales. Razón por la cual se ha impulsado el crecimiento y desarrollo de este tipo de redes en otro tipo de ambientes tales como, la industria y automatización.

Cabe señalar que hasta hace poco tiempo las redes de sensores estaban formadas por un pequeño número de estaciones conectadas por cable a una estación central de procesamiento de datos. Sin embargo, con la aparición de nuevas tecnologías en redes de comunicaciones y la necesidad de monitorear un área más extensa, el montaje se hizo cada vez más complejo debido a la cantidad de cable que se debía

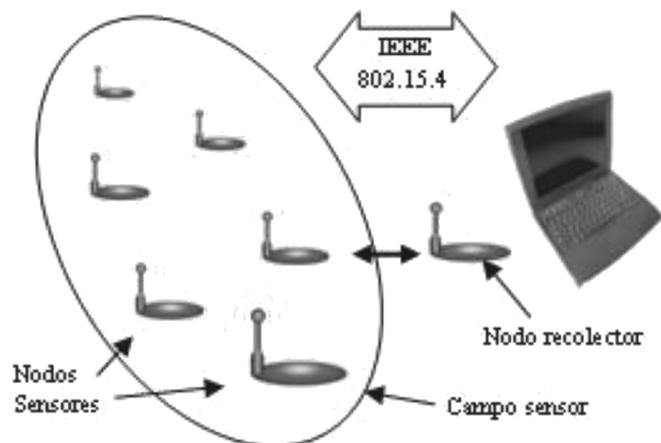


Figura 1. Esquema básico de una WSN.

B. Bases de Datos para WSN

Según [3] desde el punto de vista de almacenamiento de datos, una Base de Datos para redes de sensores, más conocida en la literatura como SensorDB, es una base de datos distribuida que recoge mediciones físicas del ambiente, las indexa, y sirve a las preguntas de los usuarios y otras aplicaciones externas con origen o destino en la red.

Adicionalmente, el modelo de arquitectura generalizado más reciente y acertado, es el propuesto por Qiong y Hejun en [2]. Básicamente esta arquitectura está apoyada en el modelo TCP/IP, el cual es bien conocido para redes LAN. En el caso de un SensorDB, el modelo comprende tres capas, MAC, enrutamiento o red, y capa de consultas o aplicaciones. Adicionalmente, hay una capa transversal que actúa para cada una de las capas anteriormente mencionadas como herramienta de control. Pues se encarga de la programación y la definición de los protocolos que interfieren en cada capa. Finalmente, un entorno que soporta toda la arquitectura y es representado por el sistema operativo.

En la Fig. 2, se representa el modelo anteriormente descrito. A continuación se da una breve descripción de las funcionalidades y características de cada una de las capas.

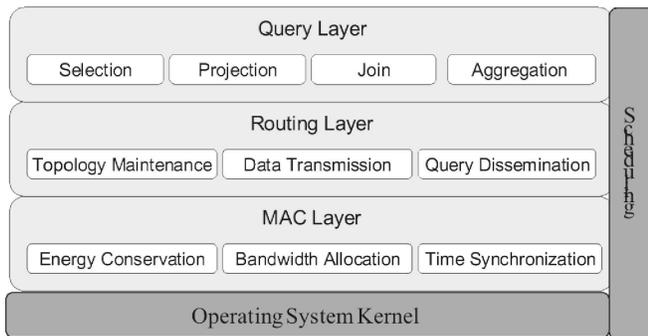


Figura 2. Arquitectura de un SensorDB. Tomado de [2].

1) Capa MAC: En esta capa se encuentra el control de acceso al medio, es decir qué nodos deben transmitir en determinado momento, quién tomará el medio, cómo controlar y sincronizar los nodos para que ninguno falle en la transmisión. Además, y una de las características más importantes, de qué forma se debe realizar la transmisión de datos de modo que el consumo de energía sea mínimo. Algunos de los protocolos definidos en esta capa son: CSMA (Carrier Sense Multiple Access), STEM (Sparse Topology and Energy Management) [4], S-MAC (Sensor Media Access Control) [5], y T-MAC (Timeout Media Access Control) [6].

2) Capa de enrutamiento: La capa de enrutamiento es la que define la forma más efectiva de flujo de datos a través de la red, el camino que se debe tomar para una petición de información en un momento determinado. Requiere por lo tanto, información de donde se encuentra cada nodo. En general, la información de ubicación se obtiene de los enlaces más cortos entre nodo pero hay protocolos especializados, que además de distancia y localización, tiene en cuenta el consumo de energía de la red, que es el principal problema para este tipo de redes. Estos son: Difusión Directa [7], MintRoute [8] y TinyAODV (Tiny Ad-hoc On-Demand Distance Vector) [9].

3) Capa de consulta: La capa de consultas se hace cargo de soportar un lenguaje declarativo, el cual se emplea comúnmente en un ACQP (ACquisitional Query Processing). Realizando de esta forma consultas de selección, proyección y agregación. Así mismo, el usuario nunca se da cuenta de las capas más bajas y sólo se restringe a conocer la abstracción de lenguaje natural a sentencias en lenguaje declarativo.

4) Programación en WSN: La programación en WSN se realiza tratando de remediar los problemas de escucha de inactividad, colisiones, control de la recepción de paquetes y garantizar una entrega en tiempo real. Por lo anterior, se tienen en cuenta varios enfoques para la programación en una WSN como lo son: Programación Distribuida (SS) [10], FPS (Flexible Power Scheduling) [11] y DCS (Distributed Cross-Layer Scheduling) [12].

5) Sistema Operativo para Entornos WSN: El sistema operativo comprende el ambiente del cual va hacer uso la base de datos para ejecutarse. Su principal tarea es dar soporte al desarrollo de las aplicaciones, proveer interfaces de programación, compilación, generación y ejecución de código, además, de recursos de administración de hardware. El Sistema Operativo de fábrica, es el TinyOS es el primer esfuerzo de investigación para las WSN.

Se han propuesto adicionalmente algunos Sistemas Operativos alternos a TinyOS, cuyo objetivo es mejorar el desempeño de la red en aspectos donde TinyOS no puede, como se verá en la sección siguiente. Algunos de estos son: Contiki [13], SOS [14], MANTIS [15], t-kernel [16], Máquinas Virtuales y DSN [17].

C. *TinyOS*

TinyOS es un entorno de libre distribución diseñado especialmente para WSN. Tiene como características que está escrito en NesC, el cual es una extensión del lenguaje de programación C optimizado, con el fin de tener en cuenta las limitaciones de memoria de las redes de sensores. Su funcionalidad está definida a través de una agrupación de tareas y procesos que se ayudan entre sí, pues su arquitectura está basada en componentes, lo que lo hace flexible a la hora de implementar nuevo hardware y software. Además, soporta diferentes plataformas de nodos de sensores para el desarrollo de aplicaciones. TinyOS está desarrollado por un consorcio liderado por la Universidad de California en Berkeley en cooperación con Intel Research [18].

Una de las falencias de TinyOS es su código estático para las aplicaciones, lo cual ocasiona la reinstalación de todo el código en un nodo para hacer uso de las mismas. Por el contrario, un código dinámico permitiría una actualización constante. Razón que ha impulsado a proponer nuevos sistemas operativos para las WSN, tales como los presentados en la sección anterior.

D. *TinyDB*

TinyDB consiste en una aplicación de TinyOS. Está diseñada específicamente para la plataforma Mica2 y la versión más estable se encuentra en la versión TinyOS 1.x. Consiste en un front end, desarrollado en JAVA y compuesto de tres objetos principales. El TinyNetwork, el cual representa la conexión con la red y éste a su vez activa el CmdFrame y QueryFrame, los cuales representan el envío de consultas a la red y el diseño de consultas respectivamente. En la Fig. 3 se muestran estos tres componentes [19].

Entre las características de TinyDB están el procesamiento de consultas de adquisición del tipo ACQP, cuyo objetivo es reducir el consumo de energía haciendo uso de sensores que tengan el control de donde, cómo y con qué frecuencia los datos adquiridos se encuentran físicamente. Muestreando y entregando resultados a los operadores de bases de datos constantemente.

Finalmente, teniendo en cuenta la arquitectura de un SensorDB anteriormente descrita, TinyDB tiene predefinidos como protocolos y características las siguientes: sistema operativo TinyOS, protocolo de capa MAC, S-MAC y como protocolo de enrutamiento el MintRoute. Con respecto a la capa de programación y consulta, se utiliza un método de operación basado en modos, el cual se describe a continuación.

El modo zombie, consiste en un período de inactividad del procesador y radio esperando que un temporizador expire o un evento externo active el dispositivo. El modo de procesamiento, donde el dispositivo se encuentra en actividad, es decir los resultados de una consulta se procesan de forma local. El modo de recepción, donde los resultados de los dispositivos vecinos son recibidos vía radio. Por último, el modo de transmisión, donde los resultados de toda la red son entregados a la estación base. Estos modos de operación aseguran una entrega confiable de los resultados de las consultas requeridas por el administrador de la base de datos.

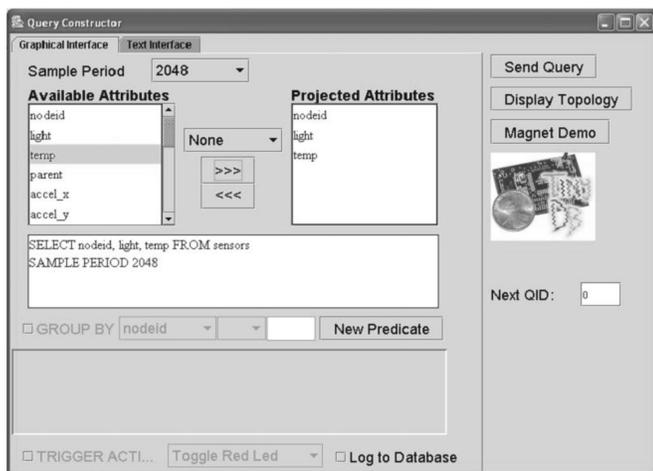


Figura 3. Interfaz gráfica de TinyDB. Tomado de [19].

E. *SwissQM*

SwissQM (por su acrónimo en inglés Scalable WIREless Sensor Query Machine) representa una nueva arquitectura para adquisición y procesamiento de datos en redes de sensores. Ha sido desarrollado con el fin de proveer una mejor funcionalidad y flexibilidad a nivel de la red. Además de una forma más adaptable para el entorno, independiente de los datos y del lenguaje de consulta. SwissQM está basado en una máquina virtual especializada que corre bytecode optimizado en vez de consultas declarativas.

Comúnmente una máquina virtual es un programa nativo, capaz de interpretar y ejecutar instrucciones en lenguaje binario especial, el cual es generado por un compilador del lenguaje Java. Incluye además algunos componentes adicionales para su funcionamiento. En el caso de SwissQM, tales componentes se encuentran en la librería de TinyOS, los cuales pueden ser copiados y utilizados en la máquina virtual.

Como resultado, SwissQM a diferencia de TinyDB o un procesador de consultas común, no toma decisiones del lenguaje de consulta usado ni tampoco de la estrategia de despliegue de la red [20]. En la Fig. 4, se muestra la arquitectura de SwissQM.

Las principales características que hacen útil y atractivo el uso de SwissQM para la consulta, procesamiento, administración y manejo de datos en WSN son las siguientes:

1) Separación de sensores e interfaz externa: El modelo de programación es independiente del lenguaje de consulta usado, como ya se mencionó anteriormente.

2) Ambiente dinámico, multi-usuario y multi-programable: No hay restricciones en la presentación de consultas ni en el número de consultas que pueden ejecutarse concurrentemente.

3) Uso optimizado de sensores: El único procesamiento que se realiza en los nodos, es el relacionado con captura, agregación y envío de datos.

4) Extensibilidad: Esta característica se refiere a la posibilidad de extender dinámicamente y modificar el comportamiento de SwissQM como sea necesario, para aplicaciones específicas que se quieran realizar. Además de que no requiere de un entorno para ejecutarse, pues posee las características de una máquina virtual común. Esto le añade por lo tanto la característica de portabilidad.

Figura 4. Arquitectura de *SwissQM*. Tomado de [20].

III. ESTADO DEL ARTE EN BASES DE DATOS PARA WSN

Las primeras bases de datos para redes de sensores presentaban un enfoque en el cual sólo era contemplado un núcleo central de almacenamiento. Por ejemplo, Giraldo y Montoya en [21] proponen un sistema de monitoreo de variables físicas para invernaderos. Este sistema implementa el protocolo de comunicación 1-wire sobre medios físicos de cobre, permitiendo únicamente una comunicación half-duplex entre la estación central y cada nodo de la red. La principal limitación en este trabajo es su enfoque centralizado, ya que todo su procesamiento ocurre en su estación base, la cual en caso de

fallar afectaría el funcionamiento de todo el sistema. Adicionalmente, para poder tomar decisiones acerca de los datos obtenidos es necesario el acceso a la base de datos y a la lectura de los mismos.

Una nueva propuesta, esta vez con un enfoque distribuido, para el monitoreo de variables físicas, se desarrolla por Montoya et. al en [22]. Dentro de las fortalezas de este trabajo, merece resaltar, el uso de instrumentos virtuales y de nodos inalámbricos distribuidos para la adquisición de datos de la red. Sin embargo, no se consideran aspectos de gran importancia para el monitoreo del sistema tales como: la visualización de la topología de la red y la posibilidad de ejecución de funciones que actúen sobre toda la red. Dichas funciones son la agregación y la organización de datos, las cuales representan el estado del sistema en tiempo real y sirven como herramienta a un administrador de bases de datos orientadas a dispositivos de WSN.

En [23] Gerhke et al. desarrollan una plataforma para la evaluación de técnicas de procesamiento de consultas de BD para sistemas embebidos (sistemas computacionales de software o hardware multi-funcionales y de recursos limitados que son construidos dentro de un sistema mayor). Esta propuesta presenta una arquitectura bastante promisoría para la adquisición de datos en una WSN, ya que implementa en su capa de enrutamiento de datos el protocolo de difusión directa considerado en un SensorDB. Adicionalmente, provee un lenguaje XML para el intercambio de mensajes a través de la red. A pesar de ello, no deja de ser una herramienta de simulación y prueba de consultas, las cuales corren bajo ciertas estrategias, más no puede ser considerado como una plataforma que pueda acercarse a las aplicaciones de bases de datos con WSN, para monitoreo de variables en entornos reales.

Finalmente, una aplicación práctica, fue desarrollada en la Universidad de Berkeley, California, por los investigadores Szewczyk et al. [24], los cuales proponen un sistema de monitoreo del habitat en la Gran Isla de Patos (Golfo de Maine, Océano Atlántico, Costa nordeste de América del Norte), cuyo objetivo es determinar los factores que interfieren en el tratamiento de un nido óptimo para el crecimiento de patos. Utilizan como herramienta de adquisición de datos TinyDB y TASK (por su acrónimo Tiny Application Sensor Kit) para la administración y manejo de datos en la red. Este proyecto todavía se encuentra vigente en la actualidad, pues una de las dificultades existentes es la gran extensión que posee la isla, lo cual no ha permitido un despliegue exitoso de motes con recursos limitados en memoria, energía, etc. que permitan suministrar la información pertinente para el buen funcionamiento del sistema.

IV. MANEJO DE BASES DE DATOS EN WSN

Físicamente, se consideran varios aspectos en relación con las BD tradicionales. Las transferencias de datos, por ejemplo, son realizadas desde datos en la memoria de los nodos, en lugar

de bloques de datos en discos. Las tablas relacionales no son estáticas, pues los datos son sensados continuamente sin interrupciones, debe considerarse añadir solo información agrupada donde no haya reordenación de operaciones, es decir, agrupar información que cumpla con determinados operadores.

Lógicamente, los datos en una red de sensores, consisten en medidas del mundo físico, lo que incluye también errores, ya sea por la interferencia o ruido del medio. Se considera para este aspecto, consultas en serie en lugar de consultas exactas, y adicionalmente consultas probabilísticas o aproximadas.

Es importante aclarar que una base de datos para WSN, es una combinación de datos de los sensores y datos de almacenamiento de cada sensor. Los datos del sensor corresponden a la toma de medidas temporales de un fenómeno físico tales como temperatura, luz, sonido, presión del aire, etc. Por su parte los datos de almacenamiento pertenecen a las características propias de cada sensor como son identificación del sensor, ubicación, tipo de sensor, etc.

Adicionalmente, para una WSN es de gran utilidad, la implementación de action triggers, es decir, operadores que puedan describir o reportar condiciones excepcionales que puedan presentarse. Como es el caso de alarmas en situaciones donde las medidas oscilen a escalas mínimas o muy altas.

A continuación se describen algunos de los conceptos que deben poseer las BD para WSN en relación con las BD tradicionales, además de las opciones de configuración que se tienen para cada una de estas características.

A. Organización de la Base de Datos

En condiciones deseables la organización de bases de datos debe tener características de persistencia, consistencia, control de acceso a los datos, escalabilidad, balanceo de carga y una topología generalizada. Por tanto para una WSN se consideran tres tipos de enfoques para los métodos de almacenamiento de datos, los cuales se describen a continuación:

1) Almacenamiento externo (ES, External Storage): Para este tipo de almacenamiento, todos los datos son enviados directamente a un AP (Access Point) externo para un procesamiento futuro. Los datos son extraídos de la red de sensores de una forma predefinida o programada y se localizan en un solo servidor. Este enfoque, presenta una reducción significativa en los gastos generales de consulta, pero el promedio de gastos en actualizaciones depende exclusivamente de la longitud de la red.

2) Almacenamiento local (LS, Local Storage): Para este tipo de almacenamiento, pequeños segmentos de información son guardados localmente en el sensor detector. Por lo anterior, los gastos generales de actualización son insignificantes, pero las consultas de los usuarios inundan toda la WSN en búsqueda de los resultados a entregar.

3) Almacenamiento de datos centrados (DCS, Data-Centric Storage): Para este tipo de almacenamiento, los datos se

almacenan en un nodo determinado por el nombre asociado con el sensado de datos. Es un mecanismo de propagación de consultas basado en árbol usado en la capa de enrutamiento. El método utilizado es con el fin de soportar consultas de cualquier nodo, suministrando un mecanismo rendez-vous [25] para datos y consultas. Por lo tanto evita inundaciones de toda la red, almacena los datos por sus atributos, recupera los datos basado en sus atributos, hace traducción de atributos en una locación del nodo o ID y realiza almacenamiento de balanceo de carga. Este método de almacenamiento es usado por TinyDB para la organización de nodos en la red [26].

En general en una WSN el almacenamiento de datos centrados se combina con varias técnicas para su optimización, como por ejemplo, algoritmos avanzados, híbridos o tablas hash. Debido a que una tabla hash es una estructura de datos que asocia llaves o claves con valores y su eficiencia esta principalmente en operaciones de búsqueda, la cual es considerada una de las operaciones principales utilizadas en el almacenamiento de datos, no solo en el contexto presente sino en general para cualquier almacenamiento de datos.

B. Propagación de consultas y agregación

El proceso de propagación de consultas en una BD tradicional es realizado mediante un software y servidores con acceso a la red. En una BD para WSN la agregación de datos se realiza considerando dos tipos de enfoques:

1) Basado en servidor: En este caso, todos los nodos envían sus datos de sensado parcial y el nodo base se encarga de realizar todo el procesamiento de la consulta.

2) Basado en red: En este caso, cada nodo calcula un registro de estado parcial (puede ser una suma o conteo), basado en sus datos y los de sus hijos, si los tiene.

Se utiliza también una técnica de agregación de datos la cual mantiene los datos usando una estructura jerárquica, y permite hacer consultas y obtener sus respuestas de una manera eficaz. Se ejecuta asumiendo que cada nodo mantiene la información de su ubicación, además cada nodo conoce el rango de la existencia física de toda la red y cada nodo tiene una función hash común para elegir sus nodos padres. El proceso de agregación consiste de los siguientes pasos: primero, toda el área donde se encuentran los nodos, es subdividida recurrentemente y virtualmente en cuatro cuadrículas y jerarquizada por varios niveles de cuadrículas. Un nodo cabeza de grupo es elegido en cada cuadrícula por una función hash común aplicada a cada nodo dentro de una misma cuadrícula. Luego la información consolidada como la media, el valor máximo, el valor mínimo, es calculada para cada cuadrícula. El nodo cabeza de grupo de cada cuadrícula, registra el ID que muestra la información geográfica de la cuadrícula a los siguientes nodos cabeza de grupo de niveles más altos. El nodo cabeza de grupo de altos niveles, mantiene la información agregada de cuatro cuadrículas del nivel anterior, y calcula la agregación de datos. Esta operación se repite recurrentemente hasta que los datos de toda la red se recogen y agregan al nodo padre de toda la red.

En el caso de TinyDB esta estructura de agregación es llamada TAG (Tiny AGgregation) [19], un servicio de agregación para redes ad-hoc basado en las características de bajo consumo de energía y entorno distribuido e inalámbrico.

C. Indexación de datos

Muchos de los atributos manejados en una WSN tienen valores escalares, por lo que una forma natural de consulta a eventos de interés es utilizar una serie de consultas multidimensionales. Un ejemplo es el siguiente: "Listar todos los eventos cuya temperatura se encuentra entre 50°F y 60°F, y cuyos niveles de iluminación se encuentran entre 10 y 15." Esas consultas son útiles para correlacionar eventos que ocurren dentro de la red. Por lo tanto DIM (Distributed Index for Multi-dimensional data), es una estructura de indexación que utiliza el método de registrar ordenadamente datos multidimensionales es decir, consultas agrupadas usadas por los usuarios, que permiten de una forma eficiente consultar detalles acerca de eventos de su interés, y por otro lado, pueden ser utilizados por las aplicaciones que se ejecutan dentro de una red de sensores para correlacionar eventos y desencadenar acciones (action triggers) [27].

D. Tipos de consultas

Para una WSN, se consideran tres tipos de consultas básicas:

1) Consultas continuas: Las cuales consisten en presentar datos cada cierto tiempo es decir, especificando un periodo de muestra. La consulta siguiente representa la sentencia de "encontrar las áreas más iluminadas".

TABLA I. Consulta continua

```
SELECT nodeid, areaNo, light
FROM sensors
WHERE light > 400
EPOCH DURATION 1s
```

2) Consultas de agregación: Las cuales corresponden a los operadores de MAX, MIN, SUM, COUNT, AVERAGE que normalmente se utilizan en un típico procesador de consultas. La consulta siguiente representa la sentencia de "cuente el número de áreas ocupadas en cada región".

TABLA II. Consulta de Agregación

```
SELECT region, CNT (occupied), AVG (sound)
FROM sensors
GROUP BY region
HAVING AVG (sound) > 200
EPOCH DURATION 10s
```

3) Consultas basadas en eventos: Las cuales consisten principalmente en la activación de adquisición de datos o un sensor específico, cuando se detecte la aparición de cierta medida o agrupación de medidas. La consulta siguiente representa la sentencia: "Señale a través de una alarma que la temperatura está demasiado alta cuando su valor sea mayor que value" (value es un valor asignado por el usuario).

TABLA III. Consulta basada en Eventos

SELECT nodeid,temp
WHERE temp > value
OUTPUT ACTION SIGNAL hot (nodeid,temp)
SAMPLE PERIOD 10s

V. INTEGRACIÓN DE BASES DE DATOS EN WSN

A. Herramientas Utilizadas

Para el sistema operativo, se utiliza XubuTOS, el cual es la versión de Linux específica para WSN. Simplifica la instalación de los sistemas operativos necesarios para la administración y manejo de WSN. Fue creado por The Toiles, el cual es un grupo de investigación de Colorado School of Mines. La versión actual es XubunTOS 2.0 contiene TinyOS 2.0.2, TinyOS 1.x desde CVS, y Xubuntu 7.04 (Feisty).

Para el servidor de bases de datos relacional se utiliza PostgreSQL, el cual está orientado a objetos y es software libre, publicado bajo la licencia BSD.

Finalmente, se utiliza la herramienta NetBeansIDE como el entorno de desarrollo integrado para construir aplicaciones en lenguaje Java.

B. Configuración

A continuación se describen las condiciones en las cuales se trabaja y como se derivó en el desarrollo del proyecto que describe en este artículo. El procedimiento que se debe seguir es el siguiente: Inicialmente se instalan las dos aplicaciones de SensorDB, TinyDB y SwissQM por separado, debido a que es necesario contar con las diferentes características y requerimientos para cada una sin que se presenten problemas de incompatibilidad entre las mismas. Luego, debido a que TinyDB es una aplicación de TinyOS y sólo se encuentra en la versión 1.x del mismo, para su instalación sólo es necesario tener configurada la librería de comunicación con Java. Finalmente, para la consulta, procesamiento, administración y manejo de datos en WSN se optó por la herramienta SwissQM que es una aplicación independiente de TinyOS. Puede ser descargada de [20] y para su ejecución es necesario tener algunas librerías, las cuales se presentan en [20].

C. Integración

La integración de aplicaciones se realiza como se mencionó anteriormente en NetBeans, bajando el sistema de archivos de las dos aplicaciones TinyDB y SwissQM. Obteniendo como resultado la aplicación que se muestra en la Fig. 5.

Esta aplicación funciona de la siguiente forma: los archivos correspondientes a las propiedades de las dos aplicaciones (tinydb.properties, swissqm.properties), los cuales en el caso de la integración son archivos de la aplicación, se escriben en la medida que el usuario presenta sus opciones de configuración de una forma más explícita, dinámica y gráfica.

El asistente de configuración cuenta con las opciones requeridas para cada una de las aplicaciones. En el caso de TinyDB sólo es permitido el uso de la plataforma Mica2. Por otro lado SwissQM tiene la flexibilidad de soportar las plataformas de Mica2 y TmoteSky. Además presenta explicaciones en cada uno de los pasos, como una guía para el usuario realizar las configuraciones necesarias. Opciones adicionales de configuración como la activación de almacenamiento de medidas (PostgreSQL) y el número de nodos de la red, pueden ser realizadas mediante esta aplicación de usuario final.

Adicionalmente, como requerimientos para una instalación exitosa de este asistente es necesario una previa instalación y verificación del conjunto de librerías para el funcionamiento de cada una de las aplicaciones TinyOS como sistema operativo y un producto de bases de datos como lo es PostgreSQL.

Finalmente, la integración de BD trata de hacer un wrapper de TinyDB y SwissQM y dar una configuración de éstas de la forma más sencilla posible, tal cual lo hace un asistente de instalación de cualquier programa.



Figura 5. Integración de bases de datos en WSN.

VI. ANÁLISIS DE RESULTADOS

A. Evaluación de aplicaciones

Inicialmente se trabaja con dos máquinas virtuales independientes, para verificar el funcionamiento de cada aplicación sin afectarse mutuamente. Cada una de las aplicaciones se evaluó teniendo en cuenta las características siguientes: Plataforma, Imagen de la aplicación, Tamaño del mensaje de transmisión de la aplicación, Número de nodos y Árbol de enrutamiento (ver Tabla IV).

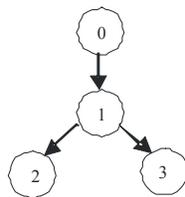
Tal como se puede observar, las aplicaciones tienen compatibilidad para plataformas diferentes, lo cual presenta una limitación para la comparación de las mismas. Pero para la evaluación en la práctica se considera principalmente el manejo y administración de BD y los aspectos de mayor relevancia para esta actividad.

Cabe señalar que los tiempos son tomados de acuerdo al número de muestras para cada uno de los lapsos de tiempo (convergencia, retardo y respuesta), los cuales fueron de 100, con el uso de un cronometro tradicional.

TABLA IV. Características de evaluación

Características	TinyDB	SwissQM
Plataforma	Mica2	TmoteSky
Imagen de la aplicación	65kB Flash 3kB SRAM	33kB Flash 3kB SRAM
Tamaño del mensaje de transmisión de la aplicación	49 bytes	36 bytes
Número de nodos	4 (con id 0, 1, 2 y 3 respectivamente)	

Arbol de enrutamiento



En la Tabla V se presentan los resultados obtenidos para las características presentadas y las configuraciones realizadas en cada una de las máquinas virtuales de cada aplicación. Los valores referentes a tiempos fueron tomados de acuerdo al tamaño de muestra de 100 y su media respectiva y en el caso de los datos en la BD, éstos corresponden al número de filas almacenadas en ésta, a los cinco minutos de tener activas cada una de las aplicaciones. Se muestran también las desviaciones para cada uno de los valores, lo que representa el grado de dispersión de los datos.

Teniendo en cuenta los cinco aspectos calificados en la práctica, TinyDB resulta ser la mejor opción para la administración de BD en una WSN, considerando que presenta tres de los cinco resultados mejores que SwissQM. Pero es de

resaltar la importancia de estos aspectos evaluados, pues aunque en TinyDB se tenga una convergencia más rápida de la red, los tiempos de respuesta de SwissQM son mejores además de la cantidad de datos capturados en la BD.

Otro aspecto importante a tener en cuenta es el fallo de los nodos, pues cada vez que un nodo sale de la red, es necesaria la reinstalación del mismo y muy probablemente la reactivación de la aplicación. Dado que cada nuevo nodo debe sincronizarse de nuevo con sus vecinos y atender las peticiones del administrador de BD. En este aspecto SwissQM no presenta buenos resultados. Su tasa de fallo de nodos es de dos en comparación con TinyDB, la cual es de solo uno. Es importante señalar que este aspecto está directamente relacionado con la capacidad de energía suministrada por las baterías de cada nodo y la cantidad de peticiones que se le realizan a uno o más nodos en particular. En este caso se tuvo en cuenta que todas las baterías tienen la misma energía y que la consulta realizada es constante para todos los nodos.

En relación con el manejo de BD, SwissQM por su parte permite flexibilidad para la implementación de nuevas aplicaciones. Además, cuenta con un intérprete para los nodos, el cual agiliza la tarea de entrega de información. También es importante resaltar que la imagen que se inyecta en el nodo para SwissQM es mucho más ligera, lo cual provee de memoria adicional a la tarea de procesamiento de consultas en operaciones como la evaluación de parámetros, agregación de un conjunto de valores y la entrega de resultados. Estas características pueden ser evidenciadas en los resultados de tiempo de respuesta y la cantidad de datos almacenados y es probable que por ello sea su tasa de fallo de nodos, dado que procesa más información en un tiempo más corto.

TinyDB cuenta con más herramientas adicionales con respecto a su arquitectura, tales como la visualización de la topología de la red, un lenguaje de detección de eventos y una interfaz gráfica más dinámica y explícita para el usuario final. Dado que esta aplicación hace parte del sistema operativo, el proceso de sincronización y convergencia de la red es realizado por el mismo y no por la aplicación como es el caso de SwissQM. Resultado que se puede evidenciar en la Tabla V con el tiempo de convergencia de la red y el retardo promedio.

TABLA V. Características de evaluación

Temporizadores	TinyDB	SwissQM
Convergencia de la red	1,15 minutos ($\sigma \pm 0,01$)	2,01 minutos ($\sigma \pm 0,06$)
Cantidad de datos almacenados en cinco minutos (con un intervalo de 2 segundos por muestra y 4 atributos en la tabla)	100 filas	120 filas
Retardo promedio de "epoch"	1,66 segundos ($\sigma \pm 0,01$)	3,39 segundos ($\sigma \pm 0,04$)
Número de nodos fallos después de una hora	1,04 nodos ~ 1 ($\sigma \pm 0,04$)	1,97 nodos ~ 2 ($\sigma \pm 0,03$)
Tiempo aproximado de respuesta	2,37 segundos ($\sigma \pm 0,01$)	1,02 segundos ($\sigma \pm 0,02$)

VII. CONCLUSIONES Y TRABAJO FUTURO

El trabajo presentado en este artículo es una aproximación a un ACQP (Acquisitional Query Processing) para realizar consultas de selección, proyección y agregación genéricas para cualquier plataforma de nodos en una WSN. Tiene como función principal presentar las consideraciones para el manejo de BD de este tipo de redes y así mismo presentar una herramienta de fácil uso para un administrador de BDs convencionales. Adicionalmente este trabajo está diseñado para cualquier aplicación que haga uso de una WSN y para cualquier distribución topológica.

Se puede concluir que una BD para una WSN es una base de datos distribuida y dinámica que representa en tiempo real los datos capturas y su flujo para una la topología de red. Lo anterior, ya que para el manejo de este tipo de almacenamiento se tienen aspectos similares en relación con las BD distribuidas convencionales, como lo es la propagación de consultas y la organización de la red, además de la adquisición y el tipo de consultas realizadas. También puede notarse que las aplicaciones mostradas en este trabajo cuentan con las características que usualmente se encuentran en un ACQP para la administración y manejo de información.

La integración de aplicaciones de BD para WSN, se hace con el fin de dar opciones y facilidades al administrador de bases de datos para la instalación y el uso de herramientas que soporten el procesamiento de información para la toma de decisiones y posterior control de una WSN. Por lo tanto, trata de reducir el tiempo de configuración de TinyDB y SwissQM sin tener que acceder al sistema de archivos de los mismos y con una interfaz grafica más clara donde el usuario pueda seleccionar sus necesidades.

Se puede concluir además que la propuesta de arquitectura del modelo de SensorDB es muy válida para entender los diferentes aspectos que implica la implementación de una red de sensores inalámbrica.

Como trabajo futuro para un caso de la vida real, donde los nodos se encuentren distribuidos en un ambiente no muy accesible, el hecho de que estas aplicaciones sean mono-thread (mono-hilo) no permite la auto-reprogramación del nodo. Esto equivale a resetear el nodo manualmente. Por lo tanto, técnicas de Inteligencia Artificial tales como los agentes móviles se deben considerar para este tipo de limitaciones y así, permitir la escalabilidad de la red.

Aunque esta propuesta equivale a reconsiderar la arquitectura de un SensorDB, brinda una mejor forma de administración. Por ejemplo, en este caso podría ser que un agente móvil cargue la información de TinyDB y otros agentes de software ejecuten tareas adicionales de optimización, tal como el modelo de Sistema Multi-Agente presentado por Piedrahita en [28]. Además, la utilización de agentes podría permitir un modo de compatibilidad entre varias plataformas de nodos para una sola aplicación y viceversa.

AGRADECIMIENTOS

El trabajo de investigación presentado en este artículo fue financiado parcialmente por el proyecto de investigación con vigencia 2008-2009, de la DIME (Dirección de Investigaciones de la Universidad Nacional de Colombia - Sede Medellín), titulado: "Modelo de sistema híbrido inteligente para el monitoreo de variables físicas usando redes de sensores inalámbricos y sistemas multi-agente" con código: 20201007312, de la Universidad Nacional de Colombia - Sede Medellín.

REFERENCIAS

- [1] J. Yick, B. Mukherjee, y D. Ghosal, "Wireless sensor network survey," Department of Computer Science, University of California, Davis, CA 95616, United States, 2008.
- [2] L. Qiong y W. Hejun, "System Design Issues In Sensor Databases," Department of Computer Science and Engineering. The Hong Kong University of Science & Technology, 2007.
- [3] L. Al-Jadir, "Sensor Network Databases - An Introduction," Advanced Research Topics in Databases Summer 2004-2005.
- [4] C. Schurgers, V. Tsiatsis, y M. B. Srivastava, "STEM: Topology Management for Energy Efficient Sensor Networks," University of California, Los Angeles. IEEEAC paper #260, Updated Sept 24, 2001.
- [5] W. Ye, J. Heidemann, y D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," In Proceedings of the IEEE Infocom, pp. 1567-1576. New York, NY, USA, USC/Information Sciences Institute, IEEE. June, 2002.
- [6] T. V. Dam y K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," Faculty of Information Technology and Systems Delft University of Technology. The Netherlands. SenSys'03, November 5-7, 2003, Los Angeles, California, USA.
- [7] C. Intanagonwiwat, R. Govindan, y D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," USC/Information Sciences Institute; and University of California, Los Angeles. MobiCom, 2000.
- [8] A. Woo, T. Tony, y D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," SenSys'03, November 5-7, 2003, Los Angeles, California, USA.
- [9] C. Perkins, E. Belding-Royer, y S. Das, "Ad hoc On-demand Distance Vector (AODV) Routing," University of California, Santa Barbara. S. Das, University of Cincinnati, RFC 3561, July 2003.
- [10] M. L. Sichitiu, "Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks," IEEE INFOCOM 2004.
- [11] B. Hohlt, L. Doherty, y E. Brewer, "Flexible Power Scheduling for Sensor Networks," IPSN'04, April 26-27, 2004, Berkeley, California, USA.
- [12] H. Wu, Q. Luo, y W. Xue, "Distributed Cross-Layer Scheduling for In-Network Sensor Query Processing," Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06) 2006.
- [13] A. Dunkels, B. Grönvall, y T. Voigt, "Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors," The 29th Annual IEEE Conference on Local Computer Networks, 2004.
- [14] C. C. Han, R. Kumar, R. Shea, E. Kohler y M. Srivastava, "A Dynamic Operating System for Sensor Nodes," International Conference on Mobile Systems, Applications, and Services, 2005.
- [15] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, y R. Han, "MANTIS OS: An Embedded Multithreaded Operating System For Wireless Micro Sensor Platforms," ACM/Kluwer Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks, vol. 10, no.

4, pp.563-579, Aug 2005.

- [16]L. Gu y J. A. Stankovic, "T-kernel: Providing Reliable OS Support for Wireless Sensor Networks," In Proc. of the 4th ACM Conf. on Embedded Networked Sensor Systems (SenSys), [online], 10 Oct 2008, Available <http://www.cs.virginia.edu/~lg6e/t-kernel/t-kernel.draft.pdf>
- [17]D. Chu, L. Popa, A. Tavakoli, J. M. Hellerstein, P. Levis, S. Shenker, y I. Stoica, "The Design and Implementation of A Declarative Sensor Network System," The 5th ACM Conference on Embedded networked Sensor Systems (SenSys 2007), Sydney, Australia, 6-9 Nov, 2007.), [Web], <http://db.cs.berkeley.edu/dsn/> [Last Access, 15 Oct 2008]. IEEE Criteria for Class IE Electric Systems (Standards style), IEEE Standard 308, 1969.
- [18]UC Berkeley. TinyOS -- An open-source OS for sensor networks. Fifth International TinyOS Technology Exchange (TTX5), February 22nd, Berkeley, CA. [Web] www.tinyos.net [Last Access, 01 Dec 2008].
- [19]S. Madden, J.Hellerstein, y W. Hong, "TinyDB: In-Network Query Processing in TinyOS," UC Berkeley. Version 0.4. September, 2003. [Web] <http://telegraph.cs.berkeley.edu/tinydb/> [Last Access, 22 Oct 2008].
- [20]R. Müeller, G. Alonso, y D. Kossmann, "SwissQM: Next Generation Data Processing in Sensor Networks," Department of Computer Science ETH. Zurich, Switzerland. In Proceedings of the 3rd Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10th 2007. [Web] <http://www.swissqm.inf.ethz.ch/> [Last update, 8 July 2008] [Last Access, 20 Oct 2008].
- [21]L. Giraldo, N. Montoya, "Monitoreo y control remoto de variables de un invernadero," Tesis de grado para optar al título de ingeniero (a) físico (a). Universidad Nacional de Colombia, Sede Medellín, 2005.
- [22]A. Montoya, D. Aristizábal, A. Piedrahita, D. González, y N. Montoya, "Monitoreo de variables medioambientales mediante redes de sensores inalámbricas usando motes," VIII Congreso Colombiano de Meteorología y Climatología, Marzo, 2008.
- [23]J. G Gerhke, M. Calimlim, W. Fu Fung, D. Sun, "COUGAR DESIGN AND IMPLEMENTATION," Computer Science Department, Cornell University, 2002.
- [24]R. Szewczyk, A. Mainwaring, J. Polastre, D. Culler y J. Anderson. Great Duck Island. [Web] www.citris-uc.org/research/projects/great_duck_island UC Berkeley.
- [25]J. Miranda, Canary Islands (Spain), 2002. The Rendezvous [Web]<http://www.iuma.ulpgc.es/users/jmiranda/gnat-rts/node20.htm> [Last Access, 15 Oct 2008].
- [26]S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, L. Yin, y F. Yu, "Data-Centric Storage in Sensornets," ICIR, UCLA, UC Berkeley. SIGCOMM 2002, February 1st, 2002.
- [27]X. Li, Y. Jin kim, R. Govindan y W. Hong, "Multi-dimensional Range Queries in Sensor Networks," [Web], <http://enl.usc.edu/projects/dim/> [Last Access, 15 Oct 2008].
- [28]A. Piedrahita, "Mecanismos de programación de una Red de sensores inalámbricos (Wireless Sensor Networks) usando inteligencia artificial," Trabajo Dirigido de Grado. Carrera de Ingeniería de Sistemas e Informática. Universidad Nacional de Colombia -Sede Medellín, 2008.



Diana Carolina Restrepo Patiño.

Estudiante de segundo semestre de Maestría en Ingeniería de Sistemas de la Universidad Nacional de Colombia - Sede Medellín. Ingeniera de Sistemas e Informática (2008) egresada de la Facultad de Minas. Integrante del grupo de Instrumentación Científica e Industrial Categoría C, Ciencias, Integrante del GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial, Categoría A de Colciencias. Trabaja actualmente como auxiliar de investigación del proyecto de

investigación titulado: "Modelo de sistema Híbrido inteligente para el monitoreo de variables físicas usando redes de sensores inalámbricos y sistemas multi-agente". El área de énfasis de su investigación es Inteligencia Artificial, más específicamente Sistemas Híbridos Inteligentes integrando Redes de Sensores Inalámbricos y Sistemas Multi-Agente aplicados a la Computación Ubicua, Simulación basada en el paradigma Multi-agente.



Demetrio Arturo Ovalle Carranza.

Profesor Asociado, Escuela de Sistemas, Universidad Nacional de Colombia - Sede Medellín. Director del GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial, Categoría A de Colciencias. Ingeniero de Sistemas y Computación, Universidad de los Andes, Bogotá, Colombia (1984). Magíster en Informática del Institut National Polytechnique de Grenoble, Francia (1987). Doctor en Informática de la Université Joseph Fourier,

Francia (1991). El área de énfasis de su investigación es Inteligencia Artificial, más específicamente Sistemas Híbridos Inteligentes integrando Redes Neuronales, Sistemas de Lógica Difusa, Redes de Sensores Inalámbricos y Sistemas Multi-Agente aplicados a la Computación Ubicua, Simulación de los Mercados de Energía, Detección de Fallas en Líneas de Transmisión, E-learning, etc. Otros tópicos de investigación que trabaja actualmente son: Inteligencia Artificial en Educación, Sistemas Tutoriales Inteligentes, Sistemas basados en CBR (Case- Based Reasoning) y Técnicas de Planificación Inteligente aplicadas a la Construcción de Sistemas de Composición de Servicios Web.



Alcides de J. Montoya Cañola.

Profesor Asistente, Escuela de Física, Universidad Nacional de Colombia - Sede Medellín. Miembro del grupo de Instrumentación Científica e Industrial Categoría C, Colciencias, Miembro del grupo GIDIA, categoría A de Colciencias. Estudiante de Doctorado en Ingeniería, Facultad de Minas, Universidad Nacional de Colombia. Físico de la Universidad de Antioquia, Medellín, Colombia (1997). Magíster en Ingeniería Informática de la Universidad EAFIT(2004) . Las áreas de énfasis de su

investigaciones son: Redes de sensores inalámbricos, Inteligencia Artificial, más específicamente Sistemas Híbridos Inteligentes integrando Redes Neuronales, Sistemas de Lógica Difusa, Redes de Sensores Inalámbricos y Sistemas Multi-Agente aplicados a la Computación Ubicua, sociedades artificiales inteligentes y física aplicada.