

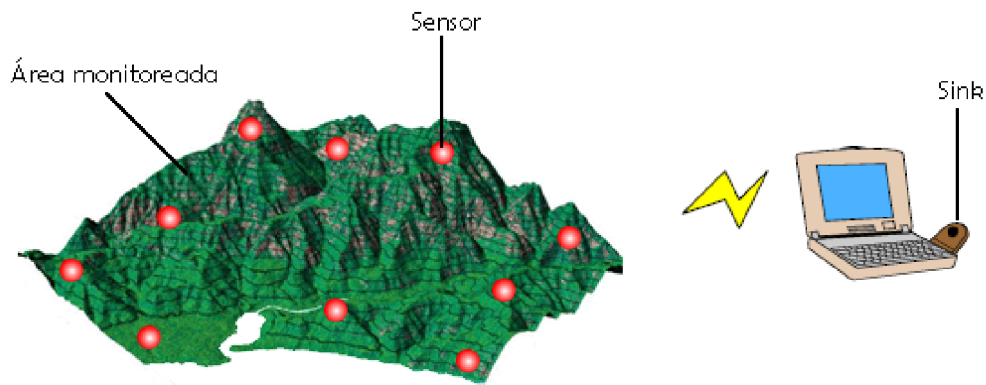
## 2 Marco teórico

Este capítulo describe el marco teórico sobre el cual se basa este trabajo. El capítulo está organizado de la siguiente manera. La sección 2.1 define una red de sensores. Primero describe el trabajo de un sensor dentro de la red. Luego muestra las topologías usadas para las redes de sensores. La sección 2.2 presenta un panorama general de un sistema de base de datos de sensores para recuperar y consultar datos censados en la red. Finalmente, la sección 2.3 concluye el capítulo.

### 2.1 Red de sensores

Una red de sensores (*Wireless Sensor Network*, WSN), es una red formada por componentes autónomos llamados nodos o sensores, los cuales tienen como tarea principal recolectar datos [BHGHT03]. Un componente necesita una cantidad mínima de energía para asegurar que las baterías duren la máxima cantidad de tiempo. Tiene procesadores empotrados, con poca capacidad de cálculo, comunicación inalámbrica mediante un protocolo de comunicación pre establecido, pueden ser emisores y receptores de datos. En la Figura 2.1 podemos observar la representación de una red de sensores, donde cada sensor procesa y manda la señal a un dispositivo conectado a la computadora que se encarga de recibir toda la información que mandan los sensores, este dispositivo se conoce como sink.





**Figura 2.1 Representación de una red de sensores**

Los sensores son distribuidos al azar en un espacio amplio donde actúan mientras los recursos energéticos no se agoten. Esta diseminación es lo que dio origen a su denominación de *smart dust* (polvo inteligente) [KKP99].

### 2.1.1 Sensor

Un sensor, también conocido como *mote*, es un nodo capaz de recibir y procesar información de un medio ambiente y, a su vez, comunicarse con otros nodos dentro de una red [ASSC02]. Un sensor está siempre en contacto con el medio y detecta y emite señales que transitan por la red. Las señales deben ser convertidas en señales estándar para poder procesarlas y observar datos asociados a variables diferentes tipos como temperatura, intensidad luminosa, distancia, aceleración, inclinación, desplazamiento, humedad y pH. En la Figura 2.2 se puede observar la arquitectura de un sensor, donde sus principales componentes son: micro controlador, transceptor, memoria, fuente de poder y uno o más sensores.



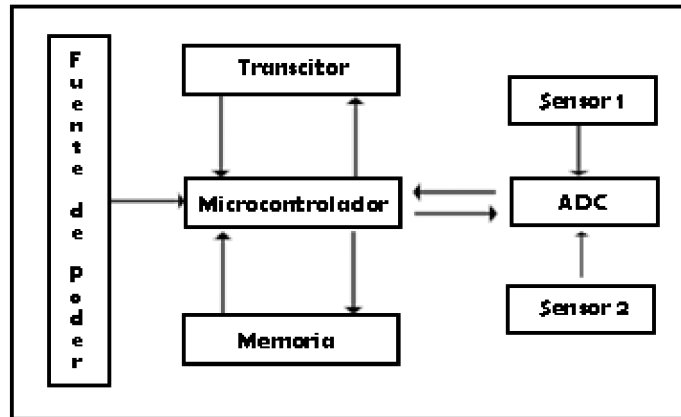


Figura 2.2 Arquitectura de un sensor

El micro controlador realiza tareas, procesa datos y controla los otros componentes del sensor. Otras alternativas que se pueden utilizar como controladores son: microprocesadores, procesadores de señales digitales, y circuitos integrados. Los micro controladores son generalmente la mejor opción para los sistemas empotrados debido a su flexibilidad para conectarlos con otros dispositivos, programarlos y porque el consumo de energía es menor, puesto que estos dispositivos pueden estar inactivos y una parte del controlador puede estar activa. Existen muchos tipos y diferentes marcas de micro controladores como el modelo OKI ARM ML67Q500x, de la compañía *Chipcon* que fue recientemente comprada por *Texas Instrument*.

Los nodos hacen uso de la frecuencia de banda ISM (*Industrial, scientific, medical*) que fueron inicialmente reservadas para uso exclusivo de las industrias y propósitos científicos. Esta frecuencia de radio tienen un espectro enorme además tiene disponibilidad global. Existen varias opciones para entablar una comunicación inalámbrica, como la radiofrecuencia, la comunicación óptica (láser) e infrarroja. El láser requiere menos



energía, pero necesita que los sensores se “vean” directamente, es decir no puede interferir factores físicos tales como paredes, puertas, etc; además de ser sensible a las condiciones atmosféricas [AK04]. La comunicación infrarroja al igual que el láser no necesitan una antena para transmitir, pero su capacidad de transmisión es muy limitada. La comunicación basada en radiofrecuencia (RF) es la que mejor se ajusta a las redes de sensores inalámbricas. Estas redes usan una frecuencia entre 433 MHz y 2.4GHz.

El transceptor en un sensor realiza las tareas de transmitir y recibir señales, los transceptores no tienen un identificador único. Los estados en los que puede operar un sensor son: transmitir, recibir, idle, sleep. Cuando un sensor está en estado idle gasta la misma cantidad de energía como si estuviera recibiendo señales. Es mejor apagar el sensor cuando no esté transmitiendo o recibiendo señales, también se consume una gran cantidad de energía cuando pases del estado apagado al estado para transmitir un paquete.

En lo que respecta a la memoria, se pueden usar dos tipos de memorias, una es la memoria en un chip que contiene el sensor, la otra es memoria flash. La memoria flash se usa generalmente dado su bajo costo y su capacidad de memoria. Los requerimientos de memoria de un sensor dependen de la aplicación, la capacidad de memoria se utiliza para almacenar datos y para programar al sensor.

El consumo de energía en el sensor se da por las tareas de sensores y comunicación. La comunicación es la que más energía consume. El consumo de energía en un sensor para transmitir 1KB a una distancia de 100 metros es aproximadamente el mismo que usa para ejecutar 3 millones de instrucciones. La energía se almacena en baterías o capacitores, las baterías son la fuente principal de alimentación para el sensor. Se usan dos tipos de



baterías, las que son recargables y las no recargables que a su vez se clasifican según el material electroquímico usado para el electrodo tales como el NiCd (nickel-cadmium), NiZn(nickel-zinc), Nimh (hidruro de metal del níquel), y Litio-Ion. Actualmente se están desarrollando sensores que puedan renovar su energía a través de la energía solar, termoeléctrica y vibraciones [LKR04].

Las dos políticas usadas para el ahorro de energía son el DPM y DVS. DPM se encarga de apagar las partes del sensor que no son utilizadas o que no están activas. El DVS (*Dynamic Voltage Scaling*) varía la energía usada dependiendo del trabajo requerido, obteniendo una variación en el voltaje junto con la frecuencia es posible obtener una reducción cuadrática en el consumo de energía [YHE02].

### 2.1.2 Diseño de una red de sensores

El funcionamiento básico de una red inalámbrica de sensores se inicia desplegando los sensores en un área. Al hacer esto, se establece una asociación entre los sensores y entre cada sensor y los objetos o ambientes que van a ser observados. Esta asociación puede ser una a una o bien destinar varios sensores a un solo objeto.

Las redes inalámbricas de sensores están diseñadas para funcionar de manera autónoma, la no ser que necesiten de algún tipo de mantenimiento (como reemplazar sensores dañados o sin energía). Los resultados de la tarea son obtenidos por el [link](#) el cual puede estar conectado con una computadora portátil o un servidor web.



Las tareas de los sensores son especificadas por consultas de alto nivel, es decir, mediante el uso de expresiones como: “reportar las condiciones de humedad y temperatura promedios del área”. Para lograr resolver este tipo de consultas, los sensores trabajan con una pequeña parte de la tarea de manera individual para posteriormente fusionarla y obtener un resultado global esperado.

Las redes de sensores pueden utilizar distintas tecnologías inalámbricas, incluyendo la IEEE 802.11, LANS inalámbricas, bluetooth e identificación de la frecuencia de radio [MPS07]. En la actualidad los nodos pueden tener un alcance de hasta 80 metros y velocidades de hasta 300 Kb/segundo. Existen características únicas para una red inalámbrica de sensores:

- Cantidad limitada de energía que se puede conservar para aprovecharse después.
- Condicionales ambientales difíciles.
- Fallas en los nodos
- Movilidad de los nodos
- Topología dinámica de la red
- Fallas en la comunicación
- Heterogeneidad de nodos
- Escalamiento de la red
- Los nodos operan sin que tengan que ser constantemente supervisados.

Los requisitos para el diseño y el proceso de las redes de sensores son: LS, RF, la autonomía, seguridad, adaptabilidad, optimización. Las redes inalámbricas de sensores



autónomas. Un número enorme de nodos en las redes de sensores requieren soluciones sofisticadas para la organización automática de la red. Por lo tanto, el software de un nodo se tiene que adaptar autónomamente a la infraestructura (ancho de banda y confiabilidad de la red) de la red para la interacción con sus nodos vecinos. El proceso final de cada tarea nos da en conjunto resultados más exactos y nos dará la pauta para otras aplicaciones. Las redes de sensores requieren mecanismos de seguridad que se puedan adaptar a las condiciones ambientales de la red. Todos los algoritmos y protocolos deben tener en cuenta que la energía de los nodos debe ser optimizada.

### 2.1.3 Funcionamiento de una red de sensores

El desarrollo del software para una red de sensores inalámbrica no es nada fácil, ya que los principios convencionales de comunicación no se aplican debido a la topología dinámica que presentan y a la necesidad de procesar tareas conjuntas en la red [LS06]. Las redes alámbricas se basan generalmente en el principio de cliente-servidor con el cliente enviando una petición y contestada por el servidor. Sin embargo, la comunicación en las redes de sensores inalámbricas se basa en los eventos ocurridos en la red. En las redes de sensores se pueden realizar consultas basadas en la posición del sensor, p.e , “*Cual es la temperatura en la posición (x,y)?*”

En las redes que tienen un gran número de sensores, los paquetes tienen que ser enrutados pasando por todos los nodos hasta llegar al *sink*, esto quiere decir que los nodos además de recopilar (censar) datos, tienen la tarea de mantener la integridad de la red. Estas tareas afectan considerablemente el tiempo de vida y la funcionalidad de un nodo. El diseño y la



implementación del software de la red es muy importante ya que se tienen que compensar la tarea específica de la que se encarga cada sensor, la funcionalidad de la red y el tiempo de vida de los nodos. La decisión para implementar un algoritmo específico de enrutamiento depende de varios criterios, tales como movilidad de la red, retraso en la transmisión (*delay*), o consumo de energía. En términos del consumo de energía, implementar un protocolo reactivo, puesto que las rutas de envío de señales se establecen solamente a por la demanda de un sensor a otro [HV03].

Una ventaja de una red de sensores inalámbrica es la posibilidad de implementar algoritmos cooperativos. Estos algoritmos reducen potencialmente el tráfico en la red por pre procesamiento y agregación de datos. Para el uso de un sensor, no es importante si la agregación de datos se realiza dentro sí mismo o por un sensor vecino. Sin embargo, la ruta de comunicación por donde la señal es enviada al *sink* tiene que ser reducida al mínimo, ya que la distancia donde el *sink* se localiza generalmente está lejos.

Un ejemplo para un algoritmo cooperativo es determinar la localización de los sensores por la triangulación. Este algoritmo necesita las medidas de por lo menos tres diferentes nodos. Las posiciones una vez procesadas se pueden utilizar para direccionar o enrutar datos [QKX02].

La selección apropiada de los mecanismos para la selección de las redes inalámbricas depende del uso que se le vaya a dar a la red y de las condiciones ambientales. Además tiene que ser considerados los recursos de los sensores (capacidad de procesamiento, capacidad de memoria y energía). Además de los requisitos generales de seguridad





como disponibilidad, confidencialidad, integridad, y la autenticación, así como también los requisitos especiales tales como claridad del envío de datos, detección de intrusos y contención de datos.

Para satisfacer las peticiones de la seguridad anteriormente mencionadas, se han adaptado mecanismos para la seguridad de las redes inalámbricas y han sido desarrollados y probados por varios grupos de investigación [LS06]. Una de las tareas principales para un software de sensores es la selección y el manejo de los mecanismos de seguridad basándose en las políticas de seguridad definidas por el operador o administrador de la red.

Los sensores tienen como fuente principal de energía una batería, sin embargo los sensores son demasiado pequeños y generalmente muy numerosos para sustituir o recargar las baterías de cada uno. Por otra parte, hay veces que las redes se despliegan en un ambiente alejado o peligroso. Por lo tanto, el maximizar el ciclo de vida de los nodos es un desafío principal del diseño y la implementación de la red. Un algoritmo de *scheduling* por ejemplo podría realizar una selección apropiada de los protocolos de comunicación para reducir el consumo de energía. Las capas que tienen un mejor funcionamiento para ahorrar energía son la física, de enlace y la de red.

#### 2.1.4 Topología

La topología de una red inalámbrica de sensores es dinámica, por lo que los nodos tienen que adaptarse para poder comunicar nuevos datos adquiridos. La topología más usada en

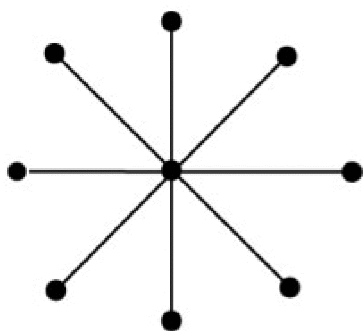


una red de sensores inalámbricas es la de malla, donde cada nodo está conectado a uno o más nodos dentro de la red, como se puede observar en la Figura 2.3. Los nodos pueden enviar y recibir información por diferente rutas [PHCSS03].



Figura 2.3 Topologías típicas de una red de sensores

La Figura 2.4 existen también la topología de estrella y de árbol, están topologías rara vez se usan debido a que el mal funcionamiento de un solo nodo puede afectar a toda la red, este problema se explica a continuación.



Estrella

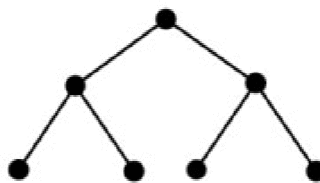


Figura 2.4 Topologías que no garantizan un buen funcionamiento de la red



Como lo mencionábamos anteriormente un aspecto muy importante dentro del diseño de la topología de la red es la tolerancia a fallas, ya que algunos nodos pueden fallar o bloquearse debido a la falta de energía, algún daño físico o que exista una interferencia ambiental. Algún fallo ocurrido en los nodos no debe de afectar la tarea global de la red. La tolerancia de error es la habilidad de sostener las funciones de la red sin ninguna interrupción debido al fallo de un nodo [WMGM07].

En la Figura 2.5 se puede observar que en la imagen (a) la red está diseñada con una topología de malla, esto nos permite ver que los nodos que están fallando no afectan a la red, ya que solo un nodo de toda la red queda inservible. Al contrario de la imagen (b) el mal funcionamiento de la misma cantidad de nodos en la red hace que por lo menos la mitad de esta queda sin funcionamiento, esto se debe a la topología usada la cual es de tipo árbol.

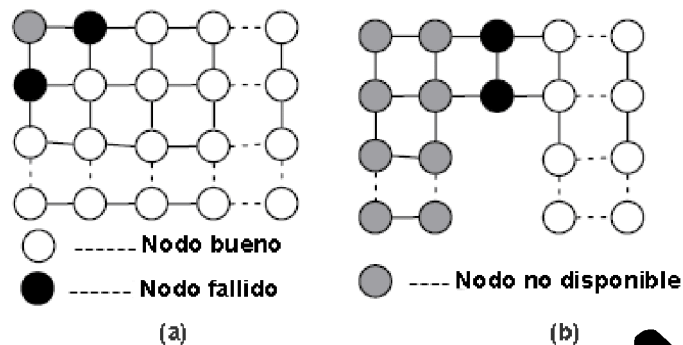


Figura 2.5 Topologías tolerantes a fallas



## 2.2 Base de datos de sensores

Una base de datos de sensores es un *middleware*, que es una capa del software entre el sistema operativo y el uso del sensor, y entre las aplicaciones del sensor con la red en general [LS06]. El objetivo primario de esta capa es ocultar la complejidad de la red aislando el manejo de las aplicaciones y del protocolo al uso de memoria, la funcionalidad y el paralelismo de la red. Un *middleware* para las redes del sensor tiene que ser: escalable, genérico, adaptable, reflectivo.

Los recursos como la memoria, velocidad en el procesamiento y el ancho de banda requieren una optimización para que puedan usarse en el nodo, esta optimización se realiza al momento de compilar el programa, nos hace tener un *middleware* escalable. Los componentes del *middleware* requieren de una interfaz genérica para reducir la tarea de diseñar otras aplicaciones o nodos. El uso de un *middleware* con componentes similares en diferentes aplicaciones nos conduce a un gran número de interfaces más complejas, la reducción de estas es el objetivo de un *middleware* genérico [RKM04].

La movilidad de los nodos y los cambios en la infraestructura requieren que el *middleware* se adapte mientras se ejecuta, dependiendo del uso de la red. El *middleware* debe ser capaz de intercambiar y funcionar dinámicamente con los componentes haciendo así el *middleware* adaptable.



La reflexión cubre la capacidad de un sistema para entender e influenciarse a sí mismo, un sistema reflexivo puede mostrar su propio comportamiento, de este modo se usan dos mecanismos, la inspección y la adaptación a su propio comportamiento. Un *middleware* que es reflexivo no intercambia componentes si no que cambia su comportamiento para adaptarse a ellos. Un ejemplo de un comportamiento reflexivo es la modificación de las estrategias de ruteo dependiendo la movilidad de los nodos.

Algunos sistemas *middleware* que presentan base de datos son los *TinyDataBases*, en este sistema los usuarios utilizan consultas de manera similar a SQL en las que se solicita que la red realice ciertas tareas. Proporciona una tabla virtual llamada “*sensors*” (sensores), en la cual cada columna corresponde al valor de un tipo específico de sensor (temperatura, luz, etc.) u otra fuente de datos de entrada (identificador del nodo, energía restante del nodo, etc.). Las lecturas de los nodos se agregan como nuevos registros [WMGM07].

El lenguaje de consulta es SQL con algunas extensiones, el sistema de consultas declarativas proveído por *TinyDB* es muy fácil de usar, además, gracias a que está basado en bases de datos éste oculta los mecanismos de distribución al usuario. Como se observa en la Figura 2.6 la recuperación de datos en la red se logra a través de un árbol de esparcimiento donde la raíz es el dispositivo del usuario, se realiza una consulta descentralizada donde cada nodo tiene su propio procesador de consultas que pre procesa y envía los datos censados solicitados directamente al dispositivo del usuario, en donde los resultados son recibidos.



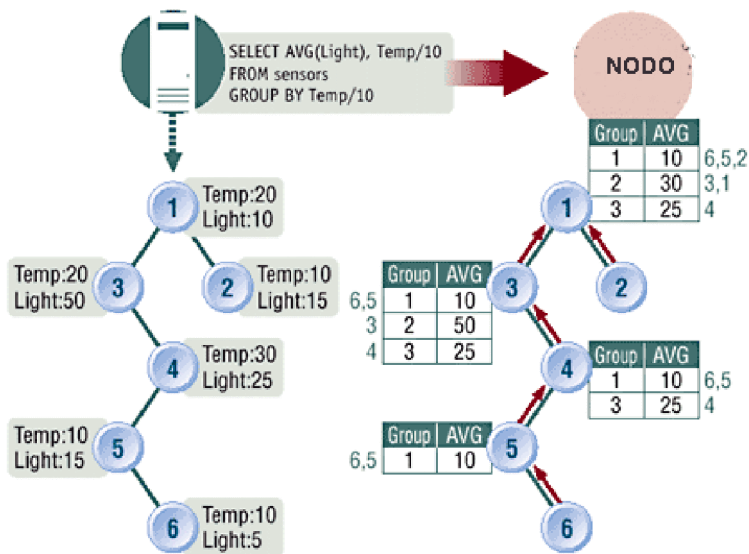


Figura 2.6 Algoritmo para recuperar datos de la red [<http://itpro.nikkeibp.co.jp/>]

Dada una consulta específica, *TinyDB* recoge esos datos de los motes, una vez que tiene los datos, realiza las operaciones necesarias y los enruta hacia una PC. *TinyDB* hace esto a través de los algoritmos los cuales aseguran la energía suficiente para realizar las tareas.

Para utilizar *TinyDB*, se instala sobre cada mote en la red *TinyOS*. *TinyDB* proporciona una API de Java para que se puedan programar aplicaciones en una PC, tales como consultas y otras aplicaciones para obtener datos de la red, también incluye una interfaz gráfica para hacer consultas y un *display* para ver los resultados.

La meta fundamental del *TinyDB* es hacer que el programador pueda realizar su tarea más fácil, y provee datos que puedan ser desarrollados en el menor tiempo posible. Con *TinyDB* no es necesaria la programación a bajo nivel, ya que nos facilita interfaces de redes de



sensores [MHH03]. Como se puede ver en la Figura 2.7 se realizan consultas a través de la interfaz de *TinyDB*.

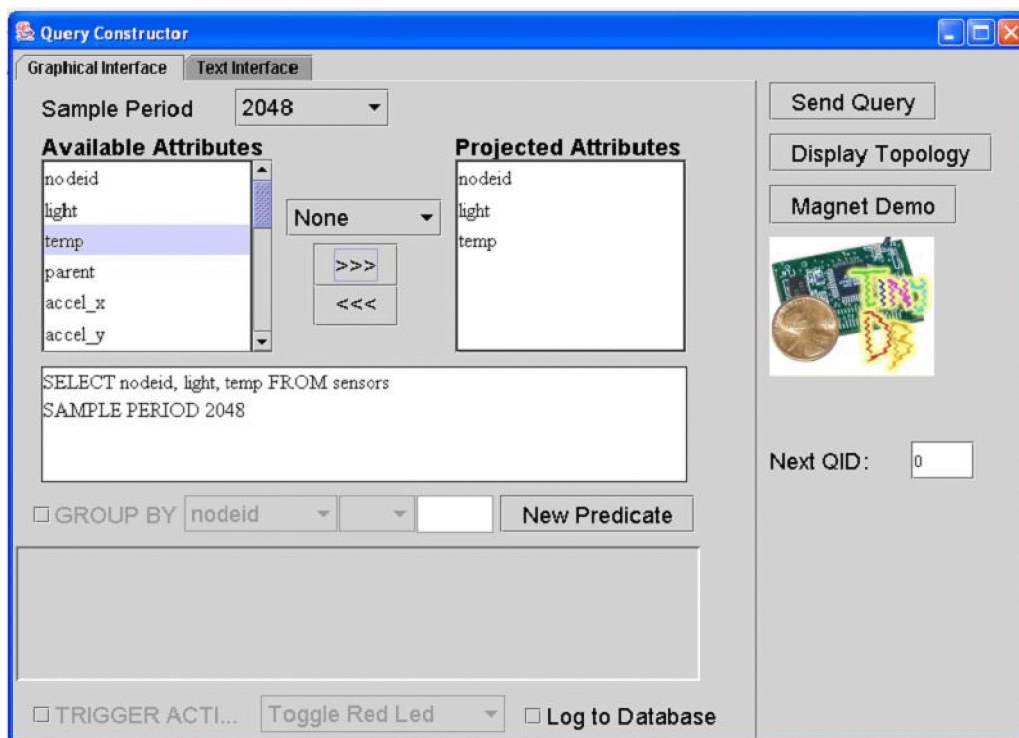


Figura 2.7 Interfaz de TinyDB [<http://www.tinyos.net/>]

Algunas características de los *TinyDB* son:

- Administración de metadatos: *TinyDB* proporciona un catálogo de metadatos, el cual describe las clases de lectura de un sensor que estén disponibles en una red de sensores.
- Consultas de alto nivel: *TinyDB* utiliza un lenguaje declarativo de consultas, que permite describir los datos que se buscan, sin ser necesario decir cómo conseguirlos.



Esto hace más fácil crear aplicaciones y nos ayuda a garantizar el buen funcionamiento de la aplicación independientemente si la red de sensores cambia o no.

- Topología de la red: *TinyDB* administra los radios adyacentes de cada nodo a través de sus nodos vecinos, manteniendo tablas de enrutamiento y asegurándose de que cada mote en la red entrega eficientemente y confiablemente los datos al usuario.
- Consultas múltiples: *TinyDB* permite consultas múltiples al mismo tiempo en los motes, las consultas pueden ser diversas y tener acceso a diversos tipos de sensores.
- Incrementar deployment a través de consultas compartidas: Para extender una red que utilice *TinyDB*, simplemente se tiene que descargar un código estándar de *TinyDB* a los motes, y *TinyDB* se encarga del resto. Los motes con *TinyDB* comparten consultas unos con otros, cuando un mote en la red recibe un mensaje de una consulta que no aun no es procesada, automáticamente pide una copia de la consulta para empezar a correrla. No se requiere ninguna otra configuración o programación en los motes nuevos, con instalarles *TinyDB* es más que suficiente [MHH05].





## 2.3 Conclusiones

Este capítulo definió el concepto de sensor y describió sus características técnicas. También describió cómo se diseñan las redes de sensores inalámbricas, sus características y funciones. Analizó las topologías que se pueden utilizar en una red de sensores para garantizar su funcionamiento. Finalmente, el capítulo describió las bases de datos de sensores, que son un *middleware* que puede procesar consultas. Como ejemplo de *middleware*, se describieron las características principales de *TinyDB*.



