

Universidad Autónoma Metropolitana
Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Diseño de una aplicación Web para visualizar los datos recolectados por una
red de sensores móviles

Modalidad: Proyecto Tecnológico

Trimestre 2018 Primavera

Carolina Salinas García

2133034467

al2133034467@correo.azc.uam.mx

Asesor: Dr. Leonardo Daniel Sánchez
Martínez

Profesor Asociado

Sistemas

ldsm@correo.azc.uam.mx

Coasesor: M. en C. Carlos Ernesto Carrillo
Arellano

Profesor Asociado

Electrónica

ceca@correo.azc.uam.mx

24 de julio de 2018

Yo, DR. LEONARDO DANIEL SÁNCHEZ MARTÍNEZ, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dr. Leonardo Daniel Sánchez Martínez

Yo, M. EN C. CARLOS ERNESTO CARRILLO ARELLANO, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



M. en C. Carlos Ernesto Carrillo Arellano

Yo, CAROLINA SALINAS GARCÍA, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Carolina Salinas García

Resumen

Una red de sensores móviles consiste en dispositivos autónomos distribuidos que poseen movilidad, llamados nodos. Estos cuentan con sensores que recolectan información del área que recorren.

Dada la movilidad de los nodos en la red, su topología sufre cambios constantes, por lo que es indispensable contar con una forma para monitorizarla, un sistema que muestre la posición de los nodos en el área de muestreo y los datos obtenidos por éstos, de igual manera, debe mostrar en tiempo real los cambios que sufren la topología y la conectividad entre los nodos. Este proyecto tuvo como objetivo desarrollar una aplicación web que genera una visualización del estado de la red y los cambios que surgen en su topología. Para esto se desarrolló un módulo de procesamiento y un módulo de visualización.

El módulo procesamiento es el encargado de dar tratamiento a los datos recolectados, además, es el intermediario entre la base de datos y el módulo de visualización, por su parte, el módulo de visualización tiene la función de presentar la información en pantalla.

Palabras Claves: desarrollo de sistemas, red de sensores móviles.

Tabla de contenido

Introducción	1
Justificación.....	2
Objetivos	2
Objetivo General.....	2
Objetivos Específicos.....	2
Marco teórico	2
Red inalámbrica de sensores móviles	2
Aplicación Web	2
Servidor web.....	3
Modelo de tres capas.....	3
Desarrollo.....	5
Base de datos	6
Módulo de visualización	6
Módulo de procesamiento de datos	8
Red inalámbrica de sensores móviles	11
Resultados	14
Análisis y discusión de resultados	19
Conclusiones	19
Referencias	20

Índice de figuras

Figura 1	3
Figura 2	4
Figura 3	5
Figura 4	5
Figura 5	6
Figura 6	7
Figura 7	8
Figura 8	8
Figura 9	9
Figura 10	9
Figura 11	10
Figura 12	10
Figura 13	11
Figura 14	12
Figura 15	12
Figura 16	13
Figura 17	14
Figura 18	14
Figura 19	15
Figura 20	16
Figura 21	16
Figura 22	17
Figura 23	17
Figura 24	18
Figura 25	18

Introducción

En los últimos años se han realizado numerosas investigaciones en torno a redes de sensores inalámbricas o WSN, por sus siglas en inglés, y sus aplicaciones prácticas. Las WSN “consisten en pequeños dispositivos autónomos distribuidos espacialmente, equipados con sensores para monitorear condiciones físicas o ambientales” [1]. Los sensores son capaces de establecer enlaces inalámbricos entre sí para poder transmitir la información recolectada hacia un nodo central denominado como el *sink* de la red [2].

Al implementar una WSN se puede obtener información del área en la que están distribuidos los sensores. Debido a los resultados que se han obtenido gracias a esta aplicación y a su versatilidad, esta tecnología se está abriendo camino en muchos campos, desde el desarrollo de sistemas inteligentes para el transporte público [3], hasta la monitorización de variables ambientales para desarrollar sistemas agrícolas inteligentes [4].

Monitorizar una WSN es indispensable como en cualquier red, por lo que se han desarrollado herramientas que permiten realizarlo, como lo es *Wireless Sensor Networks Monitoring Tool* [5], esta herramienta está desarrollada para proporcionar la visualización de la red, pero su principal objetivo es permitir que se administre la red remotamente. *SpyGlass: A Wireless Sensor Network Visualizer* [6] por otra parte es un marco modular que permite monitorizar una WSN y se centra en la importancia de la visualización de los datos generados por la red.

Una de las desventajas de las WSN es la cantidad de dispositivos autónomos o nodos utilizados que es proporcional a la superficie del territorio monitoreado, por tanto, el número de nodos crece al ampliar la superficie que se requiere monitorear, generando altos costos de infraestructura y una topología de red compleja. Con este problema en mente se está desarrollando una idea que podría disminuir las desventajas de las WSN. La propuesta es agregar movilidad a los nodos, de esta forma el área monitoreada por cada sensor se ampliaría, logrando así reducir la cantidad de nodos y sensores utilizados.

Como ya se ha mencionado anteriormente es importante el monitoreo de la red, una manera de hacerlo de forma remota es por medio de una aplicación web, que permite la visualización de la red y sus datos.

En este proyecto se desarrolló una aplicación web para el manejo de la información obtenida por una red inalámbrica de sensores móviles. El objetivo de la aplicación es generar un plano de coordenadas (x, y) donde se mostrará la ubicación de cada nodo de la red, la conectividad entre ellos, si existiera, y los datos obtenidos por cada sensor.

Justificación

Actualmente se han realizado avances importantes en el área de redes inalámbricas de sensores. Estas redes son implementadas para obtener datos del entorno, y son comúnmente monitoreadas de manera local. Distintas soluciones se han planteado para que la información recabada por los sensores pueda ser consultada desde cualquier parte en cualquier momento. Estas soluciones están implementadas para redes inalámbricas de sensores fijos, pero hasta el momento no se han enfocado en las redes inalámbricas de sensores móviles, por lo cual no se cuenta con una aplicación web que permita monitorear a estas redes de manera remota.

Objetivos

Objetivo General

Diseñar una aplicación web para la visualización de datos recolectados por una red de sensores móviles.

Objetivos Específicos

- Diseñar e implementar un sistema para administrar los datos obtenidos por una red de sensores móviles.
- Diseñar e implementar un módulo para el procesamiento de los datos recabados por la red de sensores móviles.
- Diseñar e implementar un módulo para visualizar la topología de la red, los datos recolectados por cada nodo y la conectividad entre ellos.

Marco teórico

Red inalámbrica de sensores móviles

Una red de sensores inalámbricos móviles o MWSN, por sus siglas en inglés, consiste en dispositivos autónomos distribuidos que poseen movilidad.

“Las MWSN son mucho más versátiles que las WSN estáticos, ya que los nodos se pueden implementar en cualquier escenario y hacer frente a los rápidos cambios de topología” [7].

Los nodos de una MWSN constan de varios sensores (es decir, que pueden medir luz, temperatura, humedad, presión, movilidad, etc.), un microcontrolador, y un transceptor de radio que funciona con una batería.

Actualmente las principales aplicaciones de las MWSN son la economía, monitoreo ambiental, minería, meteorología, monitoreo sísmico, detección acústica, aplicaciones de atención médica, monitoreo de procesos, protección de infraestructura, computación sensible al contexto, navegación submarina, espacios inteligentes, seguimiento de inventario y vigilancia militar táctica.

Aplicación Web

Una aplicación web es una herramienta que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador, esto quiere decir, que todos los datos y archivos con los que se trabaja son procesados y almacenados dentro de la web.

La principal ventaja de trabajar con una aplicación web es la posibilidad de ver el contenido en cualquier dispositivo independientemente del sistema operativo que este posea, ya que esta trabaja bajo el lenguaje de los navegadores web.

Servidor web

Para el desarrollo de una aplicación web es indispensable contar con un servidor web, el cual funciona de la siguiente forma:

1. El navegador web envía la petición (GET) de una página web.
2. El servidor web busca en su repositorio si está la página solicitada.
3. El servidor regresa la información solicitada (página web).
4. El navegador web en el dispositivo despliega la página.

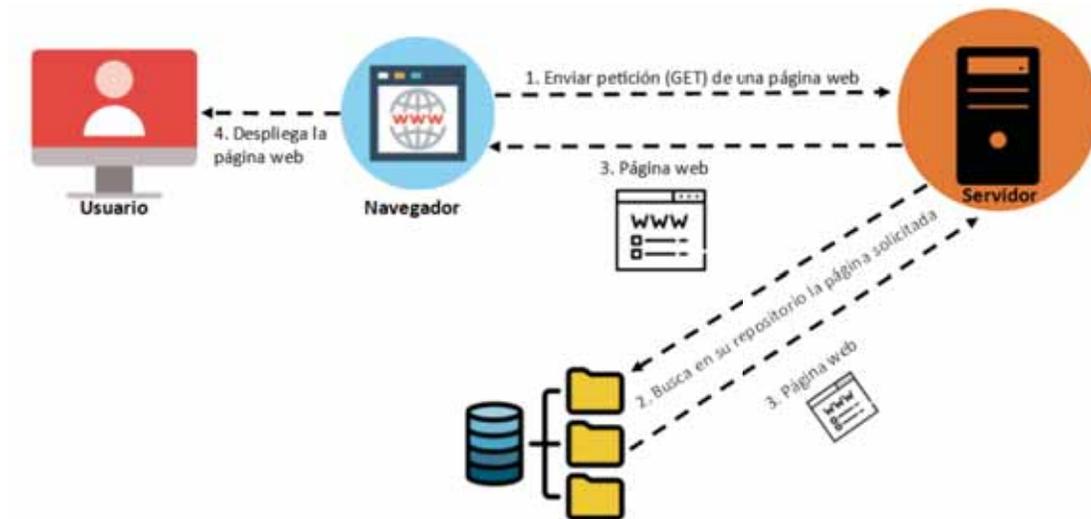


Figura 1. Funcionamiento de un servidor web.

Un servidor Web almacena los archivos de un sitio y los emite por Internet para poder ser visitado por los usuarios.

Modelo de tres capas

Los sistemas informáticos que gestionan datos requieren un repositorio donde se almacenen los datos en cuestión, comúnmente se utiliza una base de datos para ello, demandan también de una interfaz con la cual interactúan los usuarios, además de una parte donde se procesen los datos y se realicen las operaciones necesarias. El modelo de tres capas se basa en dividir el sistema en tres partes diferenciadas y cada parte, o capa, solo se comunica con la inferior. Estas capas son:

- **Persistencia:** Esta capa se encarga de guardar los datos. Será donde se gestione todo lo relativo a la base de datos y a la creación, edición y borrado de datos de ésta.
- **Negocio:** En esta capa se gestiona la lógica de la aplicación, es decir, que operaciones se realizarán con los datos.
- **Presentación:** Esta capa es la encargada de visualizar la información.

La principal ventaja al dividir el sistema en partes es que las capas solo se comunican con la inmediata y cualquier cambio que se realice en ellas solo afectará a esta y no a todo el sistema.

Muchas de las aplicaciones web están desarrolladas siguiendo el modelo de tres capas, aunque este no es recomendado para sistemas en tiempo real.

Una aplicación web orientada al monitoreo de una red inalámbrica de sensores móviles (MWSN) es un sistema que necesita ser consultado en tiempo real, por lo cual el concepto estricto de modelo de tres capas no se puede emplear. Se deben realizar algunos cambios para poderlo utilizar. En la figura 2 se muestra el modelo de tres capas estricto, enfocado para el desarrollo de una aplicación web para el monitoreo de una red inalámbrica de sensores móviles.

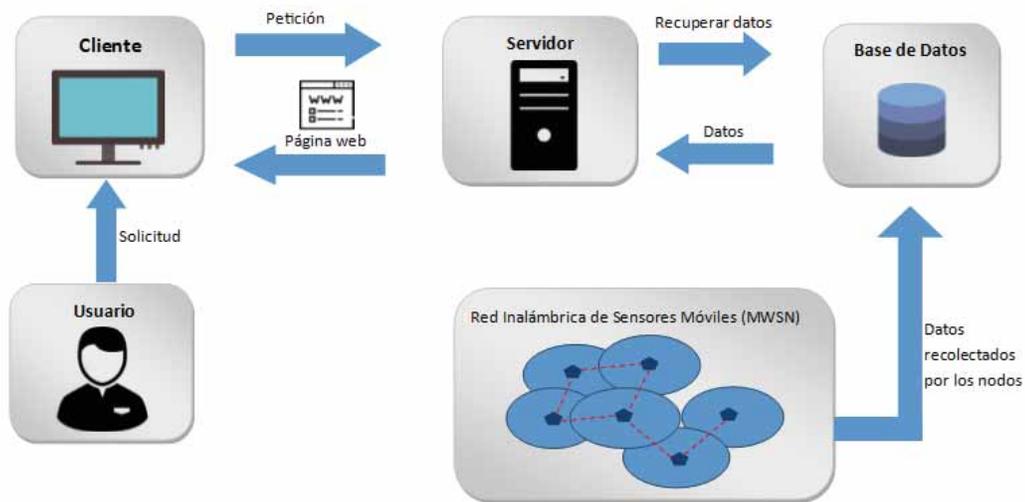


Figura 2. Modelo de tres capas enfocado a una aplicación web.

Nótese que en este caso es necesario que el usuario envíe una solicitud para obtener respuesta del servidor y se muestre la página web, esto no es útil para el monitoreo en tiempo real puesto que es necesario que el usuario actualice la página, es decir, envíe una nueva solicitud al servidor para obtener la página web nuevamente.

Conociendo las desventajas del modelo de tres capas en cuanto al monitoreo en tiempo real se modificó este modelo para poder actualizar la página web sin intervención directa del usuario. Esta modificación se presenta en la figura 3, en esta figura se puede notar que la petición solo se realizará una vez y desde ese instante la página se actualizará sin necesidad de realizar una nueva solicitud.

Teniendo en cuenta los requerimientos de una red de sensores móviles y la necesidad de crear una forma de monitorizar de forma remota, se llevó a cabo la construcción de una aplicación web orientada a el monitoreo de una red inalámbrica de sensores móviles.

La aplicación web desarrollada está enfocada a las MWSN's en general, es decir, no importa que la red sea utilizada en el monitoreo ambiental o monitoreo sísmico la aplicación web se adapta para mostrar la información recolectada para cualquier aplicación.

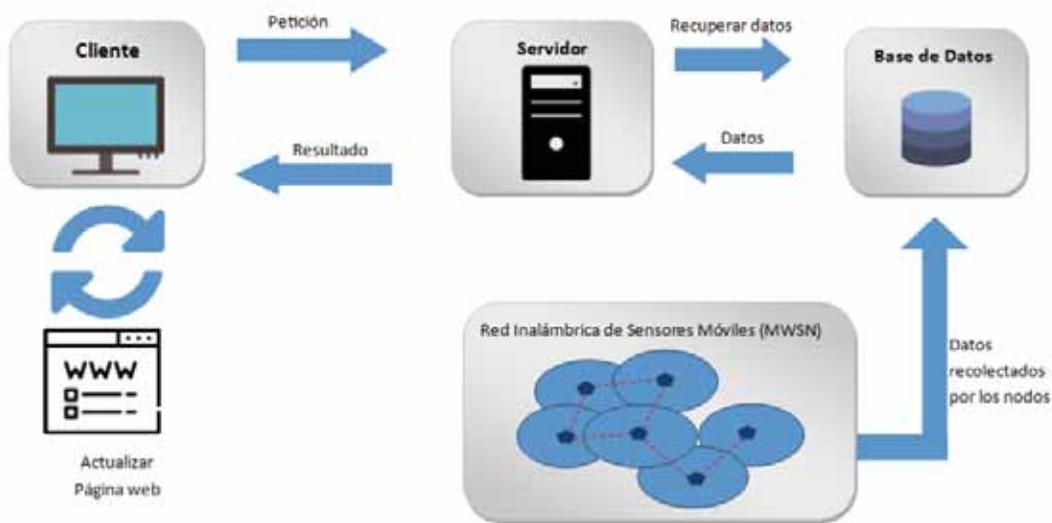


Figura 3. Modelo propuesto para una aplicación web para el monitoreo de una red de sensores móviles.

Desarrollo

La figura 4 se muestra la interacción de los módulos desarrollados, la base de datos y la red inalámbrica de sensores móviles.

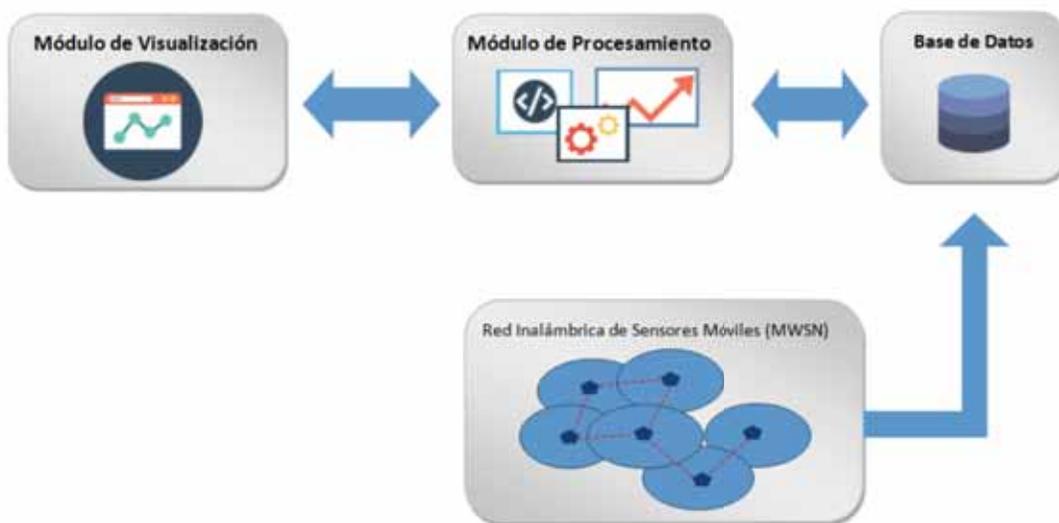


Figura 4. Interacción entre los elementos de la aplicación web.

Los detalles del desarrollo de los dos módulos, su relación e interacción entre sí y con la base de datos, se describen a continuación:

Base de datos

Se utilizó MySQL para el desarrollo de la base de datos utilizada para almacenar los datos obtenidos por la red de sensores móviles, los datos que esta almacena son:

- Identificador del nodo (idNodo)
- Posición en x (posicionX),
- Posición en y (posicionY)
- Tiempo en el que se detecta un movimiento (tiempo)
- Lectura del sensor (lectura).

Se creó una tabla, llamada *movilidad*, dentro de la base de datos llamada *reddesensores*, con 5 columnas que guardan los datos recolectados.

La figura 5 muestra el diagrama entidad-relación relacionado con la base de datos, al ser solo un medio de almacenamiento para la red de sensores móviles, no está relacionada con más entidades.

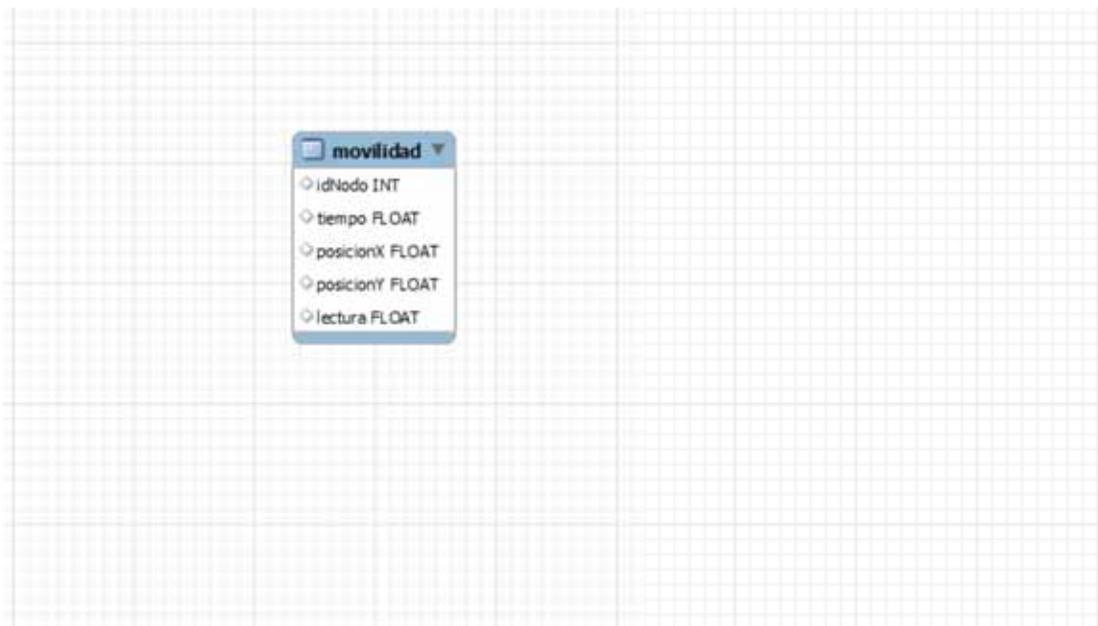


Figura 5. Diagrama Entidad-Relación.

Módulo de visualización

El módulo de visualización es el encargado de presentar la información de la red sensores móviles, este módulo se desarrolló en HTML5.

Se desarrollaron tres páginas para presentar la información de la red, la principal muestra la topología de la red, es decir los nodos, su cobertura y la conectividad, entre ellos, si es que la hay. La segunda muestra una gráfica de las lecturas obtenidas por cada nodo en función del tiempo, y la tercera muestra toda la información de la red como un histórico.

El diseño de las páginas se desarrolló con la ayuda de una plantilla gratuita llamada AdminLTE [8], en la figura 6 se muestra la página principal para realizar la descarga. De esta

se extrajo solamente la página inicial, los iconos y vínculos para navegar dentro de las páginas.

El código de la plantilla se dividió para poder obtener solamente lo necesario para el desarrollo de la aplicación, de esta forma se extrajo de este solamente el *header*, barra de navegación o lateral, barra superior y *footer*, estos elementos son idénticos para las tres páginas.



Figura 6. Página principal AdminLTE

Como la información mostrada en cada página es diferente se necesitó incluir en elementos específicos para cada una de ellas.

La página principal, llamada “mapa”, muestran la topología de la red sobre un canvasHTML5 [9], una etiqueta HTML5 que se usa para dibujar gráficos, sobre la marcha, a través de scripts, este sin embargo no tiene capacidades de dibujo propias (es solo un contenedor); depende de los scripts para dibujar los gráficos. En esta misma área se dibuja la cobertura de los nodos y la conectividad entre ellos si existiese.

La página que muestra la gráfica de las lecturas obtenidas por cada nodo en función del tiempo, llamada por simplicidad “gráfica”, emplea solo una etiqueta HTML (<div>) que será el contenedor para el despliegue de la información, mientras que todas las operaciones se realizan en el módulo de procesamiento.

Por último, la página con la información de la red como un histórico, incluye una tabla para el despliegue todos los datos obtenidos de la red, los cuales ya se mencionaron anteriormente, esta se desarrolló utilizando PHP, para la extracción de la información y HTML para el despliegue de la misma.

En la figura 7 se muestra el funcionamiento a detalle del módulo de visualización, como se puede apreciar en esta figura, solo se realiza una solicitud de la información necesaria para crear el contenido de las páginas, después las páginas se actualizan automáticamente sin necesidad que el usuario envíe una nueva solicitud.

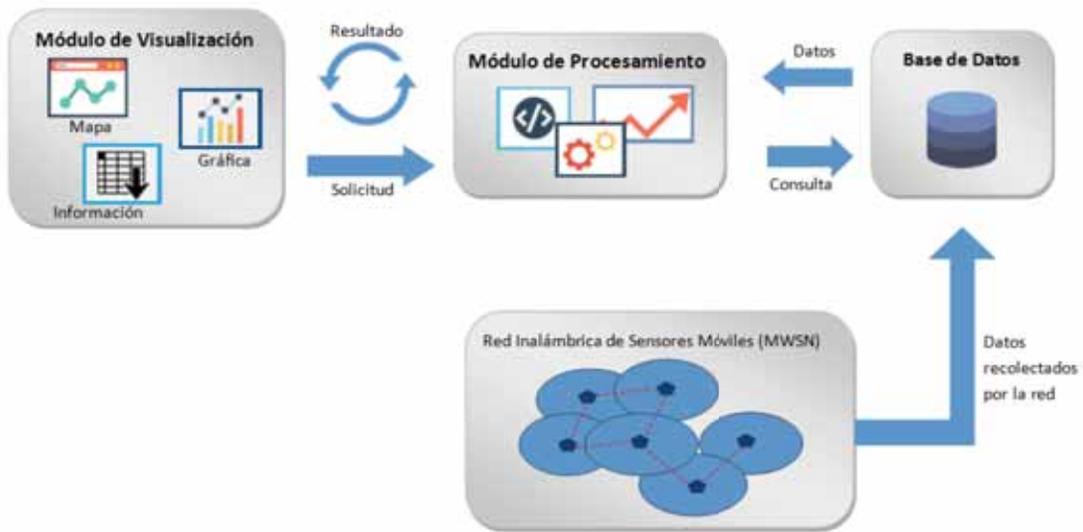


Figura 7. Detalle del módulo de visualización.

Módulo de procesamiento de datos

El módulo de procesamiento es el encargado de extraer y dar formato a los datos para finalmente ser presentados por el módulo de visualización. Este es un intermediario entre la base de datos, de donde constantemente está tomando los datos, y el módulo de visualización que presenta la información de la red de sensores móviles.

Las tecnologías utilizadas para su desarrollo fueron en PHP, para las consultas y preparación de los datos, y JavaScript y Ajax para la lógica de la aplicación.

A continuación, en la figura 8 se muestra como se comunican las clases dentro del módulo de procesamiento.

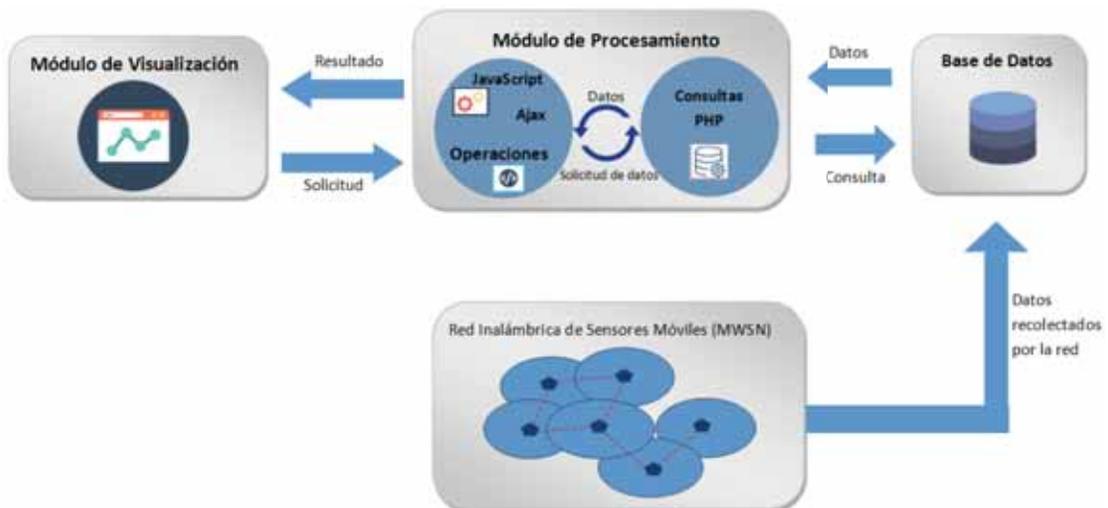


Figura 8. Detalle del módulo de procesamiento de datos.

Como se puede apreciar en la figura 8 las clases para realizar las consultas y dar formato a los datos fueron desarrolladas en PHP, mientras que las encargadas de darle tratamiento, es decir realizar las operaciones y procesos, fueron desarrolladas en JavaScript y Ajax.

La información necesaria para la visualización de la topología de la red, se extrajo de la base de datos. Para ello fue preciso realizar funciones dentro de esta para poder obtener los datos actuales de cada nodo en la red, en otras palabras, obtener los datos del nodo en el tiempo máximo registrado, la figura 9 muestra la consulta sql que se realiza para ello. Esta información es enviada a su vez a un script llamado ajaxNodos.js donde se realizan las operaciones necesarias para su futura presentación.

```
//Conexion con la base de Datos
require ('../funcionesBD/bd_conexion.php');

$sql="select idNodo,tiempo,posicionX,posicionY,lectura from movilidad inner join (select idNodo as
id,MAX(tiempo) as t from movilidad group by idNodo) tmp on movilidad.idNodo=tmp.id where
movilidad.tiempo=tmp.t;";

$resultado = $conn->query($sql);
```

Figura 9. Consulta sql para la extracción de los datos en el tiempo máximo.

En el script ajaxNodos.js se encuentran las funciones para dibujar los nodos, su cobertura y la conectividad sobre el canvasHTML5, mostradas en la figura 10 y 11 respectivamente, incluye también una función para limpiar el área de trabajo y agregar el id de cada nodo, el nodo es dibujado como un círculo relleno con centro en las coordenadas x,y de su posición en ese instante, la cobertura es dibujada como un círculo sin relleno con las mismas coordenadas x,y, la conectividad por su parte es una línea recta punteada, que conecta dos nodos con centros en (x1,y1) y (x2,y2) correspondientemente, pero esta solo será dibujada cuando la distancia entre los nodos es menor o igual a 80 pixeles, que son las unidades que maneja el contenedor canvasHTML5.

La función para determinar la conectividad entre nodos calcula la distancia euclidiana entre estos iniciando la comparación en uno contra todos los restantes, descartándose a sí mismo, continuando descartando las distancias ya calculadas, si la distancia entre los nodos es menor a 80 pixeles la conectividad es dibujada, como se mencionó anteriormente.

```
function dibujaCirculo(n,x,y,color){
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
var colorCirculo = '#00'.concat(color).concat('d60');
ctx.beginPath();
ctx.fillStyle = colorCirculo; //'#6495ED'; //'#B22222';
ctx.arc(x,y,20,0,2*Math.PI);
ctx.fill();
}

function dibujaCobertura(n,x,y){
var c = document.getElementById("myCanvas");
var ctx1 = c.getContext("2d");
ctx1.strokeStyle = '#000000';
ctx1.setLineDash([5, 10]);
ctx1.beginPath();
ctx1.arc(x,y,40,0,2*Math.PI);
ctx1.stroke();
}
```

Figura 10. Código de las funciones para dibujar los nodos sobre la etiqueta canvasHTML5.

```

function dibujaRecta(x1,y1,x2,y2){
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.strokeStyle = '#f20404';//'#B22222';#f4cc04
ctx.lineWidth =5;
ctx.setLineDash([5, 1]);
ctx.beginPath();
ctx.moveTo(x1,y1);
ctx.lineTo(x2,y2);
ctx.stroke();
}

```

Figura 11. Código de la función para dibujar la conectividad sobre la etiqueta canvasHTML5.

El script ajaxNodos.js emplea tecnología AJAX con el objeto XMLHttpRequest para hacer envíos de solicitud de información asíncrona, lo que significa que el módulo de visualización solamente realizará una petición al servidor, este a su vez solo construirá una vez la página web, que corresponde a la página principal llamada “mapa”, después de esta primera solicitud el servidor solo extraerá los datos y los enviará a modo de respuesta para actualizar solamente la etiqueta canvasHTML5 en la página mapa.

```

<div class="content-wrapper" id="contenedor">

<!-- Content Header (Page header) -->
<section class='content-header'>
<h1>RED DE SENSORES MOVILES<small>mapa</small></h1>
</section>
<form id='myValues'> </form>
<!-- Main content -->
<section class='content'>
<!-- Crear la area de tarabajo (Cava) -->
<canvas id='myCanvas' width='500' height='450' style='border:1px solid #d3d3d3;'>
Your browser does not support the HTML5 canvas tag. </canvas>

</section>
<!-- /.content -->

<script src="js/ajaxNodos.js"></script>

</div>

```

Figura 12. Sección del código donde se incluye la etiqueta canvasHTML5.

La figura 12 muestra la estructura de la página “mapa”, en ella se puede observar la etiqueta canvasHTML5, el único componente de la página que se actualiza constantemente.

Las funciones realizadas para dibujar los nodos, la conectividad y la cobertura se convocan dentro del llamado al objeto XMLHttpRequest esto permite monitorear en tiempo real los cambios que ocurren dentro de la topología cambiante de la red.

La extracción de los datos relacionados con las lecturas de los nodos, está a cargo de un archivo PHP llamado servicio-graficas.php, este extrae las lecturas registradas por cada nodo y el tiempo del registro, estos datos son tratados y enviados después al script ajaxGrafica.js que utiliza *Highcharts*, una API de gráficos gratis y sencilla, para el desarrollo de la gráfica.

La plantilla utilizada (*Basic line*) para el despliegue de información se muestra en la figura 13, esta corresponde a la gráfica visible en la página “gráfica” del módulo de visualización.

El script ajaxGrafica.js utiliza, al igual que el script ajaxNodos.js, un objeto XMLHttpRequest que permite que solamente el contenedor de la gráfica se actualice cada cierto tiempo, en esta se encuentra el código para trazar la gráfica en la página “grafica” del módulo de visualización con los datos correspondientes.

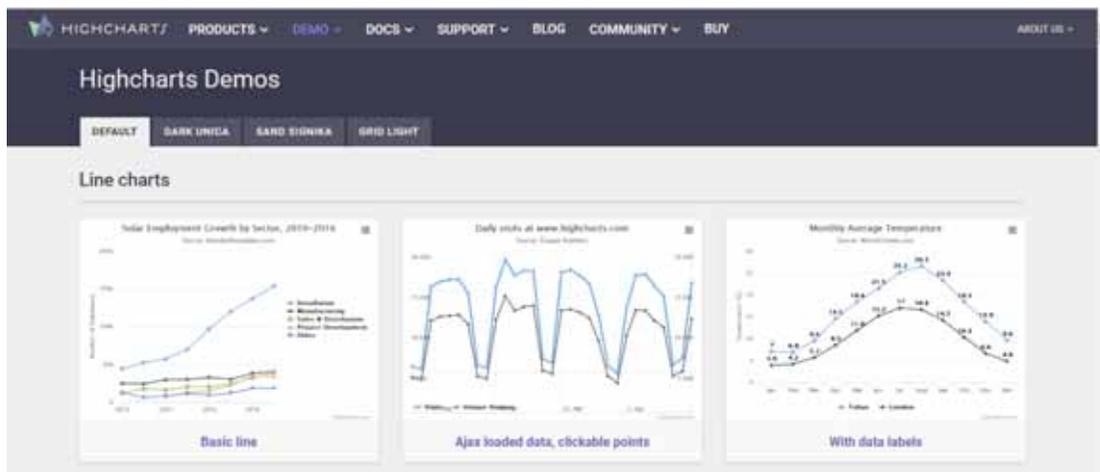


Figura 13. Página de *Highchar*

El código utilizado para el desarrollo de la gráfica es totalmente gratuito, este se adecuó para presentar las lecturas con respecto al tiempo de cada nodo de la red.

En este módulo son extraídos todos los datos de la red, estos son: identificador cada nodo, posición del nodo en ese instante(x,y), tiempo en el cual el nodo realizo un movimiento, y la lectura que este registro en ese tiempo, estos datos son extraídos en el archivo consulta.php, donde se les da tratamiento y posteriormente son enviados a ajaxTabla.js, donde se realizan las operaciones necesarias para su futura presentación, estas fueron desarrolladas utilizando Ajax y JQuery[10]. Después los datos son presentados en la página “información” del módulo de visualización, donde son mostrados en una tabla en forma de historial.

Red inalámbrica de sensores móviles

No se cuenta con una red de sensores móviles física por lo cual se simuló su funcionamiento utilizando *BonnMotion* [11], una herramienta que permite generar escenarios de movilidad aleatorios. La figura 14 muestra la página oficial de *BonnMotion* de donde se descargó la versión más reciente de este.

BonnMotion al ser software libre solo se ha adecuado para utilizarse en la plataforma Ubuntu, su manual, el cual se puede encontrar en la página oficial de *BonnMotion*, da una breve descripción de cómo debe ser utilizado y la forma de instalarse en Unix y Windows.

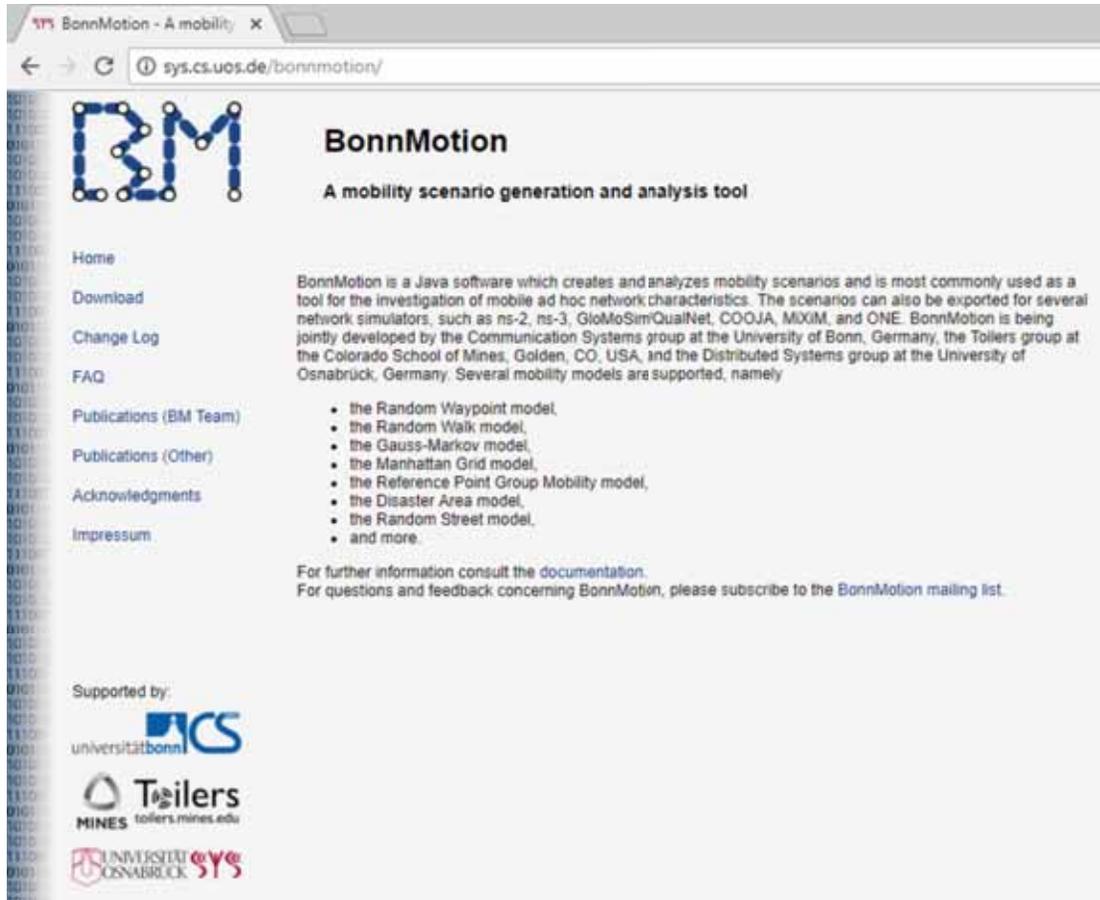


Figura 14. Página de descarga de *BonnMotion*[12].

```
carolina@carolina-VirtualBox:~/Descargas/bonnmotion-3.0.0/bin$ ./bm -f 10Nodo
s1Hora RandomWaypoint -d 3600 -n 10 -l 3600 -x 720 -y 720 -o 3
BonnMotion 3.0.0

OS: Linux 4.10.0-28-generic
Java: Oracle Corporation 1.8.0_161

Starting RandomWaypoint ...
Next RNG-Seed =7813470800138150432 | #Randoms = 667
RandomWaypoint done.
Runtime: 0 sec
```

Figura 15. Trazo de movimiento, simulación de la red de sensores móviles.

En la figura 15 podemos ver la línea de comando utilizada para la generación de la traza de movimiento, que simulará el comportamiento de una red de sensores móviles, donde:

10Nodos el nombre de la traza

RandomWaypoint el tipo de traza que va a generar, en este caso es aleatorio

-d duración

-n número de nodos

-i especifica cuantos segundos adicionales se deben omitir al principio del escenario

-x eje x

-y eje y

-o dimensión en la que se trabajar, en este caso (x, y)

Los datos generados por esta traza de movimiento son: identificador de nodo, posición en el eje x, posición en el eje y, y tiempo en el que se detecta un movimiento.

La traza de movimiento es generada en formato de texto, por tanto, es difícil de comprender. Debido a esto se exportó a formato csv con la ayuda de *BonnMotion* el cual, facilita la posibilidad de exportar la traza a este formato con una simple línea de comando, mostrada en la figura 16.

```
carolina@carolina-VirtualBox:~/Descargas/bonnmotion-3.0.0/bin$ ./bm CSVFile -
f 10Nodos1Hora
BonnMotion 3.0.0

OS: Linux 4.10.0-28-generic
Java: Oracle Corporation 1.8.0_161

Starting CSVFile ...
randomSeed (String):1528507642599
randomSeed (Long):1528507642599
CSVFile done.
Runtime: 1 sec
```

Figura 16. Línea de comando para cambiar el formato de salida de la traza.

Donde:

CSVFile formato al que se quiere convertir

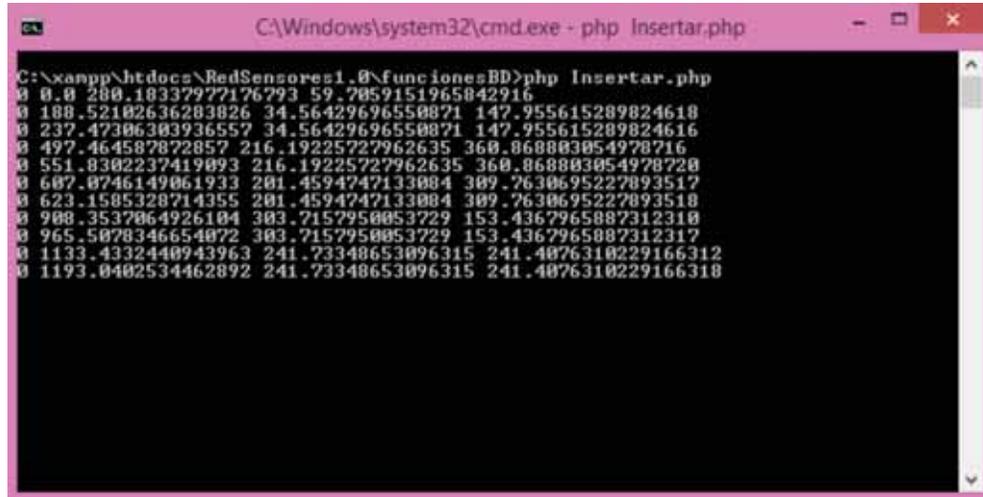
10Nodos1Hora nombre de la traza que se quiere convertir

La traza ahora en el formato cvs sigue siendo incompatible con el formato necesario para su almacenamiento por lo cual se desarrolló un pequeño programa llamado *CreadorCVS.php*, que se encarga de darle el formato adecuado, además agrega las lecturas, que son números aleatorios que simulan temperatura. Este programa, llamado *CreadorCSV.php*, se encuentra en la carpeta *funcionesBD* en el código adjunto.

Por último, los datos en el formato adecuado son almacenados en la base de datos con un programa desarrollado en PHP que los inserta con un pequeño retraso de 10 segundos para lograr realizar la simulación de una red de sensores en constante movimiento.

Resultados

Se simuló la inserción de los datos desde consola como se muestra en la figura 17, con lo cual se pudo corroborar el funcionamiento correcto de las páginas desarrolladas.



```
C:\Windows\system32\cmd.exe - php Insertar.php
C:\xampp\htdocs\RedSenceres1.0\funcionesBD>php Insertar.php
0 0.0 280.18337977176793 59.7859151965842916
0 188.52102636283826 34.56429696558871 147.955615289824618
0 237.47386383936557 34.56429696558871 147.955615289824616
0 497.464587872857 216.19225727962635 368.868883854978716
0 551.8302237419893 216.19225727962635 368.868883854978728
0 687.8746149861933 201.4594747133884 389.7638695227893517
0 623.1585328714355 201.4594747133884 389.7638695227893518
0 988.3537864926184 383.7157958853729 153.4367965887312318
0 965.5878346654872 383.7157958853729 153.4367965887312317
0 1133.4332448943963 241.73348653896315 241.4876318229166312
0 1193.8482534462892 241.73348653896315 241.4876318229166318
```

Figura 17. Inserción de datos desde consola.



Figura 18. Mapa de la red de sensores móviles solo con un nodo.

En la figura 18 se muestra el funcionamiento de la página al insertar los datos del primer nodo, en esta se observa el nodo, su nombre (id) y su cobertura, al ser solamente un nodo no tiene relación con ningún otro por lo cual no tienen conectividad.

Los nodos se actualizan cada dos segundos esto quiere decir, que se realiza una nueva consulta a la base de datos, comparando la información obtenida con la anterior, si esta es diferente se realiza una actualización en el mapa de la red.

La figura 19 muestra cinco nodos y la conectividad entre ellos, en este caso se puede ver como los nodos están conectados entre sí.

Mientras que la figura 20 se observan la topología de la red con diez nodos.

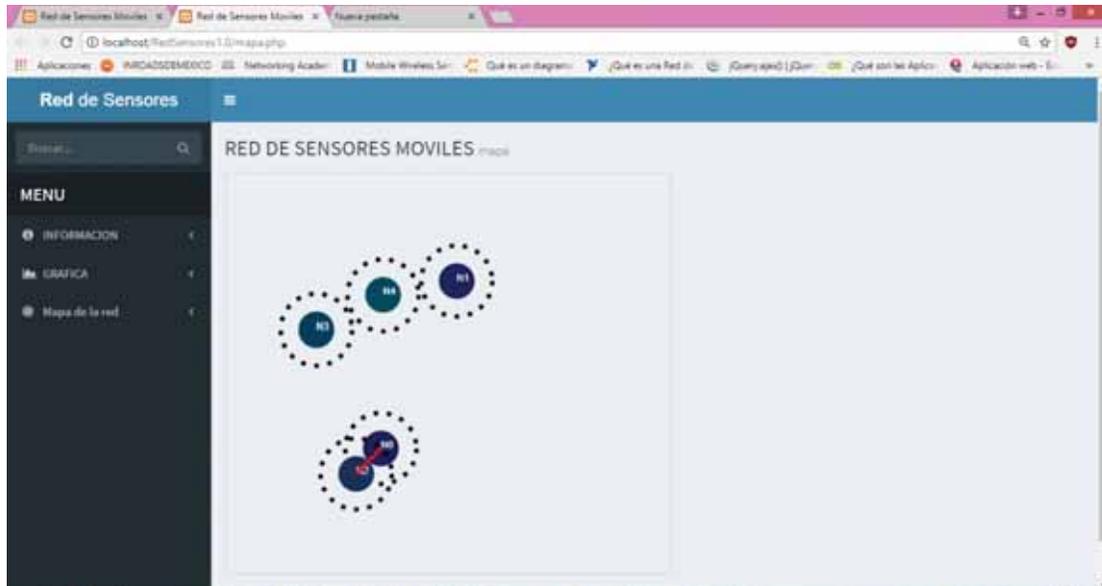


Figura 19. Mapa de la red de sensores móviles con cinco nodos.

Por otra parte, podemos observar la página que muestra la gráfica de la lectura de los nodos contra el tiempo en la figura 21, en esta solo se están recolectando las lecturas del primer nodo. En la figura 22 se muestra la página la gráfica con las lecturas de los diez nodos, cada nodo es identificado con un color diferente por lo cual es fácil identificar los nodos, de esta forma es muy fácil identificar los cambios en las lecturas de cada nodo.

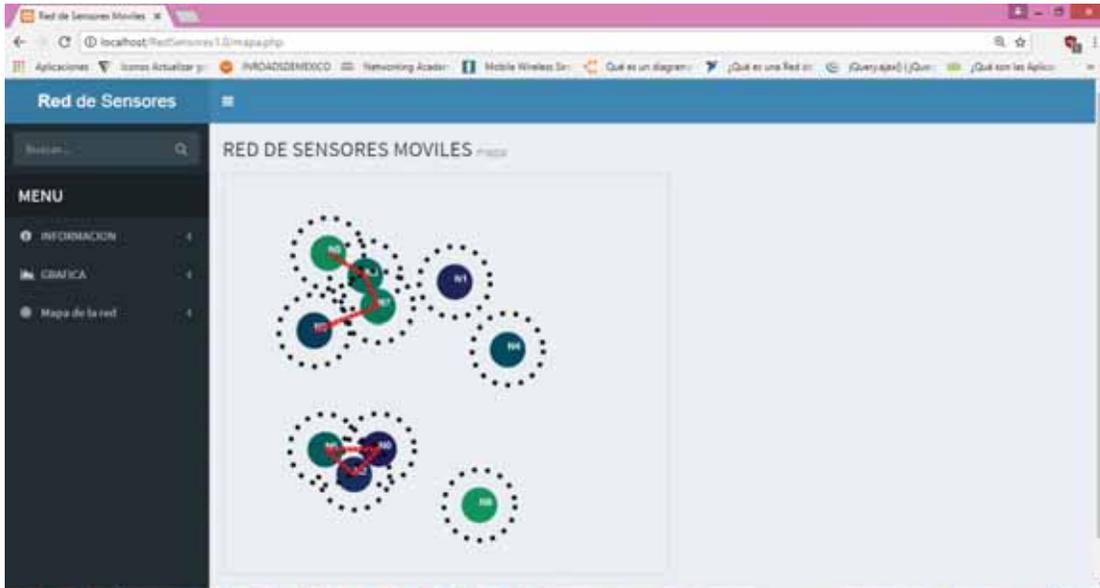


Figura 20. Mapa de la red de sensores móviles con diez nodos.

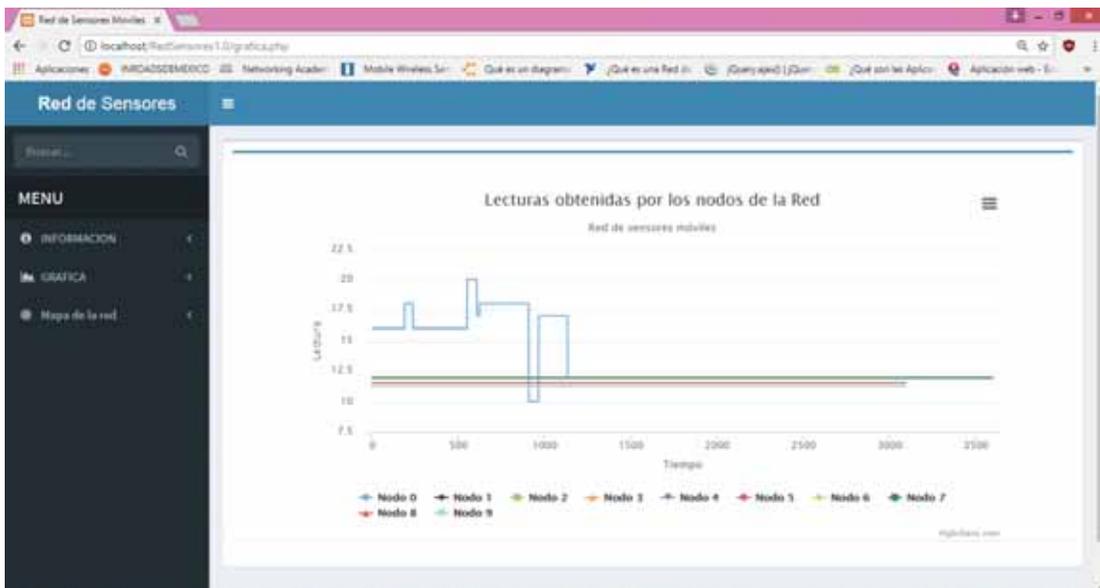


Figura 21. Gráfica de las lecturas de los nodos contra el tiempo.

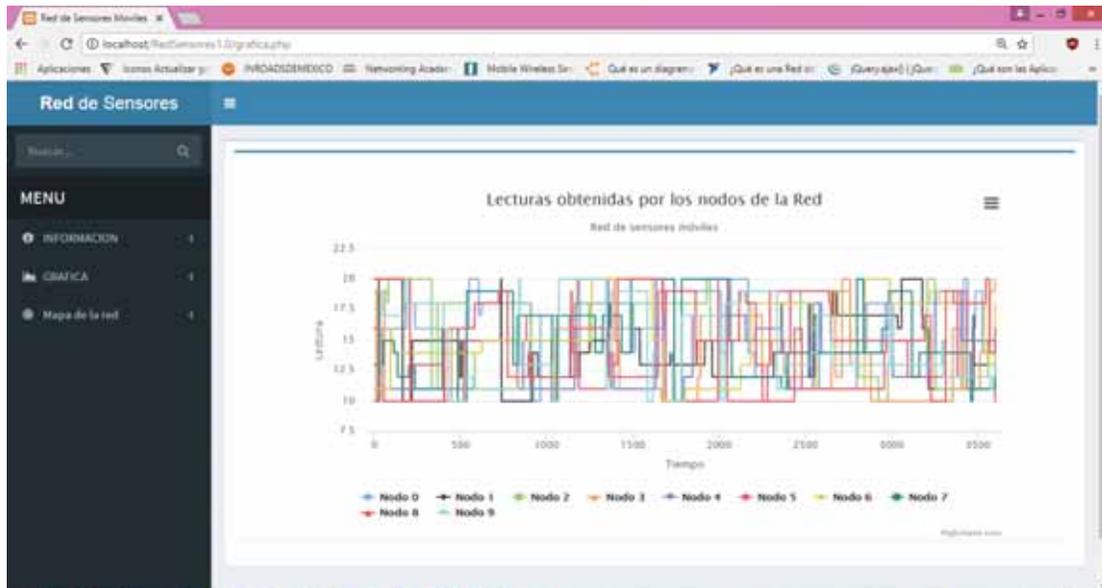


Figura 22. Gráfica de las lecturas de los nodos contra el tiempo, diez nodos.

Por último, se presenta en la figura 23 la tabla que contiene la información de la red vacía. En la figura 24 se observará cómo se van insertando los datos a la tabla, que los presenta en forma de historial. De esta forma el último registro mostrado corresponde a la última información recolectada, en la figura 25 se puede observar los últimos registros que fueron insertados a la Base de Datos.

Lectura	Tiempo	Nodo 0	Nodo 1	...

Figura 23. Tabla de información de la red vacía.

Análisis y discusión de resultados

La aplicación desarrollada tiene la ventaja de poderse adaptar a diferentes redes de sensores móviles actuales, y las que puedan ir surgiendo en el desarrollo e implementación de esta tecnología.

El método de visualización de la topología de la red al que se llegó con el diseño e implementación de esta aplicación resulta ser un método de consulta del estado de la red en tiempo actual, simple pero preciso.

La visualización de la red de sensores móviles es simple ya que muestra la topología cambiante de la red de forma didáctica y precisa, además proporciona información importante para su rápida comprensión.

Debido a que no se contaba con una red de sensores móviles física se realizó una simulación de esta, con el objetivo de obtener información útil para el diseño e implementación de la aplicación web desarrollada. Para poder implementar la aplicación web a una red de sensores móviles física se deben agregar un módulo que comunique a la red con la base de datos.

Conclusiones

La aplicación desarrollada cumple con su objetivo al mostrar los datos de una red de sensores móviles, su topología, las lecturas obtenidas por los sensores en forma de gráfica y la información completa desplegada en una tabla.

Se planeó desarrollar tres módulos para la creación de la aplicación web, pero en el proceso de desarrollo se concluyó que no era necesario la implementación del módulo para la administración de los datos de la red, puesto que solo se realizaron dos operaciones sobre la base de datos, lectura y escritura. Mientras que el módulo podría ser capaz de crear, actualizar, leer y borrar en la base de datos (CRUD por sus siglas en inglés).

Referencias

- [1] "What Is a Wireless Sensor Network? - National Instruments", Ni.com, 2009. [En línea]. Disponible en: <http://www.ni.com/white-paper/7142/es/>. [Consulta: 21-Oct-2017].
- [2] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. & Cayirci, E, Wireless sensor Networks: A survey, Computer Networks, Vol. 38, 393-422, 2002.
- [3] Katiyar, V., Kumar, P. & Chand N. An Intelligent Transportation Systems Architecture using Wireless Sensor Networks, International Journal of Computer Applications, Vol. 14, No. 2, 22-26, 2011.
- [4] Fauzi, M. Wireless Sensor Network Applications: A Study in Environment Monitoring System, International Symposium on Robotics and Intelligent Sensors, No. 41, 1204-1210, 2012.
- [5] A. Soares, "Wireless Sensor Networks Monitoring Tool", reporte final, FCTUC, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, Portugal, 2013.
- [6] C. Buschmann, D. Pfisterer, S. Fischer, S. Fekete, A.Kröller, "SpyGlass: a wireless sensor network visualizer," Acm Sigbed Review, vol. 2, pp. 1-6, 2005.
- [7] V. Ramasamy, "Mobile Wireless Sensor Networks: An Overview", Wireless Sensor Networks - Insights and Innovations, 2017.
- [8] *Adminlte.io. (2018). Free Bootstrap Admin Template | AdminLTE.IO.* [En línea] Disponible en: <https://adminlte.io/> [Consulta: 1 Jun. 2018].
- [9] W3schools.com. (2018). HTML5 Canvas. [En línea] Disponible en: https://www.w3schools.com/html/html5_canvas.asp [Consulta: 21-Mayo-2018]
- [10] js.foundation, J. (2018). jQuery. [En línea] Jquery.com. Disponible en: <https://jquery.com/> [Consulta 5-Jul-2018].
- [11] Aschenbruck, N., Ernst, R., Gerhards-Padilla, E. & Schwamborn, M., BonnMotion: a mobility scenario generation and analysis tool, Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, 2010
- [12] BonnMotion - A mobility scenario generation and analysis tool", Sys.cs.uos.de, 2017. [En línea]. Disponible en: <http://sys.cs.uos.de/bonnmotion/>. [Consulta: 30- Oct-2017].