

**APLICACIÓN WEB PARA LA CAPTURA, MONITOREO Y ANÁLISIS DE DATOS
TRANSMITIDOS A TRAVÉS DE REDES DE SENSORES INALÁMBRICOS
UTILIZANDO TECNOLOGÍA ZIGBEE**

**BIVIANA BALLESTEROS CORZO
SILVIA PAOLA CHAPARRO DURÁN**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2008



**APLICACIÓN WEB PARA LA CAPTURA, MONITOREO Y ANÁLISIS DE DATOS
TRANSMITIDOS A TRAVÉS DE REDES DE SENSORES INALÁMBRICOS
UTILIZANDO TECNOLOGÍA ZIGBEE**

**BIVIANA BALLESTEROS CORZO
SILVIA PAOLA CHAPARRO DURÁN**

Proyecto de Grado para optar al título de Ingeniera de Sistemas

Director

**Ingeniero José De Jesús Rugeles Uribe
Profesor Cátedra de la Escuela de Ingeniería Eléctrica, Electrónica y
Telecomunicaciones**

Codirector

**INGENIERO HENRY ARGUELLO
Profesor de Planta de la Escuela de Ingeniería de Sistemas e Informática**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2008

AGRADECIMIENTOS

Las autoras expresan su profundo agradecimiento a Dios por concedernos la fortuna de culminar nuestros estudios universitarios; a nuestros padres por todo el apoyo, paciencia y amor brindado durante esta etapa de nuestras vidas; a nuestros compañeros y amigos que incondicionalmente nos acompañaron durante el desarrollo de este proyecto de grado y de toda la carrera en general, brindándonos una palabra de aliento en el momento necesario; a Sergio y Leonardo por el ánimo, los consejos, la amistad y el amor que nos han ofrecido; a Luis Alejandro Osorio, por estar siempre dispuesto a colaborarnos y alimentar todo el proceso para culminar este proyecto de grado; y a todas las personas que de alguna forma estuvieron presentes en nuestra vida universitaria, alimentándola de buenos recuerdos....Gracias

TABLA DE CONTENIDO

1. GENERALIDADES	11
1.1 INTRODUCCIÓN.....	16
1.2 OBJETIVOS	20
1.2.1 OBJETIVO GENERAL	20
1.2.2 OBJETIVOS ESPECÍFICOS	21
2. MARCO TEÓRICO Y ESTADO DEL ARTE	22
2.1 TECNOLOGÍAS DE DESARROLLO WEB.....	22
2.1.1 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	22
2.1.2 APLICACIONES WEB.....	24
2.1.2.1 <i>Lenguaje del Lado del Cliente</i>	25
2.1.2.2 <i>Lenguaje del Lado del Servidor</i>	26
2.2 BASES DE DATOS.....	27
2.2.1 CLASIFICACIÓN DE LAS BASES DE DATOS.....	28
2.3 PROTOCOLOS DE COMUNICACIÓN	29
2.3.1 ZIGBEE (IEEE 802.15.4).....	30
2.3.2 BLUETOOTH (IEEE 802.15.1)	32
2.4 REDES DE SENSORES INALÁMBRICOS	33
2.4.1 TOPOLOGÍAS DE RED	34
2.4.2 APLICACIONES DE REDES DE SENSORES INALÁMBRICOS [11].....	35
2.4.2.1 <i>Aplicaciones en la Agricultura</i>	35
2.4.2.2 <i>Aplicaciones sociales y sanitarias</i>	36
2.4.2.3 <i>Aplicaciones en la naturaleza</i>	37
2.4.2.4 <i>Aplicaciones civiles y militares</i>	38
2.4.3 COMPUTACIÓN UBICUA	39
2.4.3.1 <i>Proyectos</i>	40
3. PROTOCOLOS DE COMUNICACIÓN	43
3.1 ZIGBEE	43
3.2 REDES ZIGBEE.....	47
3.2.1 APLICACIONES ZIGBEE	49
3.2.2 CONEXIÓN ZIGBEE	49
3.2.3 EVOLUCIÓN ZIGBEE	51
3.3 BLUETOOTH	52

3.3.1 CONEXIÓN ENTRE DISPOSITIVOS BLUETOOTH.....	57
3.4 ZIGBEE VS. BLUETOOTH [9] [32]	58
4. METODOLOGÍA	62
4.1 FASE DE INICIO	63
4.1.1 ANÁLISIS DE REQUISITOS	63
4.1.2 DISEÑO.....	65
4.2 FASE DE ELABORACIÓN	65
4.2.1 ANÁLISIS.....	66
4.2.2 DISEÑO.....	66
4.2.3 IMPLEMENTACIÓN	66
4.3 FASE DE CONSTRUCCIÓN.....	66
4.3.1 ANÁLISIS.....	67
4.3.2 DISEÑO.....	67
4.3.3 IMPLEMENTACIÓN	67
4.3.4 PRUEBAS.....	68
4.4 FASE DE TRANSICIÓN.....	68
5. ANÁLISIS Y DISEÑO DE LA APLICACIÓN WEB	69
5.1 DISEÑO Y DOCUMENTACIÓN DE LA APLICACIÓN WEB	71
5.2 DESCRIPCIÓN DE LA APLICACIÓN WEB POR MÓDULOS	78
5.3 DIAGRAMAS UML	80
5.3.1 ACTORES DEL SISTEMA.....	81
5.3.2 DIAGRAMA DE ACTIVIDADES.....	86
5.3.3 INTERFACES DEL SISTEMA.....	94
6. TINYOS Y NESC.....	110
6.1 TINYOS.....	110
6.1.1 CARACTERÍSTICAS TINYOS	111
6.1.2 ARQUITECTURA DE TINYOS.....	111
6.1.3 MODELO DE COMPONENTES	113
6.1.4 REQUERIMIENTOS BÁSICOS DE INSTALACIÓN	115
6.1.5 PAQUETES INSTALADOS CON TINYOS	115
6.2 NESC	116
6.2.1 ESTRUCTURA DE UNA APLICACIÓN EN NESC.....	117
6.2.2 OSCILLOSCOPE MODIFICADO.....	119
7. DESARROLLO DE ALGORITMOS Y PRUEBAS	120

7.1 DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL GENERADOR DE DATOS	121
7.2 PRUEBAS CON DATOS REALES	126
7.3 CÓDIGO JAVA COMUNICACIÓN NODOS - SERVIDOR	129
CONCLUSIONES	139
OBSERVACIONES	141
BIBLIOGRAFÍA	142

LISTA DE FIGURAS

Figura 1. Conexión con una base de datos de una red de sensores.....	20
Figura 2. Redes de sensores inalámbricos.....	19
Figura 3. Topologías comunes de red.....	20
Figura 4. Sensores para cuidado médico desarrollados en el proyecto CodeBlue.....	22
Figura 5. Sensor inalámbrico de presión para pozos de petróleo.....	23
Figura 6. Modelo de capas para la especificación ZigBee.....	29
Figura 7. Ilustración de una red ZigBee	31
Figura 8. ZigBee vs Bluetooth.....	42
Figura 9. Tabla comparativa entre Bluetooth y ZigBee	43
Figura 10. Modelo general del sistema.....	52
Figura 11. Modelo del Sistema implementado para el proyecto de grado.....	53
Figura 12. Tablas de la Base de Datos.....	55
Figura 13. Campos de la tabla info_nodos.....	55
Figura 14. Campos de la tabla mortandad.....	56
Figura 15. Campos de la tabla nodos.....	56
Figura 16. Campos de la tabla usuarios.....	56
Figura 17. Campos de la tabla variables.....	56
Figura 18. Modelo general de Base de Datos.....	57
Figura 19. Caso de uso Usuario.....	62
Figura 20. Caso de uso Administrador.....	63
Figura 21. Caso de uso Visitante.....	64
Figura 22. Caso de uso Nodos Inalámbricos.....	64
Figura 23. Caso de uso Sistema.....	65
Figura 24. Diagrama de Actividades Administrar Topología.....	66
Figura 25. Diagrama de actividades Generar Graficas.....	67
Figura 26. Diagrama de actividades Administras usuarios.....	68
Figura 27. Diagrama de actividades Registrar Mortandad.....	69
Figura 28. Diagrama de actividades Consultar Mortandad.....	70
Figura 29. Diagrama de actividades.....	71

Figura 30. Diagrama de Actividades Hacer análisis estadístico.....	72
Figura 31. Página principal REDSI.....	73
Figura 32. Página de Ingreso al sistema REDSI.....	74
Figura 33. Página de ingreso de nuevo usuario (registro).....	74
Figura 34. Menú principal REDSI.....	75
Figura 35. Menú topología y gráficas.....	76
Figura 36. Menú desplegable de cada nodo en la topología.....	77
Figura 37. Menú Crear nueva topología.....	78
Figura 38. Vista nueva topología.....	79
Figura 39. Modificar Topología.....	80
Figura 40. Página Administrar usuarios.....	81
Figura 41. Registro de mortandad.....	82
Figura 42. Información de los nodos.....	83
Figura 43. Consultas y Reportes.....	84
Figura 44. Estadísticas.....	85
Figura 45. Cerrar sesión.....	85
Figura 46. Modelo de Componentes de TinyOS y su interacción.....	89
Figura 47. Estructura de una aplicación en NesC	92
Figura 48. Estructura aplicación Blink.....	94
Figura 49. Modelo de prueba con el Generador de Datos.....	96
Figura 50. Diagrama de flujo Generador de Datos.....	98
Figura 51. Grafica con datos generados aleatoriamente de Humedad y Temperatura	100
Figura 52. Diagrama de pruebas con datos reales.....	101
Figura 53. Gráfica datos reales de temperatura y humedad.....	102
Figura 54. Código Java de Conexión Gateway-BD.....	103
Figura 55. Diagrama de flujo de captura de datos.....	104
Figura 56. Instalación TOSBase en el nodo receptor.....	105
Figura 57. Instalación Oscilloscope en el nodo transmisor.....	106
Figura 58. Resultados ejecución CapturaDatos.java.....	107
Figura 59. Modelo de prueba de envío y recepción de los datos.....	108



Figura 60. Modelo de prueba de conexión USB del nodo transmisor.....	109
Figura 61. Modelo General de Pruebas utilizando el Gateway.....	109
Figura 62. Nodo Modificado.....	110

RESUMEN

TITULO: APLICACIÓN WEB PARA LA CAPTURA, MONITOREO Y ANÁLISIS DE DATOS TRANSMITIDOS A TRAVÉS DE REDES DE SENSORES INALÁMBRICOS UTILIZANDO TECNOLOGÍA ZIGBEE.*

AUTORAS: BALLESTEROS CORZO BIVIANA**
CHAPARRO DURÁN SILVIA PAOLA**

PALABRAS CLAVES: Redsi, Administración, Procesos, Control, Web, Redes, Sensores, Inalámbrico, Zigbee

La industria avícola ha empezado a vislumbrar un futuro promisorio para su negocio con la implementación de aplicaciones que incluyan redes de sensores inalámbricos y herramientas capaces de monitorear los sistemas de producción y con base en la información obtenida disminuir costos y optimizar sus procesos.

Estudios realizados por la industria muestran que la mortalidad de las aves por estrés calórico llega a ser del orden del 6.5% generando millonarias pérdidas económicas por la muerte de las aves, pérdidas que pueden reducirse significativamente controlando las variables físicas, como humedad, temperatura, radiación solar e intensidad luminosa, que influyen en este fenómeno.

Es así como nace este proyecto de investigación y se materializa en REDSI, un sistema diseñado para capturar, almacenar y analizar datos obtenidos de una red de sensores inalámbricos implementada en un galpón, desarrollado utilizando herramientas de libre distribución como MySQL, php, ajax y java. La aplicación cuenta con una interfaz web que permite al usuario acceder a la información capturada desde cualquier lugar a través de Internet, ver lecturas en tiempo real y detectar fallas en los sensores mediante una interfaz gráfica, hacer un análisis gráfico y estadístico de la información almacenada y tener un historial de ella. Todas estas funcionalidades con el fin de ejercer un control centralizado y sistematizado del sistema y sus procesos.

A futuro este tipo de aplicaciones no solo proporcionará la información indispensable para que un administrador tome acciones al respecto sino que permitirá conectarse a un sistema de control automático que a partir de esta información y el análisis que se haga tome decisiones para estabilizar el ambiente y optimizar el sistema de producción.

* Tesis de grado en modalidad de investigación

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Ing. José de Jesús Rugeles Uribe. Codirector Ing. Henry Arguello Fuentes.

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Ing. José de Jesús Rugeles Uribe. Codirector Ing. Henry Arguello Fuentes.

SUMMARY

TITLE: APPLICATION WEB FOR THE CAPTURE, MONITOR AND ANALYZE OF DATA TRANSMITTED THROUGH NET OF WIRELESS SENSORS USING ZIGBEE TECHNOLOGY.*

AUTHORS: BALLESTEROS CORZO BIVIANA**
CHAPARRO DURÁN SILVIA PAOLA**

KEY WORDS: Redsi, Administration, Process, Control, Web, network, Sensors, Wireless, Zigbee,

The poultry industry begin to show a promising future for their business with the implementation of applications that have nets of wireless sensors and able tools of manage the production systems and with the known information to reduce costs and to improve its process. Researches realized by the industry shows that the mortality of the fowler because heat stress is around 6.5% producing millionaires economics losses which can be reduced with the control of physic variables such as humidity, temperature, solar radiation and luminous intensity that are in this phenomenon.

In that form born this research project and is materialized in REDSI, a designed system for obtaining, to store and to analyze obtained data of a net of wireless sensors using in a shed, developed using tools of free distribution such as MYSQL, PHP, AJAX Y JAVA.

The application has an interference WEB which allows to user agree to the capture information from any place through Internet, research the lectures and to know the fails in the sensors through an interval graphic, making and statistic analyze of the hoard information and to have an history of it . All those applications searching a centralized and systematic control of the system and its process.

In a future this type of applications not only will bring indispensable information to a manager which will take decisions about a particular situation; also it will allow to connect to an automatic system control which should take decisions about the analyze that on make and to stabilize the ambience and to optimize the production system.

* Tesis de grado en modalidad de investigación

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Ing. José de Jesús Rugeles Uribe. Codirector Ing. Henry Arguello Fuentes

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Ing. José de Jesús Rugeles Uribe. Codirector Henry Arguello Fuentes

1. GENERALIDADES

1.1 INTRODUCCIÓN

Con el auge de las tecnologías inalámbricas, se viene generando una revolución en las comunicaciones que ha contribuido al desarrollo de las redes de sensores inalámbricos. Se dice que este tipo de redes serán para ésta década, lo que fueron el microprocesador para los 80 y el Internet para los 90.

Las redes de sensores están emergiendo dentro del campo de las comunicaciones inalámbricas y son muchas las aplicaciones desarrolladas en torno al tema como parte de ambiciosos proyectos investigativos [1],[2],[3],[4] que buscan mejorar las condiciones actuales de la industria, convirtiéndola en uno de los sectores más beneficiados con esta revolución tecnológica.

Actualmente se cuenta con los mecanismos y la tecnología suficiente para brindar soluciones a las necesidades básicas de una organización, lo que permite que las empresas mejoren sus procesos y se hagan más productivas y por consiguiente más competitivas en el mercado.

El propósito de una red de sensores inalámbricos, es obtener información de una o más variables físicas como temperatura, humedad, radiación solar, entre otras, mediante dispositivos electrónicos que integran un elemento transductor, un procesador y un radio con el fin de centralizar la información recopilada para su posterior análisis y presentación de resultados.

Esta información que se obtiene de la red, requiere ser transportada, almacenada,

procesada y analizada. Para esto, es necesario que además de diseñar programas para los dispositivos inalámbricos, se diseñen interfaces de interconexión con Internet y sistemas robustos para el almacenamiento de la información.

En la Universidad Industrial de Santander el grupo de Investigación en Conectividad y Procesado de Señal – CPS – viene trabajando en el desarrollo de una plataforma experimental de red de sensores inalámbricos para evaluar la incidencia de variables físicas en la mortalidad de aves de corral en granjas avícolas¹, empleando tecnología ZigBee.

Esta investigación se da con el propósito de impulsar el estudio de este tema al interior de la UIS y determinar posibles aplicaciones de esta tecnología tanto en la industria avícola y en otros campos pues las organizaciones han empezado a vislumbrar un futuro promisorio para sus negocios con la implementación de aplicaciones que incluyan redes de sensores inalámbricos y herramientas capaces de monitorear los sistemas de producción y con base en la información obtenida disminuir costos y optimizar sus procesos. La industria avícola en Santander, pese a ser uno de los sectores más representativos del país, no cuenta con la tecnología necesaria para administrar adecuadamente sus recursos y evitar las pérdidas económicas generadas por realizar sus procesos artesanalmente. El panorama de las granjas avícolas se ve amenazado por el alto índice de mortalidad debido en gran parte al estrés calórico en los animales y a la tardía respuesta del operario que solo se basa en su criterio para revisar las condiciones del galpón y tomar las medidas correctivas necesarias.

Estudios realizados por la industria muestran que la mortalidad de las aves por estrés calórico llega a ser del orden del 6.5% por ciclo, generando millonarias pérdidas económicas por la muerte de las aves. Es por esto que se desarrolló este trabajo de grado interdisciplinario, como soporte a la red experimental de sensores inalámbricos

¹ REIMAGAV, Directora: Ana Ramírez

creada en el grupo de investigación CPS de la escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones y que se implementará en un galpón para monitorear la temperatura ambiente, la humedad relativa, la intensidad luminosa y la radiación solar y la incidencia de estas variables en la mortalidad de las aves de corral durante las etapas de crianza, levante y engorde.

Adicionalmente, se identificaron las posibilidades de expansión de estos sistemas y su aplicación en la industria planteando conclusiones que sirvan de base a nuevos trabajos de grado y permitan afrontar el problema en magnitudes mayores.

Como aporte al proyecto REIMAGAV del grupo CPS se diseñó una herramienta web que permite al usuario acceder a la información proveniente de la plataforma preliminar de sensores inalámbricos implementada por el grupo y así ofrecer desde la ingeniería de sistemas un marco de referencia y apoyo en el desarrollo de estas nuevas tecnologías de información. Dicha aplicación bajo plataforma web permite la captura de los datos leídos por los sensores, el almacenamiento de los mismos y el posterior análisis por parte del usuario de los resultados obtenidos en dichas mediciones, además de la administración y el monitoreo de la red misma, cuenta con una interfaz de fácil acceso y navegación, una base de datos que satisfaga las necesidades de la información que se maneja, y una serie de servicios complementarios como representación gráfica de los datos, señales de detección de fallas en los sensores y consulta de datos, entre otros.

También se realizó durante éste trabajo de grado, un estudio sobre dos de los protocolos más usados en la comunicación inalámbrica de datos, Bluetooth² y Zigbee³, para identificar las ventajas que llevaron al grupo CPS a seleccionar por los dispositivos Zigbee como base para el desarrollo de esta fase preliminar de su investigación.

² <http://www.bluetooth.com>

³ <http://www.zigbee.org/>

Adicional a esto y como un aporte complementario al trabajo de grado, se desarrolló un algoritmo en NESC⁴ y se documentó sobre este lenguaje de bajo nivel sobre el cual se programa el hardware utilizado durante el proyecto.

A continuación se mencionará el contenido del presente documento, que busca cubrir los objetivos planteados al inicio de éste proyecto:

En el capítulo 1 el lector encontrará la introducción al proyecto, evidenciando la necesidad y la problemática desde la que se partió para la realización del trabajo de grado y los objetivos generales y específicos del mismo.

En el capítulo 2 se encuentra un marco teórico sobre los principales temas que se estudiaron para adquirir la base del conocimiento relacionado con metodologías de desarrollo web, aplicaciones web y bases de datos, estándares de comunicación (zigbee y bluetooth), sensores inalámbricos y las aplicaciones de los mismos haciendo un análisis del estado del arte de estas tecnologías en los últimos años.

El capítulo 3 describe de manera más amplia los conceptos sobre los protocolos de comunicación más conocidos y profundiza sobre los estándares zigbee y bluetooth, haciendo énfasis entre sus ventajas y desventajas.

En el capítulo 4 se encuentra la metodología utilizada para el desarrollo del software, describiendo las fases de inicio, elaboración y construcción que llevan al producto final ejecutable.

⁴ <http://nesc.sourceforge.net/>

El capítulo 5 abarca un reconocimiento del problema y un planteamiento de la solución, se describen los servidores web, la interfaz de usuario y se realiza la documentación del diseño del sistema utilizando el lenguaje de modelado unificado (diagramas de casos de uso, de actividades y se muestran las interfaces del sistema final). Además se especifica el hardware y software empleados en el desarrollo del trabajo de grado (nodos inalámbricos⁵, gateway, php, MySQL, ajax).

El capítulo 6 contiene la documentación del estudio realizado al sistema operativo TinyOS⁶ y su lenguaje asociado, NesC, describiendo los componentes básicos del lenguaje mediante un ejemplo guía.

En el capítulo 7 se documentan las pruebas llevadas a cabo durante el proceso de diseño; inicialmente las pruebas se hicieron con datos obtenidos de un generador de datos aleatorio para cada variable estudiada; luego se realizaron pruebas con datos medidos directamente en el galpón proporcionados por el grupo CPS y finalmente se realizaron las pruebas con los motes conectados y transmitiendo usando los programas TOSBase y Oscilloscope contenidos en el TinyOS. Al finalizar el libro se encuentran las conclusiones y observaciones obtenidas con la realización del trabajo de grado.

1.2 OBJETIVOS

1.2.1 Objetivo General

Construir una herramienta Web que permita la captura, monitoreo y análisis de los datos recopilados provenientes de una red de sensores inalámbricos, que brinde la

⁵ Dispositivo electrónico utilizado para tomar lecturas de algunas variables físicas.

⁶ www.tinyos.net

posibilidad de estudiar el comportamiento de las variables temperatura, humedad relativa, intensidad luminosa y radiación solar en el interior de una granja avícola.

1.2.2 Objetivos Específicos

- ✓ Analizar el protocolo de comunicación IEEE 802.15.4 y establecer ventajas comparativas con IEEE 802.15.1.
- ✓ Desarrollar e implementar una aplicación Web que contenga los módulos de consulta, reportes, análisis estadístico, alarmas y visualización de los parámetros físicos Temperatura, Humedad, Intensidad Luminosa y Radiación Solar; y que permita analizar la incidencia de éstas variables en el proceso de cría, levante y engorde de aves de corral.
- ✓ Diseñar una Base de Datos escalable que satisfaga las necesidades de almacenamiento remoto de un conjunto de nodos IEEE 802.15.4, teniendo en cuenta el estado del arte de las investigaciones sobre redes de sensores inalámbricos.
- ✓ Implementar y desarrollar algoritmos en el lenguaje de programación NESC para el sistema operativo TinyOS, que permita la comunicación de un conjunto de sensores inalámbricos con un servidor Web.

2. MARCO TEÓRICO Y ESTADO DEL ARTE

2.1 TECNOLOGÍAS DE DESARROLLO WEB

2.1.1 Metodologías de desarrollo de Software [5],[6],[7],[8]

El proceso que se sigue para desarrollar software tiene un nivel alto de complejidad, y por esta razón se cuenta hoy en día con innumerables filosofías y propuestas metodológicas centradas en distintas dimensiones del proceso de desarrollo. Las propuestas mas tradicionales se basan en el control del proceso, las actividades involucradas, los productos que se desean obtener y las herramientas y notaciones que se usarán.

Un acercamiento mas reciente a la producción de software se centra en otras perspectivas, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

Este nuevo enfoque está mostrando su efectividad en proyectos que exigen reducir los tiempos de desarrollo pero manteniendo siempre una alta calidad, son estrategias de desarrollo de software basadas en procesos iterativos, orientadas hacia la entrega y de comunicación intensiva ya que el usuario actúa como un integrante más del equipo.

En la metodología de desarrollo rápido, el diseño debe ser lo más sencillo posible: “hacer todo lo posible por hacer lo menos posible”⁷ es el estilo que promueve esta filosofía que a finales de los 90’s irrumpió en las prácticas de la ingeniería del Software (junto con el tema del diseño basado en patrones), y cuya clave está en el velocidad y la simplicidad del desarrollo, promoviendo en su modelo la colaboración entre equipos de programadores y expertos del negocio.

La comunicación cara a cara entre el usuario y los desarrolladores como una manera

⁷ Booch G, Jacobson I, and Rumbaugh J. The Unified Modelling Language for Object-Oriented? Development (version 0.91) Rational Software Corporation

más eficiente de refinar el proceso continuamente; El desarrollo incremental (pequeñas entregas con ciclos rápidos); equipos de desarrollo auto organizados; maneras de estructurar código fuente y equipos de desarrollo con el fin de que los requerimientos más importantes no entren en crisis; desarrollo adaptativo, ya que incorpora cambios con facilidad, cooperativo gracias a la continua comunicación entre el usuario y los desarrolladores, y directo por la facilidad de comprender y aprender el método.

2.1.2 Aplicaciones Web

Con el avance en las comunicaciones y el crecimiento de Internet y la Web, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío para los desarrolladores, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas para ser implementadas en Internet. De ahí nace el concepto de aplicación Web, que no es más que una aplicación⁸ informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet, utilizando un cliente que popularmente suele ser un navegador Web (browser).

Las aplicaciones Web son el futuro para los sistemas compartidos ya que mientras las aplicaciones tradicionales (aplicaciones de escritorio) llegaron a una madurez, las aplicaciones Web empiezan a crecer en tecnología y funcionalidad. Algunas ventajas comparativas de las aplicaciones Web son:

- ✓ Mayor mercado al ser independientes de la plataforma (Windows, Mac, Linux) y correr sobre la red.
- ✓ Facilidad de mantenimiento y reducción de puntos de riesgo al concentrar la lógica y la información del negocio en un único sitio (el servidor Web).
- ✓ Los clientes tienen acceso a la aplicación final y al mismo tiempo son

⁸ O'Reilly Radar, Web 2.0, Principles and Best Practices

probadores del sistema ayudando a su continua alimentación.

- ✓ Toma de decisiones basadas en información obtenida en tiempo real

Afortunadamente, se cuenta con herramientas para desarrollar aplicaciones que suplan las necesidades emergentes de acceso remoto a la información y a los servicios. Han surgido nuevas tecnologías que permiten que el acceso a una base de datos desde Internet o desde Intranet, sea bastante sencillo. El único problema es decidir entre las posibilidades existentes la que mejor se adapte a cada situación.

En esencia, el Internet es un gran sistema Cliente-Servidor donde un cliente hace una solicitud y un servidor le responde, es por eso que las aplicaciones Web funcionan perfectamente bajo este sistema. Inicialmente, se solicita una información a un servidor y éste envía de vuelta un fichero que será interpretado por el navegador (el cliente) que lo formateará para visualizarlo en la máquina cliente y si se produce un cambio en la información del sistema central, inmediatamente es propagada a los receptores de la información, a la parte cliente.

Para desarrollar aplicaciones y dotar a las páginas Web de funcionalidad se puede trabajar tanto en el lado del cliente como en el lado del servidor, a continuación se describen algunas herramientas que actualmente se están utilizando de parte y parte.

2.1.2.1 Lenguaje del Lado del Cliente

HTML: Hipertext Markup Language. Lenguaje que se utiliza para la creación de páginas Web. Consta de un conjunto de elementos, denominados etiquetas o marcas, que se utiliza para incluir textos, dibujos, cabeceras, etc.

JavaScript: Lenguaje interpretado incrustado en las páginas Web JavaScript para permitir ejecutar código en el cliente a través del manejo de objetos y la captura de

eventos producidos en el navegador.

DHTML: Extensiones de HTML que cubre el uso de hojas de estilo en cascada (CSS), uso de *layers* o capas para el posicionamiento y visibilidad de elementos en el navegador, programación con lenguajes de scripts y modelado de objetos de documentos (DOM).

VBScript: Lenguaje interpretado similar a JavaScript proporcionado por Microsoft.

Applets de JAVA: Pequeña aplicación accesible desde Internet a un servidor, que se transporta por la red, se instala automáticamente y se ejecuta como parte de un documento Web.

ActiveX: Tecnología de Microsoft que permite ejecutar un programa que siga este estándar en los equipos de los usuarios al ser vinculado como objeto en una página Web. Los programas ActiveX dependen del sistema operativo Windows, por lo tanto no funcionan fuera de este ambiente.

2.1.2.2 Lenguaje del Lado del Servidor

Un servidor es el programa encargado de resolver las peticiones que los clientes realizan en la red; existen varias tecnologías de servidor, entre las cuales se destacan:

ColdFusion: Es una solución implementada por macromedia, utiliza una programación por medio de tags (como el html), es muy rápido, tiene soporte para las plataformas windows, linux, mac y Solaris, depende solo del soporte dado por su casa matriz adobe, dado que es de carácter privativo, lo cual lo tiene limitado en funcionalidades, y en la difusión para su aprendizaje.

Se descarta ColdFusion para la realización de éste proyecto debido a su carácter privativo, alto costo y la dificultad para integrarlo a las otras tecnologías seleccionadas.

Perl: Este lenguaje es la base de todo el software de servidores. Un punto a favor de perl es la madurez de su desarrollo y sus implementaciones en las plataformas mas conocidas, además tiene apoyo de grupos opensource, pero se considera que su hijo mas cercano, PHP, tiene algunas ventajas sobre el.

PHP: Este software opensource se deriva de perl y su código esta basado en C, este producto, tiene implementaciones en las plataformas mas conocidas, el soporte de millones de usuarios en la Web, con infinidad de librerías funcionales, y miles de empresas que desarrollan y corren sus aplicaciones en el, con una curva de aprendizaje a la medida de la necesidad del usuario, e implementaciones en cuanta tecnología haya tocado el opensource. Php es un servidor muy adaptable y su vida comercial está apoyada en su gran difusión en la web actual.

2.2 BASES DE DATOS

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada brindando al usuario del sistema recursos para realizar diversas operaciones sobre los archivos almacenados, incluidas entre otras las siguientes:

- ✓ Agregar archivos nuevos a la base de datos.
- ✓ Insertar datos nuevos en archivos ya existentes.
- ✓ Obtener datos de archivos ya existentes.
- ✓ Actualizar datos en archivos ya existentes.
- ✓ Eliminar archivos ya existentes de la base de datos.

2.2.1 Clasificación de las bases de datos

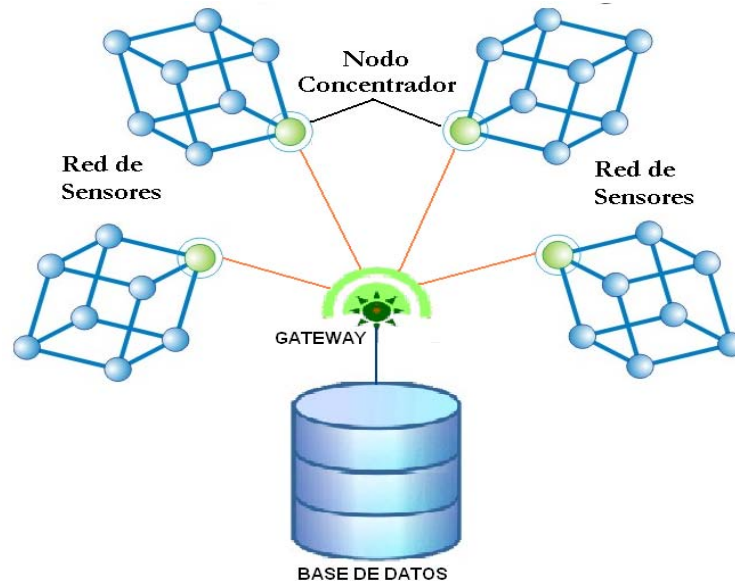
Las bases de datos pueden clasificarse según la variabilidad de los datos almacenados (Bases de datos estáticas o dinámicas) o según su contenido (Bases de datos bibliográficas y Bases de datos de texto completo).

Las bases de datos para redes de sensores inalámbricos deben estar en capacidad de recibir y almacenar enormes cantidades de datos producidas por las lecturas hechas por los sensores de magnitudes físicas diferentes, además de proporcionar un acceso eficiente a dicha información. Aunque la cantidad de datos del sensor recogidos depende de las limitaciones del ancho de banda que utilice la red para transmitir, la información tomada en tiempo real puede recargar el servidor si no se diseña correctamente la base de datos.

En la figura 1 se muestra el proceso que seguirán los datos capturados por la red de sensores inalámbricos para finalmente, llegar a ser almacenados en la base de datos.

Cada red de sensores inalámbricos que se creó, envía los datos que captura al nodo concentrador y es el encargado de enviarlos al gateway para que desde ahí sea accedido a través de una aplicación que lee los datos, realiza un proceso de verificación para luego almacenarlos en la base de datos.

Figura 1. Conexión con una base de datos de una red de sensores



Fuente: Autoras del proyecto

2.3 PROTOCOLOS DE COMUNICACIÓN

Para que dos nodos de una red se puedan comunicar entre si es necesario que ambos empleen el mismo “idioma”, es decir, que entre ellos exista la misma configuración de protocolos de comunicación que son las reglas que especifican el intercambio de datos y órdenes generados de la transmisión y recepción de la información.

En sistemas de comunicación de alto impacto, los protocolos de comunicación se convierten en estándares; OSI (Organización Internacional para la estandarización) es la entidad encargada de definir cómo pueden los fabricantes crear sus productos de tal forman que funcionen con los de otros fabricantes sin necesidad de limitarlos a controladores o equipos adicionales.

Entre los protocolos propios de una red de área local se pueden distinguir dos principales grupos. Por un lado están los protocolos de los niveles Físico y de Enlace, (niveles 1 y 2 del modelo OSI), que definen las funciones asociadas con el uso del

medio de transmisión: envío de los datos a nivel de bits y trama, y el modo de acceso de los nodos al medio. Estos protocolos vienen unívocamente determinados por el tipo de red (Ethernet, Token Ring, etc.). El segundo grupo de protocolos se refiere a aquellos que realizan las funciones de los niveles de Red y Transporte, (niveles 3 y 4 de OSI), es decir los que se encargan básicamente del enrutamiento de la información y garantizar una comunicación extremo a extremo libre de errores. Estos protocolos transmiten la información a través de la red en pequeños segmentos llamados paquetes. Si un ordenador quiere transmitir un fichero grande a otro, el fichero es dividido en paquetes en el origen y vueltos a ensamblar en el ordenador destino⁹. Cada protocolo define su propio formato de los paquetes en el que se especifica el origen, destino, longitud y tipo del paquete, así como la información redundante para el control de errores. Los protocolos de los niveles 1 y 2 dependen del tipo de red, mientras que para los niveles 3 y 4 hay diferentes alternativas, siendo TCP/IP la configuración mas extendida.

2.3.1 ZigBee (IEEE 802.15.4)

Inicialmente la abolición de cables en la transmisión de datos se inició con la implementación del Bluetooth, un estándar creado para la transmisión de voz y datos entre dispositivos móviles y accesorios (como los manos libres), pero pronto surgieron inconvenientes con ésta nueva tecnología como su elevado costo, su alto consumo de energía y su limitado alcance de transmisión, llevando a las compañías fabricantes a aliarse en busca de una solución económica que aunque no tuviera la misma velocidad de transmisión que Bluetooth, cumpliera con los requerimientos que estaban buscando.

De esta forma nace el estándar ZigBee, pensado para transmitir pequeños paquetes de datos utilizando dispositivos cuyo tamaño es cada vez mas reducido. Es un estándar

⁹ <http://www.angelfire.com/mi2/Redes/protocolo.html>

tecnológico, basado en la norma IEEE 802.15.4, que fue creado específicamente para el control y monitoreo de redes de sensores. ZigBee maneja una baja velocidad de transmisión de datos, bajo consumo de energía y baja complejidad y está diseñado para redes de área personal (WPAN). Este estándar fue creado por una organización conocida como la Alianza ZigBee, que se compone de un gran número de empresas y líderes de la industria informática y de comunicaciones. [9]

Una de las preocupaciones que pueden surgir alrededor de Zigbee es la relacionada con la banda de frecuencia específica que utiliza, (2.4 GHz), que es la misma banda utilizada por Bluetooth y WiFi, pero ZigBee puede tener acceso hasta a 16 diferentes canales de 5 MHz en la banda de 2.4 GHz, varios de los cuales no se superponen a los de Bluetooth y WiFi y en caso de interferencia los paquetes de datos se retransmiten automáticamente. Por lo tanto, ZigBee, interactúa perfectamente con otras tecnologías inalámbricas.

Su ventaja radica en la disminución del consumo energético pues la mayor parte del tiempo se encuentra suspendido y solo se activa en el momento necesario de iniciar la transmisión, y aunque ZigBee ha sido diseñada para transmitir a 250 kbps (que comparado con Wi-Fi que transmite a 200 Mbps o más resulta bastante lenta), es esta característica la que le permite a los transmisores ZigBee funcionar de seis meses a dos años con un par de baterías AA¹⁰ sin necesidad de recarga; sin contar además con la disminución en los costos de fabricación y un alcance igual o que puede superar a una tecnología tan diferente como Wi-Fi, calculando rangos que van desde los 10 hasta los 100 metros de propagación de su señal.

Inicialmente, la investigación de la tecnología relacionada con las redes de sensores inalámbricas se llevó a cabo en los equipos de la Universidad de California Berkeley

¹⁰ Batería AA: 13.9 milímetros de diámetro y 50 milímetros de altura total

donde se han desarrollado prototipos de nodos y aplicaciones para demostraciones e investigaciones. Allí, los investigadores utilizan las redes de sensores inalámbricas para recibir información detallada sobre el efecto de los movimientos sísmicos en los edificios, entre otras aplicaciones.

2.3.2 Bluetooth (IEEE 802.15.1)

La tecnología Bluetooth es automática e inalámbrica, y tiene un número de características interesantes que pueden facilitar la vida cotidiana, como por ejemplo, ser una tecnología de ondas de radio de corto alcance, desarrollada e impulsada en el mercado por el Bluetooth Special Interest Group, cuyo objetivo primordial es establecer comunicación simultánea entre grupos de dispositivos o entre productos individuales con Internet.

Permite comunicaciones incluso a través de obstáculos, a distancias de hasta 10 metros dependiendo de la versión. También sirve para crear una conexión a Internet inalámbrica desde un portátil usando el teléfono móvil. Un caso aún más práctico es el poder sincronizar libretas de direcciones, calendarios etc. en la PDA, teléfono móvil, computador de escritorio y portátil automáticamente y al mismo tiempo. Los promotores de Bluetooth incluyen Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia y Toshiba, y cientos de compañías asociadas.

Esta tecnología está diseñada para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura y globalmente libre (2,4 GHz.). Los principales objetivos que se pretende conseguir con esta norma son:

- ✓ Facilitar las comunicaciones entre equipos móviles y fijos.

- ✓ Eliminar cables y conectores.
- ✓ Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

2.4 REDES DE SENSORES INALÁMBRICOS

Una red de sensores inalámbricos está conformada por un grupo de pequeños dispositivos de última tecnología, distribuidos espacialmente, con mecanismos diminutos, baratos y además de bajo consumo de energía que están organizados en función de una tarea común [10].

Estos dispositivos electrónicos, están dotados de una serie de sensores, encargados de tomar medidas de variables predeterminadas como temperatura, humedad, sonido, vibración, presión, etc. y son capaces tanto de procesar información localmente, como de comunicarse inalámbricamente para transmitir los datos.

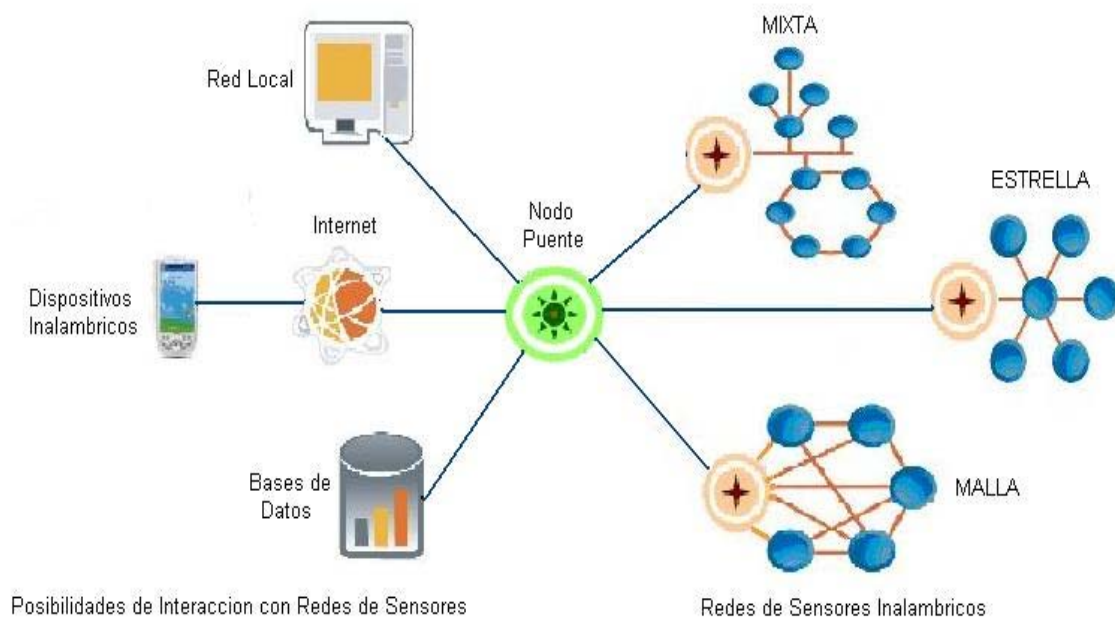
Características de una red de sensores:

- ✓ Topología Dinámica
- ✓ Variabilidad del canal inalámbrico
- ✓ No se utiliza infraestructura de red
- ✓ Tolerancia a errores
- ✓ Comunicaciones multisalto o broadcast
- ✓ Consumo energético bajo
- ✓ Hardware limitado

La funcionalidad de este tipo de redes radica en que mediante un nodo puente, se puede establecer conexión con redes locales, con dispositivos con tecnología

inalámbrica que se comuniquen a través de internet y con bases de datos que reciban y envíen información capturada localmente, o desde un servidor de datos, o desde internet, como muestra la figura 2.

Figura 2. Redes de sensores inalámbricos



Fuente: Autoras del proyecto

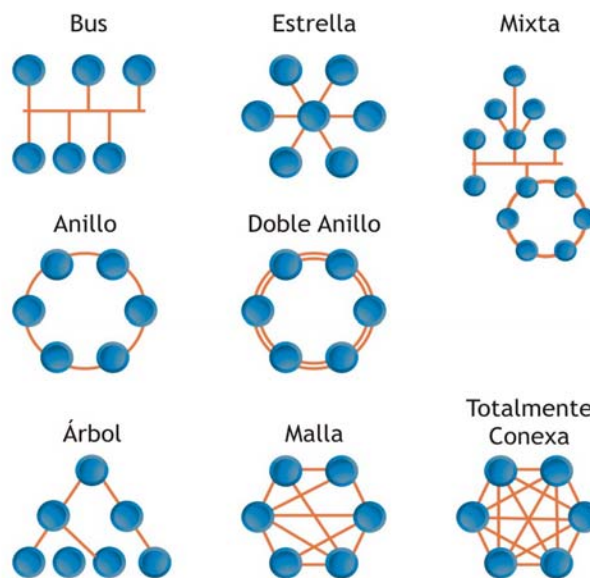
2.4.1 Topologías de Red

La forma como se disponen físicamente los nodos de una red de computadores o en este caso específico, una red de sensores y su conexión mediante el uso de estándares y protocolos, se denomina topología o arquitectura de red.

La topología de red la determina únicamente la configuración de las conexiones entre

nodos. La distancia entre los nodos, las interconexiones físicas, las tasas de transmisión, definen una red y las reglas de interacción entre los componentes de la misma. La figura 6 muestra diferentes topologías de red existentes y de uso común:

Figura 3. Topologías comunes de red



Fuente: http://es.wikipedia.org/wiki/Imagen:Topolog%C3%ADa_de_red.png Modificada por las autoras de:

2.4.2 Aplicaciones de Redes de Sensores Inalámbricos [11]

2.4.2.1 Aplicaciones en la Agricultura

Las aplicaciones de redes inalámbricas han encontrado en la agricultura una de las áreas mas importantes para implementarse. Las redes de sensores contribuyen a la optimización de procesos y el control de las variables que afectan el entorno de

desarrollo. Con los avances en la tecnología inalámbrica y los desarrollos que se han producido en las redes de sensores inalámbricos en los últimos años, especialmente la miniaturización de los dispositivos, han surgido nuevas tendencias en el sector agrícola como la llamada agricultura de precisión.

Las aplicaciones de precisión están enfocadas al control y manipulación de las variables que afectan un entorno, a la detección a tiempo de problemas y a la gestión y optimización de procesos. Existen algunas aplicaciones interesantes entre las que se encuentran las aplicaciones para el control de plagas y enfermedades. Los sensores inalámbricos permiten monitorear variables como el clima, la temperatura o la humedad de las hojas, con el fin de detectar rápidamente situaciones adversas y desencadenar los tratamientos apropiados.

Otra aplicación interesante es el desarrollo de un sistema de monitoreo inalámbrico de caudales en una red canales de riego. Este sistema Web permite ver el caudal a partir de las imágenes de Google Maps, evitando así que el personal se tenga que desplazar directamente a un sitio predeterminado para medir el terreno.

2.4.2.2 Aplicaciones sociales y sanitarias

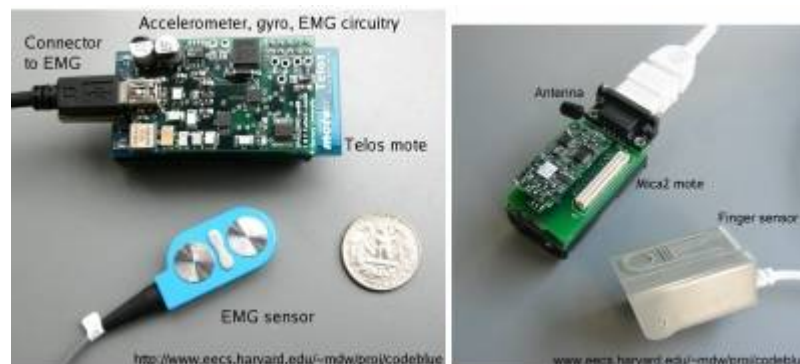
Las enfermedades y sus efectos no solo en el organismo humano sino también en la vida cotidiana de las personas que las padecen han suscitado el uso de la tecnologías inalámbricas en el proceso de seguimiento y cuidado que requiere cada caso, sin afectar la privacidad de los pacientes y sin dejar de lado todos los cuidados médicos necesarios para llegar a la recuperación total.

Mediante el uso de una red de sensores inalámbricos situados en puntos estratégicos del domicilio de los pacientes, así como en objetos de uso cotidiano, las personas

encargadas del cuidado de los enfermos, pueden monitorear en tiempo real el comportamiento de las pacientes y su evolución, evitando la realización de tareas tediosas y concentrando sus esfuerzos en aspectos más importantes como es la mejora de la calidad de vida del paciente.

CodeBlue¹¹ es un buen ejemplo de este tipo de aplicaciones. Este proyecto fue desarrollado en la Universidad de Harvard, donde implementaron distintos tipos de sensores para monitorear parámetros vitales como la tasa de latidos del corazón, la concentración de oxígeno en sangre, los datos EKG de electrocardiograma, etc. Toda esta información se recoge por los sensores y se distribuye de forma inalámbrica a una PDA o computador portátil para su procesamiento. De este modo, cualquier señal de alerta puede detectarse a distancia en tiempo real.

Figura 4. Sensores para cuidado médico desarrollados en el proyecto CodeBlue



Fuente: <http://www.radioptica.com/Radio/wsn.asp?pag=3>

2.4.2.3 Aplicaciones en la naturaleza

Las aplicaciones de redes de sensores inalámbricas en la naturaleza se hace un poco

¹¹ Prof. Dr. Francisco Ramos Pascual. Artículo publicado en la revista ELECTRÓNICA & COMUNICACIONES nº 217

compleja debido a las grandes dimensiones que se manejan, a la flora y fauna que no deben verse afectadas por la implementación de tecnología y a las condiciones variables que interactúan con el medio.

Debido a estas limitaciones, los avances de este tipo de tecnología en cuanto a tamaño, capacidad de almacenamiento y transmisión hacen de las redes inalámbricas de sensores una herramienta que puede resultar de gran ayuda en este tipo de medios. Los sensores, de pequeño tamaño, pueden disimularse con en el entorno, procesando los datos de diversos parámetros ecológicos y transmitiendo la información de forma inalámbrica hasta un centro de control. Entre los parámetros a monitorizar están: temperatura, humedad, crecimiento de árboles y arbustos, desplazamientos de especies, conteo de animales, caudales de ríos, etc.

2.4.2.4 Aplicaciones civiles y militares

Existen distintas aplicaciones prácticas para implementar en ámbitos civiles y militares, entre ellas se incluyen las siguientes:

- ✓ Localización de espacios libres en parqueaderos.
- ✓ Vigilancia de niños y personas mayores mediante videocámaras.
- ✓ Seguridad del hogar.
- ✓ Detector de riesgo de epidemias (gripe, fiebres, etc.)
- ✓ Monitoreo de redes de computadores.
- ✓ Observatorios terrestres y marítimos (costas)

Figura 5. Sensor inalámbrico de presión para pozos de petróleo



Fuente: <http://www.radioptica.com/Radio/wsn.asp?pag=5>

Además de estas aplicaciones civiles, las redes de sensores inalámbricos encuentran un importante campo de aplicación en misiones militares. Por ejemplo, en la identificación y seguimiento de tropas o vehículos militares, así como en la detección de armas químicas y biológicas.

2.4.3 Computación Ubicua

El concepto de computación Ubicua surgió en 1991 a cargo de Mark Weiser, quien visionaba tener entornos completos de computación y comunicación integrados conjuntamente con las personas; es decir, introducir en los objetos cotidianos la capacidad de computación haciendo que las personas se concentren en lo que deben hacer y no en las herramientas que utilizan para hacerlo, porque se pretende que esas herramientas pasen desapercibidas. La idea de Weiser se convertiría en el tercer paradigma de la computación (la primera fue un sistema de cómputo de tiempo compartido por muchas personas y la segunda era un sistema personal donde para una persona hay un computador de escritorio), en la que se tienen varios sistemas de cómputos con diferentes características al servicio de una persona.

En ese entonces era difícil concebir la idea de Weiser como algo tangible a corto plazo, sin embargo, mas de 10 años después de sus primeras observaciones, la interacción hombre-máquina se consolida cada vez más debido a la cantidad de dispositivos al alcance del usuario como los ordenadores de bolsillo, las redes inalámbricas, los sensores cada vez mas pequeños y con mayor potencia y en general, los grandes avances tecnológicos que hoy en día pueden posibilitar el acceso a gran cantidad de información sin importar la ubicación del usuario.

Esta disciplina, también llamada Inteligencia Ambiental tiene ramas asociadas como las aplicaciones sensibles al contexto, la computación pervasiva, los hogares digitales o domótica entre otras.

Actualmente la integración de la informática con el entorno que nos rodea se logra insertando dispositivos inteligentes en los aparatos de uso diario para que la interacción con estos sea natural, desinhibida y transparente para las personas. Sin embargo, la computación ubicua es aún un campo por explorar y por seguir investigando así como todos los conceptos que la complementan (sensores, comunicaciones, materiales y componentes) para seguir avanzando conforme se avanza en estos aspectos; ya que el objetivo primordial es que pase tan desapercibida la utilización de estos dispositivos que el usuario no note toda la tecnología que hay embebida en ellos sino simplemente las acciones que pueden realizar sin importar donde se encuentre, es decir que los servicios que prestan los aparatos sean tan móviles como los usuarios y ya no tener que depender de los cables que mantenían atados a las personas con los dispositivos.

2.4.3.1 Proyectos

- ✓ Paraguas con Internet: En la universidad de Tokio se está trabajando en un prototipo de paraguas que se conecta a Internet por una conexión inalámbrica

que permite proyectar en el interior del mismo las imágenes del ciberespacio. Esto resulta novedoso luego del desarrollo de algunas compañías estadounidenses que desarrollaron un paraguas que a través de un receptor radial mostraba los estados del tiempo iluminando el mango según el boletín meteorológico recibido. Cuanto mas fuerte se iluminara el mango, mas indicio de lluvia o nieve se pronosticaba.

- ✓ En los últimos años el campo de los sensores se ha desarrollado a pasos agigantados, reduciendo cada vez más su tamaño y su consumo energético. Gracias a estos desarrollos se consiguen hoy en día cámaras y micrófonos cada vez mas pequeños, detectores de huellas digitales, sensores de localización y reconocimiento de voz entre otras.
- ✓ Construcción de la ciudad ubicua de New Songlo City, en una isla a 60 kilómetros al oeste de Seúl y 680 hectáreas de superficie. Se presume que esté terminada para el 2014 y que albergue a 65000 personas donde todos los sistemas de información estén interconectados y los computadores estén integradas a las viviendas, calles, edificios y zonas comerciales.
- ✓ Oxygen desarrollado por el Instituto Tecnológico de Massachusetts, el Aura del CMU y el CoolTown de HP son algunos proyectos de computación ubicua centrados en desarrollar y perseguir los objetivos que hagan de este concepto una realidad a corto plazo.
- ✓ El software Facet analiza los acontecimientos grabados por dispositivos móviles y proporciona una plataforma para que estos dispositivos transmitan por ejemplo la detección de intrusos, haciendo de este software un sistema de vigilancia para interpretar lo que ocurre comparándolo con las imágenes captadas por las cámaras haciendo innecesaria la implementación de una persona que interprete lo que ocurre en la pantalla como sucedía en investigaciones anteriores similares.
- ✓ **REIMAGAV** (Red Inalámbrica de Monitorización Ambiental en Galpones Avícolas). Es el proyecto desarrollado por el grupo de investigación en

Conectividad y Procesado de Señal, CPS, de la Universidad Industrial de Santander que propone el diseño de una red para la monitorización de variables físicas como temperatura, humedad e intensidad luminosa, y que junto con el uso de termografía infrarroja busca identificar la incidencia de estas variables y los cambios térmicos corporales que afectan el ciclo productivo de los pollos de engorde y la relación con la mortandad de las mismas; brindándole a la industria los argumentos suficientes para optimizar sus procesos y disminuir las pérdidas económicas que se registran por la alta tasa de mortalidad presente durante la producción de los pollos. Este proyecto de grado hace parte de REIMAGAV junto con otros desarrollos de la escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones¹².

¹² Andescon 2008.pdf. Este documento se encuentra en el CD adjunto a este documento

3. PROTOCOLOS DE COMUNICACIÓN

Desde hace algunos años, se han venido desarrollando diferentes protocolos de comunicación para redes inalámbricas que han permitido la evolución de la tecnología de redes de sensores inalámbricos existentes. Entre las más comerciales está el estándar IEEE 802.15.1 Bluetooth y el estándar IEEE 802.15.4 en el cual se basa la tecnología ZigBee.

Los protocolos ZigBee minimizan el tiempo de actividad de la radio para evitar el consumo de energía. En las redes encargadas de transmitir paquetes con información, los nodos sólo necesitan estar activos mientras se transmiten las tramas (o cuando se les asigna tiempo para transmitir). Si no hay paquetes para transmitir en la red, el consumo es asimétrico repartido en dispositivos permanentemente activos y otros que sólo lo están esporádicamente.

Para conocer y utilizar las comunicaciones inalámbricas es necesario hacer un análisis de los estándares que las rigen y establecer algunas ventajas y desventajas entre ellos para determinar cual es la más adecuada para el modelo de desarrollo que se quiera seguir; basados en que en el proyecto se usaron nodos inalámbricos con tecnología ZigBee y haciendo el estudio comparativo con Bluetooth para entender la razón por la cual es más útil implementar una u otra en los dispositivos utilizados en la red de sensores inalámbricos en los galpones

3.1 ZIGBEE

La especificación ZigBee aparece en el año 2004 y nace soportada en el protocolo IEEE 802.15.4. Es un estándar tecnológico creado específicamente para el control y

monitoreo de redes de sensores.

ZigBee maneja una baja velocidad de transmisión de datos, bajo consumo de energía con maximización de la vida útil de las baterías, baja complejidad y está diseñado para redes de área personal (WPAN). Este estándar fue creado por una organización conocida como la Alianza ZigBee, que se compone de un gran número de empresas y líderes de la industria informática y de comunicaciones.

Se destacan empresas como Invensys, Mitsubishi, Honeywell, Philips y Motorola que trabajan para crear un sistema estándar de comunicaciones, vía radio y bidireccional, para usarlo dentro de dispositivos de domótica, automatización de edificios (inmótica), control industrial, periféricos de PC, juguetería, sensores médicos. Los miembros de esta alianza justifican el desarrollo de este estándar para cubrir el vacío que se produce por debajo del Bluetooth.

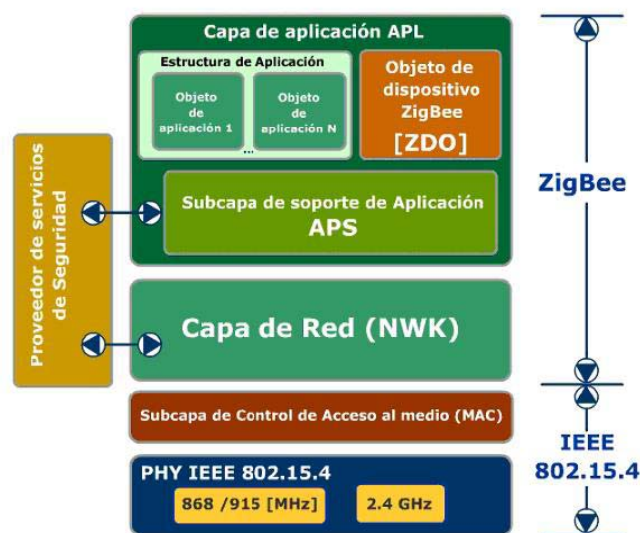
Los dispositivos ZigBee se rigen por el estándar IEEE 802.15.4-2003 de WPAN de baja tasa de transmisión. Éste define los niveles más bajos: el nivel físico (PHY) y el control de acceso al medio (MAC, parte del nivel de enlace de datos, DLL).

ZigBee utiliza la banda ISM para usos industriales, científicos y médicos. En Europa, 868 MHz, en Estados Unidos ,915MHz y 2,4 GHz en todo el mundo. Sin embargo la mayoría de empresas a la hora de diseñar dispositivos, utilizan por lo general la banda de 2,4 GHz, por ser libre en todo el mundo. Se definen hasta 16 canales en el rango de 2,4 GHz, cada uno de ellos con un ancho de banda de 5 MHz.

La arquitectura ZigBee, que se ilustra en la figura 9, se basa en el modelo estándar de siete-capas del Open Systems Interconnection OSI. El estándar IEEE 802.15.4-2003 define las dos capas más bajas: la capa física PHY y la subcapa de control de acceso al

medio MAC. La alianza ZigBee [9] define la capa de red NWK y la estructura para la capa de aplicación, que incluye la subcapa de soporte de aplicación APS, los objetos del dispositivo ZigBee ZDO y los objetos de aplicación definidos por el fabricante.

Figura 6. Modelo de capas para la especificación ZigBee



Fuente: Tesis Maestría, Ingeniero Jose de Jesús Rugeles Uribe

La primera versión de este estándar se llama ZigBee 2004. La segunda versión se da en el año 2006 se denomina ZigBee 2006, y reemplaza la estructura MSG/KVP con una librería de clusters, dejando obsoleta a la anterior versión.

La última versión es de octubre de 2007 y se creó para que se adecue a la actual versión de la especificación, centrándose en optimizar funcionalidades de nivel de red (como agregación de datos). También se incluyen algunos perfiles de aplicación nuevos, como lectura automática, automatización de edificios comerciales y automatización de hogares en base al principio de uso de la librería de clusters.

El nivel de red de ZigBee 2007 o ZigBee PRO no es compatible con el de ZigBee 2004-2006, aunque un nodo RFD puede unirse a una red 2007 y viceversa. No pueden combinarse routers de las versiones antiguas con un coordinador 2007.

En la tabla se muestran las características principales de la versión más actual del estándar ZigBee.

Tabla1. Características ZigBee PRO

Característica	Descripción
Escalabilidad de red	Un mejor soporte para las redes más grandes, ofreciendo más opciones de gestión, flexibilidad y desempeño.
Fragmentación	Nueva capacidad para dividir mensajes más largos y permitir la interacción con otros protocolos y sistemas.
Agilidad de Frecuencia	Las redes cambian los canales en forma dinámica en caso que ocurra interferencia.
Gestión automatizada de direcciones de dispositivos	El conjunto fue optimizado para grandes redes con gestión de red agregada y herramientas de configuración.
Localización grupal	Ofrece una optimización adicional del tráfico necesaria para las grandes redes.
Puesta de Servicio Inalámbrico	El conjunto fue mejorado con capacidades seguras para poner en marcha el servicio inalámbrico.
Recolección centralizada de datos	El conjunto fue sintonizado específicamente para optimizar el flujo de información en las grandes redes.

Fuente: Tesis de Maestría, Ingeniero Jose de Jesús Rugeles Uribe

3.2 REDES ZIGBEE

ZigBee permite tres topologías de red:

- ✓ Topología en estrella: el coordinador se sitúa en el centro.
- ✓ Topología en árbol: el coordinador será la raíz del árbol.
- ✓ Topología de malla: al menos uno de los nodos tendrá más de dos conexiones.

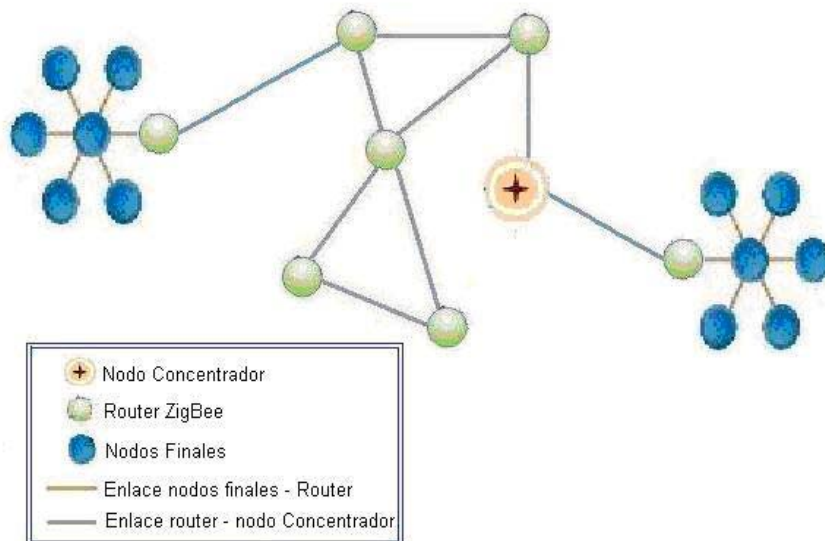
En cualquiera de estas arquitecturas siempre existe un nodo de red que asume el papel de coordinador y que siempre está listo para comenzar a trabajar en el momento en que cualquier otro nodo salga del estado de suspensión para optimizar el consumo.

La topología más interesante es la topología de malla. Ésta permite que si, en un momento dado, un nodo del camino falla y se cae, pueda seguir la comunicación entre todos los demás nodos debido a que se rehacen todos los caminos. La gestión de los caminos es tarea del coordinador, quien además también es responsable de inicializar la red y de elegir los parámetros de la misma, aunque se debe tener en cuenta que la red puede ser ampliada a través del uso de routers ZigBee.

En la configuración en estrella, uno de los dispositivos denominado nodo concentrador, asume el rol de coordinador de red y es el responsable de inicializar y mantener los dispositivos en la red. Todos los demás dispositivos que hacen parte de las redes zigbee, se conocen con el nombre de dispositivos finales, y hablan directamente con el coordinador.

Cualquier nodo ZigBee puede hacer llegar los datos a cualquier parte de la red, siempre que ésta esté lo suficientemente dispersa como para que todos los dispositivos tengan un vecino dentro de su rango de cobertura.

Figura 7. Ilustración de una red ZigBee



Fuente: Autoras del proyecto

Los componentes que observamos de la grafica se describen claramente a continuación:

- **Nodo Concentrador ZigBee:** Este dispositivo se encarga de controlar la red y administrar los caminos que deben seguir los nodos para conectarse y transmitir la información entre ellos. Debe existir uno por red.
- **Router ZigBee** Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.
- **Nodo final:** Posee la funcionalidad necesaria para comunicarse con su router

ZigBee, pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Este tipo de dispositivos tienen requerimientos mínimos de memoria y son por lo tanto significativamente más baratos.

El algoritmo de encaminamiento utiliza un protocolo de pregunta-respuesta (request-response) para eliminar las rutas que no sean óptimas, La red final puede tener hasta 254 nodos y se pueden crear hasta 255 redes de este tipo para configurar una red de mas de 65000 nodos.

3.2.1 Aplicaciones Zigbee

Los protocolos ZigBee están definidos para su uso en aplicaciones embebidas con requerimientos muy bajos de transmisión de datos y consumo energético. Se pretende su uso en aplicaciones de propósito general con características auto organizativas y de bajo coste (redes en malla, en concreto). Puede utilizarse para realizar control industrial, albergar sensores empotrados, recolectar datos médicos, ejercer labores de detección de humo o intrusos.

Como ejemplo de aplicación en Domótica, en una habitación de la casa se tienen diversos dispositivos finales (como un interruptor y una lámpara) y una red de interconexión realizada con routers ZigBee y gobernada por el coordinador.

La red en su conjunto utilizará una cantidad muy pequeña de energía de forma que cada dispositivo individual pueda tener una autonomía de hasta 5 años antes de necesitar un recambio en su sistema de alimentación.

3.2.2 Conexión Zigbee

Las redes ZigBee han sido diseñadas para conservar la potencia en los nodos finales, de esta forma se consigue el bajo consumo de potencia. La estrategia consiste en que, durante mucho tiempo, un dispositivo final está en modo “dormido”, de tal forma que solo se “despierta” por una fracción de segundo para confirmar que está “vivo” en la red de dispositivos de la que forma parte. Esta transición del modo “dormido” al modo “despierto” (modo en el que realmente transmite), dura unos 15ms.

En las redes Zigbee, se pueden usar dos tipos de entornos o sistemas, aquellos que transmiten paquetes de datos y los que no.

El entorno en el cual se hace necesario el envío de paquetes, ejerce un mecanismo de control del consumo de potencia en la red. Este entorno permite a todos los dispositivos saber cuándo pueden transmitir. En este modelo, los dos caminos de la red tienen un distribuidor que se encarga de controlar el canal y dirigir las transmisiones.

Este sistema o modo de transmisión es más recomendable cuando el coordinador de red trabaja con una batería. Los dispositivos que conforman la red, escuchan a dicho coordinador durante el envío de mensajes y cuando alguno quiera intervenir, lo primero que tendrá que hacer es registrarse para el coordinador, y es entonces cuando mira si hay mensajes para él. En el caso de que no haya mensajes, este dispositivo vuelve a “dormir”, y se despierta de acuerdo a un horario que ha establecido previamente el coordinador. En cuanto el coordinador termina el envío del paquete, vuelve a “dormirse”.

Cuando se trabaja con sistemas en donde no es necesario el envío de paquetes de información, se usa el acceso múltiple al sistema Zigbee en una red punto a punto cercano. En este tipo, cada dispositivo es autónomo, pudiendo iniciar una conversación, en la cual los otros pueden interferir. A veces, puede ocurrir que el dispositivo destino no escuche la petición, o que el canal esté ocupado.

Este sistema se usa típicamente en los sistemas de seguridad, en los cuales sus dispositivos (sensores, detectores de movimiento o de rotura de cristales), permanecen inactivos prácticamente todo el tiempo (el 99,999%). Para que se les tenga en cuenta, estos elementos se “despiertan” de forma regular para anunciar que siguen en la red. Cuando se produce un evento (detección de movimiento, humo, temperaturas superiores a un margen establecido, etc.), el sensor “despierta” instantáneamente y transmite la alarma correspondiente. Es en ese momento cuando el coordinador de red, recibe el mensaje enviado por el sensor, y activa la alarma correspondiente. En este caso, el coordinador de red se alimenta de la red principal durante todo el tiempo.

3.2.3 Evolución Zigbee¹³

- ✓ Las redes de la familia de ZigBee se conciben hacia 1998, al tiempo que se hizo claro que Wi-Fi y Bluetooth no serían soluciones válidas para todos los contextos. En concreto, se observó una necesidad de redes ad-hoc inalámbricas.
- ✓ El estándar IEEE 802.15.4 se aprobó en mayo de 2003.
- ✓ En el verano de 2003, Philips Semiconductors puso fin a su inversión en redes de mallas. Philips Lighting ha perpetuado la participación de Philips, que sigue siendo un miembro prominente de la ZigBee Alliance.
- ✓ ZigBee Alliance anunció en octubre de 2004 una duplicación en su número de miembros en ese año a más de 100 compañías en 22 países. En abril de 2005 había más de 150 miembros corporativos, y más de 200 en diciembre del mismo año y actualmente es una organización muy estable en la industria.
- ✓ La especificación se aprobó el 14 de diciembre de 2004.
- ✓ ZigBee 2004 se puso a disposición del público el 13 de junio de 2005.
- ✓ En diciembre de 2006 se publicó la actual revisión de la especificación.
- ✓ En Noviembre de 2007 se publicó el perfil HOME AUTOMATION de la

¹³ www.zigbee.org

especificación.

Se espera que los módulos ZigBee sean los transmisores inalámbricos más baratos de la historia, y además producidos de forma masiva. Tendrán un coste aproximado de alrededor de los 6 euros, y dispondrán de una antena integrada, control de frecuencia y una pequeña batería. Ofrecerán una solución tan económica porque la radio se puede fabricar con muchos menos circuitos analógicos de los que se necesitan habitualmente.

3.3 BLUETOOTH

La tecnología Bluetooth originalmente fue desarrollada por Ericsson en 1994. En febrero de 1998, se formó un grupo llamado Bluetooth Special Interest Group (Bluetooth SIG) con más de 200 compañías, dentro de las cuales se encontraban Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia y Toshiba. Esta tecnología está integrada por hardware, software y requerimientos de operatividad, por lo que el objetivo de estas empresas era desarrollar las especificaciones para Bluetooth 1.0, que se publicaron en julio de 1999 y trabajar conjuntamente en los dispositivos que van a brindar seguridad en la transmisión y facilidad de interconexión.

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura y globalmente libre (2,4 GHz.). Los principales objetivos que se pretende conseguir con esta norma son:

- ✓ Facilitar las comunicaciones entre equipos móviles y fijos.
- ✓ Eliminar cables y conectores entre éstos.
- ✓ Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales.

Esta tecnología se utiliza con mayor intensidad en el sector de las telecomunicaciones y la informática personal, como PDAs, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras y cámaras digitales, evitando así el uso de conexiones cableadas.

Pero no solo la eliminación de los cables es el punto fuerte de esta tecnología inalámbrica, también se encuentra el intercambio de archivos, tarjetas de visita, citas del calendario entre usuarios de bluetooth; la sincronización y transferencia de archivos etc.

Bluetooth se encuentra dividido en dos clases que dependen de la velocidad de transmisión, siendo la clase 2 el estándar más común y barato, que permite tener dispositivos conectados a unos 10 metros de distancia, transmitiendo a velocidades de aproximadamente 1 Mbps; la clase 1 no es tan común y tiene un alcance de unos 100 metros.

Tabla 2. Clases Bluetooth

CLASES	POTENCIA (PERDIDA DE SEÑAL)	ALCANCE
CLASE 1	100 mW (20 dBm)	100 METROS
CLASE 2	2.5 mW (4 dBm)	10 METROS

Fuente: Autoras del proyecto

Del mismo modo que WiFi, bluetooth utiliza la técnica FHSS (Frequency Hopping Spread Spectrum, o Espectro ensanchado por saltos de frecuencia), que consiste en

dividir la banda de frecuencia de 2.402 - 2.480 GHz en 79 canales (denominados saltos) de 1 MHz de ancho cada uno y, después, transmitir la señal utilizando una secuencia de canales que sea conocida tanto para la estación emisora como para la receptora. Por lo tanto, al cambiar de canales con una frecuencia de 1600 veces por segundo, el estándar Bluetooth puede evitar la interferencia con otras señales de radio con las que comparte la frecuencia de transmisión.

La siguiente tabla muestra el ancho de banda que usa Bluetooth para transmitir, dependiendo de la versión:

Tabla 3. Anchos de Banda utilizados por Bluetooth

VERSION	ANCHO DE BANDA
Versión 1.2	1 Mbit / s
Versión 2.0 + EDR	3 Mbit / s
UWB (Ultra Wide Band) Bluetooth	53 – 480 Mbit / s

Fuente: Autoras del proyecto

Fue Ericsson quien desde 1994 inició estudios para investigar la viabilidad de una nueva interfaz de bajo costo y consumo para la interconexión vía radio (eliminando de esta forma los cables) entre dispositivos como teléfonos móviles y otros accesorios. El estándar Bluetooth ha madurado y evolucionado su concepto a tal punto que hoy contamos con la versión ULP (Ultra Low Power Band).

Desde sus inicios hasta la actualidad, las versiones de Bluetooth que han salido al mercado son las siguientes:

- ✓ Bluetooth v.1.1
- ✓ Bluetooth v.1.2
- ✓ Bluetooth v.2.0
- ✓ Bluetooth v.2.1
- ✓ UWB (Ultra Wide Band) Bluetooth
- ✓ ULP (Ultra Low Power)Bluetooth

La diferencia entre la versión 1.1 y la versión 1.2, es la solución inalámbrica complementaria que se desarrolló para permitir que Bluetooth y Wi-Fi co-existieran en el espectro de los 2.4 GHz, sin que se presentara interferencia entre ellos.

La versión 1.2 usa la técnica "Adaptive Frequency Hopping (AFH)", que ejecuta una transmisión más eficiente y un cifrado más seguro. Esta versión mejora la calidad de voz (Voice Quality - Enhanced Voice Processing) obteniendo menor ruido ambiental, y proveyéndole una más rápida configuración de la comunicación con los otros dispositivos Bluetooth dentro del rango del alcance, como pueden ser PDAs, HIDs (Human Interface Devices), portátiles, computadores de escritorio, Headsets, impresoras y celulares.

La versión 2.0 es prácticamente una especificación separada; incorpora la técnica Enhanced Data Rate" (EDR) que le permite mejorar las velocidades de transmisión en hasta 3Mbps a la vez que intenta solucionar algunos errores de la especificación 1.2.

La versión 2.1, simplifica los pasos para crear la conexión entre dispositivos, además el consumo de potencia es 5 veces menor.

Ultra Wide Band Bluetooth

El 28 de marzo de 2006, La Ultra Wide Band Bluetooth fue anunciada por el Bluetooth SIG y se planteó la posibilidad de utilizar Ultra-Wideband/MB-OFDM como capa física

para futuras versiones de Bluetooth.

La integración de UWB crea una versión de la tecnología Bluetooth se hizo para poder utilizar grandes anchos de banda. Esta nueva versión permite alcanzar los requisitos de sincronización y transferencia de grandes cantidades de datos así como de contenidos de alta definición para dispositivos portátiles, proyectores multimedia, televisores y teléfonos VOIP.

Al mismo tiempo, la tecnología Bluetooth continuará satisfaciendo las necesidades de aplicaciones de muy bajo consumo como mouse, teclados o auriculares mono, permitiendo a los dispositivos seleccionar la capa física más apropiada para sus requisitos.

Ultra Low Power Bluetooth

En junio de 2007, Nokia y el Bluetooth SIG anunciaron esta nueva versión con la característica fundamental del bajo consumo, usada en diferentes aplicaciones como dispositivos sensores o envió de órdenes a distancia. La propuesta de Nokia es utilizar esta tecnología como enlace de bajo coste hasta un teléfono móvil que actúe de puerta de enlace hacia otras tecnologías como UMTS, Wi-Fi o incluso el mismo Bluetooth.

Arquitectura Hardware

El hardware que compone un dispositivo Bluetooth consta de dos partes fundamentales:

- ✓ **un dispositivo de radio**, encargado de modular y transmitir la señal.
- ✓ **un controlador digital**, compuesto por una CPU que se encarga de atender las instrucciones relacionada con el dispositivo anfitrión, por un procesador de señales digitales (DSP - Digital Signal Processor) llamado Link Controller (o controlador de enlace) que esta encargado de hacer el procesamiento de la

banda base del manejo de los protocolos ARQ y FEC de capa física. Además, se encarga de las funciones de transferencia (tanto asíncrona como síncrona), codificación de audio y cifrado de datos y de los interfaces con el dispositivo anfitrión.

3.3.1 Conexión entre dispositivos Bluetooth

El establecimiento de una conexión entre dos dispositivos Bluetooth sigue un procedimiento relativamente complicado para garantizar un cierto grado de seguridad, como el siguiente:

1. Modo pasivo
2. Solicitud: Búsqueda de puntos de acceso
3. Paginación: Sincronización con los puntos de acceso
4. Descubrimiento del servicio del punto de acceso
5. Creación de un canal con el punto de acceso
6. Emparejamiento mediante el PIN (seguridad)
7. Utilización de la red

Durante el uso normal, un dispositivo funciona en "modo pasivo", es decir, que está escuchando la red. El establecimiento de una conexión comienza con una fase denominada "solicitud", durante la cual el dispositivo maestro envía una solicitud a todos los dispositivos que encuentra dentro de su rango, denominados puntos de acceso. Todos los dispositivos que reciben la solicitud responden con su dirección.

El dispositivo maestro elige una dirección y se sincroniza con el punto de acceso mediante una técnica denominada paginación, que principalmente consiste en la sincronización de su reloj y frecuencia con el punto de acceso.

De esta manera se establece el enlace que le permite al dispositivo maestro ingresar a una fase de descubrimiento del servicio del punto de acceso, mediante un protocolo denominado SDP (Service Discovery Protocol).

3.4 ZIGBEE vs. BLUETOOTH [9] [32]

Tanto el estándar ZigBee como el estándar Bluetooth están diseñados para ser implementados en redes WLAN y se consideran tecnologías complementarias pero se trata de dos tecnologías distintas con diferentes campos de aplicación.

A continuación se presenta un comparativo entre estos dos estándares:

- ✓ Una red ZigBee puede constar de un máximo de 65535 nodos distribuidos en subredes de 255 nodos, frente a los 8 máximos de una subred Bluetooth.
- ✓ Menor consumo eléctrico que el de Bluetooth. En términos exactos, ZigBee tiene un consumo de 30mA transmitiendo y de 3uA en reposo, frente a los 40mA transmitiendo y 0.2mA en reposo que tiene el Bluetooth. Este menor consumo se debe a que el sistema ZigBee se queda la mayor parte del tiempo dormido, mientras que en una comunicación Bluetooth siempre se está transmitiendo y/o recibiendo.
- ✓ Tiene una velocidad de hasta 250 kbps, mientras que en Bluetooth es de hasta 1 Mbps.
- ✓ Debido a las velocidades de cada estándar, uno es más apropiado que el otro para ciertas aplicaciones. Por ejemplo, mientras que el Bluetooth se usa para aplicaciones como los teléfonos móviles y la informática casera, la velocidad del ZigBee se hace insuficiente para estas tareas, desviándolo a usos tales como la Domótica, los productos dependientes de la batería, los sensores médicos, y en artículos de juguetería, en los cuales la transferencia de datos

es menor.

- ✓ Existe una versión que integra el sistema de radiofrecuencias característico de Bluetooth junto a una interfaz de transmisión de datos vía infrarrojos desarrollado por IBM mediante un protocolo ADSI y MDSI.
- ✓ ZigBee, se centra en el control y la automatización, mientras que la tecnología Bluetooth se centra en la conectividad entre computadoras portátiles, PDA's, y cosas por el estilo.
- ✓ Mientras que ZigBee utiliza una baja velocidad de transmisión, bajo consumo de energía, y trabaja con pequeños paquetes de dispositivos; Bluetooth utiliza una mayor velocidad de transmisión de datos, mayor consumo de energía, y trabaja con grandes paquetes dispositivos.

Las redes ZigBee alcanzan números mayores de dispositivos y mayor alcance entre estos que el que proporciona Bluetooth. Debido a estas diferencias, las tecnologías no sólo están orientadas a diferentes aplicaciones, sino que no tienen la capacidad de extenderse a otras aplicaciones. Como un ejemplo, Bluetooth debe cargar con frecuencia sus baterías, mientras que uno de los objetivos principales del estándar ZigBee es que un usuario ponga un par de baterías en los dispositivos y se olvide de ellas durante meses o años

Figura 8. ZigBee vs. Bluetooth



Fuente: Autoras del proyecto

Cuando una aplicación de cualquier tipo que use alguna de estas dos tecnologías y colapse o esté pasando por un momento crítico, ZigBee está diseñado para responder con rapidez, mientras que Bluetooth demora más tiempo y podría verse reflejado negativamente en la respuesta a una solicitud urgente.

Tanto Bluetooth como ZigBee pueden emitir en la banda ISM de 2.4 Ghz. pero ZigBee solo ofrece una velocidad de transmisión de entre 20 y 250 Kbps ya que está pensado para transmitir cifras mas modestas (a diferencia de Bluetooth que tiene una velocidad de hasta 1Mbps).

En la figura 12 se evidencian algunas diferencias entre zigbee y bluetooth en cuanto a las aplicaciones a las cuales se enfocan estos dos estándares, los recursos que utilizan y la autonomía para funcionar sin batería. Además deja claro los anchos de banda

usados por cada una de estas tecnologías y su alcance de transmisión.

Dentro del cuadro comparativo también se dejan ver las características de otras tecnologías existentes en el mercado y que se usan para transmisiones inalámbricas como son WiFi y GSM/GPRS, lo cual permite aclarar los rangos de aplicación y los fines para los que se han desarrollado estas tecnologías.

Figura 9. Tabla comparativa entre Bluetooth y ZigBee

	ZIGBEE	GSM/GRPS	WI-FI	BLUETOOTH
ESTÁNDAR	802.15.4	GSM/GRPS	802.11b	802.15.1
APLICACIONES	Monitoreado, control	Transmisión de voz y datos	Web, e-mail, vídeo	Sustituto conexión por cable
RECURSOS DEL SISTEMA	4 kb - 32kb	+ 16 mb	+ 1 mb	+ 250 kb
AUTONOMÍA (EN DÍAS)	100 -+ 1.000	1 - 7	5	5
ANCHO DE BANDA (KB/S)	20 -250	64 -+ 128	+11.000	720
ALCANCE DE TRANSMISIÓN	1 -+ 100	+1.000	1 - 100	1 -+ 10

Fuente: Adaptada por las autoras de ZigBee Alliance (www.zigbee.org)

Finalmente se puede concluir que cada una de las tecnologías desarrolladas para implementarse en comunicaciones inalámbricas tiene funcionalidades distintas y sus campos de aplicación varían de acuerdo a sus características técnicas, por lo que se hace importante conocer a fondo cada estándar para determinar cual es el más apropiado para el desarrollo de un proyecto o aplicación específico teniendo en cuenta las necesidades del problema.

4. METODOLOGÍA

Después de realizarse en el capítulo 2 una pequeña introducción a las metodologías más comunes en el desarrollo de software, que mediante un conjunto de actividades a seguir muestran una guía que permite obtener un producto de calidad, se establecen los parámetros que se seguirán en el desarrollo de REDSI.

El desarrollo de la aplicación requiere dividir el trabajo en pequeñas partes o etapas y por esto se usó el criterio del Proceso de Desarrollo Unificado (RUP) descrito anteriormente, que permite aplicar las pautas descritas por la metodología que mejor se adapten a las necesidades o implementar ciertos cambios para mejorar la práctica del desarrollo del software. Cada requisito corresponde a una iteración que finalmente representa un incremento del producto (avance) y como complemento al proceso de desarrollo unificado se usan algunas pautas del modelo en cascada evidenciado en el Análisis, Diseño del Sistema, Codificación, Pruebas y Mantenimiento. El uso del modelo en cascada aporta orden a cada etapa y establece claridad en cada requisito, sin obligar a que sean totalmente rígidos pues es de anotar que cada etapa propuesta con el RUP maneja implícitamente el control de cambios y la presentación de cada avance del software (que en el modelo en cascada no se aprecia sino al final del desarrollo).

Con esta adaptación de la metodología buscada para el desarrollo de esta tesis, se dividió la aplicación web en mini-tareas para reducir tiempos y costos al poder identificar en cada iteración los posibles errores y actualizaciones en los requerimientos del cliente

y así, poder encontrar los problemas en etapas tempranas del desarrollo y no al final del producto cuando intentar modificar algo acarrearía demoras injustificadas y poco recomendadas.

Esta metodología de desarrollo utiliza UML (lenguaje de modelado unificado) para representar todos los esquemas del software aportando eficiencia al desarrollo en términos de calidad, tiempos y costos. Estos diagramas UML se observan con detenimiento en el capítulo 5 del libro.

Como se describe la metodología RUP, a continuación se analizan las fases de Inicio, Elaboración, Construcción y Transición planteadas e implementadas durante el desarrollo de la aplicación REDSI.

4.1 FASE DE INICIO

El objetivo de esta fase es identificar y justificar el desarrollo del proyecto a partir del análisis de requisitos. Para complementar esta etapa se hizo un análisis del problema y planteamiento de la solución (ver capítulo 5 del libro).

4.1.1 Análisis de Requisitos

En esta fase se realizaron entrevistas con los integrantes del grupo CPS (clientes y usuarios del sistema final) para desarrollar una descripción de producto final y obtener claridad en las necesidades de la aplicación web. Los requisitos también se lograron obtener del estudio del estado del arte para ofrecer un producto acorde con la actualidad en la rama de los sensores inalámbricos.

En la siguiente tabla se muestra una aproximación a los requerimientos de la aplicación web:

Tabla 4. Requerimientos de la Aplicación Web

<p>TOPOLOGÍA</p>	<p>El sistema debe tener una vista gráfica de la topología usada en el galpón con la posibilidad de crear una nueva o modificar la existente, según sea el caso.</p>
<p>GRÁFICAS</p>	<p>El sistema debe presentar gráficamente los datos capturados por los sensores inalámbricos o nodos ubicados en el galpón.</p>
<p>ADMINISTRACIÓN DE USUARIO</p>	<p>El sistema debe diferenciar 2 tipos de usuarios: Administrador y Usuario, cada uno con sus permisos y restricciones. Además de contar con la seguridad mínima de validación de dichos usuarios.</p>
<p>REGISTRO DE MORTANDAD DE AVES</p>	<p>El sistema debe contener un formulario para registrar diariamente (o según convenga el administrador de la red) las pérdidas de aves en el galpón.</p>
<p>INFORMACIÓN DE LOS NODOS</p>	<p>El sistema debe presentar una información actualizada del estado de los nodos que conforman la red.</p>
<p>CONSULTAS</p>	<p>El sistema debe permitir al usuario realizar consultas sobre los parámetros de interés.</p>
<p>ESTADÍSTICAS</p>	<p>El sistema debe permitir obtener información estadística descriptiva,</p>

	<p>(tomando medidas como la media, moda, mediana) que brinde información valiosa para la toma de decisiones. Además de poder disponer de esta información para exportarla y manejarla fuera de la aplicación y así ampliar los análisis que se requieran.</p>
--	---

Fuente: Autoras del proyecto

4.1.2 Diseño

Para llevar a cabo la fase de diseño y cumplir con los requisitos establecidos por el usuario se partió de los diagramas UML para contribuir al entendimiento global de la herramienta. En éste momento ya se tiene claridad de la mayor parte de los requisitos y de los riesgos que obstaculicen la consecución de los objetivos; además de tener identificado el lenguaje de programación que puede resultar útil, los sistemas operativos, las tecnologías de interfaz de usuario y demás componentes que interfieran en el desarrollo de la aplicación.

4.2 FASE DE ELABORACIÓN

En esta fase el objetivo es buscar claridad en la arquitectura del sistema, es decir la estructura o la organización que permite al desarrollador y al usuario tener una visión común del sistema completo, involucrando los aspectos más significativos sobre cómo tiene que ser construido y en qué orden el sistema; todo esto basado en la calidad del software (rendimiento, reducción de costos, tiempos, etc.)

4.2.1 Análisis

Para esta etapa ya se ha analizado el dominio del problema y se ha complementado la visión del sistema, obteniendo los inicios de la arquitectura (para esto, se han identificado los casos de uso más significativos).

Entre los resultados obtenidos de esta etapa están la identificación de por lo menos el 80% de los casos de uso (todos los actores identificados y una gran parte desarrollados), la inclusión de requisitos adicionales no funcionales, la descripción inicial de la arquitectura y un prototipo de la misma.

4.2.2 Diseño

En esta fase se adquiere una comprensión en detalle (brindada por el análisis) de los requisitos y su modelo básico como punto de inicio para las fases siguientes. Se diseñan e implementan los aspectos significativos del sistema, casos de uso, interfaces y subsistemas, que son los aspectos relevantes de la vista de la arquitectura.

4.2.3 Implementación

En esta etapa se desarrolla la arquitectura y se define el sistema como un todo, asignando componentes ejecutables que se utilizarán en la siguiente fase.

4.3 FASE DE CONSTRUCCIÓN

En esta fase se realizan las iteraciones necesarias para la consecución del producto; se refinan (si es necesario) el análisis y diseño de algunos casos de uso. Todo esto para obtener operabilidad del producto software, implementando, integrando y probando todos los componentes y características definidos con los requisitos.

Al finalizar esta fase se deben tener los modelos completos y la arquitectura en una versión aceptable del producto.

4.3.1 Análisis

Partiendo de los requisitos obtenidos de la fase de Elaboración se analizan principalmente los encaminados a desarrollar la interfaz del usuario ya que ayuda a comprender la interacción entre el sistema y el usuario. Con esto se obtiene finalmente un prototipo operativo del software. En esta etapa se debe tener detalladamente el 100% de los diagramas de casos de uso identificados y listos para la etapa de diseño e implementación.

4.3.2 Diseño

Partiendo de los diagramas de caso de uso que aún no se han diseñado, se complementan para presentar la versión ejecutable inicial del sistema.

4.3.3 implementación

El objetivo de esta fase es desarrollar paso a paso (iteración a iteración) los casos de uso establecidos y detallados al 100% de las fases anteriores; realizar pruebas donde determine resultados satisfactorios o desfavorables y finalmente entregar la versión terminada de la herramienta.

En pocas palabras es la fase que implementa el código desarrollado, partiendo de los prototipos y realizando correcciones (iteraciones) para obtener el sistema final.

4.3.4 Pruebas

En esta etapa del desarrollo, las pruebas toman su importancia, pues en las fases anteriores se iba probando los diagramas UML y los pequeños desarrollos pero en esta fase se prueba el correcto funcionamiento de los módulos del sistema para validar su operabilidad. Inicialmente se pobló la base de datos con información ficticia utilizando un generador de datos aleatorios y después con datos obtenidos directamente desde el escenario de la granja; para terminar las pruebas con la información obtenida de los nodos conectados al gateway.

4.4 FASE DE TRANSICIÓN

Con esta fase se busca validar que el producto esté listo para ser entregado al usuario, que la documentación requerida esté completa y que la capacitación básica del manejo del sistema (instalación, configuración y uso) se lleve a cabo; todo esto para conseguir un producto que deje satisfecho al usuario y que cumpla con los requisitos esperados por el.

La única etapa evidenciada es la implementación pues se debe hacer entrega y dejar operando el sistema, sin embargo es la etapa más larga en el ciclo de vida del proyecto. Si es necesario, se deben hacer ajustes ya sea para corregir errores de operabilidad que se presenten o para introducir alguna mejora resultante del trabajo por parte del usuario sobre la aplicación.

5. ANÁLISIS Y DISEÑO DE LA APLICACIÓN WEB

Una de las industrias de mayor crecimiento en la última década es la industria avícola nacional, que creció 9.57%. La producción de carne de pollo pasó de 762 mil toneladas en el 2006 a 850 mil toneladas en el 2007. [13]

Sin embargo, a pesar de su florecimiento, la industria se encuentra ante expectativas inciertas de crecimiento y la latente posibilidad de la firma y aprobación del TLC que se negocia con Estados Unidos, hace temer el posible fracaso de la industria avícola debido a la falta de competitividad en desarrollo, tecnología y recursos frente a la industria extranjera. Ante este panorama, los inversionistas han prendido las alarmas y están estudiando todas las posibilidades de salir de esta encrucijada bien librados.

Para optimizar los resultados en la industria avícola, es necesario manejar entornos adecuados, que proporcionen condiciones ambientales óptimas para el desarrollo de esta actividad económica.

El desempeño productivo del pollo de engorde se ve afectado por factores ambientales como la temperatura y la humedad relativa. Estos factores con valores por encima o debajo de unos rangos normales producen estrés calórico en los animales y los lleva a reducir el consumo de alimentos para minimizar la cantidad de calor generado por la digestión y el metabolismo energético, obteniendo como resultado final, reducción del crecimiento, de la eficiencia en la conversión de alimentos y mortalidad [20].

De esta problemática nace la propuesta de este proyecto, de la necesidad que surge en

la industria avícola de optimizar sus sistemas de producción y ejercer control sobre las variables que afectan el proceso de cría y engorde de las aves.

La idea es incluir la naciente tecnología inalámbrica, en el desarrollo de la actividad avícola con el fin de ejercer un control más específico sobre todas estas variables que afectan la producción.

El grupo de investigación CPS, pretende hacer un análisis para encontrar una topología adecuada e implementar una red de sensores inalámbricos en los galpones de las granjas avícolas, de manera que se puedan tomar medidas precisas de temperatura, humedad, intensidad luminosa y radiación solar, para luego capturarlas y enviarlas inalámbricamente a un servidor de datos.

Como aporte de este trabajo de grado a la industria avícola se diseñó e implementó una aplicación web (REDSI), una base de datos y una serie de algoritmos para leer y almacenar los datos capturados por los nodos inalámbricos en la base de datos. A través de la aplicación web se tiene acceso a toda la información obtenida de esta red de sensores y su interfaz permite al usuario (en este caso la industria avícola) hacer un análisis de los datos, tomar decisiones e implementar soluciones que optimicen la producción, ya sean en infraestructura o automatización de procesos.

Esta aplicación también permitirá mantener un historial de datos, generar consultas y reportes a partir de la información y sacar estadísticas de los datos, además de cualquier otro tipo de manipulación deseada sobre la información almacenada en la base de datos.

Este sistema propuesto, desarrollado e implementado durante el desarrollo del proyecto de grado, es una solución al problema encontrado y planteado inicialmente, permitiendo ejercer un control sobre las variables que afectan el sistema estudiado mediante la captura, el almacenamiento y posterior análisis detallado de los datos que se obtienen

de las medidas tomadas por los sensores inalámbricos instalados en los galpones de las granjas avícolas. La aplicación Web actúa como interfaz entre el usuario y la red de sensores inalámbricos.

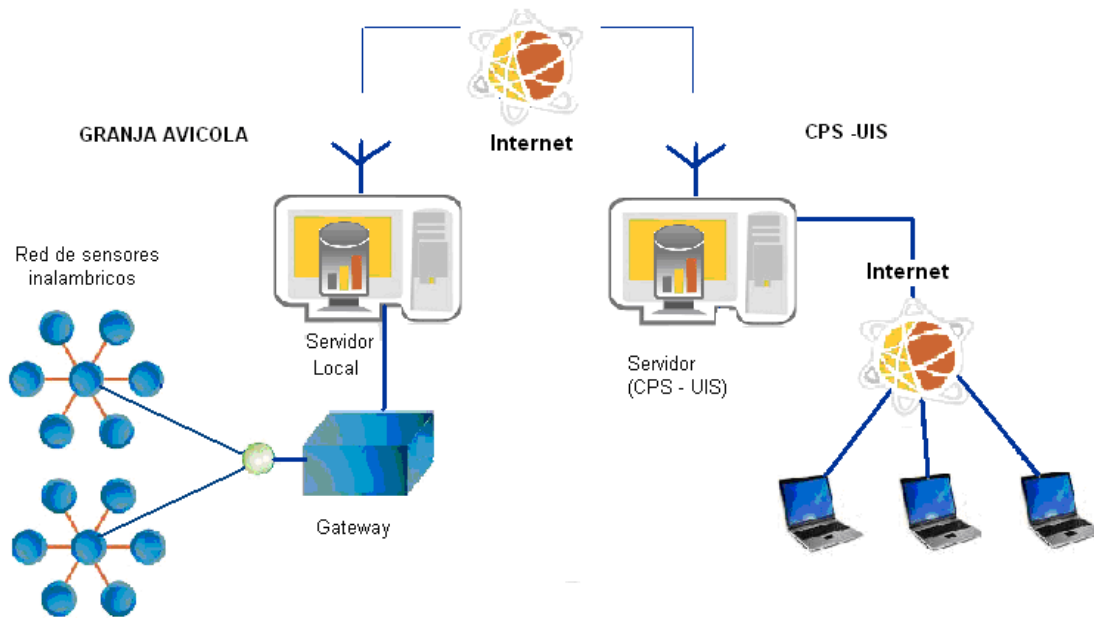
5.1 DISEÑO Y DOCUMENTACIÓN DE LA APLICACIÓN WEB

El modelo completo del sistema planteado en el proyecto de investigación propuesto por el grupo CPS pretende implementar una red de sensores inalámbricos en un galpón de una granja avícola de la región, con el fin de medir las variables que afectan la producción de aves por ciclo elevando a niveles inaceptables para la economía de esta industria, la mortandad de las aves durante su proceso de cría, levante y engorde.

En la granja avícola se adecuará un espacio para ubicar el servidor web local que contará con una base de datos, la aplicación web REDSI y los algoritmos desarrollados para capturar y almacenar los datos. También se instalara un gateway con un nodo concentrador conectado recibiendo datos desde la red de sensores inalámbricos. A su vez este servidor estará conectado a un modem celular para poder establecer comunicación y enviar los datos al servidor remoto ubicado en las instalaciones de la universidad industrial de Santander y que es al que accederán los usuarios desde Internet, como se ve en la figura a continuación.

El servidor web, cuenta además con una interfaz gráfica que permite la interacción entre los usuarios y la información obtenida por los nodos inalámbricos. Esta interfaz permite almacenar, consultar y manipular dicha información.

Figura 10. Modelo General del Sistema

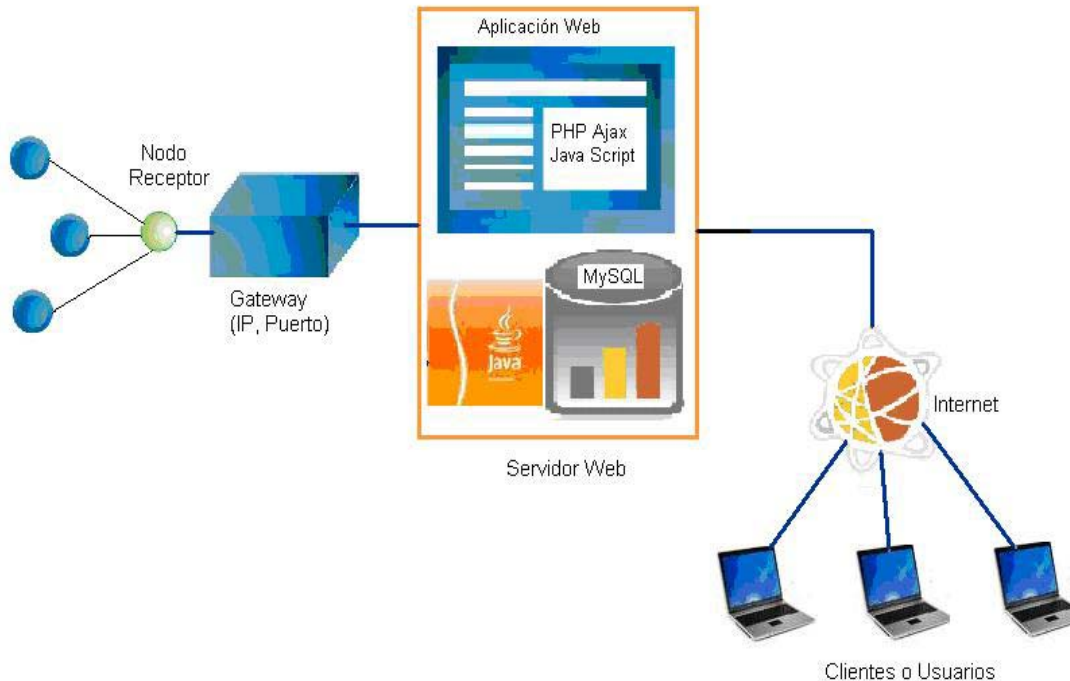


Fuente: Autoras del proyecto

Al término de esta tesis de grado, el dispositivo final que se implementará en los galpones de las granjas avícolas se encuentra aun en proceso de desarrollo, ya que se requiere de un contenedor para los nodos inalámbricos con un diseño especial que contemple y evite las condiciones hostiles del galpón donde finalmente deben quedar ubicados.

El proceso de desarrollo, pruebas e implementación del sistema propuesto en esta tesis de grado, se hizo con un nodo inalámbrico funcionando como nodo concentrador, conectado a un gateway con una IP pública y un puerto asignado y un par de nodos funcionando con pila y transmitiendo inalámbricamente en un ambiente de prueba (laboratorios de electrónica) en la UIS. Además se adecuo un equipo como servidor web en la oficina del grupo CPS, y se le instaló la base de datos, los algoritmos de captura de datos y la aplicación web REDSI, desde donde se puede acceder vía Internet a la información del gateway como se ilustra en la figura 14.

Figura 11. Modelo sistema implementado para la tesis de grado



Fuente: Autoras del proyecto

Para propósitos del sistema final, que es almacenar la información recopilada por los motes que se determinen en la red, y suponiendo una lectura por nodo cada 5 minutos (tiempo que puede variar dependiendo del administrador del sistema), durante los 45 días que dura el ciclo de levante de los pollos, se obtuvieron datos muy aceptables en cuanto a capacidad de almacenamiento (para el caso de backups) y de tiempos de respuesta en las consultas; confirmando la decisión de trabajar con mysql.

5.1.1 Base de Datos

El diseño de la Base de Datos de esta tesis de grado es de tipo estático de sólo lectura, seguro y escalable, donde no se aplica el modelo relacional. Las bases de datos estáticas son utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar su comportamiento a través del tiempo,

realizar proyecciones y tomar decisiones.

Por las características del software de bases de datos utilizado para el desarrollo de esta tesis de grado (MySQL), se puede asegurar que la base de datos funciona en diferentes plataformas, Windows, linux, mac, entre otras. Además, MySQL cuenta con un sistema de privilegios y contraseñas que es muy flexible y seguro y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de las mismas se encripta cuando se conecta con un servidor.

En cuanto a la escalabilidad y los límites, MySQL soporta grandes bases de datos con hasta 50 millones de registros, y se conocen casos registrados y comprobados por Sun Microsystems, en donde se usó este software de base de datos con 60.000 tablas y cerca de 5.000.000.000.000 de registros.

Además, el sistema de indexación incluido en este software permite usar hasta 64^{14} índices por tabla, lo que permite encontrar rápidamente los registros que tengan un determinado valor de sus columnas. Sin un índice, MySQL tiene que iniciar la búsqueda desde el primer registro y recorrer toda la tabla para encontrar los valores buscados. Por lo tanto los índices son eficaces para acelerar la búsqueda de los datos reduciendo los tiempos de CPU y las operaciones de lectura/escritura en disco.

A continuación se muestran las tablas de la base de datos redsi, creada para el desarrollo de esta tesis de grado. En la figura 12, se muestran todas las tablas de la base de datos, en las figuras 13,14,15,16 y 17 se muestran los campos de cada tabla.

Figura 12. Tablas de la Base de Datos

¹⁴ <http://dev.mysql.com/doc/refman/5.0/es/index.html>

```
mysql> use redsi
Database changed
mysql> show tables;
+-----+
| Tables_in_redsi |
+-----+
| info_nodos      |
| mortandad       |
| nodos           |
| usuarios        |
| variables       |
+-----+
5 rows in set (0.00 sec)
```

Fuente: Autoras del proyecto

Figura 13. Campos de la tabla info_nodos

```
mysql> describe info_nodos;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Registro       | bigint(20)    | NO   | PRI | NULL     | auto_increm |
| IdNodos        | int(11)       | NO   |     |          |            |
| Temperatura    | decimal(5,2)  | NO   |     |          |            |
| Humedad        | decimal(5,2)  | NO   |     |          |            |
| Intensidad_Luminosa | decimal(10,2) | NO   |     |          |            |
| Radiacion_Solar | decimal(5,2)  | NO   |     |          |            |
| Tiempo         | timestamp     | YES  |     | CURRENT_TIMESTAMP |            |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.02 sec)
```

Figura 14. Campos de la tabla mortandad

```
mysql> describe mortandad;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| IdMortandad    | int(11)       | NO   | PRI | NULL     | auto_increment |
| Fecha          | date          | NO   |     |          |            |
| Cantidad       | int(11)       | NO   |     |          |            |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.05 sec)
```

Figura 15. Campos de la tabla nodos

```
mysql> describe nodos
-> ;
```

Field	Type	Null	Key	Default	Extra
IdNodos	int(11)	NO	PRI	NULL	auto_increment
posx	int(11)	NO			
posy	int(11)	NO			
FechaActivacion	datetime	NO			

4 rows in set (0.02 sec)

Figura 16. Campos de la tabla usuarios

```
mysql> describe usuarios;
```

Field	Type	Null	Key	Default	Extra
Cedula	varchar(64)	NO	PRI		
Password	varchar(64)	NO			
Email	text	NO			
Nombres	text	NO			
Apellidos	text	NO			
Telefono	int(11)	NO			
Cargo	text	NO			
Tipo	varchar(20)	NO		Usuario	
Habilitado	tinyint(4)	NO		0	

9 rows in set (0.03 sec)

Figura 17. Campos de la tabla variables

```
mysql> describe variables;
```

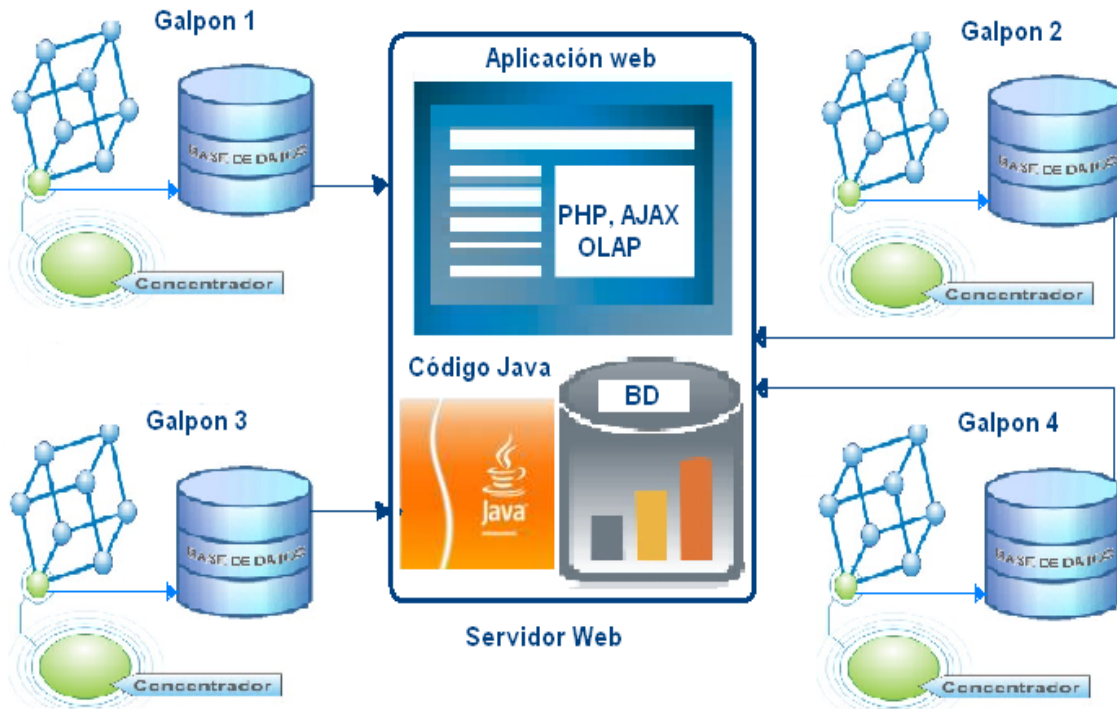
Field	Type	Null	Key	Default	Extra
IdVariable	int(11)	NO	PRI	NULL	auto_increment
Nombre	text	NO			
Min	double	NO			
Max	double	NO			
Unidad	text	NO			

5 rows in set (0.02 sec)

Fuente: Autoras del proyecto

El modelo general que se presenta a continuación contempla el escenario a futuro, con la creación de redes de sensores inalámbricos en varios galpones de una misma empresa monitoreados centralmente desde un servidor principal al que llegan todos los datos capturados por cada una de las redes y almacenados en una base de datos principal.

Figura 18. Modelo General Base de Datos



Fuente: Autoras del proyecto

Durante el desarrollo de este proyecto se diseñó y creó un sitio Web con una interfaz de usuario que permite la interacción entre los usuarios y la información que se almacena en la base de datos obtenida de la información tomada por la red de sensores inalámbricos. A través de esta aplicación el usuario puede monitorear, consultar y hacer análisis de la información con el fin de optimizar el desarrollo de ciertas tareas o actividades en este caso específico, el fin es optimizar los procesos de cría y engorde de aves.

Esta aplicación fue pensada y desarrollada como base de una aplicación más robusta contemplando a futuro ambientes reales y no solo los utilizados en esta primera fase del

proyecto de investigación. Permite además, solucionar el problema fundamental de capturar los datos enviados por los sensores inalámbricos o nodos y almacenarlos para posteriormente hacer los análisis correspondientes a los datos y en base a esto dar herramientas a la industria para adoptar medidas que optimicen los procesos de producción ya sea sobre la implementación de nuevas tecnologías, sobre la infraestructura o sobre la automatización de los procesos.

Para que el modelo funcione correctamente es necesario contar con un ancho de banda mínimo de 250 kbps según los requerimientos del Gateway utilizado para establecer la conexión entre la red de sensores inalámbricos y la aplicación web.

5.2 DESCRIPCIÓN DE LA APLICACIÓN WEB POR MÓDULOS

Para iniciar el sistema se solicita que el usuario se registre y luego de que sus datos se validaren por el administrador del sistema, que ingrese a la aplicación usando un nombre de usuario y una contraseña. Inmediatamente después del ingreso se entra a la pantalla de inicio al sistema, y ahí se encuentran los primeros dos módulos de la aplicación, el modulo de topología y el módulo de gráficas.

MÓDULO TOPOLOGÍA Y GRÁFICAS

En el módulo de topología se encuentra la ventana de visualización de la topología de red creada para recrear a escala la red de sensores implementada en un galpón de 100 metros de largo por 12 metros de ancho (medidas tomadas de un galpón perteneciente a Distraves, ubicado en Girón Santander). Aquí se puede crear una topología de red nueva, ingresando el número de nodos que van a conformar la red para luego arrastrar en el panel gráfico los nodos hasta la posición donde se quieren ubicar de acuerdo a su posición real. También se puede modificar la topología existente, añadiendo o eliminando nodos.

En el módulo desarrollado para graficar los datos almacenados en la base de datos, se encuentra un formulario inicial donde se debe escoger la variable a graficar, el rango en tiempo de datos a graficar (la fecha y hora inicial y final) y se debe escoger además si se quiere graficar el promedio entre todos los nodos, si se quiere graficar la información de un solo nodo o si se quiere hacer la grafica comparativa entre uno y otro nodo. Luego de llenar el formulario y enviar los datos, el graficador se encarga de procesar la información y genera una nueva ventana con la gráfica correspondiente y un encabezado que contiene la información del nodo o los nodos que se están graficando y la variable escogida, además de los datos máximos y mínimos encontrados en el rango de tiempo seleccionado.

MÓDULO ADM. USUARIOS

A este módulo solo tiene acceso el administrador del sistema, quien debe encargarse de habilitar los registros de los usuarios y asignarles el tipo de usuario (Administrador o usuario normal), con el fin de definir los permisos a los diferentes módulos de la aplicación. También el administrador en este módulo debe llenar el formulario con los picos máximos y mínimos aceptables para las variables que se están midiendo con la red de sensores, esto con el fin de activar las alarmas cuando los datos leídos se salgan de estos rangos.

MÓDULO REGISTRO MORTANDAD

A través de este módulo el administrador del galpón en la granja, diariamente (o con la frecuencia deseada) puede ingresar un registro con el número de aves muertas a la fecha escogida. Además de eso también se puede consultar el historial de estos registros.

MÓDULO INF. NODOS

En el módulo Inf. Nodos se encuentra una consulta estática con la fecha de activación de los nodos, su identificador, su posición y su estado.

MÓDULO CONSULTAS

En el módulo de consultas se debe llenar un formulario donde se seleccionan los parámetros de los datos a mostrar. Se escoge la variable que se quiere consultar, se decide si se quiere graficar los datos y se escoge el rango de tiempo. Finalmente, se escoge si se quiere mostrar los datos y la gráfica o si se exportan los datos a Excel para desde esta herramienta hacer un análisis más exhaustivo de esta información.

MÓDULO ESTADÍSTICAS

El análisis a los datos que se hace en este módulo corresponde a un análisis estadístico descriptivo. Este análisis es muy básico, pero fundamental en todo estudio y prácticamente es una representación gráfica y en una serie de medidas de los datos tomados.

Este análisis estadístico descriptivo de los datos que se hace dentro de la aplicación web permite escoger en un formulario si se quiere utilizar uno o todos los nodos de la red, seleccionar una variable, una fecha inicial y final, un tipo de medida (porcentaje o numérico), un modo (normal o acumulado), un tipo de gráfica (barra, lineal o área) y un color para dicha gráfica. Luego de enviar esta información el desarrollo encargado de hacer el procesamiento de los datos, genera una gráfica y un encabezado con los datos relevantes del análisis hecho y con cálculos de medidas estadísticas básicas como la moda, la media y la mediana .

5.3 DIAGRAMAS UML

De acuerdo a las necesidades de este proyecto y tomando como base la aplicación web que se está construyendo, se escogieron los siguientes diagramas:

- ✓ Diagrama de casos de uso
- ✓ Diagrama de Actividades
- ✓ Diagrama de Navegación

Para la elaboración de los diagramas de casos de uso y de actividades se hace necesaria la identificación de los actores que interaccionan en el sistema y establecer sus funciones y atribuciones dentro del mismo.

5.3.1 Actores del Sistema

Los actores del sistema son todos los seres humanos u otros sistemas que interaccionan directa o indirectamente con el sistema. Para la aplicación web REDSI tenemos el usuario, el administrador del sistema, el visitante, los motes o nodos inalámbricos y el sistema como tal.

El administrador del sistema dentro de la aplicación se define como el ente encargado de llevar a cabo todas las operaciones de dirección, mantenimiento y control sobre la aplicación web.

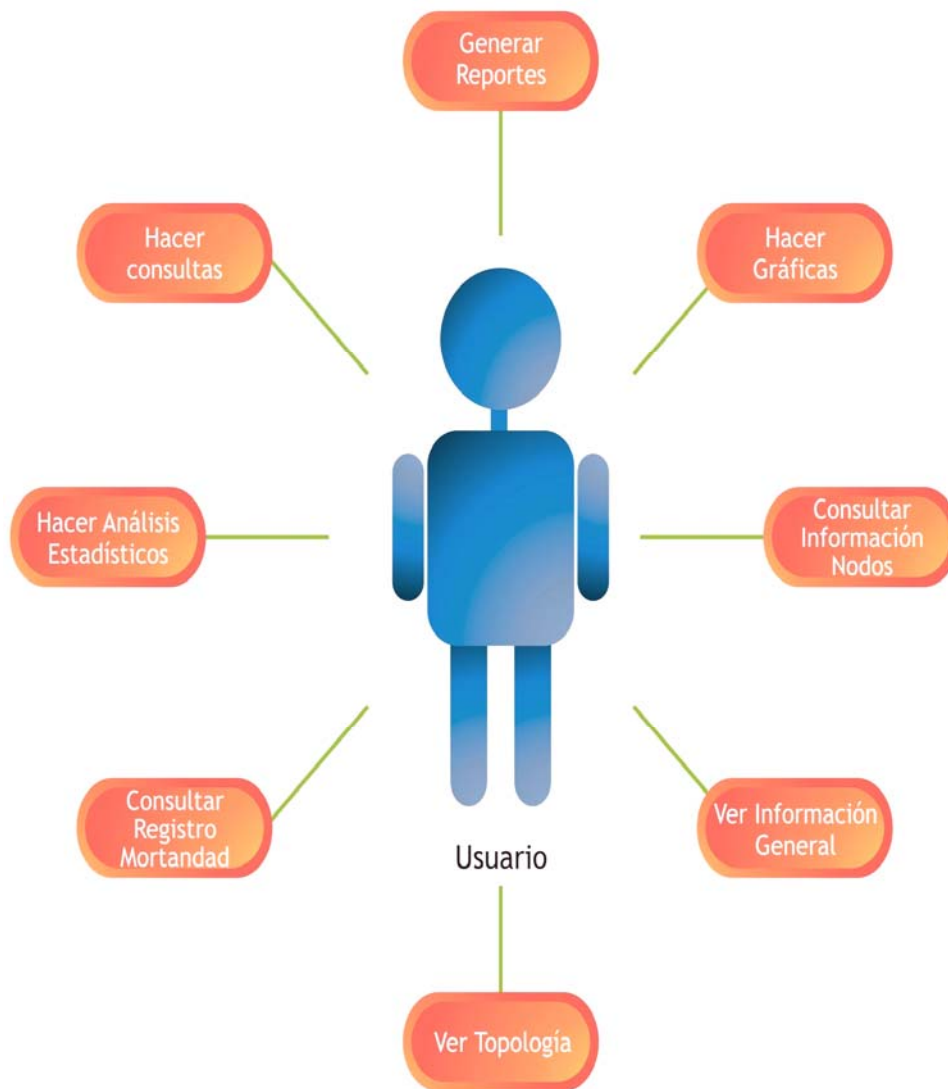
El usuario de la aplicación se define como el individuo al cual el administrador le habilita su usuario y contraseña y le permite acceder restringidamente a la aplicación concediéndole acceso a una cantidad limitada de módulos. Finalmente el visitante es aquella persona que ingresa a la web y solo puede acceder a la información general visible para cualquier individuo, y a pesar de registrarse no es habilitado por el administrador para permitir su ingreso al resto de los módulos.

En la figura 22 se ve un típico diagrama de caso de uso en donde se muestra de forma gráfica los permisos del usuario para acceder a los módulos generar reportes, hacer consultas, hacer análisis estadísticos, consultar registro mortandad, ver topología, ver información general, consultar información nodos y hacer gráficas.

De aquí en adelante, para cada actor del sistema, se describirán sus permisos de acceso a los módulos de la aplicación web a través de representaciones gráficas como este diagrama de actividades que se muestra a continuación.

USUARIO: El usuario de la aplicación web REDSI tiene permisos para realizar todo tipo de consultas y reportes y por ende tiene la potestad de manipular la información que obtiene para hacer análisis adicionales.

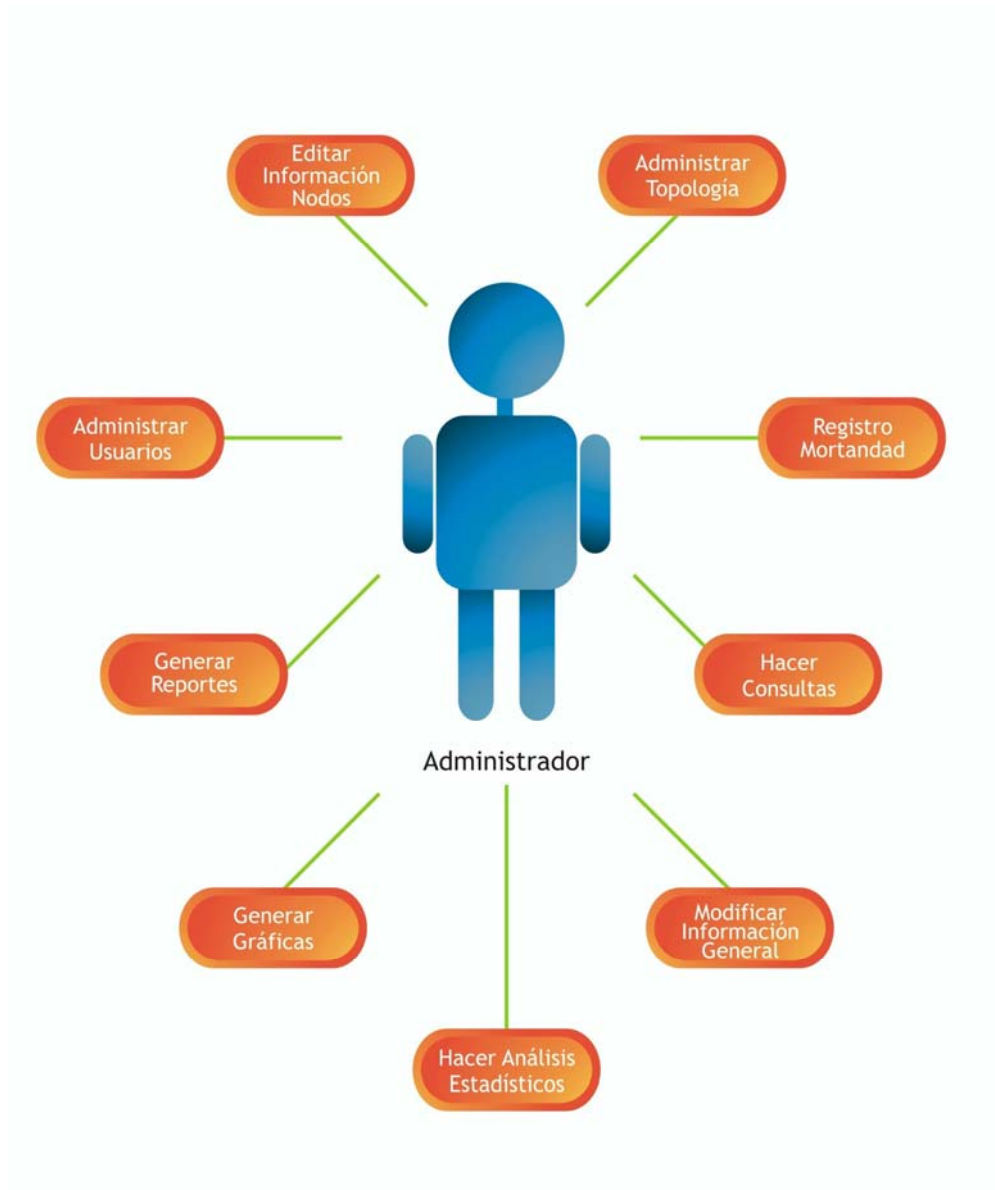
Figura 19. Caso de uso Usuario



Fuente: Autoras del proyecto

ADMINISTRADOR: El administrador del Sistema tiene acceso a todos los módulos de la aplicación y es el encargado de alimentar, modificar y consultar la Base de Datos además de ejercer una supervisión y control sobre los datos que se obtienen de los sensores y tomar medidas o decisiones en base a esa información.

Figura 20. Caso de uso Administrador.



Fuente: Autoras del proyecto

VISITANTE: El visitante solo podrá acceder a la información general que se publica antes del ingreso como usuario autorizado del Sistema.

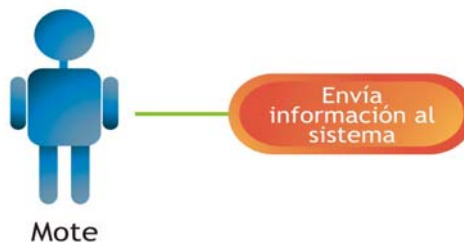
Figura 21. Caso de uso Visitante.



Fuente: Autoras del proyecto

MOTES (NODOS INALÁMBRICOS): Los nodos interactúan con el sistema, enviando la información que capturan sus sensores para que estos datos sean almacenados en la BD y posteriormente analizados.

Figura 22. Caso de uso Nodos Inalámbricos.



Fuente: Autoras del proyecto

SISTEMA: El sistema recibe la información enviada por los nodos y la información o los cambios que el administrador hace en el sistema y lo guarda, actualizando la Base de Datos y la interfaz de la aplicación que se alimenta de ella.

Figura 23. Caso de uso Sistema.



Fuente: Autoras del proyecto

5.3.2 Diagrama de Actividades

ACTOR: Administrador

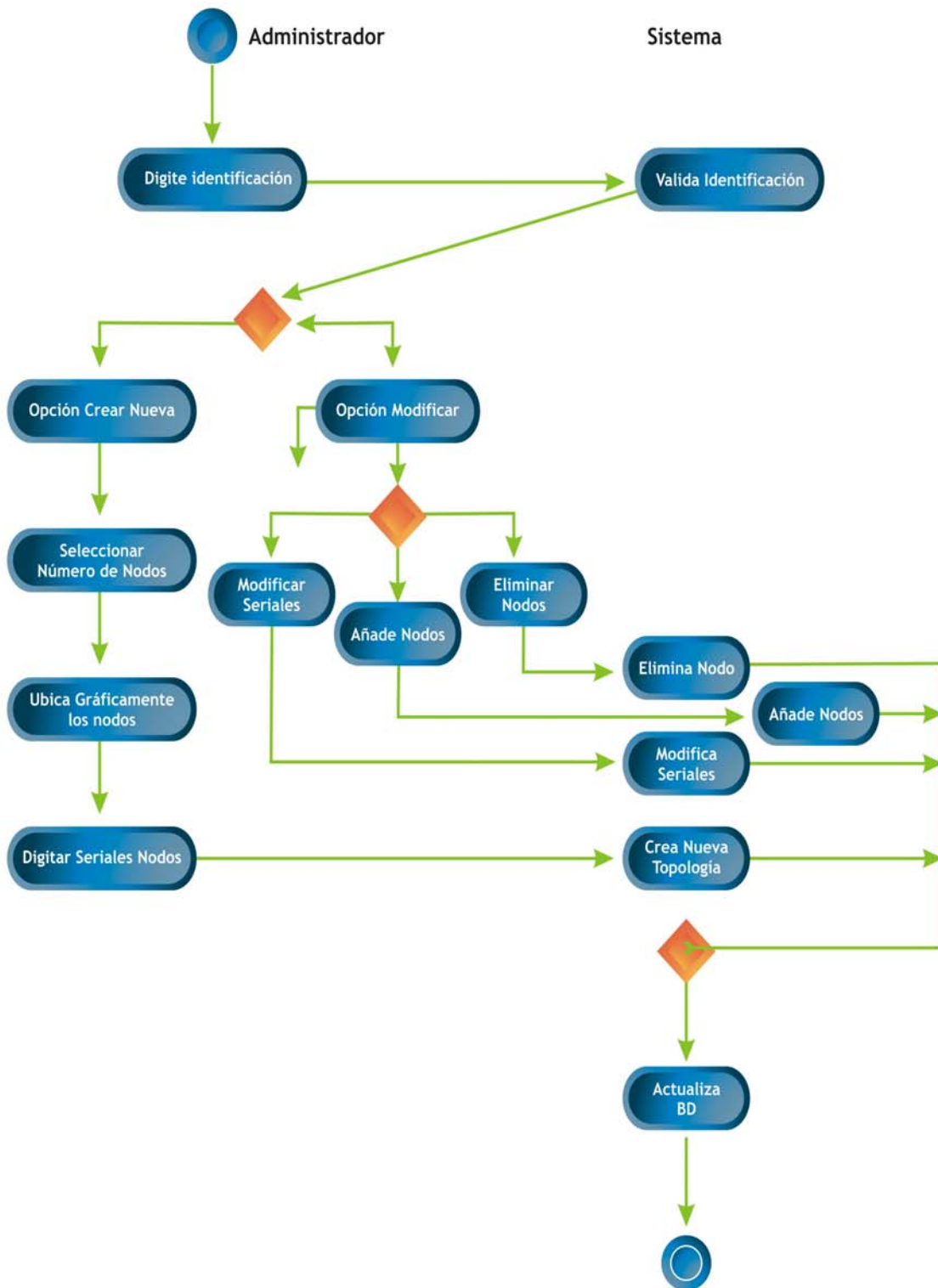
CASO DE USO: Administrar Topología

En este y todos los diagramas de actividades siguientes se describen las acciones que toman los actores del sistema en cada caso de uso.

En el diagrama de actividades que se muestra en la figura 20, se describe gráficamente la secuencia de acciones de la interacción entre el usuario y el administrador. Este diagrama en específico evidencia las actividades para administrar la topología.

Figura 24. Diagrama de Actividades Administrar Topología.

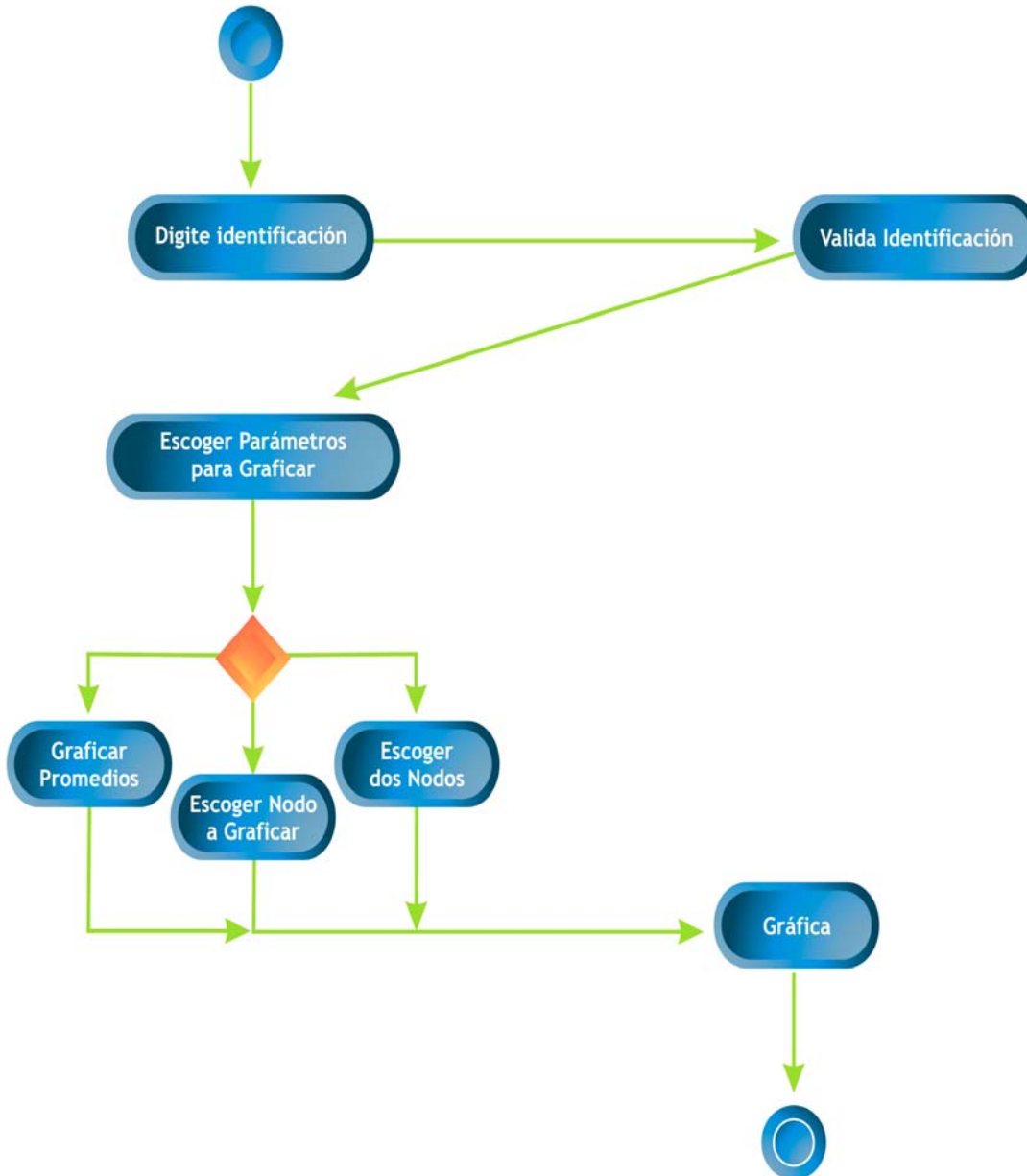
Fuente: Autoras del proyecto



ACTOR: Administrador, Usuario

CASO DE USO: Generar Gráficas

Figura 25. Diagrama de actividades Generar Graficas.

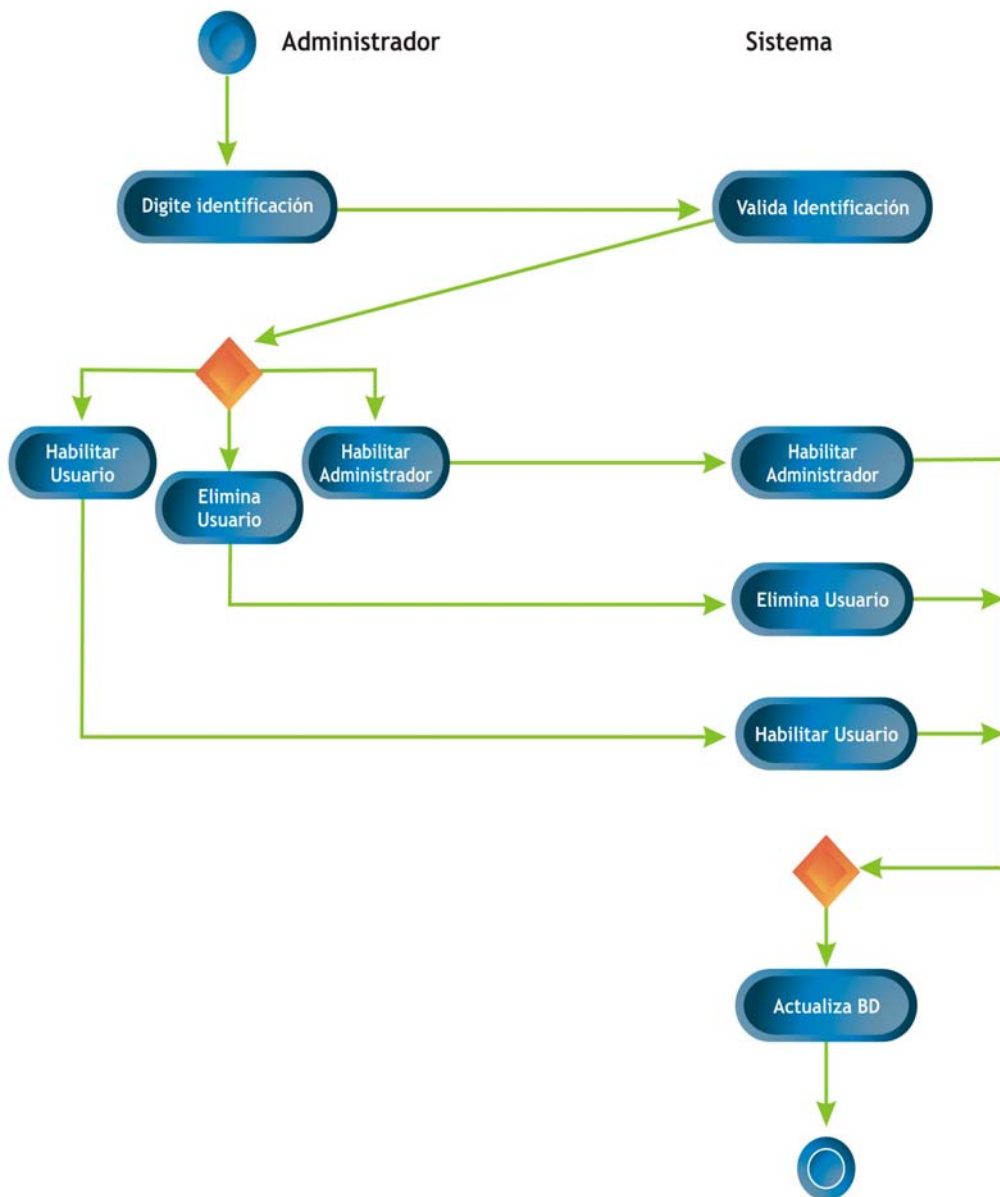


Fuente: Autoras del proyecto

ACTOR: Administrador

CASO DE USO: Administrar Usuarios

Figura 26. Diagrama de actividades Administras usuarios

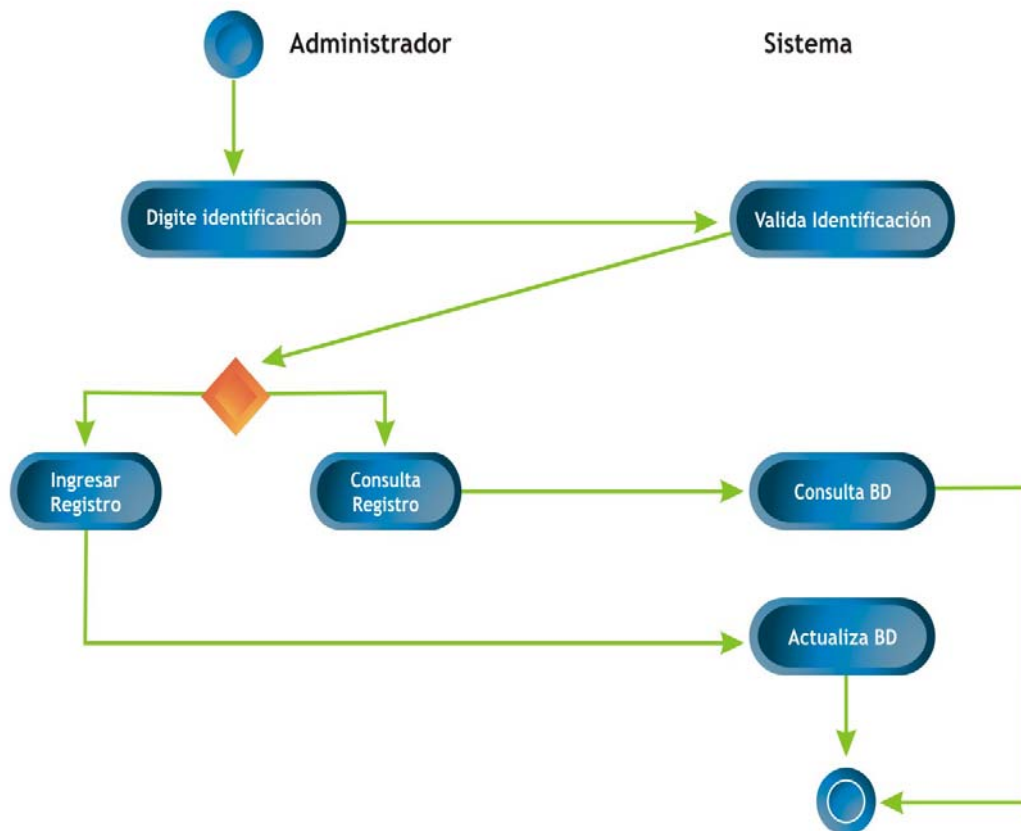


Fuente: Autoras del proyecto

ACTOR: Administrador

CASO DE USO: Registrar Mortandad

Figura 27. Diagrama de actividades Registrar Mortandad.

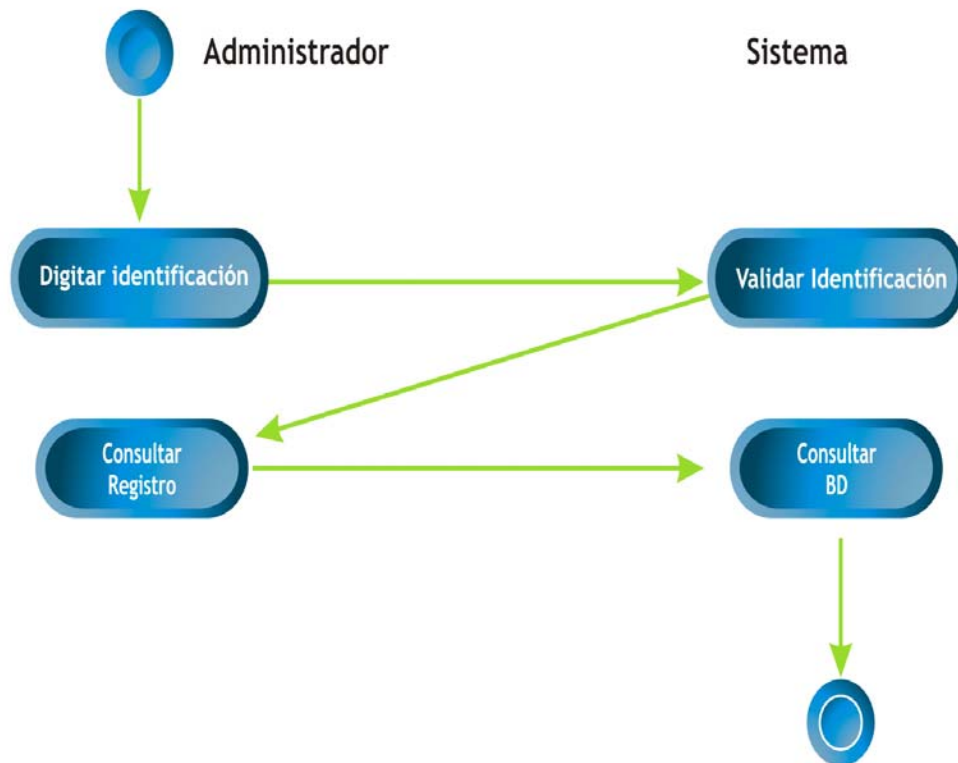


Fuente: Autoras del proyecto

ACTOR: Usuario

CASO DE USO: Consultar Mortandad

Figura 28. Diagrama de actividades Consultar Mortandad





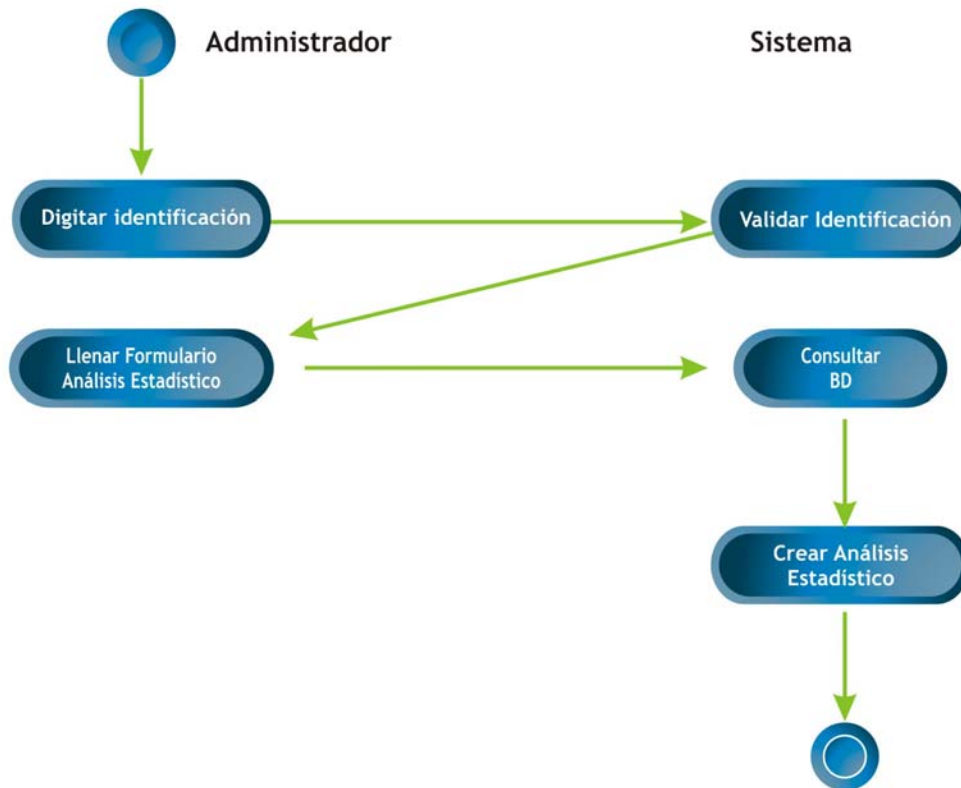
Fuente: Autoras del proyecto

ACTOR: Usuario

CASO DE USO: Hacer Consultas y Reportes

Figura 29. Diagrama de actividades

Figura 30. Diagrama de Actividades Hacer análisis estadístico



Fuente: Autoras del proyecto

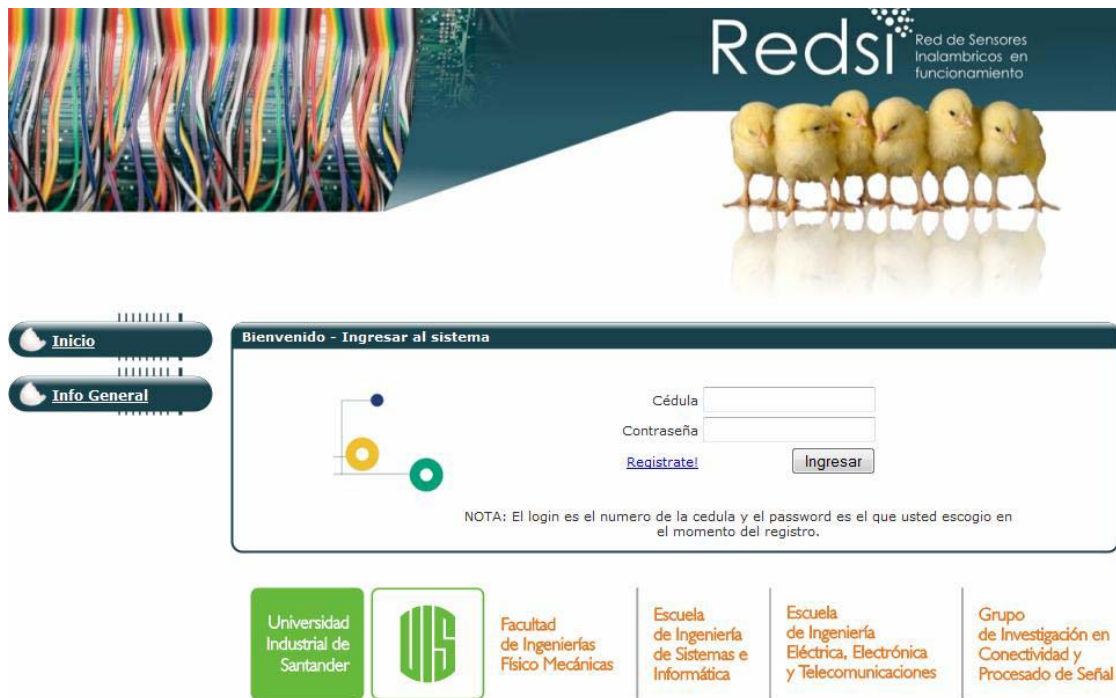
5.3.3 Interfaces del Sistema

Las interfaces de usuario están optimizadas para una resolución de 1024x768 o 1280x1024 para ofrecer mayor campo visual y gran espacio para información. Así mismo se tiene en cuenta el manejo de colores, formas e iconos para familiarizar rápidamente al usuario con la aplicación.

PANTALLA DE BIENVENIDA

Se compone de 3 Zonas o “frames”, el frame superior que contiene un banner con el logo del sistema, el menú principal a la izquierda y el contenido en el frame central.

Figura 31. Página principal REDSI



Fuente: Autoras del proyecto

INGRESO AL SISTEMA

El ingreso al sistema consta de un formulario en el frame de contenido para el ingreso del usuario si éste ya se ha registrado; si aún no lo ha hecho, puede hacerlo a través del enlace Regístrate

Figura 32. Página de Ingreso al sistema REDSI

Bienvenido - Ingresar al sistema

Cédula

Contraseña

[Regístrate!](#)

NOTA: El login es el numero de la cedula y el password es el que usted escogio en el momento del registro.

Fuente: Autoras del proyecto

REGISTRO DE UN NUEVO USUARIO

Consiste en un formulario de registro en donde se solicitan algunos datos personales del usuario a registrar, tales como cedula, que será su usuario a la hora de ingresar a la aplicación, contraseña, nombre y apellido, e-mail, teléfono y cargo dentro de la empresa. Luego del registro el usuario deberá esperar que el administrador del sistema habilite su registro para poder acceder a la aplicación.

Figura 33. Página de ingreso de nuevo usuario (registro)

Inicio

Info General

Registrarse en el sistema

CÉDULA

CONTRASEÑA

NOMBRES

APELLIDOS

E-MAIL

TELEFONO

CARGO

CANCELAR REGISTRARME

NOTA: El sistema diferencia entre Mayúsculas y minúsculas para el password.

Fuente: Autoras del proyecto

MENÚ PRINCIPAL

Una vez el administrador habilite el registro del usuario y este ingrese correctamente al sistema, se activará el menú de acuerdo a los permisos que el administrador haya habilitado para ese usuario específico.

Figura 34. Menú principal REDSI



Fuente: Autoras del proyecto

TOPOLOGÍA Y GRÁFICAS

La topología de la red la pueden ver tanto los usuarios como el administrador, al igual que la opción de generar gráficas, pero la potestad para modificar la topología de red solo la tiene el administrador.

En la opción Gráficas se puede escoger la variable, el tiempo y el tipo de gráfica que se necesita ver para analizar el comportamiento de los parámetros seleccionados.

En la opción topología, se puede ver gráficamente la ubicación de los nodos y los datos que cada nodo está emitiendo en tiempo real.

Figura 35. Menú topología y gráficas

Gráficas

Por favor, seleccione los parámetros a graficar:

Tamaño de la gráfica: Pequeña(720x288)

Variable a graficar: Temperatura

Fecha Inicial: Ver

Hora Inicial: 00h : 00m : 00s

Fecha Final: Ver

Hora Final: 00h : 00m : 00s

Graficar todos los nodos

Graficar sólo éste nodo: 1

Comparar los nodos: 1 y 1

Graficar

Topología **Gráficas**

Por favor espere mientras carga...

10

2

4

9

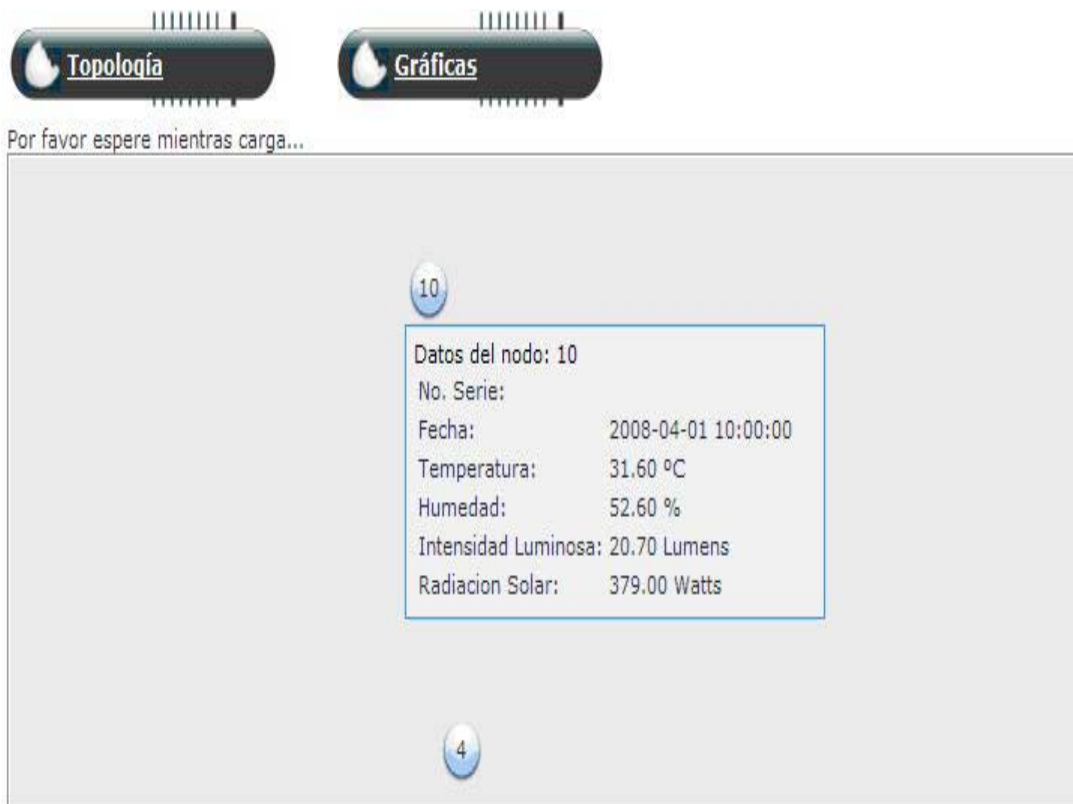
1

Crear Nueva Modificar

Fuente: Autoras del proyecto

La figura 36 muestra el menú que se despliega al dar clic sobre cada uno de los nodos de la topología de red creada por el usuario administrador. Este menú contiene los últimos datos registrados en la base de datos del nodo seleccionado.

Figura 36. Menú desplegable de cada nodo en la topología.



Fuente: Autoras del proyecto

CREAR NUEVA TOPOLOGÍA DE RED

Dentro de Topología se puede crear o modificar la topología establecida. Si lo que se desea es crear, el sistema inicia preguntando con cuantos nodos se desea crear la

nueva Topología de Red.

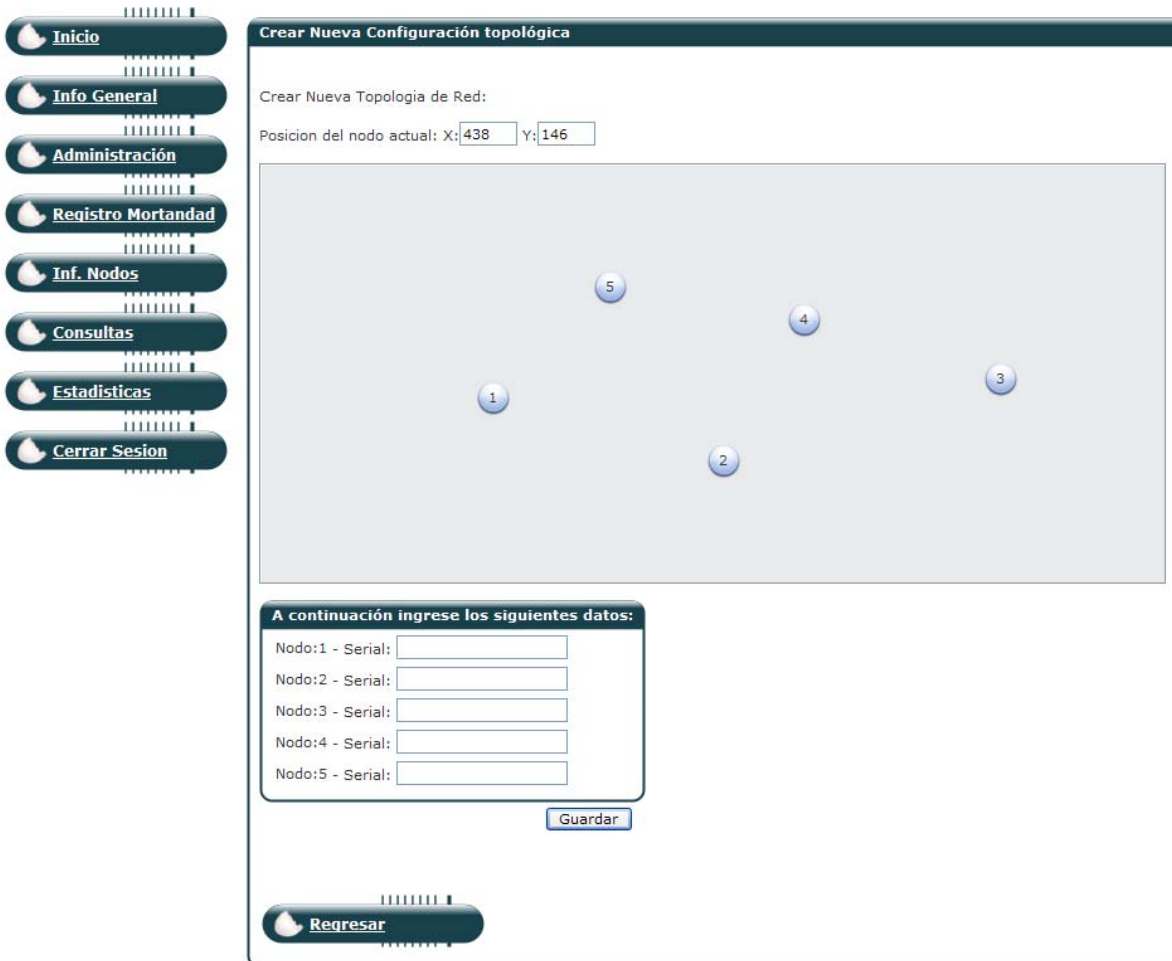
Figura 37. Menú Crear nueva topología



Fuente: Autoras del proyecto

Luego de digitar el número de nodos, se crea la topología y se habilita un formulario gráfico para posicionar los nodos, ya sea arrastrando el óvalo que representa el nodo hasta la posición deseada, o digitando en los recuadros de posición X y Y, las coordenadas exactas de la ubicación deseada para cada nodo.

Figura 38. Vista nueva topología

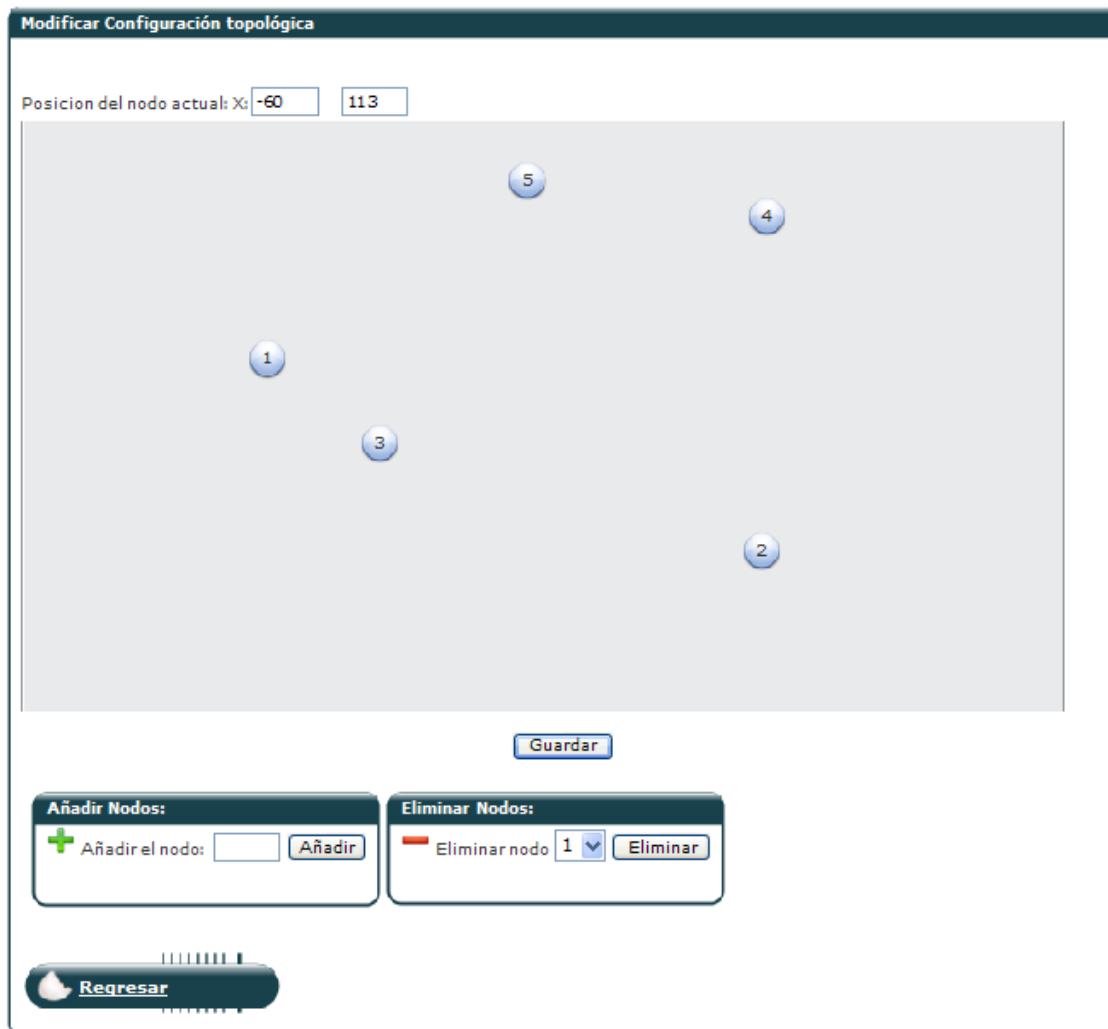


Fuente: Autoras del proyecto

MODIFICAR TOPOLOGÍA DE RED

Si se escoge la otra opción, la que permite modificar la topología, se puede no sólo agregar o eliminar nodos, si no también cambiar de posición los existentes, y modificar su información.

Figura 39. Modificar Topología



Fuente: Autoras del proyecto

ADMINISTRAR USUARIOS

Esta opción sólo se habilita para el administrador y permite habilitar los registros de los usuarios, eliminarlos y darles permisos de administrador. Al momento de que exista un nuevo registro (solicitud de una persona para ingresar al sistema) el Administrador puede habilitarlo o no, y establecer sus privilegios.

Figura 40. Página Administrar usuarios



Administración de usuarios

Nombres	Apellidos	Tipo	Habilitar	Eliminar	Administrar
qwq		Usuario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Administrador		Admin	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Guardar Usuarios

Administración de Variables

Nombre de la variable	Mínimo	Máximo	Unidades
Temperatura	1	5	d
Humedad	2	6	b
Intensidad Luminosa	3	7	c
Radiaci	4	8	d

Guardar Variables

Fuente: Autoras del proyecto

REGISTRO MORTANDAD

El ingreso de registros de los pollos muertos sólo se habilitará cuando el usuario

registrado tenga permisos de administrador, mientras que la consulta al registro de mortandad estará habilitada tanto para usuarios como para administradores.

Se puede ingresar o ver los registros desde el mismo menú. Para registrar solo se debe digitar o seleccionar la fecha y la cantidad de pollos muertos.

Para ver los registros anteriores se pueden filtrar los registros de un solo día o de todos los realizados por orden cronológico.

Figura 41. Registro de mortandad





Fuente: Autoras del proyecto

INFORMACIÓN NODOS

Figura 42. Información de los nodos



Fuente: Autoras del proyecto

CONSULTAS Y REPORTE

A este módulo tienen acceso usuarios y administradores. El formulario permite escoger las variables a consultar, los tiempos y la opción de que se grafique o no, la consulta establecida.

En este menú se puede ver (Consultar) o generar un Reporte de la consulta filtrada, para este último caso, se genera un archivo en Excel con una sencilla representación de los datos consultados para su posterior análisis y manipulación por parte del usuario.

Figura 43. Consultas y Reportes

Consultas

Seleccione los parámetros de los datos a mostrar:

Variables:

- Temperatura
- Humedad
- Intensidad Luminosa
- Radiacion Solar

Tiempo:

- Mostrar solo los ultimos datos registrados
- Mostrar los datos en este periodo de tiempo:

Fecha Inicial: Ver

Hora Inicial: : :

Fecha Final: Ver

Hora Final: : :

Fuente: Autoras del proyecto

ESTADÍSTICAS

Este módulo se habilita para usuarios y administradores. El formulario permite escoger las variables a analizar, los nodos, los tiempos y el tipo de medida estadística a utilizar; además del tipo de gráfico estadístico deseado. Luego genera las respectivas gráficas estadísticas y los datos con la medidas básicas como la media, la moda, la mediana y la desviación estándar.

Figura 44. Estadísticas

Fuente: Autoras del proyecto

CERRAR SESIÓN

Finalmente se sale de la sesión mediante el menú CERRAR SESIÓN que regresa al usuario a la página de Ingreso al sistema.

Figura 45. Cerrar sesión



Fuente: Autoras del proyecto

6. TINYOS Y NES C

6.1 TINYOS

TinyOS [1] es un Sistema Operativo de código abierto, desarrollado por la Universidad de Berkeley útil para trabajar con redes de sensores inalámbricas y diseñado especialmente para los nodos y las aplicaciones de red propias de la comunicación entre ellos; y a su vez, estos nodos están programados en lenguaje NesC. La biblioteca de componentes de TinyOS incluye protocolos de red, servicios distribuidos, drivers para los sensores y herramientas de adquisición de datos. Todas estas utilidades se pueden usar exactamente como están diseñadas o pueden ser modificadas para adaptarlas a una aplicación personalizada.

La actividad de un nodo sensor está basada en los impulsos externos que recibe, es decir, los mensajes radio que recibe de otros nodos, la detección de sonido, de temperaturas fuera del rango habitual que debe generar un mensaje de alerta etc. TinyOS utiliza el concepto de “wiring” que lo que hace es enlazar componentes software

para producir un programa final. Este modelo de programación además pone requisitos sobre cómo deben ser escritos los programas.

6.1.1 Características TinyOS

- ✓ Arquitectura basada en componentes.
- ✓ Capas de abstracción bien establecidas, limitadas claramente a nivel de interfaces, a la vez que se pueden representar los componentes automáticamente a través de diagramas.
- ✓ Amplios recursos para elaborar aplicaciones.
- ✓ Adaptado a los recursos limitados de las motas: energía, procesamiento, almacenamiento y ancho de banda.
- ✓ Operaciones divididas en fases (Split-phase).
- ✓ Dirigido por eventos (Event Driven): reacciona ante sensores y mensajes.
- ✓ Concurrencia de tareas y basada en eventos.
- ✓ Implementación en Nesc.
- ✓ Las interfaces realizan servicios.
- ✓ Interfaces bidireccionales, con comandos y eventos.
- ✓ Los comandos los implementa el proveedor.
- ✓ Los eventos son implementados por el usuario.
- ✓ Un módulo implementa una interfaz.
- ✓ Los componentes proveen y usan interfaces (representado en el código por las etiquetas “provide and use”).
- ✓ Una configuración enlaza las interfaces internas y externas.
- ✓ Una aplicación consiste en una configuración de alto nivel y todos los módulos asociados.

6.1.2 Arquitectura de TinyOS

TinyOS soporta sistemas embebidos en la red, por lo que puede mantener el sistema en estado inactivo o dormido el mayor tiempo posible pero en estado de escucha, esperando por cualquier estímulo que detecte para despertar y realizar los cálculos y tareas para los cuales se han programado los nodos de la red y volver a dormir lo más rápido posible.

Además este sistema operativo se diseñó para ser escalable y poder trabajar con una gran cantidad de nodos. También es compatible con los avances tecnológicos debido a su código abierto y con el fin de trabajar cada vez con dispositivos más pequeños y baratos y que cada vez consumen menos recursos y menos energía.

El núcleo multihilo de TinyOS trabaja a base de dos tipos de estructuras, eventos y tareas. Los eventos actúan cuando se produce actividad en el hardware (motes o nodos inalámbricos) y debe realizarse algún tipo de cálculo. Los eventos pueden interrumpir a las tareas. Las tareas por su parte, utilizan más tiempo para hacer sus cálculos que los eventos y solo se ejecutan cuando la tarea que la precede termina.

La actividad de un nodo o mote, está basada en los impulsos externos que recibe, es decir, la detección de sonido, la detección de una temperatura más alta o más baja de lo habitual y que debe generar un mensaje de alerta que se propague por la red, etc.

La implementación del prototipo planteado para llevar a cabo esta parte de la investigación, requiere de la utilización de dispositivos electrónicos o nodos utilizados para la medición de determinadas variables, conformando una red inalámbrica dentro de un ambiente de prueba, que para este caso es un galpón. El sistema operativo TinyOS, todas sus librerías y aplicaciones se diseñaron para trabajar con los motes y permitir la comunicación entre los nodos de la red. Como se mencionó anteriormente, TinyOS es un sistema operativo hecho en NesC y es precisamente con este lenguaje, con el cual se programan los nodos para que realicen ciertas funciones.

Para empezar a usar TinyOS, es necesario instalar y utilizar la herramienta Cygwin, que emula el entorno Linux en Windows, ya que este sistema operativo recurre a comandos y funcionalidades Linux en su funcionamiento. Cygwin puede estar incluida dentro del paquete instalador de TinyOS 1.x y 2.x o puede ser descargada e instalada independientemente.

Las redes de sensores inalámbricos cuentan con pocos recursos, por lo que se debe hacer una utilización muy eficiente de los recursos disponibles. Estos sensores cuentan con pequeña memoria RAM, poca capacidad de procesamiento e insuficiente espacio disponible para almacenamiento y TinyOS está diseñado para adaptarse a estas necesidades.

6.1.3 Modelo de Componentes

El modelo de componentes de TinyOS, es un modelo de programación característico de los sistemas embebidos. Este modelo está compuesto de un conjunto de componentes, los cuales proveen la base para la creación de aplicaciones.

TinyOS funciona a través de tareas y eventos, los componentes entregan tareas al planificador para que las ejecute. Las tareas programadas deben ser cortas y de necesitar procedimientos grandes, debe dividirse en múltiples tareas.

A continuación se detallan los cuatro elementos que conforman una componente:

- ✓ Manejador de comandos.
- ✓ Manejador de eventos.
- ✓ Un frame¹⁵ de tamaño fijo y estáticamente asignado, en el cual se representa

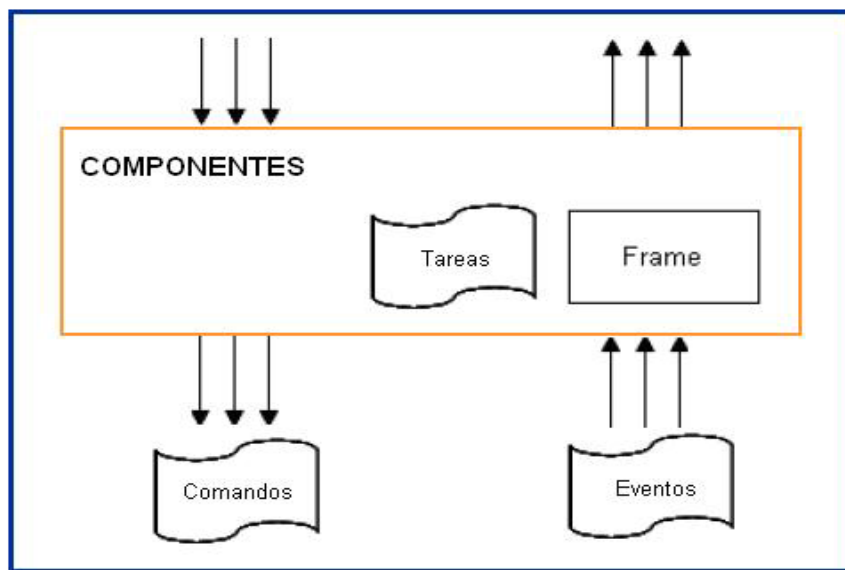
¹⁵ En este caso nos referimos a un "bloque" que proporciona el contexto en cual se ejecuta el programa y se almacenan las variables

el estado interno de la componente.

- ✓ Un bloque con tareas simples.

En la figura a continuación mostramos el modelo de componentes sobre el cual funciona TinyOS.

Figura 46. Modelo de Componentes de TinyOS y su interacción



Fuente: Autoras del proyecto

Los comandos son peticiones hechas a componentes de capas inferiores y son solicitados para ejecutar alguna operación. Hay dos tipos de comando, bifase y no bifase. El primero envía una petición al planificador, quien se encargará de ejecutar la solicitud generando un evento que indique el fin de la operación. En el segundo caso, la operación solicitada por el comando se realizará completamente, sin necesidad de crear un evento.

El frame que indica el estado interno del componente se modifica así como lo hacen los comandos y agrega las tareas al componente para que sean llevadas a cabo.

Los eventos tienen mayor prioridad que las tareas, es por eso que las tareas pueden ser interrumpidas por los eventos de ser necesario. El planificador es el encargado de recibir y ejecutar secuencialmente las tareas, y para esto utiliza el método FIFO (First In – First Out) que quiere decir que las tareas que ingresen primero, serán atendidas inmediatamente y saldrán primero de la cola.

6.1.4 Requerimientos Básicos de Instalación

- ✓ CD-ROM de herramientas de soporte de TinyOS proporcionado por el fabricante.
- ✓ Sistema Operativo: Microsoft Windows (XP, 2000, NT)
- ✓ 1GB o más de espacio libre en el disco.
- ✓ 550 MB o más de espacio libre en el disco C, para el directorio TEMP.
- ✓ Software Terciario:
 - WinZip.
 - Adobe Acrobat PDF Reader, para ver los manuales.
 - Bloc de notas, para ver, modificar y crear aplicaciones en NesC.

Debido a que este sistema operativo se creó inicialmente sobre Unix, para que funcione correctamente en Windows, se hace necesaria la instalación de Cygwin, un simulador de plataformas Unix.

6.1.5 Paquetes Instalados con TinyOS

La instalación de TinyOS se hace con un asistente que durante el proceso, instala los

siguientes paquetes:

- ✓ TinyOS 1.x o 2.x, de acuerdo a la versión que se tenga.
- ✓ Herramientas de TinyOS
- ✓ nesC
- ✓ Cygwin (aparece el icono en el escritorio cuando se instala TinyOS)
- ✓ SurgeView (aplicación para mostrar en pantalla la red de sensores creada)
- ✓ Herramientas de soporte
- ✓ Java 1.4 JDK y Java COMM 2.0
- ✓ Graphviz
- ✓ Herramientas AVR

La instalación del sistema operativo con todos los paquetes adicionales que trae, puede demorar entre 15 minutos y 1 hora, dependiendo de la velocidad del computador donde se este instalando.

En el anexo de instalación se describe paso a paso este proceso y el directorio donde se encuentran los instaladores o las direcciones web para su respectiva descarga.

6.2 NES C

NesC (network embedded system C) es un lenguaje de C basado en componentes. Cada componente de NesC se parece a un objeto de C++ o java, con la diferencia que los componentes hacen referencia a funciones y variables de nivel local, en C++ o java, los objetos lo hacen a nivel global. Esto significa que además de declarar las funciones que implementa, un componente debe declarar las funciones que llama.

Las aplicaciones en NesC están construidas por Interfaces, módulos y configuraciones.

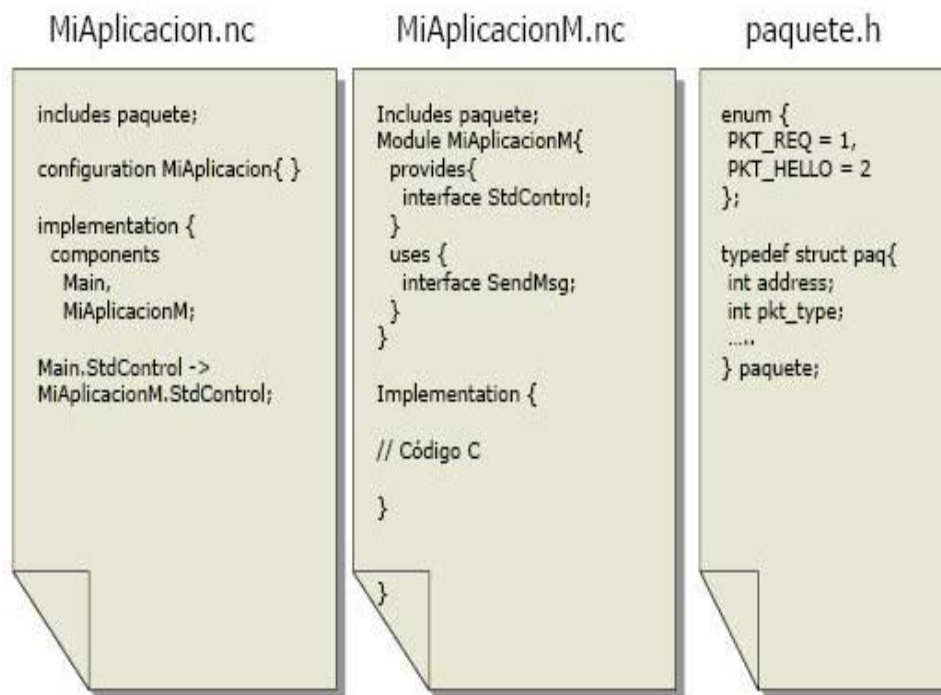
- ✓ Interfaces: Son el único acceso al componente (el componente proporciona y

usa interfaces). Especifican la funcionalidad a quien lo vaya a utilizar y especifican los comandos que se pueden llamar y los eventos que deben ser implementados por quien utilice la interfaz.

- ✓ Módulo: Es donde se implementa la interfaz, es decir debe implementar cada comando de las interfaces que proporciona y cada evento de las interfaces que usa.
- ✓ Configuración: es donde se especifican la unión de componentes, es decir, dice que interfaces se van a utilizar, por quien se va a utilizar y quien las implementa.

6.2.1 Estructura de una Aplicación en NesC

Figura 47. Estructura de una aplicación en NesC



Fuente: Curso de NesC de Jose María Alcaraz Calero

Cuando se escribe una aplicación en NesC se deben crear tres archivos, uno con la extensión .nc que contiene el código referente a la configuración del componente. Un segundo archivo con la extensión .nc pero con el nombre de la aplicación terminado en M, que contiene el código correspondiente al modulo del componente y un tercer archivo con la extensión .h que contiene la cabecera de la aplicación.

El directorio de TinyOS, donde se encuentran todas las aplicaciones predefinidas en NesC para los nodos es el siguiente:

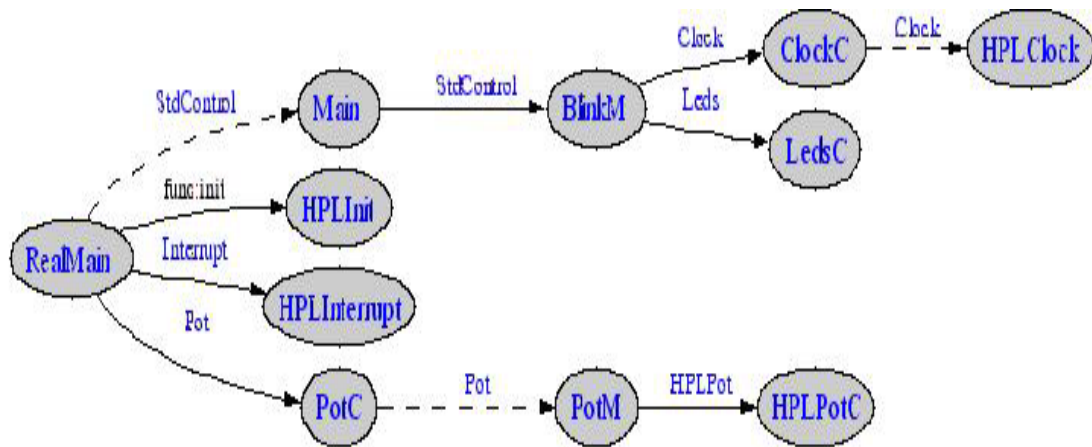
- ✓ /apps: Están los componentes que son aplicaciones
- ✓ /tos/interfaces: Están las interfaces que ofrecen los componentes.
- ✓ /tos/system: Están los componentes primitivos del TinyOs
- ✓ /tos/types: Están los ficheros que contiene los tipos de datos (Pej: AM.h contiene la estructura de un mensaje TOSMsg)

Cualquier aplicación que desee crear se debe poner en el directorio C: /Cygwin/opt/Moteiv/apps y así se puede utilizar el resto de las aplicaciones y adaptarlas a necesidades propias.

A continuación mostraremos y explicaremos una aplicación ejemplo contenida dentro de las apps de TinyOS llamada Blink. La tarea que realiza el programa blink cuando es instalada en un nodo es prender uno de los led's del mote cada segundo (el led se escoge especificando el color que se desea que se ilumine en el código de la aplicación).

La figura muestra la estructura que siguen las instrucciones hechas a través del código Blink y el proceso que se sigue para que dichas instrucciones sean llevadas a cabo.

Figura 48. Estructura aplicación Blink



Fuente: <http://nescs.sourceforge.net/>

6.2.2 Oscilloscope Modificado

Como se enunció anteriormente, los componentes del código nesC son los módulos y las configuraciones; y para apreciar mejor dichos elementos, se usó el oscilloscope, una aplicación del tinyos y mediante modificaciones realizadas permite leer otras variables adicionales a las inicialmente estudiadas, amoníaco, monóxido de carbono y velocidad del viento.

En el anexo 4 se puede ampliar un poco más la información relacionada con este tema además de contener bibliografía útil sobre el mismo.

7. DESARROLLO DE ALGORITMOS Y PRUEBAS

Este proyecto de grado hace parte de REIMAGAV (Red Inalámbrica de Monitorización Ambiental en Galpones Avícolas), una investigación que está llevando a cabo el grupo de Investigación CPS de la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones, apoyado por COLCIENCIAS, razón por la cual, la mayor parte de las pruebas con hardware se llevaron a cabo con la asesoría y el acompañamiento de compañeros electrónicos de este grupo de investigación.

El desarrollo de los algoritmos y pruebas sobre la aplicación web REDSI, se llevaron a cabo siguiendo las fases de la metodología escogida inicialmente para el desarrollo del sistema. La metodología y su descripción se hicieron en el capítulo 4.

Durante la ejecución y puesta en marcha de este proyecto de grado, se desarrollaron e implementaron una serie de pruebas, tanto con el hardware como con el software. A continuación se hará una breve descripción de estas actividades ya que sólo de ese modo se aprecia la evolución y el proceso que se siguió para llegar a una versión final de esta parte del trabajo de grado.

Inicialmente las pruebas y el desarrollo empezaron con un generador de datos hecho para simular el envío de datos de los sensores a un concentrador y de ahí se almacenaba la información en una base de datos. Luego se inició la construcción de la aplicación web REDSI y se estableció la conexión entre la aplicación y la base de datos para probar los diferentes módulos en desarrollo en ese momento.

Finalmente y luego de hacer el análisis del modelo general del proyecto para determinar las necesidades básicas de hardware y software requeridas, se tomaron algunos nodos para la toma de datos con los sensores de las variables a estudiar, un gateway que tenía conectado el nodo receptor y al cual se accedía para leer los datos capturados en cualquier momento y desde Internet, gracias a la IP pública asignada al gateway, y un equipo que actuó como servidor web donde se instaló la base de datos, la aplicación web y el código en java para leer los datos del gateway, para hacer las pruebas necesarias con los dispositivos que al final, serán ubicados en los galpones.

También se reutilizaron algunas librerías del sistema operativo TinyOS en el desarrollo de los algoritmos en NesC para programar los nodos y Java para establecer la comunicación con los dispositivos electrónicos que conforman la red y almacenar la información que se capture en una base de datos.

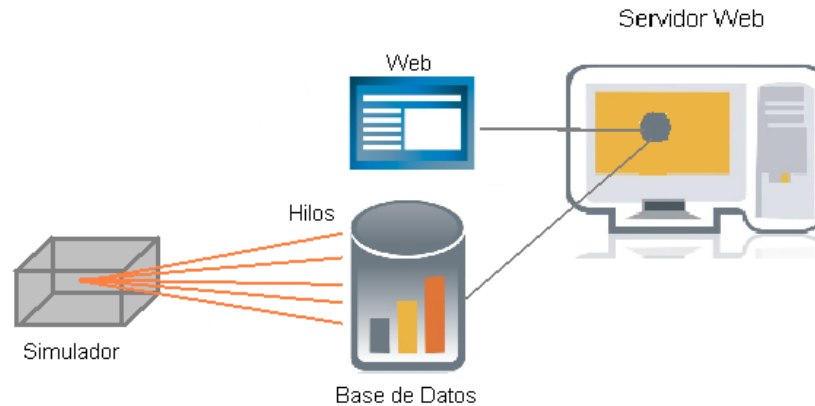
Este modelo físico de pruebas se implementó en los laboratorios de redes de la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad Industrial de Santander.

7.1 DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL GENERADOR DE DATOS

Al inicio del proyecto, y ante la necesidad de obtener datos similares a los que se manejarán mas adelante con el uso de los nodos, se desarrolló un algoritmo que simula el envío de datos cada cinco segundos (tiempo establecido al hacer análisis de los datos reales que brindó el grupo CPS) de cuatro variables: temperatura, humedad,

intensidad luminosa y radiación solar; insertando esta información en la base de datos junto con el tiempo en que se tomó la muestra según el generador de datos.

Figura 49. Modelo de prueba con el Generador de Datos



Fuente: Autoras del proyecto

Este código fue desarrollado en Java utilizando hilos para simular una cantidad determinada de dispositivos enviando datos. El generador de datos se creó con el fin de representar un modelo similar al real, por lo que su fin primordial era poblar la base de datos con toda la información generada por un supuesto grupo de nodos, para posteriormente manipularla a través de la interfaz Web creada.

Después de poner en marcha el generador de datos, cualquier nodo que se adicione a través de la interfaz Web es reconocido automáticamente por el programa debido a su constante lectura de cuántos nodos se encuentran transmitiendo, y dinámicamente se crea un hilo mas para simular el envío de datos de ese nuevo nodo en la red. Este trabajo lo hace una clase llamada Administrador de nodos.

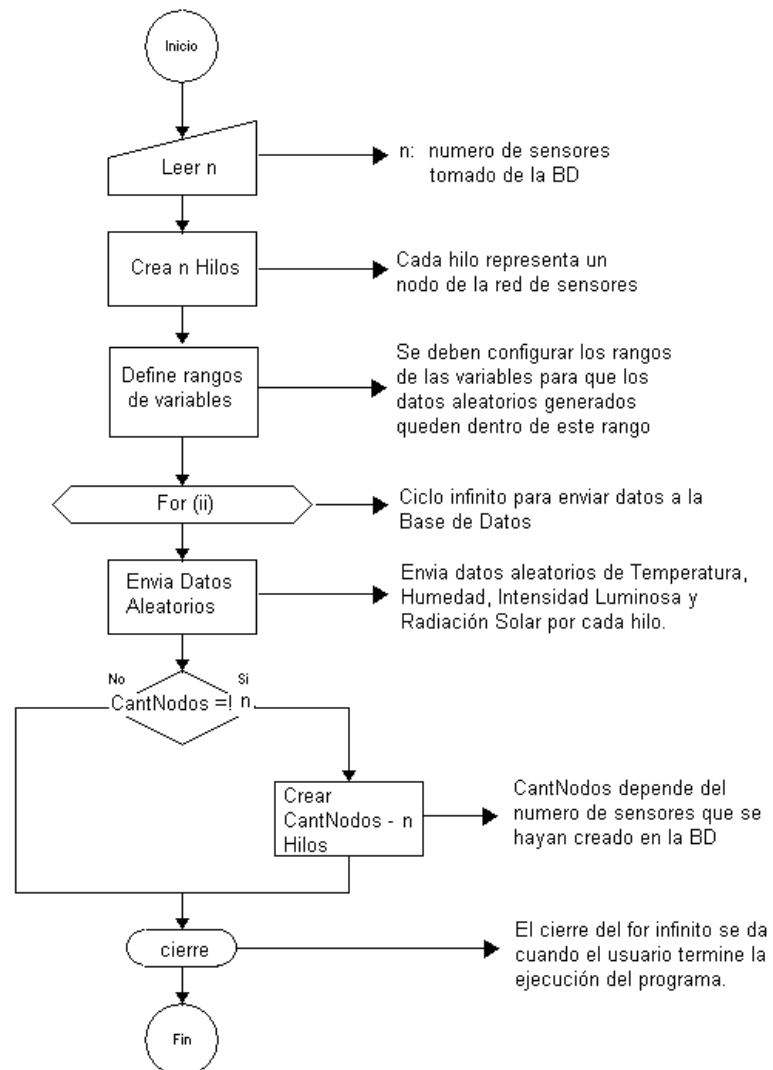
Los rangos en los que oscilan los datos de las variables simuladas, se configuran directamente en el código y este generador de datos solo detiene su ejecución cuando el usuario lo detiene el programa manualmente.

El ciclo de cría, levante y engorde de las aves es de 45 días, durante el cual se establece en la investigación la toma de muestras cada cinco minutos (tiempo que puede aumentar o disminuir según las pruebas que se realicen en la investigación), lo que equivale a 388000 registros durante este periodo de tiempo. Este escenario fue simulado enviando datos utilizando el generador de datos, para observar el comportamiento de la base de datos con estas cantidades de información y determinar que tan rápidas son las consultas desde la aplicación web.

Durante 18 horas este simulador estuvo enviando datos a la base de datos cada cinco segundos con el fin de recrear los 45 días que dura el ciclo de cría, levante y engorde de las aves, tomando muestras cada cinco minutos. Esta prueba se hizo simulando 30 sensores y como resultado se obtuvieron 388000 registros en la base de datos, que a la hora de hacer backup ocupan 38.25 MB de espacio en disco. Con esto se comprobó que la base de datos soporta esta cantidad de datos sin problemas y las consultas siguen siendo muy rápidas. A pesar de repetir el procedimiento con el generador de datos varias veces, nunca variaron los resultados obtenidos debido a que este algoritmo nunca envía paquetes dañados por lo que la cantidad de registros generados y almacenados en la base de datos en este rango de tiempo siempre fue la misma.

A continuación se muestra un diagrama de flujo que representa la funcionalidad del código y sus diferentes procesos.

Figura 50. Diagrama de flujo Generador de Datos



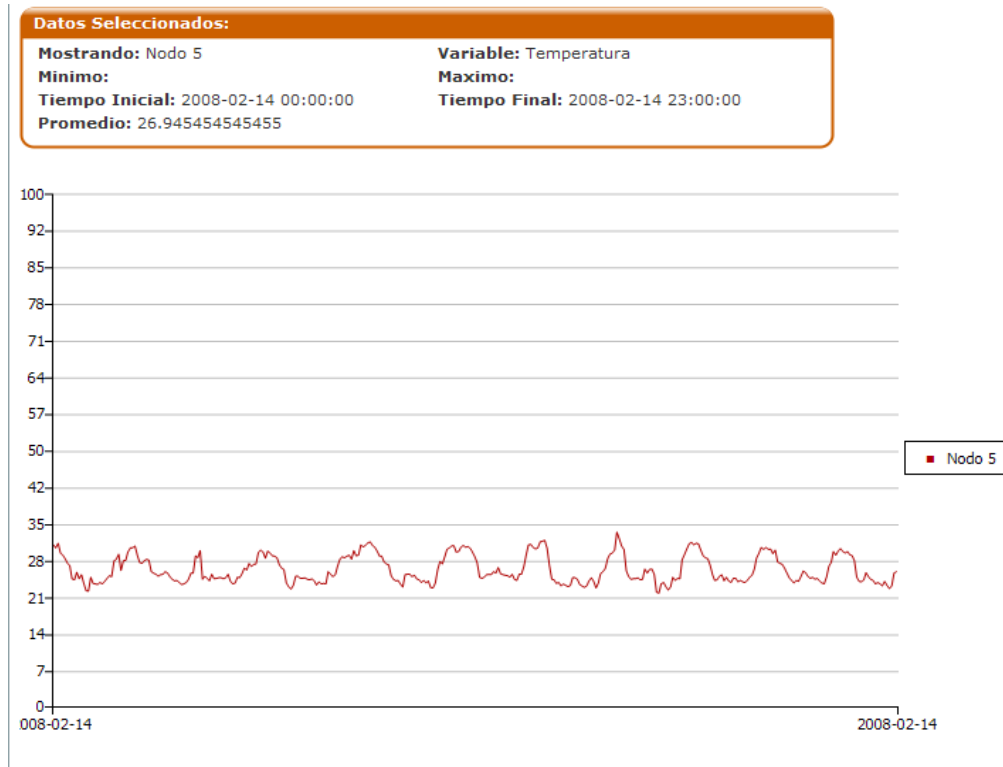
Luego de realizar con éxito esta prueba, de los registros almacenados en la base de datos durante estas 18 horas, se generó desde la aplicación web un análisis gráfico de la información para determinar mas adelante que tan alineado esta el simulador respecto a los datos reales.

A continuación se muestran las gráficas obtenidas con los datos del generador para la variables temperatura y humedad. Estas gráficas se dan en un rango de tiempo de 18 horas, por lo que solo cubren un día y como ejemplo para mostrar se toman los datos

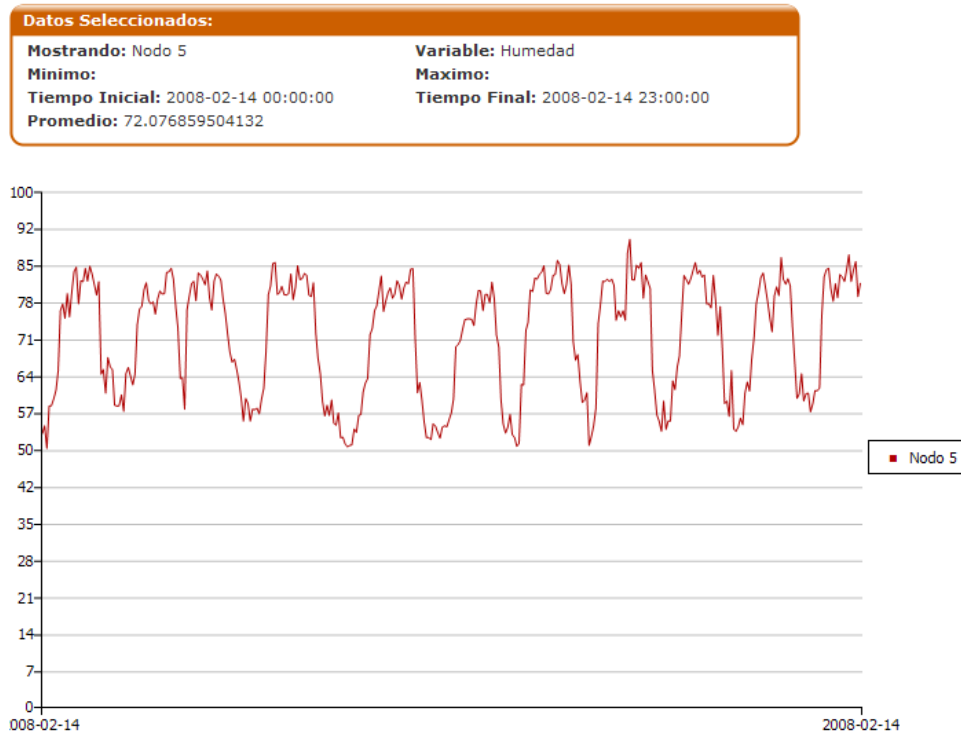
del nodo 5.

Para generar datos de temperatura para un nodo en el algoritmo se estableció un rango mínimo de 20 grados y un rango máximo de 50 grados. Para la variable humedad se estableció un rango mínimo de 50% de humedad y un máximo de 90%.

Figura 51. Grafica con datos generados aleatoriamente de Humedad y Temperatura



Como se puede observar el comportamiento de la grafica para temperatura y humedad es semejante al comportamiento de la función senoidal. Y como se comprobara mas adelante, se asemeja también al comportamiento de los datos reales.



Fuente: Autoras del proyecto

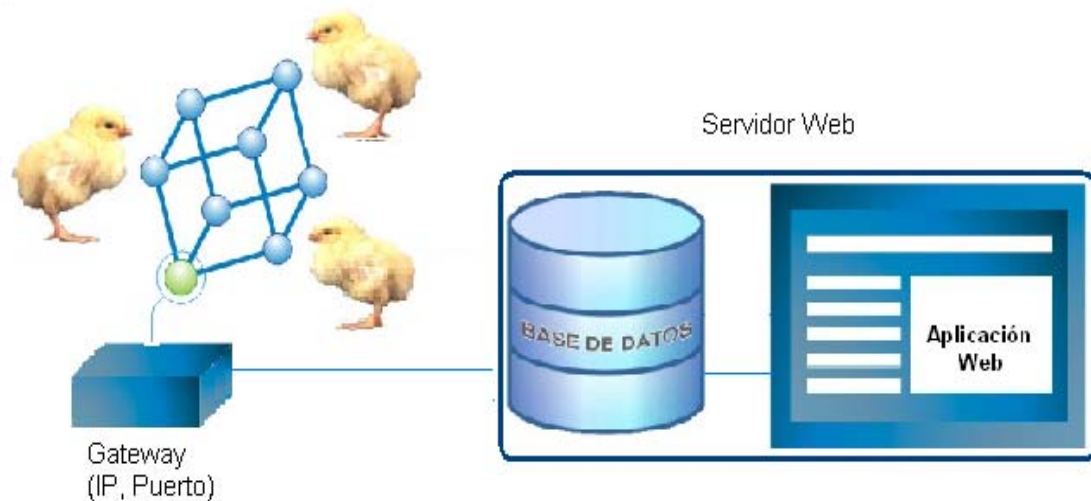
7.2 PRUEBAS CON DATOS REALES

El grupo de investigación CPS, ubicó un nodo en uno de los galpones de prueba, con el fin de tomar muestras de temperatura y humedad, directamente en el medio en el que finalmente va a estar expuesta la red de sensores inalámbricos. Las muestras obtenidas corresponden a doce días de prueba y a un solo nodo denominado nodo1.

Estas muestras de datos se pasaron a un documento en Excel y fueron exportadas a MySQL con el fin de hacer análisis de los datos, compararlos gráficamente con los datos aleatorios, sacar promedios y finalmente ajustar el generador de datos, asignando

unos rangos que se adapten a los promedios obtenidos al manipular estas muestras, además de obtener rangos adecuados para activar las alarmas dentro de la aplicación web.

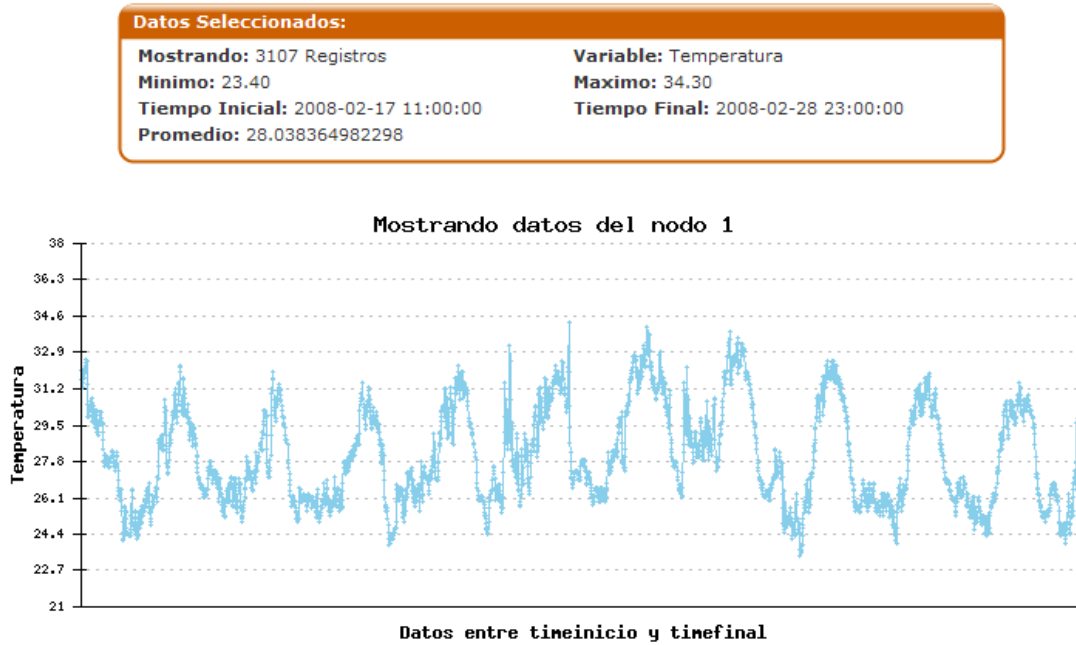
Figura 52. Diagrama de pruebas con datos reales



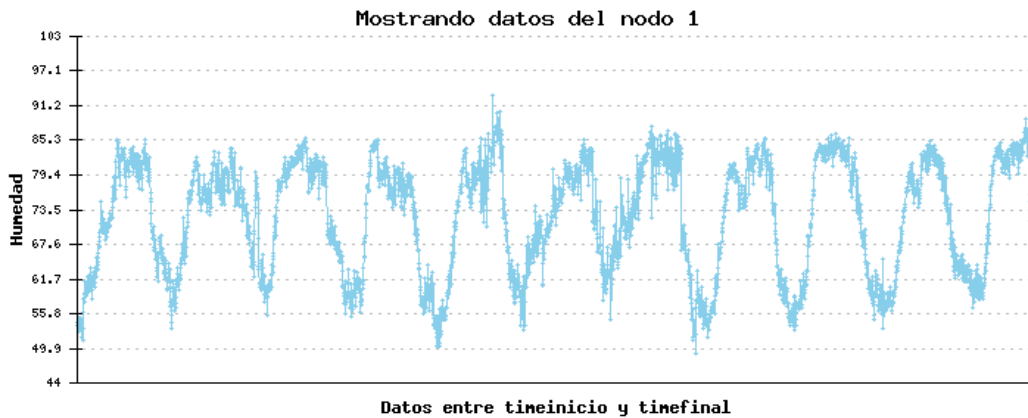
Fuente: Autoras del proyecto.

Después de cargar los datos del sensor puesto en funcionamiento en el galpón a la base de datos, se accedió a la aplicación web REDSI y se obtuvieron las siguientes gráficas correspondientes a las variables de temperatura y humedad.

Figura 53. Gráfica datos reales de temperatura y humedad



Datos Seleccionados:	
Mostrando: 3107 Registros	Variable: Humedad
Mínimo: 49.10	Máximo: 92.90
Tiempo Inicial: 2008-02-17 11:00:00	Tiempo Final: 2008-02-28 23:00:00
Promedio: 72.438847763115	



Fuente: Autoras del proyecto

Como se observa, las graficas de humedad y temperatura de los datos reales obtenidos de un sensor tomando información de un galpón, muestran un comportamiento similar a las obtenidas con datos generados aleatoriamente y los rangos de datos para estas variables en el generador de datos son bastante coherentes.

7.3 CÓDIGO JAVA COMUNICACIÓN NODOS - SERVIDOR

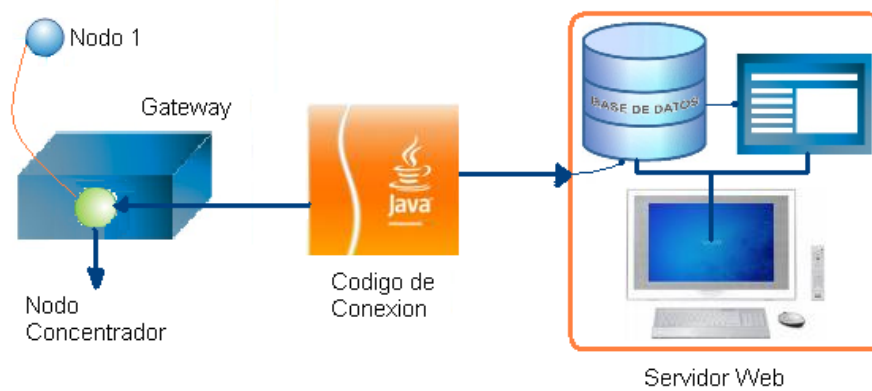
Además de este desarrollo se generó un código adicional en java que utiliza algunas de las librerías y clases java de TinyOS. Este código se denomino “CapturaDatos”, nombre alusivo a la función que cumple, ya que se encarga de traer del gateway la trama de datos¹⁶ enviada por los nodos de la red al nodo concentrador (conectado directamente al gateway).

Luego de capturarla y almacenarla en un vector, se convierten los datos del formato hexadecimal original a decimal; se identifica cual es la información asociada a cada

¹⁶ Ver anexo 2

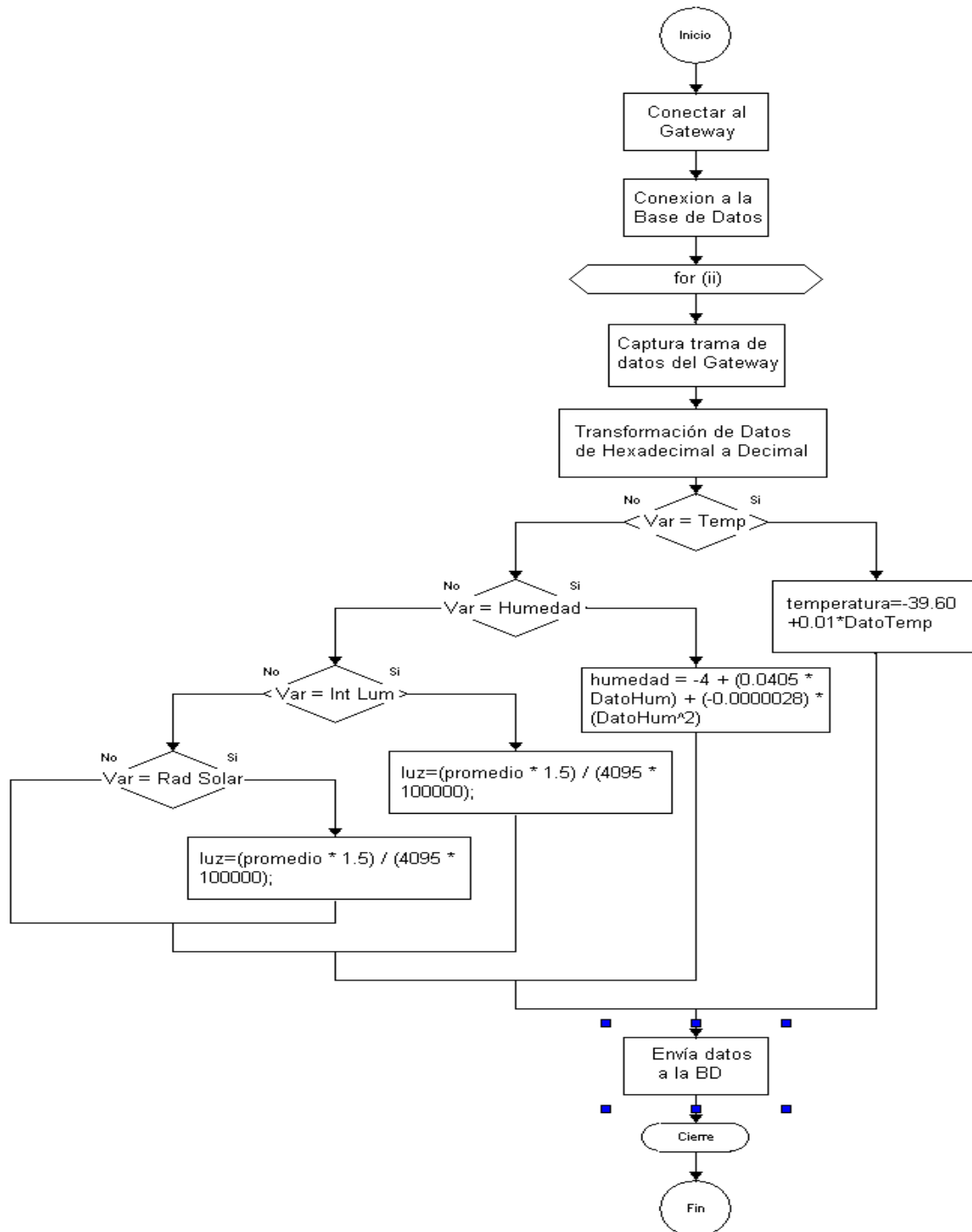
variable según el canal (que se reconoce en la trama recibida) y se aplica una fórmula específica (las formulas están descritas para cada variable en el anexo 2), para obtener el valor real de las lecturas y así almacenarla en la base de datos. Este código de conexión en java se ejecuta desde el servidor web donde tenemos la aplicación REDSI y la base de datos.

Figura 54. Código Java de Conexión Gateway-BD



Fuente: Autoras del proyecto

Figura 55. Diagrama de flujo de captura de datos

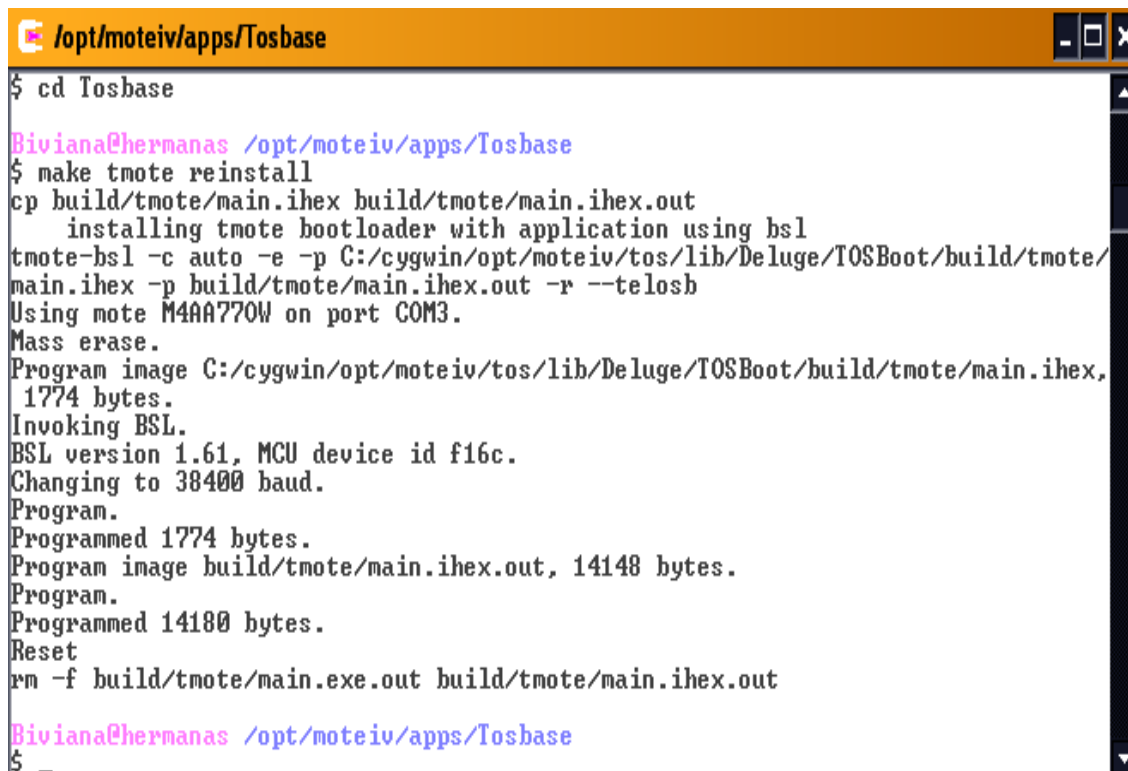


Fuente: Autoras del proyecto

Para iniciar con las pruebas del código que permite la captura y almacenamiento de los datos leídos por los motes, se debe instalar en los estos dispositivos sus respectivos programas.

Al dispositivo que actúa como nodo receptor, se le instala el programa TOSBase, este programa se encuentra en C:\cygwin\opt\motziv\apps\TOSBase. Para instalarlo en el mote, se abre Cygwin, se busca el directorio y se ejecuta la instrucción “ make tmote reinstall ”, se verifica que no haya errores en la instalación y se procede a instalar el mote transmisor.

Figura 56. Instalación TOSBase en el nodo receptor



```

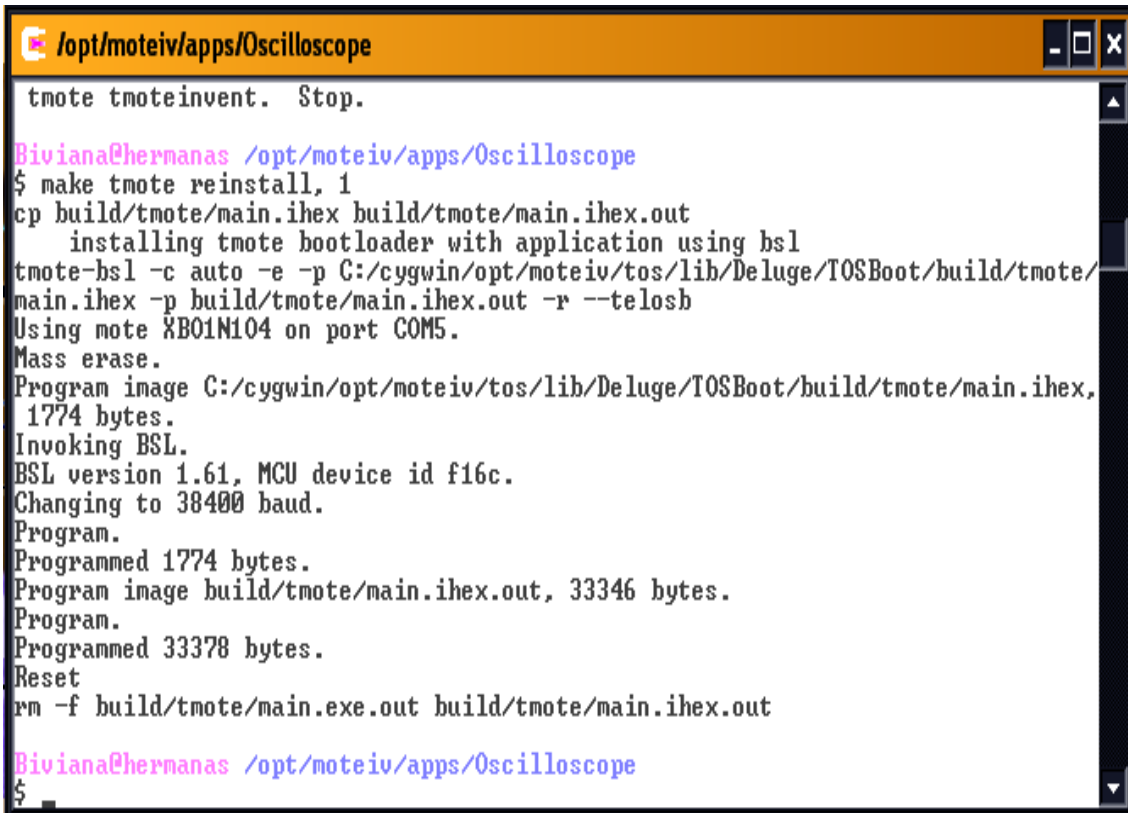
/opt/motziv/apps/Tosbase
$ cd Tosbase
Biviana@hermanas /opt/motziv/apps/Tosbase
$ make tmote reinstall
cp build/tmote/main.ihex build/tmote/main.ihex.out
installing tmote bootloader with application using bsl
tmote-bsl -c auto -e -p C:/cygwin/opt/motziv/tos/lib/Deluge/TOSBoot/build/tmote/
main.ihex -p build/tmote/main.ihex.out -r --telosh
Using mote M4AA770W on port COM3.
Mass erase.
Program image C:/cygwin/opt/motziv/tos/lib/Deluge/TOSBoot/build/tmote/main.ihex,
1774 bytes.
Invoking BSL.
BSL version 1.61, MCU device id f16c.
Changing to 38400 baud.
Program.
Programmed 1774 bytes.
Program image build/tmote/main.ihex.out, 14148 bytes.
Program.
Programmed 14180 bytes.
Reset
rm -f build/tmote/main.exe.out build/tmote/main.ihex.out
Biviana@hermanas /opt/motziv/apps/Tosbase
$

```

Al mote que actúa como nodo transmisor, se le instala el programa Oscilloscope que se

encuentra en C:\cygwin\opt\motelv\apps\Oscilloscope. Se abre Cygwin y se ejecuta la instrucción “ make tmote reinstall, id del nodo”.

Figura 57. Instalación Oscilloscope en el nodo transmisor



```

tмотe tмотeinvent. Stop.

Biviana@hermanas /opt/moteiv/apps/Oscilloscope
$ make tmote reinstall, 1
cp build/tмотe/main.ihex.out build/tмотe/main.ihex.out
installing tмотe bootloader with application using bsl
tмотe-bsl -c auto -e -p C:/cygwin/opt/moteiv/tos/lib/Deluge/TOSBoot/build/tмотe/main.ihex -p build/tмотe/main.ihex.out -r --telosh
Using mote XB01N104 on port COM5.
Mass erase.
Program image C:/cygwin/opt/moteiv/tos/lib/Deluge/TOSBoot/build/tмотe/main.ihex,
1774 bytes.
Invoking BSL.
BSL version 1.61, MCU device id f16c.
Changing to 38400 baud.
Program.
Programmed 1774 bytes.
Program image build/tмотe/main.ihex.out, 33346 bytes.
Program.
Programmed 33378 bytes.
Reset
rm -f build/tмотe/main.exe.out build/tмотe/main.ihex.out

Biviana@hermanas /opt/moteiv/apps/Oscilloscope
$

```

Luego de tener el nodo receptor conectado al computador y el nodo transmisor enviando datos inalámbricamente, se ejecuta el código java para empezar a almacenar datos en la base de datos. En la figura a continuación se muestra la trama de datos que se captura del gateway y el valor obtenido después de la conversión a decimal y de aplicarle la formula correspondiente de acuerdo a la variable que se está midiendo.

Finalmente la información es almacenada en la base de datos y accedida a través de la aplicación web que se creo para este fin.

Para efectos del desarrollo de este proyecto de investigación se usó el programa TOSBase, instalado en el nodo receptor y Oscilloscope, instalado en los nodos transmisores. Inicialmente, las pruebas fueron hechas localmente, conectando el nodo receptor a un computador de escritorio usando un puerto USB y colocando un nodo transmisor a enviar datos. El modelo de pruebas reales se ve representado en la siguiente figura.

Figura 59. Modelo de prueba de envío y recepción de los datos



Fuente: Autoras del proyecto

En esta figura 62 el nodo transmisor utiliza baterías AA para transmitir inalámbricamente la información al nodo receptor. El nodo transmisor puede conectarse USB a un equipo y usarlo como fuente de energía, en caso de no tener las baterías AA que necesita para transmitir inalámbricamente.

Figura 60. Modelo de prueba de conexión USB del nodo transmisor



Fuente: Autoras del proyecto

Finalmente, se adquirió el Gateway para conectar el nodo receptor; se adjudicó la dirección IP 192.168.45.58 en primera instancia para efectos de las pruebas preliminares y luego se asignará una IP pública, para permitir capturar los datos y acceder a ellos desde Internet.

Figura 61. Modelo General de Pruebas utilizando el Gateway



Gateway – Nodo Receptor



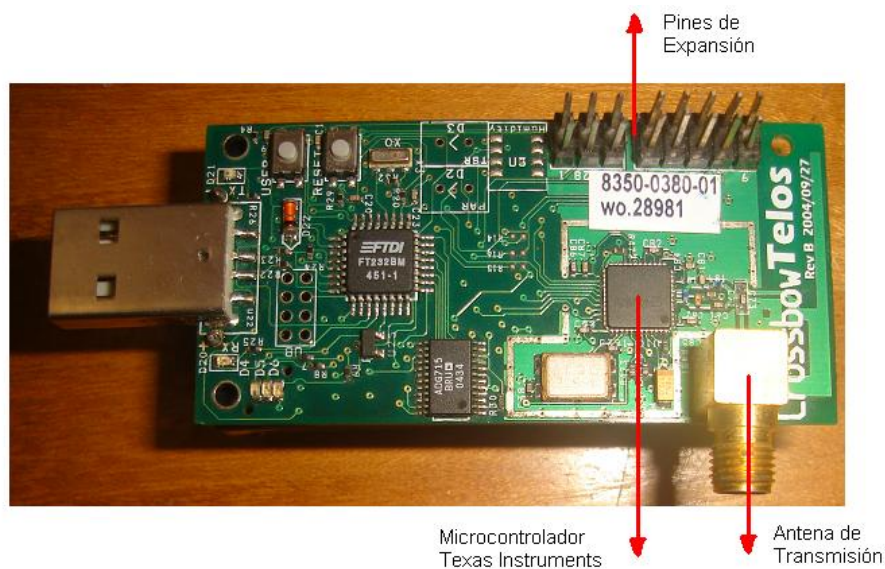
Nodos Transmisores

Fuente: Autoras del proyecto

Luego de tener este modelo acoplado a la interfaz Web desarrollada, el proyecto de investigación tomó un nuevo giro; debido a que la red de sensores inalámbricos se visualiza ubicada en un galpón, se pensó en un diseño adecuado para salvaguardar los dispositivos electrónicos de los daños que puedan causarles las aves. De ahí la necesidad de utilizar una caja ergonómicamente diseñada para ubicar cada uno de los nodos y que permita sacar los sensores de ella, de manera que puedan tener contacto con el medio y tomar medidas fiables de las variables en estudio.

Contemplando esta necesidad, se modificó el nodo y se utilizaron los pines adicionales que trae la placa base para incluir los sensores de tal forma que puedan sacarse de la caja diseñada como contenedor de los nodos. El equipo de electrónicos del grupo de Investigación CPS, se hizo cargo de las modificaciones del hardware. Además se modificó el código Oscilloscope en NesC, que es el encargado de tomar las medidas de los sensores, empaquetarlas y enviarlas al nodo receptor. Este código está referenciado en el capítulo anterior.

Figura 62. Nodo Modificado





Fuente: Autoras del proyecto

De esta forma se completa el modelo hardware y software utilizado durante las pruebas hechas para obtener los valores finales que alimentarán el sistema.

CONCLUSIONES

- ✓ Con el desarrollo del proyecto se hizo un análisis sobre los protocolos de comunicación IEEE 802.15.4 (ZigBee) e IEEE 802.15.1 (Bluetooth), documentando sus principales características, su funcionamiento, sus campos de aplicación y estableciendo una comparación entre estas dos tecnologías.
- ✓ Se implementaron los algoritmos de NesC, contenidos en TinyOS para tomar muestras de motes con sensores de variables predeterminadas como temperatura y humedad, y en vista de las necesidades propias del ambiente para el cual fue concebido este proyecto de investigación, se desarrolló un algoritmo en NesC, que luego de modificar el software (mote) e incluirle nuevos sensores, permitiera leer datos de variables como velocidad del viento, amoníaco y monóxido de carbono.
- ✓ Se diseñó una base de datos adaptada totalmente a las necesidades de almacenamiento del sistema, garantizando la confiabilidad y seguridad de la información. El diseño sencillo de la base de datos, permite la escalabilidad, ya que se puede crecer no sólo en cantidad de nodos y variables, si no también, en cantidad de galpones monitoreados, sin afectar drásticamente la estructura de la base de datos. Para el caso de implementar nuevas variables a estudiar, se debe modificar la aplicación web para realizar consultas con estos nuevos datos.
- ✓ Se diseñó y construyó una aplicación web como interfaz entre los usuarios y la red de sensores inalámbricos que permite el acceso y manipulación de los datos capturados desde cualquier sitio a través de Internet. Este diseño cumple con las condiciones deseadas de fácil acceso y navegación, contando con una interfaz grafica amigable y de fácil navegación.



- ✓ Dentro de la aplicación web se creó una interfaz gráfica para la visualización de los nodos y los datos que se están emitiendo, además de permitir detectar visualmente cualquier falla en alguno de los sensores. También se crearon los módulos de consultas y reportes gráficos, el modulo de análisis estadístico y un modulo de control para registro y consulta de la mortandad de las aves.

OBSERVACIONES

- ✓ La aplicación web creada durante el desarrollo de este proyecto de grado es una versión base para nuevos desarrollos. Como se evidencia en el contenido del libro, al finalizar la investigación, se busca implementar la aplicación web para monitorear un grupo de galpones repartidos en una amplia zona geográfica. Por lo que se hace necesario plantear un nuevo proyecto de grado que continúe con este desarrollo y logre aportar mayor funcionalidad y apoyo a la industria avícola de nuestra región.
- ✓ Es importante asignar una dirección IP pública al servidor Web para que la aplicación pueda ser accedida finalmente desde cualquier lugar.
- ✓ Se debe empezar a visualizar un escenario de mayor desarrollo tecnológico en la industria Avícola Santandereana, que establezca controles automáticos del galpón.
- ✓ Con esta tesis de grado se pudo evidenciar la necesidad de trabajar interdisciplinariamente con otras ramas académicas y así lograr mayor conocimiento y desempeño en otros temas que antes se catalogaban de uso exclusivo de determinada carrera. De este análisis se deja planteada la idea de crear al interior de la Escuela de Ingeniería de Sistemas e Informática un grupo de investigación en redes y telecomunicaciones que abra las puertas a otros estudios relacionadas con las nuevas tecnologías que resulten tan enriquecedores como el presente proyecto.

BIBLIOGRAFÍA

- [1] TinyOS, (vía Internet)
<http://www.tinyos.net/>
- [2] Manual de Usuario de Nodo-view (Vía Internet)
<http://www.davidson.com.au/products/wireless/crossbow/mote-view.asp>
- [3] Proyecto con redes de sensores inalámbricos (Vía Internet)
www.revistaecosistemas.net/articulo.asp?Id=343&Id_Categoria=2&tipo=portada-36k
- [4] Proyecto con redes de sensores inalámbricos (Vía Internet)
<http://www.eecs.harvard.edu/~werner/projects/volcano/>
- [5] LETELIER, Patricio y PENADES, M^a Carmen. Metodologías Ágiles para el Desarrollo de Software, Extreme Programming (XP). Universidad Politécnica de Valencia
- [6] Extreme Programming (Vía Internet).
http://www.xprogramming.com/what_is_xp.htm
- [7] Metodología Scrum (Vía Internet).
<http://www.scrumalliance.org/>
- [8] Comparación entre metodologías de desarrollo. (Vía Internet).
www.objectmentor.com/publications/RUPvsXP.pdf
- [9] ZigBee Alliance Web Site (Vía Internet)
<http://ZigBeeAlliance.org>
- [10] Redes de sensores inalámbricos. (Vía Internet).

- [http://library.abb.com/GLOBAL/SCOT/scot271.nsf/VerityDisplay/A019E9833DCF2819C1257199004E5DD2/\\$File/39-42%202M631_SPA72dpi.pdf](http://library.abb.com/GLOBAL/SCOT/scot271.nsf/VerityDisplay/A019E9833DCF2819C1257199004E5DD2/$File/39-42%202M631_SPA72dpi.pdf)
- [11] Aplicaciones de redes de sensores inalámbricos. (Vía Internet).
<http://www.radioptica.com/Radio/wsn.asp>
- [12] CrossBow Web Site (Vía Internet).
<http://www.chipcon.com>
- [13] TLC: primer pulso es por control del debate. Artículo publicado por el partido Liberal Colombiano en su página de Internet. (Vía Internet).
http://www.partidoliberal.org.co/root/index.php?option=com_content&task=view&id=849&Itemid=16
- [14] <http://wsnplanet.com/content/view/13/9/> (tutoriales de redes de sensores)
- [15] MySQL AB. *MySQL*. <http://www.mysql.com>
- [16] Borland Software Corporation. *Interbase*. <http://www.borland.com/interbase>
- [17] PostgreSQL Global Development Group. *PostgreSQL*.
<http://www.postgresql.org>
- [18] <http://www.ece.uah.edu/~jovanov/whrms/>
- [19] CrossBow Web Site: <http://www.chipcon.com>
- [20] <http://www.xbow.com/Technology/UserInterface.aspx>
- [21] www.rational.com
- [22] Using the IBM Rational Unified Process for Small Projects: Expanding Upon eXtreme Programming
- [23] www.objectmentor.com/publications/RUPvsXP.pdf
- [24] INSIGHT: Internet-Sensor Integration for Habitat Monitoring, Autores: Murat Demirbas, Ken Yian Chow, Chieh Shyan Wan; Dept. of Computer Science & Engineering, University at Buffalo Buffalo, NY 14226.
- [25] ATEMU: A Fine-grained Sensor Network, Simulator, Jonathan Polley, Dionysys

- Blazakis, Jonathan McGee, Dan Rusk, John S. Baras; Center for Satellite and Hybrid Communication Networks Department of Electrical and Computer Engineering & Institute for Systems Engineering University of Maryland, College Park, MD 20742, USA .
- [26] A Spreadsheet Approach to Programming and Managing Sensor Networks, Alec Woo
Arched Rock Corp, 657 Mission St. Ste. 600, San Francisco, CA 94105;
Siddharth Seth, Dept. of EECE, Indian Institute of Technology, Kharagpur, India;
Tim Olson, Jie Liu, Feng Zhao, Microsoft Research, One Microsoft Way,
Redmond, WA 98052.
- [27] <http://www.domodesk.com/content.aspx?co=97&t=21&c=47>
- [29] IEEE 802.15, <http://www.ieee802.org/15/pub/TG4.html>
- [30] Descripción de ZigBee, <http://www.radio-electronics.com/info/wireless/zigbee/zigbee.php>
- [31] Tutorial, <http://www.tutorial-reports.com/wireless/zigbee/Wikipedia>
- [32] www.bluetooth.com

ANEXO 1

INSTALACIÓN REDSI

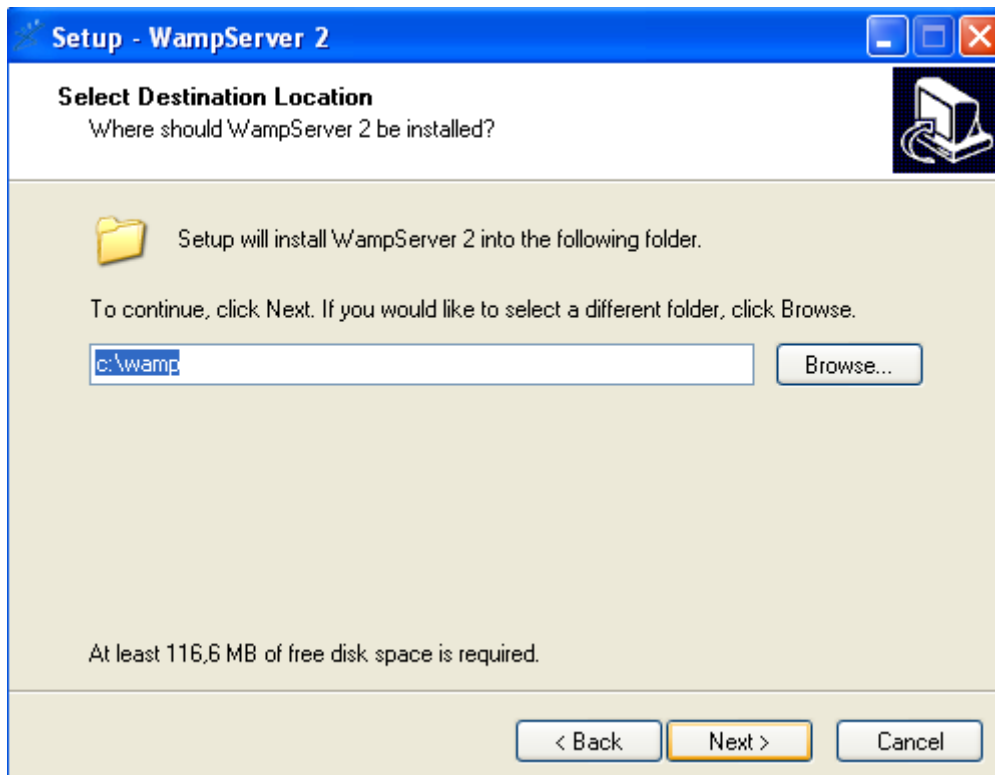
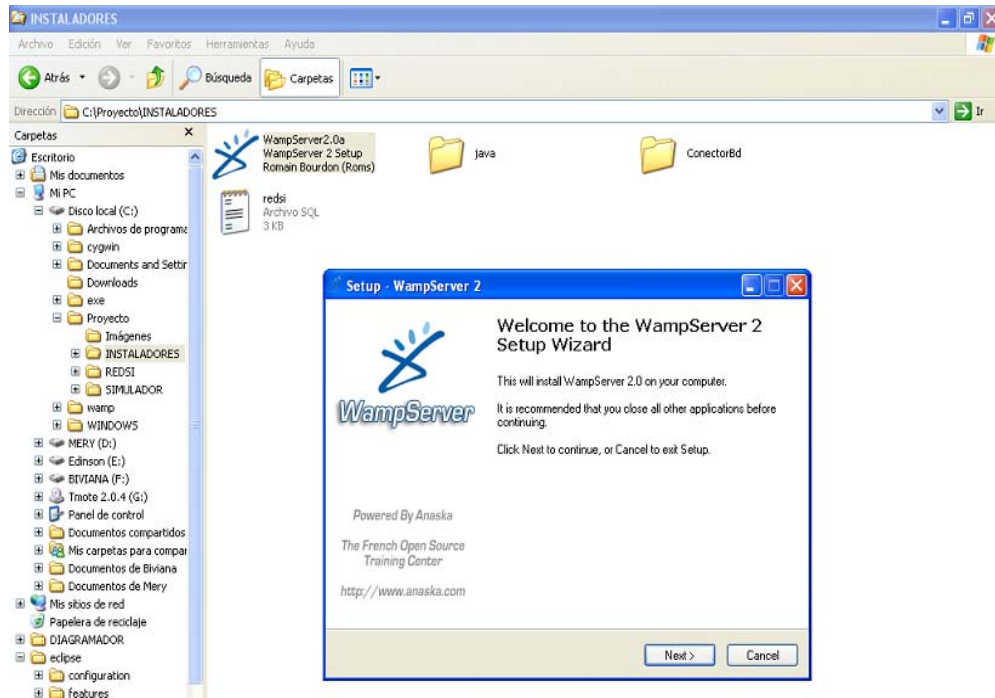
Para la instalación y puesta en marcha de la aplicación REDSI, se debe seguir el procedimiento descrito a continuación para garantizar su correcto funcionamiento.

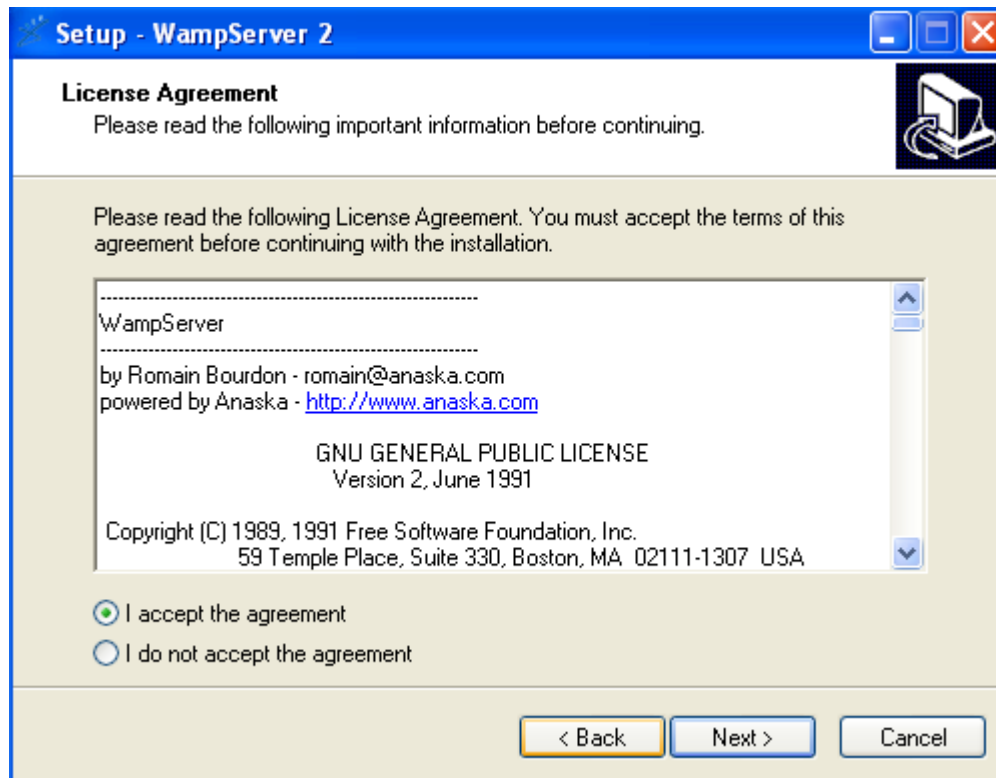
INSTALACIÓN DEL SERVIDOR WEB Y BASE DE DATOS

Se inicia instalando la versión 2.0 del Wamp Server. El instalador se encuentra en el CD adjunto a este documento, en el directorio "...Proyecto \ INSTALADORES \ Wamp \ ". Luego se copia la carpeta REDSI en el directorio www que se encuentra dentro del directorio Wamp. La ruta será la siguiente: "C: \ wamp \ www \ REDSI"

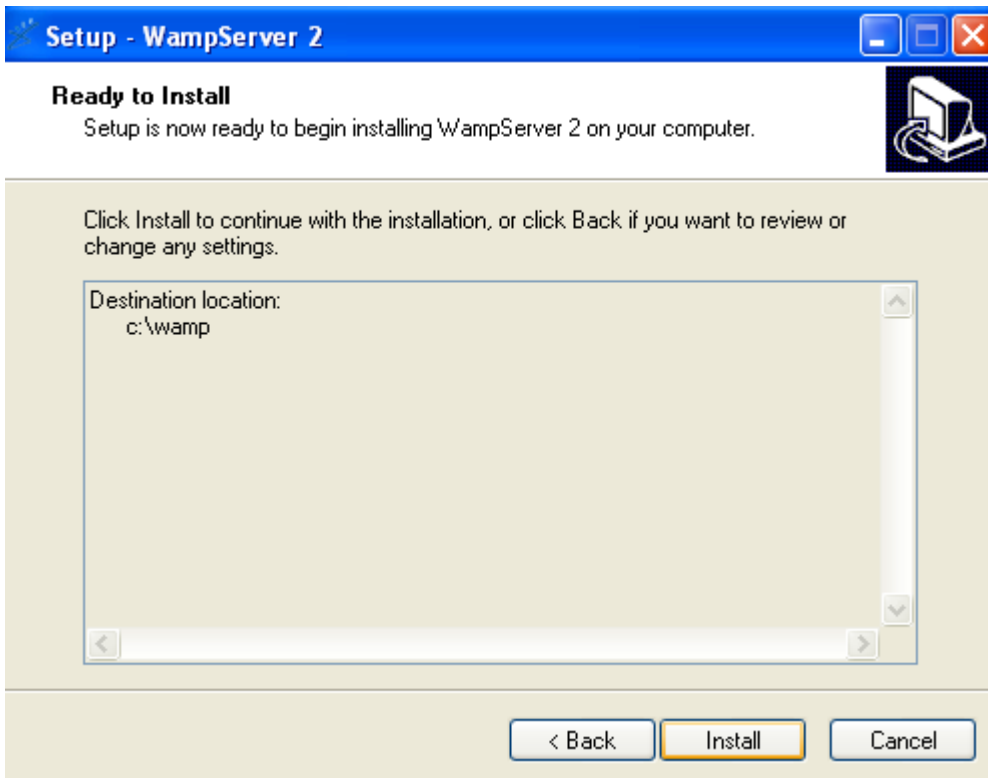
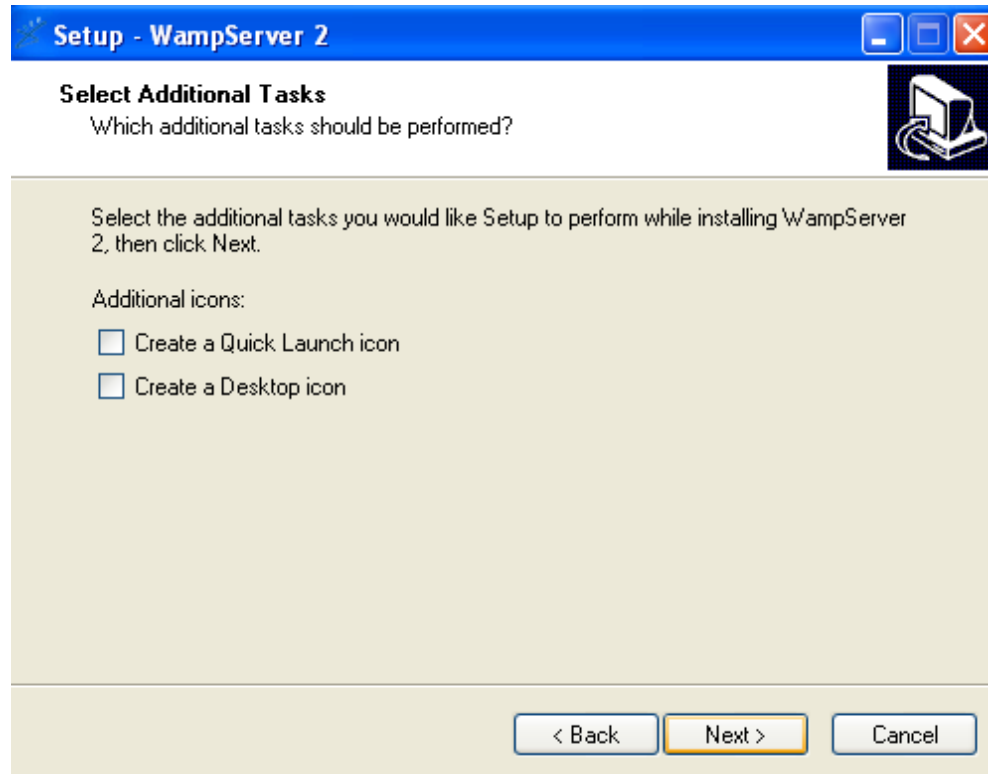
Pasos de instalación del WampServer

1. Haga doble clic sobre el archivo WampServer2.0a.exe incluido junto con los demás instaladores en el CD adjunto a este documento en la carpeta INSTALADORES. Esta versión contiene e instala las últimas versiones de Apache, MySQL y PHP y las configura para permitir la comunicación entre ellas.



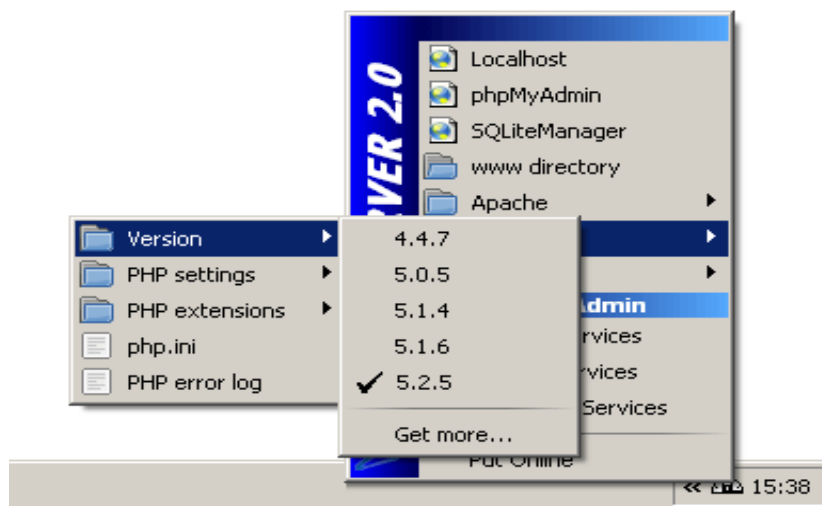


Si se desea se puede incluir el icono de acceso directo en el escritorio.



En caso de tener instaladas varias versiones de Apache, MySQL o PHP, aparecerá un menú como el que se muestra en la figura 1, donde se podrá escoger la versión con la que se desea trabajar. Cada versión de Apache, MySQL y PHP tiene su propia configuración y sus propios ficheros.

Figura 1. Selección versión PHP instalada



Fuente: Autoras del proyecto

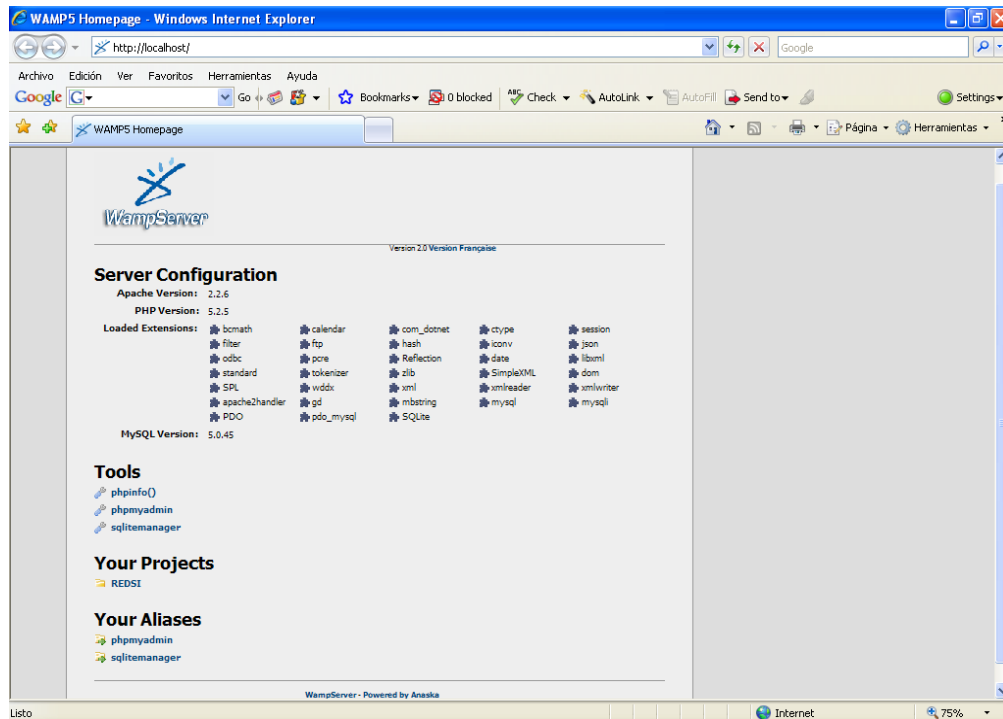
Hay que tener en cuenta que no todas las versiones de Apache y PHP son compatibles unas con otras, pero se resalta la habilidad de WampServer para identificar y notificar cuales son exactamente las versiones incompatibles.

Al instalar WampServer, se crea automáticamente un directorio "www" en c: \ Wamp \ www, es ahí donde debe copiarse la carpeta REDSI que contiene todo el proyecto y los archivos PHP de la aplicación web.

Haga clic en el vínculo "localhost" del menú desplegable de WampServer y abra su navegador con la dirección <http://localhost>, ahí encontramos la página principal de

WampServer y desde ahí, adicionando a la URL, la palabra REDSI, se podrá acceder directamente a la aplicación web creada, como se muestra en las figuras 2 y 3.

Figura 2. Página de configuración de WampServer



Fuente: Autoras del proyecto

Figura 3. Página principal REDSI

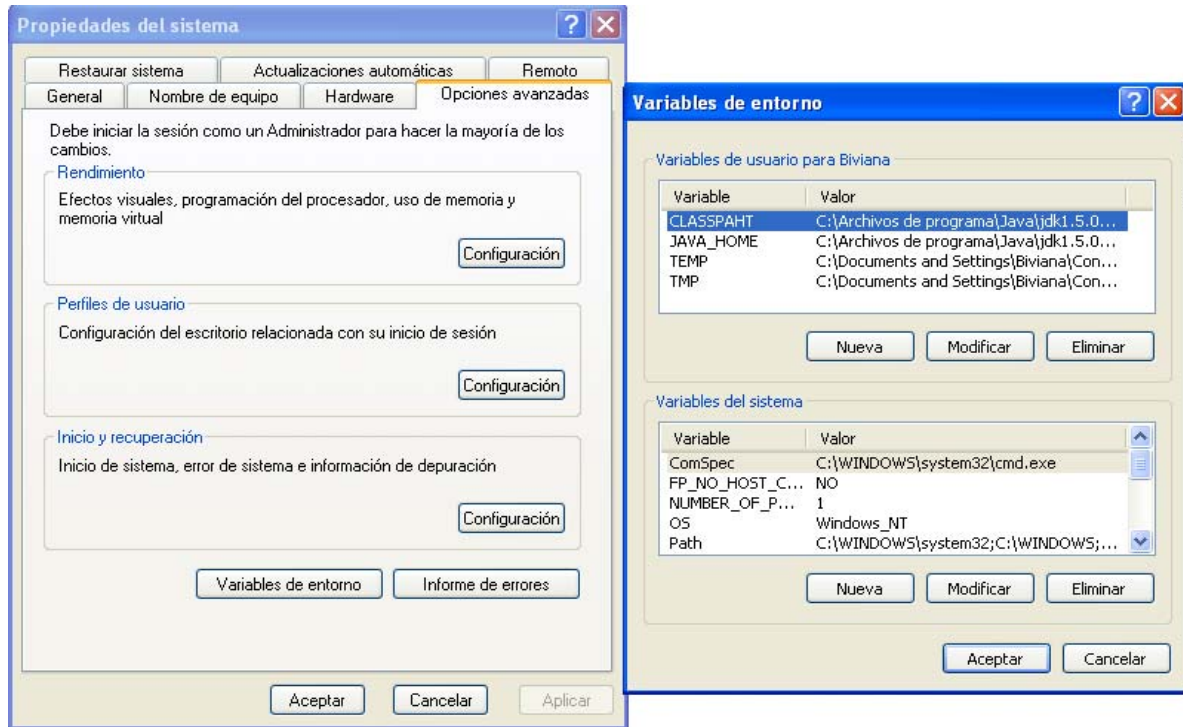


Fuente: Autoras del proyecto

Luego se instala Java, en este caso se usó la versión 1.5.0_09 que se encuentra en la carpeta INSTALADORES, contenida en el CD y cuya ruta fue referenciada anteriormente o que puede ser descargada de la pagina oficial de java: www.java.com.

Además, se debe modificar y crear algunas variables de entorno; para esto, desde MI PC se da clic derecho, se accede a *propiedades*, luego se escoge la pestaña *opciones avanzadas*, y finalmente *variables de entorno*.

Figura 4. Variables de Entorno



Fuente: Autoras del proyecto

Estando ahí, se verifica si están las siguientes variables en la ventana variables de usuario y de no ser así, deben ser creadas:

Nombre de la Variable CLASSPAHT

Valor de la Variable C:\Archivos de programa\Java\jdk1.5.0_09\lib\dt.jar;C:\Archivos de programa\Java\jdk1.5.0_09\lib\tools.jar

Nombre de la Variable JAVA_HOME

Valor de la Variable C:\Archivos de programa\Java\jdk1.5.0_09

En las Variables de Sistema, se verifica si esta la variable path y si tiene la ruta al bin de java, de no ser así, se copia esta ruta y se adiciona a esta variable de entorno.

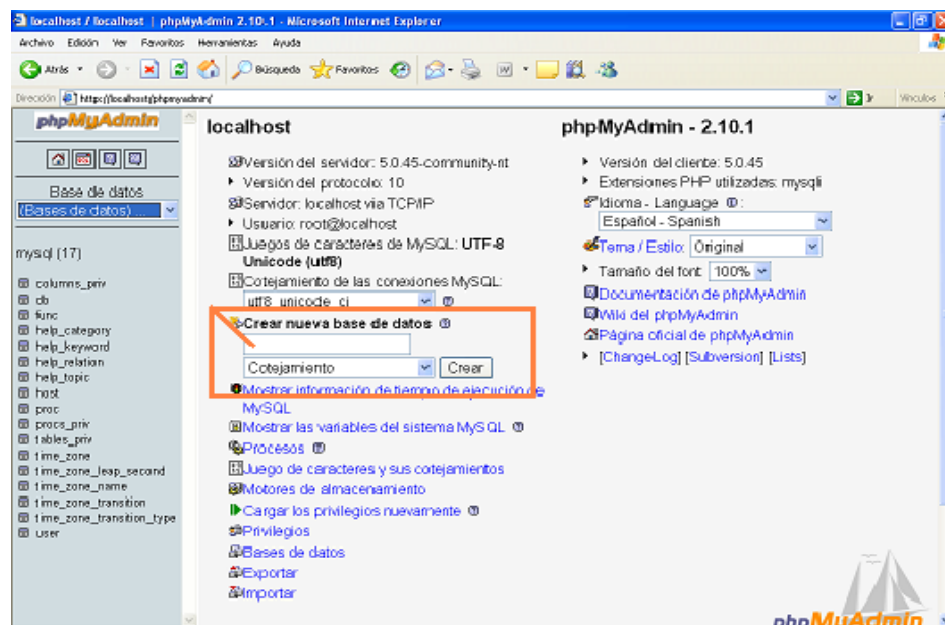
Nombre de la Variable path

Valor de la Variable C:\Archivos de programa\Java\jdk1.5.0_09\bin

Finalmente se debe descargar el controlador para la base de datos que se desee utilizar y se debe copiar en la máquina de java, siguiendo la ruta: C:\Archivos de programa\Java\jdk1.5.0_09\jre\lib\ext. El controlador que se usó para este proyecto es el siguiente: mysql-connector-java-5.0.6-bin, y se encuentra en el directorio INSTALADORES \ ConectorBD\.

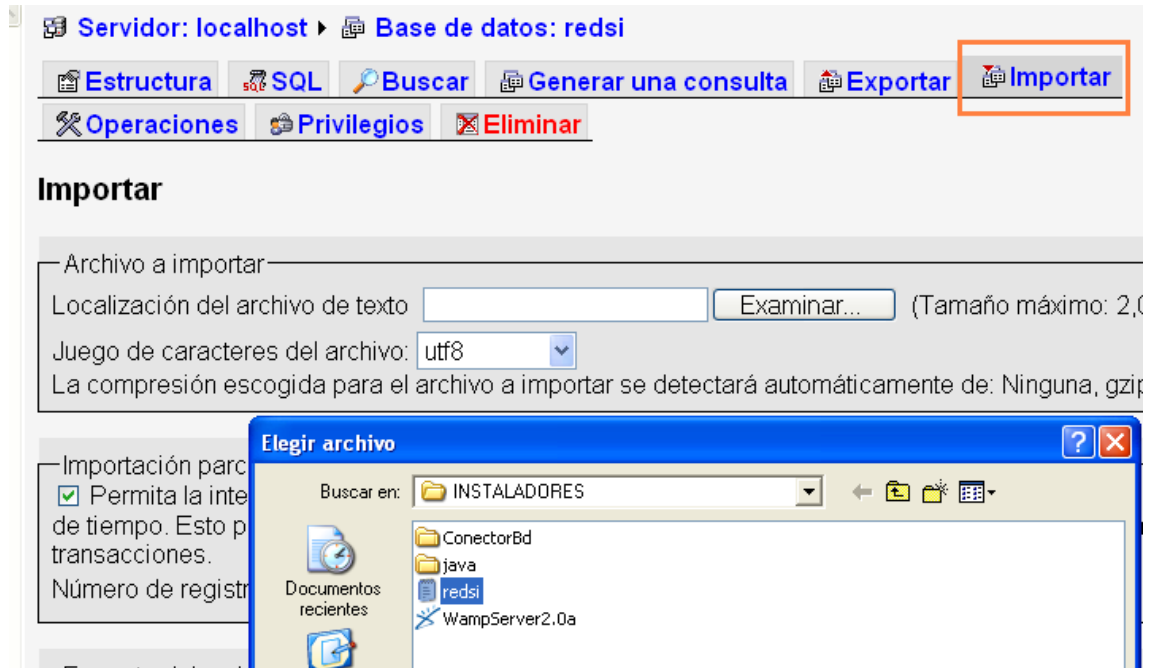
Ya instalados todos los programas necesarios para el funcionamiento de la aplicación se debe cargar el archivo de texto “redsi.sql” a la base de datos, archivo que se encuentra en la carpeta INSTALADORES directamente. Se abre PhpMyAdmin, el administrador que se instala con WampServer para administrar la base de datos, y se crea una base de datos llamada redsi, como se muestra en la figura 5. Luego se selecciona la base de datos que se acaba de crear y se importa el archivo redsi.sql, como se muestra en la figura 6.

Figura 5. Crear Base de Datos



Fuente: Autoras del proyecto

Figura 6. Importar archivo de texto plano con la Base de Datos



Fuente: Autoras del proyecto

Después de instalar el software necesario para que la aplicación REDSI funcione correctamente, se debe acceder a la aplicación web desde la URL <http://localhost/redsi/>, se ingresa usuario y contraseña y se accede al módulo gráfico de crear una topología de red. En el capítulo 5 del libro se describe como crear una topología. Luego de tener la topología creada, se puede ejecutar el código del generador de datos si se quiere probar o poner en funcionamiento los dispositivos que conforman la red de sensores inalámbricos (un nodo receptor y uno o varios nodos transmisores enviando datos) y ejecutar el código `CapturaCatos.java` para capturar los datos leídos por estos dispositivos y almacenarlos en la base de datos. Este programa se ejecuta desde el archivo `Run.bat` que se encuentra en el directorio `C:\Proyecto\SIMULADOR\redsi\exe`.

Finalmente se puede formar físicamente la red en el escenario real (el galpón), crear la topología en la aplicación web y probar el funcionamiento del modelo completo al que

se hace alusión en todo el libro.

ANEXO 2

ANÁLISIS DE REQUISITOS

El análisis de requisitos contempló tanto el Software como el Hardware necesario para el desarrollo y la implementación de este proyecto de grado. En cuanto al Software, se llevó a cabo el análisis y la documentación necesaria para finalmente escoger los lenguajes más adecuados para las necesidades de desarrollo. De igual forma con el Hardware se hizo un estudio preliminar bajo el acompañamiento del grupo de investigación CPS, con el fin de analizar las diferentes tecnologías existentes en el mercado y en especial de las marcas con que cuentan el grupo.

Finalmente se hace un análisis a fondo de los requerimientos necesarios para llevar a cabo el desarrollo e implementación del sistema pensado para este proyecto de grado y se identifican los lenguajes que se usan tanto para programar los dispositivos electrónicos o motes (encargados de realizar las lecturas de las variables estudiadas) como la aplicación web y la comunicación entre la base de datos y la red de sensores.

HARDWARE UTILIZADO

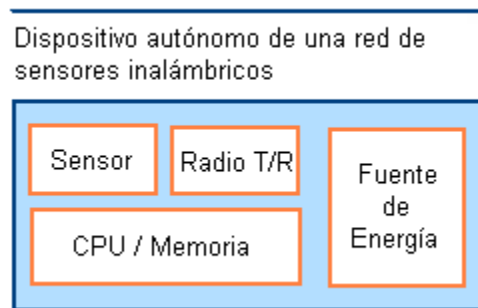
Para la implementación de la red de sensores inalámbricos en los galpones, se adquirieron 30 motes, se desarrollaron las pruebas necesarias y se hicieron los análisis requeridos para determinar la topología mas adecuada para el desarrollo experimental.

Además se adecuaron los equipos necesarios para la recepción de los datos transmitidos por los sensores, un equipo portátil que hace las veces de Servidor Web o servidor de datos, un gateway dotado de un IP y un puerto por donde se comunican los

moten con él y a su vez él con el servidor Web. También se adecuó el área de trabajo, el galpón, y se implementó la infraestructura necesaria para establecer la comunicación entre los sensores, el servidor de datos y los usuarios que acceden a la información a través de Internet.

MOTES

Figura 7. Estructura interna de los motes



Fuente: Autoras del proyecto

Los motes son los dispositivos electrónicos o nodos que conforman una red de sensores inalámbricos. Cada nodo de la red consta de un dispositivo con microcontroladores, sensores y transmisores/receptores. Los motes se comunican utilizando el puerto USB o a través de su radio transmisor que le permite comunicarse inalámbricamente y aunque son capaces de procesar una limitada cantidad de datos, permiten ser programados para que envíen cada cierto tiempo los datos a través de la tecnología ZigBee.G

El medio en el cual se implementan las redes de sensores inalámbricos que se quieren monitorear, no tiene la infraestructura para el suministro energético, ni para la comunicación, es por eso, que los motes funcionan con baterías recargables y se comunican inalámbricamente.

TIPOS DE MOTES

A pesar de ser una tecnología relativamente nueva, los motes ya están siendo desarrollados por varios fabricantes, entre ellos:

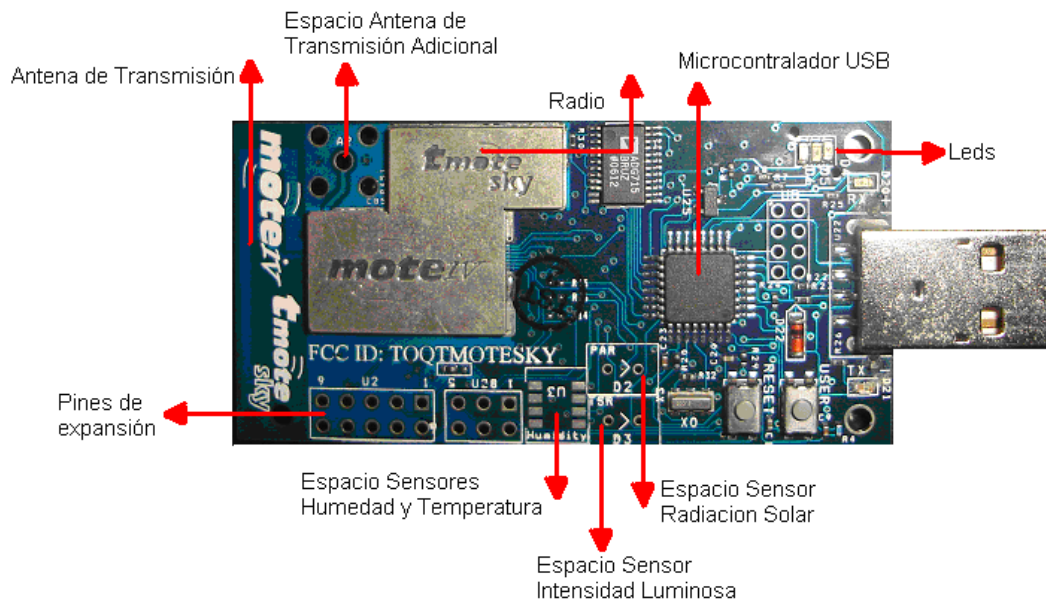
CROSSBOW [12]: Esta empresa desarrolla plataformas hardware y software que dan soluciones para las redes de sensores inalámbricas. Su especialidad, son los sensores, Entre sus productos encontramos las plataformas Mica, Mica2, Micaz, Mica2dot, telos y telosb.

MOTEIV: Joseph Polastre, antiguo doctorando de un grupo de trabajo de la Universidad de Berkeley formó la compañía Moteiv. Ha desarrollado la plataforma Tmote Sky y Tmote Invent. El tipo de mota Tmote Sky será detallado en el siguiente punto, dado que es una de las marcas utilizadas en el desarrollo del proyecto de grado.

SHOCKFISH: Empresa suiza que desarrolla TinyNode.

TMOTE SKY

Figura 8. Estructura de los mote Tmote Sky



Fuente: Autoras del proyecto

Tmote Sky es un dispositivo diseñado para consumos bajos de energía y altos niveles de recolección de datos en redes de sensores. Lleva integrados los sensores de luz, humedad y temperatura, la radio, la antena y el microcontrolador, y es de fácil programación. Las operaciones de baja energía en las Tmote Sky son realizadas gracias al microcontrolador MSP430 F1611. Este procesador RISC de 16 bits consume muy poca energía tanto en el estado activo como durante el sueño o hibernación.

Para reducir al máximo este consumo, permanece en modo ahorro (bajo consumo) dante la mayoría del tiempo, y cuando necesita procesar, recibir o enviar información, se despierta rápidamente, realiza la actividad correspondiente y vuelve a “dormirse”.

Entre las principales características de Tmote Sky se encuentran:

- Transmisor Chipcon inalámbrico de 250Kbps 2.4GHz IEEE 802.15.

- Interactúa con otros dispositivos IEEE 802.15.4
- Microcontrolador MSP430 Texas Instruments de 8MHz (10Kb de RAM y 48 Kb de Flash)ADC, DAC, supervisor de voltaje y controladora DMA integrada
- Antena, sensores de humedad, temperatura y luz
- Muy bajo consumo
- Rápido en despertar del sueño (<6 μ s)
- Hardware para encriptación y autenticación de la capa de enlace
- Programación y recogida de datos por USB
- 16 pines para soportar una expansión y conector de antena opcional SMA
- Ayuda de TinyOS: enrutamiento de malla e implementación de las comunicaciones
- Cumple con los requisitos FCC Parte 15

Los sensores de humedad/temperatura están manufacturados por Sensiron AG, producidos con procesadores CMOS y emparejado con un dispositivo ADC de 14 bits. Los coeficientes para estos sensores se guardan en la EEPROM. La precisión del sensor es de 0.5°C.

MOTEIV Telos

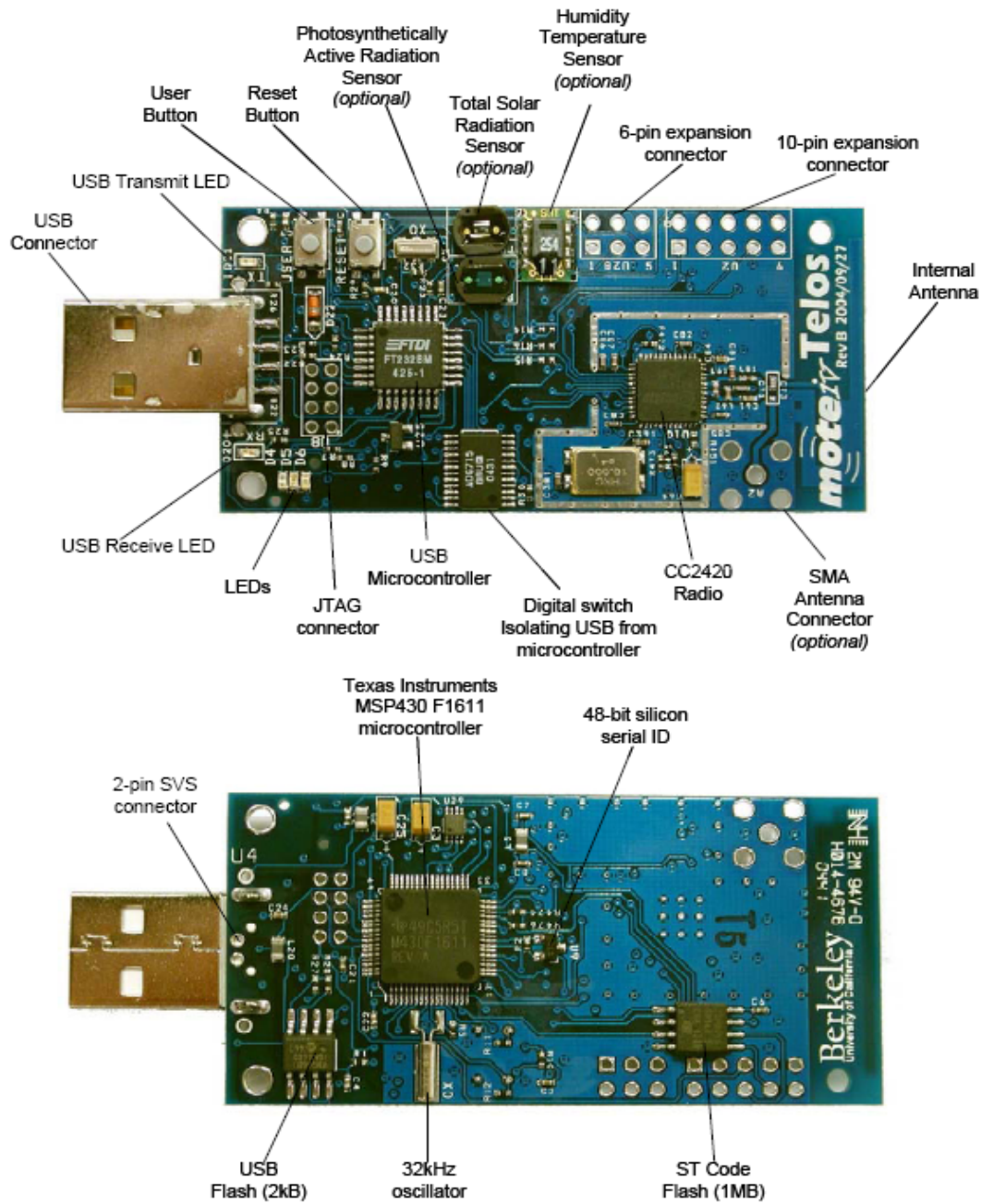
La Universidad de Berkeley diseño una serie de módulos para las investigaciones de sensores y les dio el nombre de motes. Estos dispositivos son construidos y comercializados por la empresa Crossbow convirtiéndose en una de las plataformas mas utilizadas en la actualidad por los grupos de investigación en sus trabajos con redes de sensores inalámbricos.

Los motes de referencia Telos funcionan bajo la especificación ZigBee para comunicarse. Tienen un procesador MSP430 fabricado por Texas Instruments y un transceiver Chipcon CC2420 [12]. En la figura a continuación observamos la estructura



y la composición de estos dispositivos.

Figura 9. Estructura de un mote Moteiv Telos



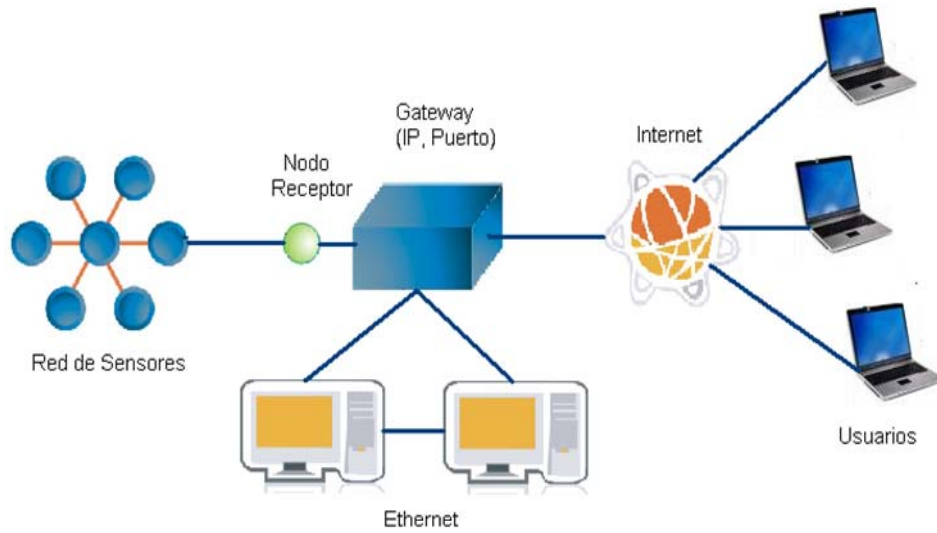
Fuente: www.moteiv.com

GATEWAY

Una puerta de enlace o Gateway es un dispositivo o equipo informático configurado para interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Generalmente realizan operaciones de traducción de direcciones IP (NAT: Network Address Translation). Esta capacidad de traducción de direcciones permite aplicar una técnica llamada IP Masquerading (enmascaramiento de IP), usada para dar acceso a internet a los equipos de una red de área local compartiendo una única conexión a Internet, y por tanto, una única dirección IP externa.

Para efectos del prototipo desarrollado en ésta tesis de grado, se utilizó un dispositivo Linksys NSLU2 Network Attached Storage como gateway utilizando el software Tmote Connect. Los motes Tmote Sky que actúan como receptores, se conectan USB a dos puertos asignados por defecto: el 9001 y el 9002. Además el gateway tiene una interfaz gráfica desarrollada para configurar este dispositivo en la que se asigna la dirección IP pública por medio de la cual se accede a la información que los motes transmiten a través de internet. La infraestructura de comunicación del gateway inalámbrico permite la conexión entre la red de sensores y una red local (Ethernet) o Internet, como se puede apreciar en la figura a continuación.

Figura 10. Prototipo de uso del gateway



Fuente: Autoras del proyecto

El hardware del gateway utilizado durante la etapa final de pruebas del proyecto de grado se muestra en la figura 13.

Figura 11. Gateway



Fuente: <http://www.sentilla.com/pdf/eol/tmote-connect-datasheet.pdf>

Características del Gateway

- ✓ Es un puente entre la red de sensores inalámbricos y el internet
- ✓ Soporta hasta dos motes conectados (nodos receptores) vía USB
- ✓ Conexión bi-direccional para transferencia de datos desde y hacia e mote receptor a través de sockets TCP / IP
- ✓ Integración con el sistema operativo TinyOS en sus versiones 1.x y 2.x y con todas sus herramientas
- ✓ Interface web (mostrada en la figura) donde se identifica el estado de los motes receptores conectados a los dos puertos disponibles del gateway. Además se puede resetear y revisar los contadores de la cantidad de paquetes leídos.
- ✓ Funciona en redes con DHCP.
- ✓ Actualizable para las nuevas funciones del software de Moteiv.

Figura 12. Interface Web Tmote connect

File Edit View Go Bookmarks Tools Help

http://192.168.4.104/

Getting Started Latest Headlines

moteiv
Devices Consulting Applications
Wireless Sensor Networks

PRODUCTS SERVICES SUPPORT ABOUT

Welcome to **tmote connect**: Ethernet/tmote gateway Refresh: off | 30 seconds

tmote connect

LKG7DD7D4
192.168.4.104
00:0F:66:7D:D7:D4

Reset SF/BSL Servers
Forcibly disconnects all clients and restarts the servers.

Reboot System
Takes about 1 minute

Device 1 - M4MWCJF0		Device 2 - M49X80SS	
Moteiv Telos (Rev A 2004-04-27)		Moteiv tmote sky	
Clients	0	Clients	0
Packets read	0	Packets read	4105434
Packets written	0	Packets written	0
Serial forwarder	port 9001	Serial forwarder	port 9002
Control, status	port 10001	Control, status	port 10002
Reset device 1: M4MWCJF0		Reset device 2: M49X80SS	

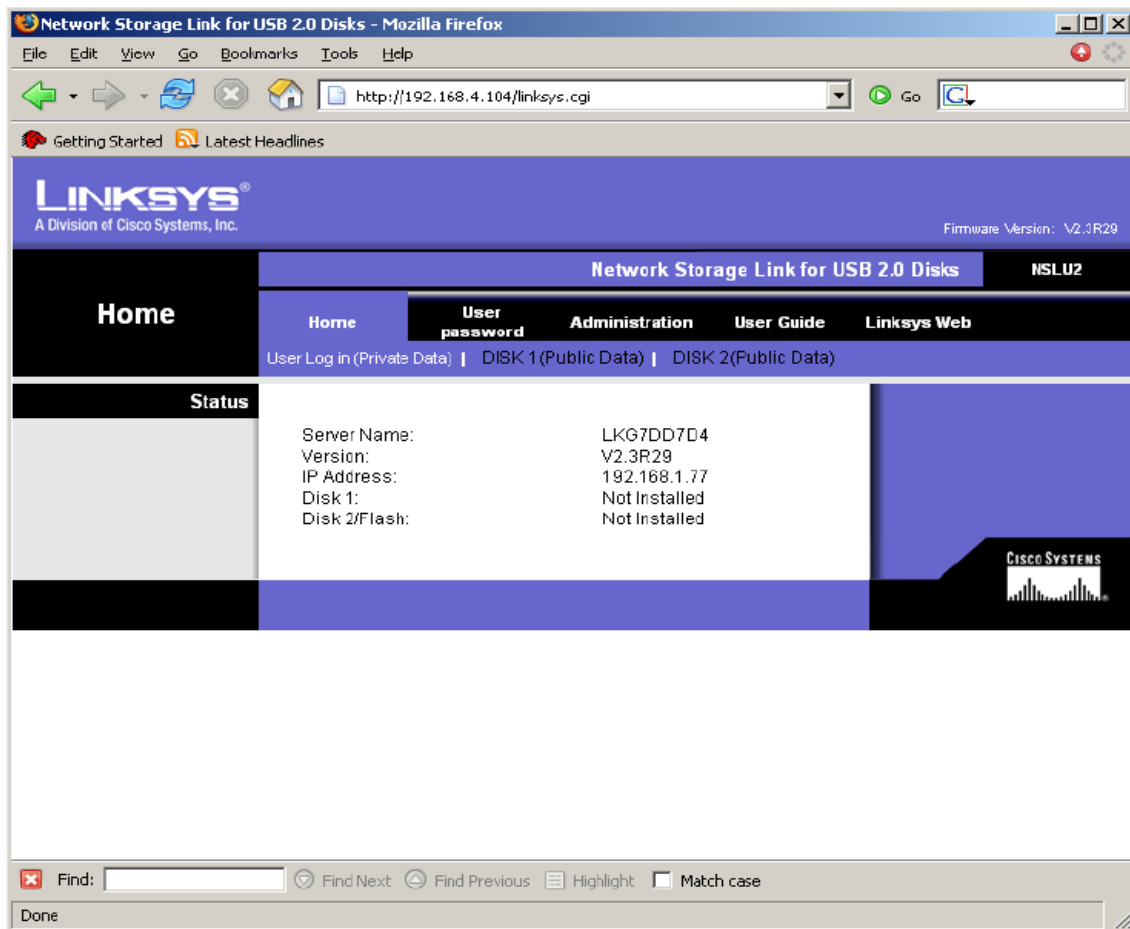
tmote connect version 1.0 510.965.1312 | info@moteiv.com

Instalación del Gateway

1. Conectar el puerto de ethernet del gateway al punto en el switch o hub que permite la conexión con la red local o con Internet.
2. Conectar los motes receptores en los puertos USB.
3. Conectar el gateway a una fuente de energía.
4. Después de encendido, el gateway realizará una prueba automática de su funcionamiento seguida del proceso de arranque que al finalizar emitirá un zumbido.
5. Si se conecta un mote al puerto USB 1 del gateway, se debe encender el led1 e igualmente sucede si se conecta un mote al puerto USB 2.
6. Luego de hacer la conexión física, se puede acceder localmente a la interfaz web utilizando la dirección IP que trae configurada por defecto el gateway como se muestra

en la siguiente figura. La URL de acceso a la página de configuración es <http://192.168.4.104/linksys.cgi>.

Figura 13. Interfaz de configuración



Por medio de la interfaz es posible cambiar el usuario *admin* y la contraseña *admin* que vienen por defecto al igual que la dirección IP establecida para la transmisión de la información.

7. Finalmente y después de poner uno o más motes a transmitir, se accede al gateway desde la consola Cygwin con las siguientes instrucciones:

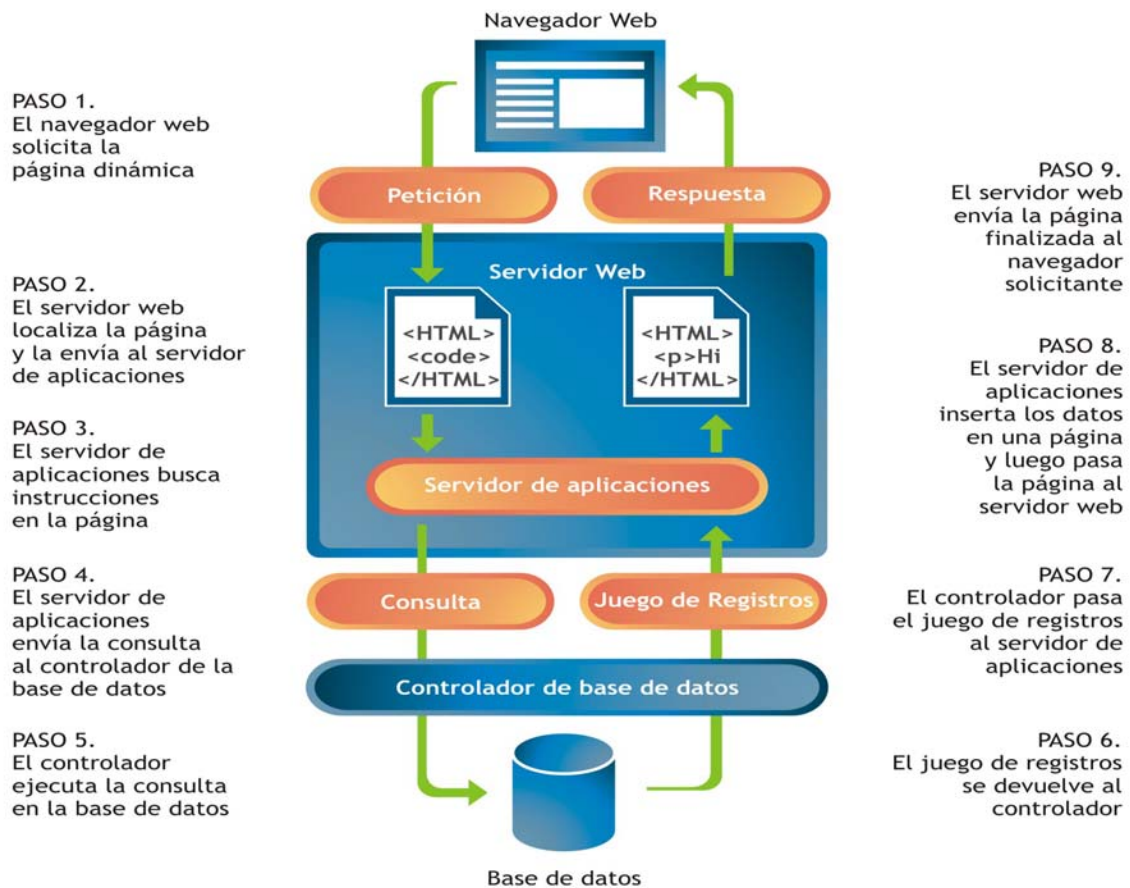
export MOTECOM=serial @ IP-Gateway: Puerto de conexión (9001 ó 9002)

MOTECOM=serial @ IP-Gateway: Puerto de conexión java net.tinyos.tools.Listen

SOFTWARE UTILIZADO

Una aplicación Web está comúnmente estructurada como una aplicación de tres capas, donde el navegador Web es la primera capa, un motor usando alguna tecnología Web dinámica (por ejemplo CGI, PHP, Java Servlets o ASP) es la segunda capa y una base de datos es la última capa. El navegador Web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

Figura 14. Proceso de consulta de base de datos y de devolución de los datos al navegador



Fuente: Adaptada por las autoras de:
http://livedocs.adobe.com/dreamweaver/8_es/using/images/process4.jpg

ESTUDIO DE HERRAMIENTAS DE DESARROLLO

1. JSP

Solución implementada por Sun Microsystems, utiliza el lenguaje java y una extensa investigación en estándares del software para brindar un perfecto paradigma de objetos un diseño de 3 capas correctamente definido (algo normal en la casa Sun y su lema para java, “codifica una vez, corre donde sea”). Jsp tiene soporte para todas las plataformas populares, desde el principio a tenido una amplia variedad de librerías funcionales gracias al carácter open-source de los proyectos realizados en su padre

java, esto se encuentra en aumento, desde que liberaron el código de la mismísima plataforma, estas características lo han convertido en los preferidos de las corporaciones, pero lo han alejado de los usuarios comunes, debido a que la implementación de toda esa teoría del software hace que la curva de aprendizaje de esta tecnología sea casi una cultura de vida.

Se descartó Jsp para la realización del proyecto debido a que la curva de aprendizaje es muy elevada y extendería el proceso de desarrollo.

2. ASP

Solución implementada por Microsoft, nacido del VisualBasic, tiene una excelente interoperabilidad entre los productos de Microsoft, como los son ODBC, VBscript, además de un gran soporte dado por se casa matriz, su IDE (Internas de desarrollo) y características adicionales, facilitan el uso, y tiene una curva de aprendizaje que permite una escalabilidad moderada, cabe destacar que asta ahora se esta trabajando en el soporte para otras plataformas diferentes a windows, pero las demás tecnologías le tienen ventaja en este campo.

Se descarta Asp para la realización del proyecto debido a su carácter privativo, costo y poca portabilidad.

Selección de PHP

PHP es un lenguaje apropiado para el desarrollo de esta tesis por las ventajas comparativas con otros lenguajes; por ejemplo, la forma fácil y rápida de diseñar aplicaciones web (finalidad con la que fue desarrollado este lenguaje), además PHP es un lenguaje multiplataforma, garantía de que cualquier código se pueda ejecutar en servidores Linux, Windows, etc. Finalmente por ser código abierto se consigue gratuita y rápidamente información, tutoriales y recursos de apoyo para desarrollar en PHP.

ESTUDIO DE HERRAMIENTAS PARA BASES DE DATOS

1. *PostgreSQL:*

Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características interesantes, entre las más relevantes para el proyecto a desarrollar incluyen manejo de objetos, Almacenamiento y búsqueda de XML de forma nativa, además de ofrecer tamaños ilimitados de la base de datos y un rendimiento muy elevado en las consultas. Permite instalar el sistema de administración en entorno Web o aplicación de escritorio. Es libre y con miles de fuentes de documentación, aunque no es muy fácil de manejar si no se es un usuario experto.

2. *Oracle:*

Es el sistema de Bases de datos de tipo propietario con las funciones más avanzadas, está enormemente adelantado a otros sistemas como MySQL y PostgreSQL debido a que además de incluir todas las funciones que manejan éstos, incluyendo el manejo nativo de XML con más de 1 año de ventaja, añade todo un sistema de “Advisors” (Consejeros) que son monitores del sistema que permiten una administración sencilla en medio de un complejo sistema. Al igual que PostgreSQL, hace uso del modelo Objeto-Relacional para almacenar los datos, un punto fuerte a favor. Tiene en contra que se debe tener licencia para usarlo, la cual es muy costosa.

3. *MySQL:*

MySQL es un sistema de gestión de base de datos, multiplataforma, multihilo y

multiusuario con más de seis millones de instalaciones, lo cual lo hace el más popular. En éstos momentos no contempla manejo de XML de forma nativa, sin embargo se planea implementar éstas funciones en una próxima versión. Tiene un límite máximo de 50 millones de registros, lo cual es más que suficiente para una base de datos pequeña o mediana. Su rendimiento es muy bueno, aunque no comparable al de PostgreSQL u Oracle. En cuanto a facilidad de uso, MySQL tiene una enorme ventaja con respecto a PostgreSQL, está pensado para que una persona con conocimientos muy básicos de bases de datos pueda manejarlo.

Selección de MySQL

MySQL es un sistema de gestión de bases de datos recomendable para el proyecto, por su gran facilidad de uso, por disponer de las capacidades necesarias para la ejecución del proyecto, y por la amplia documentación existente sobre ésta herramienta.

En este modelo se visualiza el proyecto en versiones futuras y permite ampliar el tema del diseño y la construcción de la base de datos, pensada en un escenario de múltiples granjas avícolas separadas geográficamente conteniendo cada una un volumen de información considerable y almacenando dicha información en un único servidor; para este escenario que cuenta con mayor información a la trabajada en esta primera etapa de prueba con esta tesis de grado, se debe pensar en motores mas grandes de bases de datos y la implementación de tecnologías para optimizar el acceso a la información como son los Cubos de Información.

La base de datos diseñada y construida para esta tesis se acomodó a las necesidades del proyecto de investigación del equipo CPS y resultó en un modelo sencillo conformado por cuatro tablas: usuarios, nodos, info_nodos y mortandad.

En la tabla de usuarios, se almacena la información solicitada en el formulario de

registro de la aplicación web. Después del registro el administrador valida la información y establece un tipo de usuario que lleva consigo la asignación de los permisos de acceso a los módulos de la aplicación y lo almacena dentro de esta tabla en la base de datos.

La tabla nodos contiene información sobre la posición en la que fue ubicado un nodo, el ID y la fecha en que fue activado y puesto en funcionamiento. Esta tabla es alimentada con la información que adquiere la aplicación cuando se crea gráficamente la topología y se insertan los ID programados a cada mote creado.

En la tabla info_nodos se registra la información de las variables que están midiendo los motes a través de los sensores instalados y que son capturadas por los algoritmos desarrollados e implementados durante esta fase de la investigación. En este caso se insertan datos de temperatura, humedad, intensidad luminosa y radiación solar. Además de esto, se almacena la fecha y hora en que se recibió e insertó el dato a la base de datos.

Finalmente se creó la tabla mortandad que contiene la información del formulario de registro de aves muertas encontrado en la aplicación web. Esta inserción contiene la fecha en que se hace el registro y la cantidad de aves que murieron en esa fecha.

Este diseño se hizo pensando en hacer una base de datos escalable, que a futuro permita no solo incluir datos de otros galpones si no también jugar con el número de motes implementado en cada red. De igual forma es posible acceder a nuevas variables dependiendo de los sensores y las aplicaciones utilizadas, siempre y cuando la aplicación web se modifique para tal fin.

Después de hacer el estudio de la cantidad de registros que la base de datos debe almacenar en un día, un mes y un año, y conociendo que MySQL almacena hasta 2 TB

de información se determinó que la base de datos escogida, MySQL y su motor InnoDB eran adecuados para esta primera fase del desarrollo, pues con este modelo experimental los motes transmitirán 8.640 registros por mes, lo que equivale a 103.680 registros por año y por nodo y si se habla de 30 nodos por galpón, se tendrán 3.110.400 datos. Haciendo el mantenimiento normal que se le debe realizar a todo sistema con la eliminación frecuente de archivos corruptos, backups periódicos y demás procedimientos de administración rutinarios la elección de MySQL se considera adecuada.

El diseño de la base de datos además de ser escalable, requiere rapidez en las consultas y debido a las grandes cantidades de registros, los tiempos de acceso a la información se hacen importantísimos a la hora de hablar de la base de datos y el motor usado.

Cubos De Información (Olap)

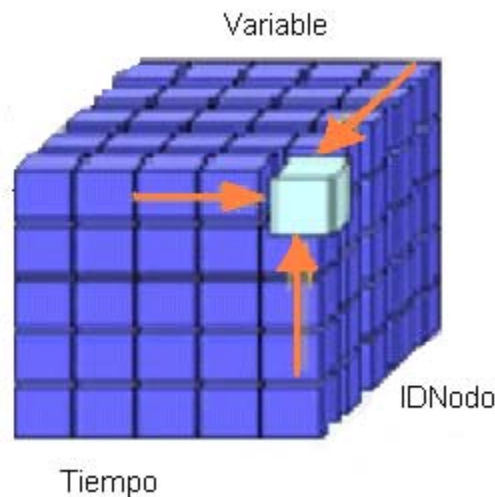
A medida que el modelo planteado para el proyecto de investigación va creciendo y se acerca al final, se debe pensar en la forma de organizar grandes cantidades de información para que el acceso a los datos históricos no sea demasiado lento. Los cubos de información son una metodología utilizada para la optimización del almacenamiento y por ende del acceso a la información. Organiza los datos que se deben acceder constantemente en una sola tabla y la visualiza como un cubo con tres campos por los que guío las búsquedas.

El crecimiento del cubo de información o de las tablas como tal es dinámico y permite crecer tanto como se necesite en cantidad de registros almacenados. La base de datos que se diseñó para este proyecto se adapta fácilmente a este concepto, ya que el número de registros de la tabla principal, info_nodos, crece rápidamente con las lecturas tomadas por los motes. Esta tabla se creó para que contenga toda la

información enviada por los “n” nodos que conforman la red, por lo que es fácil pensar en formar un cubo de información con estos datos, de tal modo que facilite el acceso a la información y los tiempos de consulta se hagan cortos.

El cubo puede tener como base, tres campos principales de esta tabla: la variable, el tiempo y el identificador del nodo, y partiendo de esta información se puede encontrar cualquier registro que se necesite rápidamente. A continuación se muestra gráficamente como se vería el diseño de la base de datos actual aplicando el concepto de cubos de información.

Figura 15. Modelo Cubos de información aplicado a la Base de Datos del proyecto.



Además, cuando se llegue al final de toda la investigación, el modelo general planteado va a contener mucha mas información que la que tenemos ahora y será de gran utilidad aplicar este concepto a ese diseño final de la base de datos y tener en cuenta no solo la variable y el nodo si no también el galpón donde se encuentra ese nodo de donde se esta registrando un dato.

Actualmente, MySQL no incluye este módulo para la creación de cubos de información a partir de una base de datos, pero otros motores pagos como Oracle o el mismo

SQLServer tienen esta propiedad y automáticamente se puede hacer de las bases de datos ordinarias, bases de datos multidimensionales y óptimas para almacenar cualquier cantidad de información.

UTILIZACIÓN DEL WAMP 5 VERSIÓN 2.0

Después de escoger las herramientas MySQL y PHP como base para el desarrollo de la aplicación web REDSI, se pensó en utilizar una herramienta que las contuviera junto con la configuración necesaria para que interactuasen entre ellas y se escogió WampServer que es una plataforma de desarrollo Web bajo Windows. Esta solución permite desarrollar aplicaciones web dinámicas utilizando el servidor Apache2 versión 2.2.6, el lenguaje PHP versión 5.2.5 y la base de datos MySQL versión 5.0.45. También incluye phpMyAdmin y SQLite, útiles para administrar las bases de datos.

WampServer tiene una interfaz de fácil manejo y su uso es muy intuitivo ya que las configuraciones son predeterminadas y los drivers necesarios para la conexión a la base de datos, instalados por defecto. WampServer también tiene un "TrayIcon" (icono con un menú desplegable), para administrar y configurar su base de datos, su aplicación web y las herramientas que trae consigo esta solución, sin tocar los ficheros de configuración. Además admite añadir otras versiones de Apache, MySQL y PHP.

Figura 16. TRYICON (Icono con menú desplegable para administrar WampServer)



El proceso de instalación y manejo de esta herramienta se encuentra descrito en el anexo 1.

ANEXO 3

CAPTURA Y TRANSFORMACIÓN DE DATOS A TRAVÉS DEL GATEWAY

En el modelo general de éste proyecto de grado es claro que la red de sensores ubicada en un determinado ambiente y con una topología previamente estudiada y determinada, se comunica inalámbricamente con un nodo receptor para enviarle una trama de datos TCP-IP con la información sobre las lecturas realizadas por los sensores que tiene cada nodo.

La orden para que los nodos lleven a cabo esta tarea se le programa através del sistema operativo Tinyos, en su lenguaje asociado, NesC. A cada nodo trasmisor se le instala un programa llamado Oscilloscope, incluido dentro de TinyOS en el directorio apps de Moteiv.

El procedimiento se lleva a cabo utilizando Cygwin, una aplicación que simula un entorno linux y que sirve de interfaz para ejecutar todas las aplicaciones incluidas en el sistema operativo TinyOS.

Se inicia abriendo Cygwin y se ubica el directorio principal de TinyOS. Allí se encuentra la carpeta *opt* y dentro de ella los directorios *Moteiv* y *TinyOS-1.x*. Desde ahí es posible acceder al *apps* y a todas las aplicaciones desarrolladas en java y que se encuentran en estos sub-directorios.

Dentro de la carpeta *Moteiv* se encuentran dos aplicaciones importantes que se utilizaron dentro del desarrollo de la tesis: Oscilloscope y TOSBase. La primera es la aplicación que se programa a los nodos para que funcionen como transmisores de la información captada por sus sensores y la segunda aplicación se le programa al nodo designado como el nodo receptor y que se conecta a un Gateway para poder recibir y

manipular dicha información.

En la figura 17 se muestra el directorio apps que contiene las aplicaciones mencionadas y al que debo ingresar desde cywing para programar el nodo.

Figura 17. Directorio apps de TinyOS

```

Trend@vm-trend ~
$ cd ..

Trend@vm-trend /home
$ cd ..

Trend@vm-trend /
$ cd opt/moteiv/apps

Trend@vm-trend /opt/moteiv/apps
$ dir
Count Delta Oscilloscope RadioDemo TOSBase invent

Trend@vm-trend /opt/moteiv/apps
$
  
```

Para programar el *Oscilloscope* en los nodos se entra al directorio `moteiv / apps / oscilloscope`, y se ejecuta la instrucción:

make tmote reinstall, ID del Nodo

En donde el ID del nodo es el número que debe asociarse a cada nodo para su identificación en la sarta de datos que transmitirá.

```

/opt/moteiv/apps/oscilloscope
Biviana@emery /opt
$ cd moteiv
Biviana@emery /opt/moteiv
$ dir
apps doc tinyos-1.x tools tos
Biviana@emery /opt/moteiv
$ cd apps
Biviana@emery /opt/moteiv/apps
$ dir
Count Delta Oscilloscope RadioDemo TOSBase invent
Biviana@emery /opt/moteiv/apps
$ cd oscilloscope
Biviana@emery /opt/moteiv/apps/oscilloscope
$ dir
Makefile                               OscilloscopeTmoteSky.nc   README.TmoteSky
OscilloscopeTmoteInvent.nc           OscilloscopeTmoteSkyM.nc
OscilloscopeTmoteInventM.nc         README.TmoteInvent
Biviana@emery /opt/moteiv/apps/oscilloscope
$ ~
$

```

```

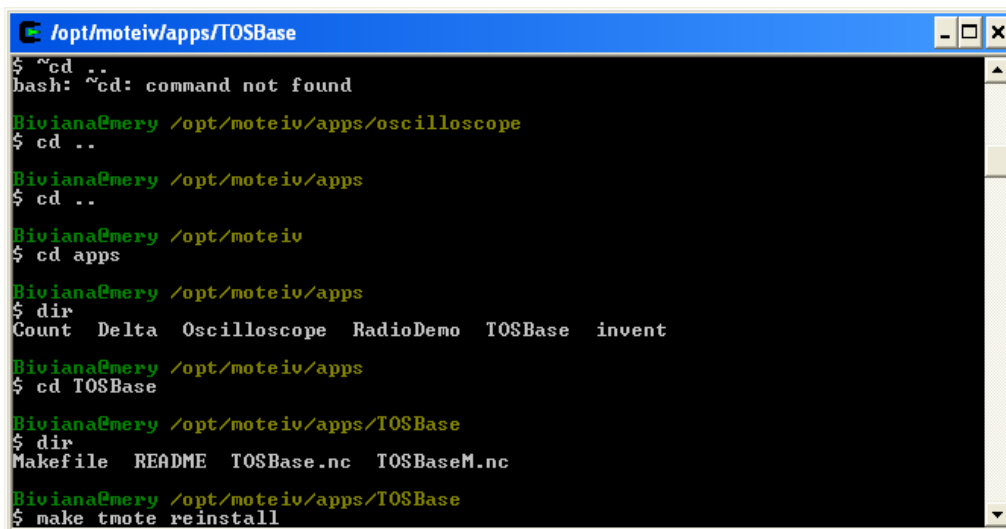
/opt/moteiv/apps
Trend@vm-trend ~
$ cd ..
Trend@vm-trend /home
$ cd ..
Trend@vm-trend /
$ cd opt/moteiv/apps
Trend@vm-trend /opt/moteiv/apps
$ make tmote reinstall, 1

```

Luego de instalar el programa oscilloscope en los nodos trasmisores, se debe escoger el nodo receptor y se le debe instalar el programa TOSBase ubicado en el directorio: moteiv / apps / TOSBase. La instrucción a ejecutar es:

make tmote reinstall

En este caso no se asigna un ID de nodo porque este nodo no envía ningún tipo de información, no está diseñado para tomar muestras y no tiene sensores dentro de su infraestructura física.



```

/opt/moteiv/apps/TOSBase
$ ~cd
bash: ~cd: command not found
Biviana@emery /opt/moteiv/apps/oscilloscope
$ cd ..
Biviana@emery /opt/moteiv/apps
$ cd ..
Biviana@emery /opt/moteiv
$ cd apps
Biviana@emery /opt/moteiv/apps
$ dir
Count Delta Oscilloscope RadioDemo TOSBase invent
Biviana@emery /opt/moteiv/apps
$ cd TOSBase
Biviana@emery /opt/moteiv/apps/TOSBase
$ dir
Makefile README TOSBase.nc TOSBaseM.nc
Biviana@emery /opt/moteiv/apps/TOSBase
$ make tmote reinstall

```

Luego de instalar los programas, el nodo receptor, se conecta al Gateway vía USB y se puede acceder a el y a la información que le esta llegando de toda la red de sensores inalámbricos, ejecutando el siguiente comando:

export MOTECOM=sf @ IPGateway : Puerto de comunicación del Gateway

La IP del gateway debe ser una IP pública a la que se pueda acceder vía Internet, y el puerto de comunicación del gateway viene definido por defecto y para efectos del desarrollo del proyecto, se utiliza el 9001 o el 9002, dependiendo del puerto físico al que se conecte el nodo receptor.

A continuación se ejecuta la siguiente instrucción para poder ver en la interfaz de Cygwin los datos que están llegando al sensor receptor y que se pueden acceder y capturar através del Gateway.

MOTECOM=sf @ IPGateway : Puerto Gateway java net.tinyos.tools.Listen

```

/opt/moteiv/apps/oscilloscope
$ java net.tinyos.tools.Listen
serialCOM6:57600: resynchronising
1A 01 08 5D FF FF FF FF 0A 7D 04 00 50 00 00 00 C9 04 C9 04 CA 04 CA 04 C9 04 C9
04 CA 04 C9 04 C9 04 C8 04
1A 01 08 5E FF FF FF FF 0A 7D 04 00 50 00 01 00 99 19 99 19 99 19 99 19 99
19 99 19 99 19 99 19 9B 19
1A 01 08 5F FF FF FF FF 0A 7D 04 00 50 00 02 00 12 00 1A 00 20 00 11 00 13 00 16
00 11 00 16 00 1D 00 17 00
1A 01 08 60 FF FF FF FF 0A 7D 04 00 50 00 03 00 15 00 16 00 12 00 1A 00 23 00 12
00 10 00 16 00 12 00 14 00
1A 01 08 61 FF FF FF FF 0A 7D 04 00 50 00 04 00 8A 0B 87 0B 8A 0B 8B 0B 8A 0B 8A
0B 8A 0B 8B 0B 8B 0B 8A 0B
1A 01 08 62 FF FF FF FF 0A 7D 04 00 50 00 05 00 27 09 27 09 27 09 27 09 27 09 27
09 27 09 27 09 28 09 27 09
1A 01 08 63 FF FF FF FF 0A 7D 04 00 5A 00 00 00 C8 04 C7 04 C6 04 C4 04 C2 04 C0
04 BF 04 BF 04 BF 04 BE 04
1A 01 08 64 FF FF FF FF 0A 7D 04 00 5A 00 01 00 9E 19 9E 19 A0 19 A1 19 A2 19 A2
19 A4 19 A5 19 A6 19 A7 19
1A 01 08 65 FF FF FF FF 0A 7D 04 00 5A 00 02 00 0F 00 13 00 16 00 0D 00 10 00 15
00 19 00 1D 00 07 00 11 00
1A 01 08 66 FF FF FF FF 0A 7D 04 00 5A 00 03 00 1C 00 23 00 18 00 07 00 11 00 11
00 0E 00 13 00 17 00 12 00
1A 01 08 67 FF FF FF FF 0A 7D 04 00 5A 00 04 00 8B 0B 8B 0B 8B 0B 8B 0B 8E 0B 8E
0B 8E 0B 8E 0B 8B 0B 8E 0B
1A 01 08 68 FF FF FF FF 0A 7D 04 00 5A 00 05 00 29 09 27 09 27 09 29 09 27 09 29
09 29 09 27 09 27 09 27 09
1A 01 08 69 FF FF FF FF 0A 7D 04 00 64 00 00 00 BC 04 BA 04 B8 04 B7 04 B6 04 B6
04 B5 04 B6 04 B6 04 B6 04 B6 04
1A 01 08 6A FF FF FF FF 0A 7D 04 00 64 00 01 00 A8 19 AA 19 AB 19 AD 19 AD 19 AE
19 AF 19 AF 19 B0 19 B3 19
1A 01 08 6B FF FF FF FF 0A 7D 04 00 64 00 02 00 11 00 0E 00 13 00 17 00 1C 00 0A
00 11 00 0E 00 12 00 17 00
1A 01 08 6C FF FF FF FF 0A 7D 04 00 64 00 03 00 0B 00 0F 00 12 00 0E 00 12 00 18
00 1E 00 09 00 11 00 0E 00
1A 01 08 6D FF FF FF FF 0A 7D 04 00 64 00 04 00 8B 0B 8E 0B 8E 0B 8E 0B 8E 0B 8E
0B 8E 0B 8F 0B 8F 0B 8E 0B
1A 01 08 6E FF FF FF FF 0A 7D 04 00 64 00 05 00 29 09 27 09 29 09 29 09 27 09 29
09 29 09 29 09 29 09 29 09
1C 01 08 6F FF FF FF FF 0A 7D 04 00 02 00 FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF 00 00 FF FF 00 FF 06 00
1A 01 08 70 FF FF FF FF 0A 7D 04 00 6E 00 00 00 B6 04 B6 04 B6 04 B5 04 C7 04 D2
05 72 06 96 06 E4 06 02 07
1A 01 08 71 FF FF FF FF 0A 7D 04 00 6E 00 01 00 B2 19 B4 19 B6 19 B5 19 BB 19 C1
19 C5 19 C8 19 CD 19 D2 19
1A 01 08 72 FF FF FF FF 0A 7D 04 00 6E 00 02 00 19 00 0C 00 08 00 05 00 08 00 09
00 04 00 06 00 06 00 06 00
1A 01 08 73 FF FF FF FF 0A 7D 04 00 6E 00 03 00 19 00 17 00 0F 00 04 00 0F 00 0F

```

Luego de ejecutar la última instrucción, las tramas de datos enviados por los nodos empiezan a llegar cada cierto tiempo, tiempo que también debe ser programado en los nodos de acuerdo a las necesidades que se tengan. Cada lectura consta de cinco tramas de datos que representan las variables que se están midiendo de acuerdo a los sensores que tiene cada dispositivo y las variables internas, propias del nodo. La trama de datos recibida por cada variable medida es la siguiente:

```

1A 01 08 FF FF FF FF FF 0A 7D 04 00 4A 01 00 00 6B 04 6C 04 6C 04 6B 04 6C 04 6C
04 6C 04 6B 04 6B 04 6B 04

```

El 04 encerrado entre dos líneas azules representa el numero del nodo que está

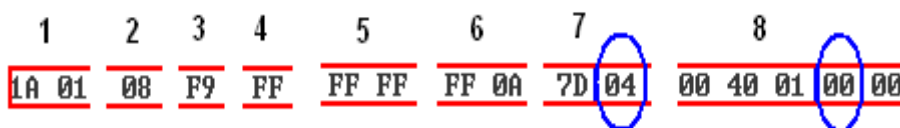
enviando la trama de datos, este número, se le da al nodo cuando se le esta instalando el programa con el que se quiere que funcione.

El 00 que también está señalado entre dos líneas azules, representa el canal de trasmisión o la variable de la cual es la muestra que se esta capturando. En este caso 00 representa la variable Humedad Relativa y esto se sabe por la documentación del programa osciloscope de Tinyos que es el que se le instala al nodo para la implementación y desarrollo de este proyecto de grado.

Esta primera trama de datos de la sarta, representa el encabezado y de la cual solo se tiene en cuenta los datos para almacenar en la base de datos que son el ID del nodo y el ID de la variable.

1A 01 08 F9 FF FF FF FF 0A 7D 04 00 40 01 00 00

1. Destination address (2 bytes)
2. Active Message handler ID (1 byte)
3. Group ID (1 byte)
4. Message length (1 byte)
- 5-6-7-8. Payload (up to 29 bytes):
 - source mote ID (2 bytes)
 - sample counter (2 bytes)
 - ADC channel (2 bytes)



El resto de la trama contiene las lecturas de las variables tomadas por los nodos. Esta trama se divide en 10 lecturas de 2 bytes cada una, que están dadas en Hexadecimal y deben ser transformadas a decimal y promediadas para obtener un solo dato, que es el que finalmente se almacena en la base de datos.

- ADC data readings (10 readings of 2 bytes each)

70 04 6F 04 6F 04 6E 04 6E 04 6E 04 6E 04 6E 04 6C 04 6C 04

Para llevar la trama completa a un solo dato, se divide y se hace la conversión a decimal cada dos bytes. Por ejemplo:

70 04

De los dos primeros bytes, el más significativo es 04 y el menos significativo es 70, y debido al formato “little endian” en que los nodos transmiten la información, se deben leer en este sentido para hacer la transformación de cada lectura, por lo que para efectos de este ejemplo, 70 04 debe ser leído como 0470 en Hexadecimal y que equivale a 1136 en decimal. Este procedimiento debe realizarse para las nueve lecturas restantes. Luego de tener los diez datos en decimal, se promedian para poder aplicar la fórmula respectiva a cada variable leída y obtener el dato que se guardará en la base de datos.

Las fórmulas para calcular la humedad relativa, la temperatura, la intensidad luminosa y la radiación solar a partir de los datos promedios obtenidos son las siguientes:

HUMEDAD, el canal o ID que lo representa es el “0”y la fórmula para obtener el dato ajustado a la realidad es la siguiente:

$$\text{humidity} = -4 + 0.0405 * \text{prom} + (-2.8 * 10^{-6}) * (\text{prom}^2)$$

Donde prom, es el dato promedio obtenido de las diez lecturas del sensor de humedad dentro del nodo.

TEMPERATURA, el canal o ID que la representa es el “1” y la fórmula obtener el dato ajustado a la realidad, en grados Celsius es la siguiente:

$$\text{Temperatura} = -39.60 + 0.01 * \text{prom}$$

Donde prom, es el dato promedio obtenido de las diez lecturas de este canal del sensor de temperatura dentro del nodo.

INTENSIDAD LUMINOSA Y RADIACIÓN SOLAR, los canales o ID que los representan son “2” y “3” respectivamente y la fórmula obtener el dato ajustado a la realidad es la siguiente:

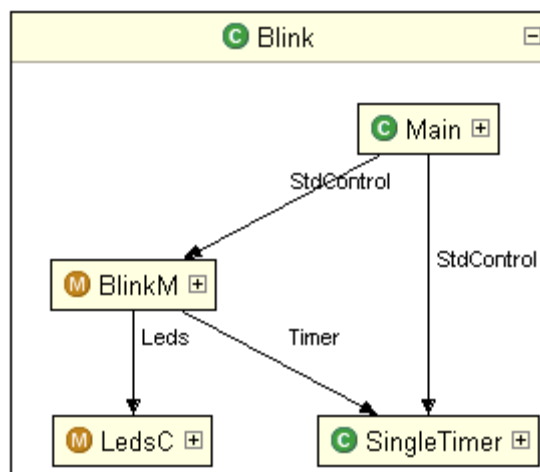
$$\text{Luz} = (\text{prom} * 1.5) / (4095 * 100,000),$$

Donde prom es el dato promedio obtenido de las diez lecturas enviadas por estos canales.

ANEXO 4

Código fuente del archivo Blink.nc

Figura 1. Componente gráfico de Blink.nc



```

configuration Blink {
}

implementation {
    components Main, BlinkM, ClockC, LedsC;

    Main.StdControl -> BlinkM.StdControl;
    BlinkM.Clock -> ClockC;
    BlinkM.Leds -> LedsC;
}

```

Código fuente del archivo BlinkM.nc



```

módulo BlinkM (
    proporciona la interfaz StdControl;
    utiliza la interfaz de reloj;
    utiliza la interfaz Leds;
)

aplicación (
    bool estado; /* el estado del LED rojo (encendido o apagado) */
    /* Iniciar el componente.
    * @ Retorno siempre devuelve ÉXITO */
    comando result_t StdControl.init () (
        state = false;

```

```

Leds.init llamada ();
Devuelve un valor verdadero;
)
/* Iniciar las cosas. Esto sólo establece la tasa para el componente del reloj.
* @ Retorno siempre devuelve ÉXITO */
comando result_t StdControl.start () (
    Clock.setRate llamada de retorno (TOS_I1PS, TOS_S1PS);
)

/* Detener la ejecución de la aplicación.
* La presente sólo desactiva el componente del reloj.
* @ Retorno siempre devuelve ÉXITO */
comando result_t StdControl.stop () (
    Clock.setRate llamada de retorno (TOS_I0PS, TOS_S0PS);
)

/* Cambiar el LED rojo en respuesta a la Clock.fire evento.
* @ Retorno siempre devuelve ÉXITO */
caso result_t Clock.fire () (
    state =! estado;
    if (estado) (
        Leds.redOn llamada ();
    ) Else (
        Leds.redOff llamada ();
    )
    Devuelve un valor verdadero;
)
)

```

6.2.2 Oscilloscope Modificado

Como se enunció anteriormente, los componentes del código nesC son los módulos y las configuraciones; y para apreciar mejor dichos elementos, se usó el osciloscopio, una aplicación del tinys y mediante modificaciones realizadas permite leer otras variables adicionales a las inicialmente estudiadas, amoniaco, monóxido de carbono y velocidad del viento.

A continuación se muestra mediante su componente gráfico la estructura del código de osciloscopio modificado.

Figura 2. Estructura de Osciloscopio



ANEXO 5

EVIDENCIAS FOTOGRÁFICAS

En este anexo mostramos algunas fotografías que permiten dimensionar el escenario actual del galpón donde el grupo CPS está realizando sus estudios.

Inicialmente el galpón es dividido en 4 zonas, cada una de 25m x 12m y con dos círculos demarcados dentro de ellas para almacenar aproximadamente 2000 aves.



Durante este tiempo los integrantes del grupo realizan visitas y registran las observaciones de los estados de los termómetros y de las aves.





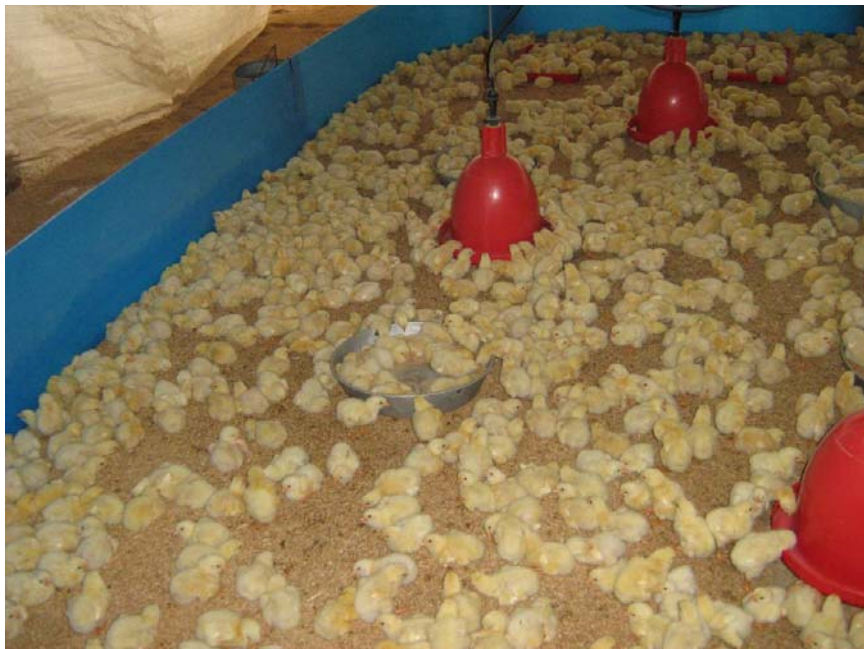
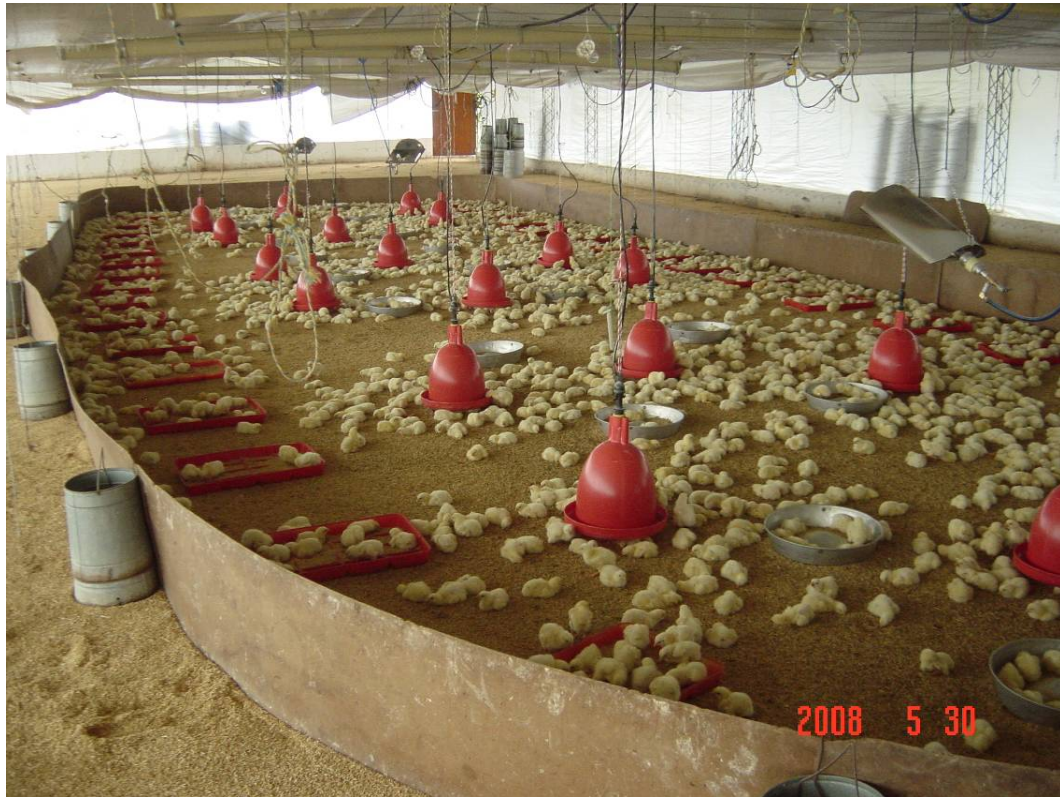
Durante los primeros 10 días del ciclo de cría de las aves se hace necesario instalar unos quemadores de gas o calefactores que proporcionen el calor que los pollitos necesitan para superar el frío intenso que sufren durante sus primeros días de vida.



Actualmente se encuentran ubicados termómetros de mercurio que identifican los cambios y necesidades climáticas de las aves. El operario que está en el galón supervisa regularmente estos termómetros y toma las medidas necesarias según las lecturas que estén marcando.



Con el paso de los días, los animales crecen a tal punto que se hace necesario unir los dos “círculos” iniciales de cada zona del galón y dejarlo de un tamaño mucho mayor que contenga las 4000 aves.



El operario debe revisar periódicamente y retirar del galpón cada día las pérdidas

avícolas que se presenten.



Estas evidencias se presentan para que se identifique el medio donde se están haciendo las investigaciones y pruebas y donde finalmente debe ser implementada la

red de sensores inalámbricos y la aplicación web desarrollada durante este proyecto de grado.