

**MONITOREO DE SENSORES DE GASES CONTAMINANTES DEL AIRE A
TRAVÉS DE REDES INALÁMBRICAS IEEE 802.15.4**

**MILDRED ARROYAVE BUENO
JENNIFER MENDOZA VELANDIA**

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

2013

**MONITOREO DE SENSORES DE GASES CONTAMINANTES DEL AIRE A
TRAVÉS DE REDES INALÁMBRICAS IEEE 802.15.4**

**MILDRED ARROYAVE BUENO
JENNIFER MENDOZA VELANDIA**

**Trabajo de Grado para optar al título de
Ingeniería de Sistemas**

**Director
PEDRO JAVIER TRUJILLO TARAZONA
Ing. De Sistemas, Mg. Informática**

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

2013

*A Dios, que ha sido mi motor y mi fuerza cada día,
mis padres Ramón e Inés, que siempre estuvieron a mi lado dándome todo su
amor y apoyo en cada paso de este largo camino lleno de retos,
mi hermana Mayury, por estar ahí cuando más la necesité,
mi compañera Jennifer por la oportunidad de emprender este reto juntas,
por todo lo que aprendí a su lado, y su constante motivación para lograrlo.*

Mildred

*A Dios por guiarme y llenar mi vida de bendiciones,
mis padres Wilson e Irene por ser mi apoyo incondicional en el
logro de mis metas y la fuerza para seguir adelante cada día de mi vida,
mis hermanos Jessica y Brandon por su apoyo y orientación y mi compañera
Mildred por su amistad y comprensión.*

Jennifer

AGRADECIMIENTOS

A la Universidad Industrial de Santander, en especial a la Escuela de Ingeniería de Sistemas e Informática por formarnos profesionalmente.

A nuestro director de proyecto de grado, Ingeniero Pedro Javier Trujillo, por su apoyo en la planeación y estructuración del presente proyecto.

Al Ingeniero Juan Carlos Escobar por la colaboración brindada para la culminación del proyecto.

A todos los familiares y amigos, gracias por sus sabios consejos en los momentos que más lo necesitamos.

CONTENIDO

	pág.
INTRODUCCIÓN	19
1. DESCRIPCIÓN DEL PROYECTO	21
1.1. JUSTIFICACIÓN	21
1.2. DESCRIPCIÓN DEL PROBLEMA	21
1.3. OBJETIVO GENERAL	22
1.4. OBJETIVOS ESPECÍFICOS	22
2. MARCO TEÓRICO	24
2.1. CONTAMINACIÓN DEL AIRE	24
2.2. RED DE SENSORES INALÁMBRICOS	25
2.2.1. ARQUITECTURA DE UN DISPOSITIVO SENSOR INALÁMBRICO	25
2.3. ESTÁNDAR IEEE 802.15.4	27
2.4. ZIGBEE	29
2.5. MÓDULOS DE TRANSMISIÓN INALÁMBRICA XBEE	32
2.5.1. OPERACIÓN API	37
2.6. SENSORES DE GAS	38
3. LENGUAJE DE PROGRAMACIÓN	41
3.1. PYTHON	41
3.2. BASES DE DATOS	42
3.2.1. MYSQL	42
3.3. DJANGO	43
4. METODOLOGÍA	46

5. DESARROLLO DE LA HERRAMIENTA SOFTWARE	52
5.1. FASE DE EXPLORACIÓN	52
5.1.1. HISTORIAS DE USUARIO.....	53
5.1.2. LENGUAJES DE PROGRAMACIÓN Y TECNOLOGÍAS	57
5.2. FASE DE PLANIFICACIÓN.....	59
5.3. FASE DE ITERACIONES.....	60
5.3.1. PRIMERA ITERACIÓN.....	60
5.3.2. SEGUNDA ITERACIÓN.....	69
5.3.3. TERCERA ITERACIÓN	74
5.3.4. CUARTA ITERACIÓN	81
5.3.5. QUINTA ITERACIÓN.....	85
5.4. FASE DE PRODUCCIÓN.....	85
5.5. FASE DE MANTENIMIENTO	87
5.6. FASE FINAL DEL PROYECTO	90
6. CONCLUSIONES	91
7. RECOMENDACIONES.....	92
BIBLIOGRAFÍA.....	93
ANEXOS.....	98

LISTA DE FIGURAS

pág.

Figura 1. Arquitectura general de un dispositivo sensor inalámbrico	26
Figura 2. Topologías ZigBee.....	30
Figura 3. Módulo XBee	32
Figura 4. Módulo montado sobre una tarjeta de interfaz RS-232	33
Figura 5. Adaptador Sensor XBee	33
Figura 6. Conexiones mínimas requeridas para el XBee	34
Figura 7. Modos de operación del módulo XBee	36
Figura 8. Estructura de trama de datos API	38
Figura 9. Funcionamiento patrón Modelo-Plantilla-Vista de Django	44
Figura 10. Sincronización de tablas entre Django y MySQL.....	63
Figura 11. Descubrimiento de nodos y, envío y recepción de trama.	67
Figura 12. Envío de tramas a través de rutinas y almacenamiento de los datos en MySQL.....	69
Figura 13. Almacenamiento de los datos tratados y depurados	74
Figura 14. Interfaz de ingreso y registro de usuario.....	78
Figura 15. Menú interfaz web.....	79
Figura 16. Formulario de consulta de datos depurados.....	79
Figura 17. Visualización del despliegue de la red	80
Figura 18. Formulario de consulta de reporte diario	81
Figura 19. Interfaz de gestión del administrador.....	84
Figura 20. Asignación de IP y visualización de la página web desde el campus universitario.....	84
Figura 21. Despliegue de los sensores de dióxido de nitrógeno y ozono	86
Figura 22. Despliegue de coordinador y router	86
Figura 23. Despliegue de la red inalámbrica de sensores	88
Figura 24. Ubicación Coordinador. Oficina 159.	88
Figura 25. Ubicación sensor de dióxido de Nitrógeno. Oficina 160.....	89
Figura 26. Ubicación sensor de ozono. Oficina 131.....	89
Figura 27. Diseño del sistema.....	90

LISTA DE TABLAS

pág.

Tabla 1. Enrutamientos ZigBee.....	31
Tabla 2. Características de los protocolos de operación de interfaz serial	35
Tabla 3. Historia de usuario No 1. Creación y administración de base de datos ...	54
Tabla 4. Historia de usuario No 2. Construcción de rutinas para adquisición de datos	54
Tabla 5. Historia de usuario No 3. Implementación de procesos para la depuración de datos	55
Tabla 6. Historia de usuario No 4. Diseño interfaz web	55
Tabla 7. Historia de usuario No 5. Diseño interfaz de administrador	56
Tabla 8. Historia de usuario No 6. Creación de un servidor de alojamiento web ...	56
Tabla 9. Historia de usuario No 7. Definición de procedimientos para el despliegue de la red	57
Tabla 10. Especificación del número de iteraciones	59
Tabla 11. Estructura de la tabla Inf_modulo	60
Tabla 12. Estructura de la tabla principal_datosensor	61
Tabla 13. Estructura de las tablas principal_dato_analizado	61
Tabla 14. Estructura de las tablas principal_dato_ICA	62
Tabla 15. Prueba 1 para la historia de usuario No 1	62
Tabla 16. Tarea 1 para la historia de usuario 2.....	63
Tabla 17. Tarea 2 para la historia de usuario 2.....	65
Tabla 18. Prueba 1 para la historia de usuario 2	67
Tabla 19. Prueba 2 para la historia de usuario 2	68
Tabla 20. Prueba 3 para la historia de usuario 2	68
Tabla 21. Prueba 4 para la historia de usuario 2	69
Tabla 22. Tarea 1 para la historia de usuario 3.....	70
Tabla 23. Tiempos de exposición para contaminantes criterio	70
Tabla 24. Tarea 2 para la historia de usuario 3.....	71
Tabla 25. Puntos de corte del ICA	72
Tabla 26. Prueba 1 para la historia de usuario 3	73
Tabla 27. Prueba 2 para la historia de usuario 3	73
Tabla 28. Tarea 1 para la historia de usuario 4.....	75
Tabla 29. Tarea 2 para la historia de usuario 4.....	75
Tabla 30. Tarea 3 para la historia de usuario 4.....	76
Tabla 31. Tarea 4 para la historia de usuario 4.....	76
Tabla 32. Tarea 5 para la historia de usuario 4.....	77
Tabla 33. Prueba 1 para la historia de usuario 4	77
Tabla 34. Prueba 1 para la historia de usuario 4	78
Tabla 35. Prueba 3 para la historia de usuario 3	79

Tabla 36. Prueba 3 para la historia de usuario 380
Tabla 37.Prueba 5 para la historia de usuario 480
Tabla 38.Tarea 1 para la historia de usuario 5.....81
Tabla 39.Tarea 2 para la historia de usuario 5.....82
Tabla 40. Tarea 1 para la historia de usuario 6.....82
Tabla 41. Prueba 1 para la historia de usuario 583
Tabla 42.Prueba 1 para la historia de usuario 684
Tabla 43. Ubicación de los dispositivos inalámbricos dentro del campus
universitario.....87

LISTA DE ANEXOS

	pág.
ANEXO A. Especificación técnica de los sensores de gas Alphasense	98

LISTA DE SIGLAS

ASCII: American Standard Code for Information Interchange
ANSI: American National Standards Institute
API: Application Programming Interface
APL: Layer Application
AT: Agile and Transparent
AODV: Ad hoc On-Demand Distance Vector Routing
CSMA-CA: Carrier Sense Multiple Access with Collision Avoidance
CSS: Cascading Style Sheets
CVS: Concurrent Versions System
FFD: Full-Function Device
IEEE: Institute of Electrical and Electronics Engineers
MAC: Medium Access Control
MTV: Model-Template-View
ORM: Object-Relational mapping
PAN: Personal Area Network
PHY: Physical Layer
RDBMS: Relational Database Management System
RF: Radio Frequency
RFD: Reduced-Function Device
RSSI: Received Signal Strength Indication
RXD: Received Data
TCP: Transmission Control Protocol
TXD: Transmit Data
UART: Universal Asynchronous Receiver-Transmitter
VOCs: Volatile Organic Compounds
WSN: Wireless Sensor Network
XP: Extreme Programming
ZDO: ZigBee Device Objects

GLOSARIO

CO (Monóxido de carbono): Gas inflamable, incoloro e insípido que se produce por la combustión de combustibles fósiles.

Interferencia: Efecto de una energía no deseada debida a una o varias emisiones, radiaciones, inducciones o sus combinaciones sobre la recepción en un sistema de radiocomunicación, que se manifiesta como degradación de la calidad, falseamiento o pérdida de la información que se podría obtener en ausencia de esta energía no deseada.

Módulo: En el hardware es una tarjeta del circuito electrónico que incluye el sensor.

NO₂ (Dióxido de Nitrógeno): Gas de color pardo rojizo fuertemente tóxico cuya presencia en el aire de los centros urbanos se debe a la oxidación del nitrógeno atmosférico que se utiliza en los procesos de combustión en los vehículos y fábricas.

O₃ (Ozono): Gas azul pálido que, en las capas bajas de la atmósfera, se origina como consecuencia de las reacciones entre los óxidos de nitrógeno y los hidrocarburos (gases compuestos de carbono e hidrógeno principalmente) en presencia de la luz solar.

SO₂ (Dióxido de Azufre): Gas incoloro, no inflamable que posee un fuerte olor en altas concentraciones.

RESUMEN

Título: MONITOREO DE SENSORES DE GASES CONTAMINANTES DEL AIRE A TRAVÉS DE REDES INALÁMBRICAS IEEE 802.15.4*

Autores: ARROYAVE BUENO, Mildred
MENDOZA VELANDIA, Jennifer**

Palabras Claves: Calidad del aire, XBee, ZigBee, red de sensores inalámbrica, IEEE 802.15.4.

DESCRIPCIÓN

En este documento se presenta el proceso y los resultados del desarrollo e implementación de una solución software para monitorear algunos gases contaminantes del aire por medio de una red inalámbrica de sensores; esta red se encuentra conformada por dispositivos XBee integrados con tecnologías ZigBee y el estándar IEEE 802.15.4.

El desarrollo del proyecto se fundamentó en la problemática sobre la contaminación del aire que se ha venido evidenciando desde hace años y que en la actualidad es una de las mayores preocupaciones debido al impacto que está generando sobre la salud humana y el medio ambiente, afectando en mayor proporción a la población más vulnerable los niños, ancianos y personas con sistemas inmunológicos débiles.

Los datos capturados en tiempo real por medio de la red inalámbrica de sensores son almacenados en una base de datos y depurados de acuerdo a la normatividad vigente de la calidad del aire en Colombia; esta información es alojada en un servidor y visualizada a través de una interfaz web.

Por medio de esta interfaz el usuario puede conocer los puntos donde se encuentran ubicados los sensores, efectuar consultas tomando como parámetros los gases monitoreados y las fechas en que estas mediciones fueron realizadas; como resultado se obtienen gráficas que muestran el comportamiento del gas y según su nivel de concentración cuales son los efectos que producen en la salud y las acciones preventivas a seguir.

* Proyecto de grado en la modalidad de investigación

** Facultad de Ingenierías físico-mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director Pedro Javier Trujillo Tarazona.

ABSTRACT

Title: MONITORING OF GAS SENSORS AIR POLLUTANTS THROUGH IEEE 802.15.4 WIRELESS NETWORKS*

Authors: ARROYAVE BUENO, Mildred
MENDOZA VELANDIA, Jennifer**

Keywords: Air quality, XBee, ZigBee, Wireless Sensor Network, IEEE 802.15.4

DESCRIPTION:

This document presents the process and the results of the development and implementation of a software solution to monitor some polluting gases of the air by means of a wireless sensor network; this network is conformed of XBee devices integrated with ZigBee technologies and the IEEE 802.15.4 standard.

The development of the project was based on the problematic on the air pollution that has been evidencing for years and is now one of the greatest concerns due to the impact that is generating on human health and the environment, affecting in greater proportion to the population more vulnerable children, the elderly and people with weak immune systems.

The captured data in real time through the wireless sensor network are stored in a database and deputed in accordance with the regulations of the air quality in Colombia; this information is hosted on a server and displayed through a web interface.

By means of this interface the user can know the points where are located the sensors, perform queries taking as parameters the gases monitored and the dates on which these measurements were carried out; as a result are obtained graphs that show the behavior of the gas and according to your level of concentration which are the effects on health and the preventive actions to follow.

* Thesis degree in the modality of research.

** Faculty of Physical-Mechanical Engineerings.School of Systems and Computer Engineering.
Director Pedro Javier Trujillo Tarazona.

INTRODUCCIÓN

En los últimos años la calidad del aire se ha venido deteriorando debido al alto grado de contaminación en el ambiente generando efectos secundarios como el calentamiento global, el deterioro de la capa de ozono y las lluvias acidas, una de las causas de esta problemática se debe a la emisión de gases tóxicos como NO₂, SO₂, CO y O₃, que en concentraciones altas producen diversos efectos perjudiciales en la salud humana y en casos críticos la muerte, afectando en mayor proporción a la población más vulnerable los niños, ancianos y personas con sistemas inmunológicos débiles.

Conscientes de esta problemática ambiental, se desarrolló un software que permite monitorear valores de la concentración de algunos gases contaminantes en el aire, por medio de una red inalámbrica de sensores, con la cual se busca que el usuario pueda determinar los lugares donde existe mayor grado de contaminación de acuerdo a la ubicación de los puntos de medición.

En el presente documento se describe cómo se realizó el desarrollo del proyecto el cual se encuentra organizado de la siguiente manera: en el capítulo 1, se da una descripción del proyecto, que contiene la justificación, descripción del problema, objetivo general y objetivos específicos; en el capítulo 2, se expone el marco teórico donde se presentan los conceptos sobre contaminación ambiental, redes de sensores inalámbricas, estándares implementados y las herramientas utilizadas en el desarrollo del proyecto; en el capítulo 3, se describen los lenguajes de programación bajo los que se desarrolló la herramienta software; en el capítulo 4, se explica la metodología aplicada; en el capítulo 5; se expone el desarrollo de la investigación de acuerdo a la metodología escogida; en el capítulo 6, se muestra el resultado final del sistema; en el capítulo 7, se presentan las conclusiones que se lograron verificar durante el avance del proyecto; finalmente en el capítulo 8, se despliegan unas breves recomendaciones.

El software fue diseñado y desarrollado aplicando la metodología ágil programación extrema, la cual permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la ocurrencia de un error y su detección, debido a la realización constante de pruebas.

1. DESCRIPCIÓN DEL PROYECTO

1.1. JUSTIFICACIÓN

Viendo la problemática que se está presentando en la actualidad sobre la deficiente calidad del aire generada por la emisión de gases contaminantes, que es ocasionada por diversos factores como los vehículos automotores, las plantas de generación eléctrica, las fuentes industriales, comerciales y residenciales que queman combustibles, y los efectos perjudiciales que causa en la salud humana; siendo una amenaza con la que nos enfrentamos diariamente, se optó por plantear una posible solución a la problemática desde la percepción de ingeniería de sistemas con el enfoque del área de redes y el uso de herramientas electrónicas facilitadas por el director de proyecto.

1.2. DESCRIPCIÓN DEL PROBLEMA

El avance y el continuo desarrollo del entorno industrial, han cambiado significativamente las condiciones de la calidad del aire; en la actualidad el problema de la contaminación del aire se ha convertido en uno de los factores de mayor preocupación que ha afectado la salud y el bienestar humano.

Esta realidad se ha venido divulgando en diversos medios de comunicación, haciendo un llamado a la conciencia de los ciudadanos y conseguir así mejores resultados en la búsqueda de una solución a esta problemática; para lograr este objetivo, diferentes entidades que se encargan de controlar la calidad ambiental, suman sus esfuerzos para corregir esta situación, y así generar condiciones óptimas para los ciudadanos y reducir las emisiones de gases.

Uno de los problemas que originan la contaminación atmosférica, es la concentración de gases contaminantes en el aire, que se caracterizan por ser inodoros e incoloros, por lo tanto imperceptibles al ser humano, de gran amenaza y riesgo para la salud de las personas y el medio ambiente. Otras características de estos gases es que son tóxicos, inflamables, explosivos o volátiles; los cuáles se pueden encontrar en diversos sitios como hogares, industrias y parqueaderos de automotores.

Desde nuestra formación académica se realizó el desarrollo de una interfaz web que por medio de una red de inalámbrica de sensores permita monitorear valores de la concentración de gases contaminantes en el aire, que facilite al usuario consultar información precisa, oportuna y útil de los diferentes niveles de contaminación en los puntos sensados, con la cual logre identificar los lugares que implican un riesgo para su salud.

Esta red puede ser ubicada en diferentes lugares con facilidad debido al reducido tamaño de los dispositivos; transmite en tiempo real los datos capturados de los sensores, y posteriormente, al ser almacenados en la base de datos son depurados de acuerdo a la normatividad vigente de la calidad del aire.

1.3. OBJETIVO GENERAL

Elaborar una herramienta software para el monitoreo de algunos gases contaminantes del aire mediante una red inalámbrica de sensores.

1.4. OBJETIVOS ESPECÍFICOS

- Definir procedimientos para despliegue de la red inalámbrica.

- Elaborar la interfaz de usuario para visualizar los datos recibidos de los sensores y para la interacción del usuario en la gestión de la red.
- Diseñar y codificar las rutinas para la transmisión de tramas IEEE 802.15.4.
- Desarrollar y codificar las rutinas para el tratamiento y almacenamiento de los datos.

2. MARCO TEÓRICO

2.1. CONTAMINACIÓN DEL AIRE

Según una evaluación de la carga de morbilidad debida a la contaminación atmosférica hecha por la Organización Mundial de la Salud¹, cada año se producen más de 2 millones de muertes prematuras atribuibles a los efectos de la contaminación atmosférica urbana y de la contaminación del aire de interiores, causada por la utilización de combustibles sólidos, o cualquier agente químico, físico o biológico que modifica las características naturales de la atmósfera. Más de la mitad de esa carga recae sobre la población de los países en desarrollo [20].

Actualmente en Colombia esta problemática atmosférica es la que ocasiona los mayores costos sociales y ambientales después de los generados por la contaminación del agua y los desastres naturales; las concentraciones de gases contaminantes en la atmósfera de mayor preocupación para la salud pública que monitorean las entidades gubernamentales de gestión y control ambiental a nivel nacional son el monóxido de carbono, ozono, dióxido de nitrógeno y dióxido de azufre, que en largos periodos de exposición son analizadas de acuerdo a las normas ambientales con el fin de proteger la salud de la población y el medio ambiente [18].

Estas normatividades de la calidad del aire son difundidas por el Ministerio del Medio Ambiente, Vivienda y Desarrollo Territorial (MAVDT), dentro de las cuales se encuentra la Resolución 610 del 24 de Marzo de 2010 [23] en la que se establece los nuevos niveles permisibles de concentración en el aire de cinco contaminantes criterio² que podrían afectar significativamente la salud de la población en general; y la Resolución 650 del 29 de Marzo del 2010 [22] que da a

¹ Autoridad directiva y coordinadora de la acción sanitaria en el sistema de las Naciones Unidas

² Contaminantes del aire que tienen el mayor impacto en la salud humana.

conocer el Protocolo para el Monitoreo y Seguimiento de la Calidad del Aire, donde se indica una estandarización en los niveles de contaminación de todo el territorio Colombiano mediante la implementación del Índice Nacional de Calidad del Aire (ICA).

2.2. RED DE SENSORES INALÁMBRICOS

Una red de sensores inalámbricos (WSN) consiste en varios dispositivos autónomos distribuidos de forma espacial, los cuales hacen uso de sensores que captan cierta información de condiciones físicas o ambientales, como temperatura, sonido, vibración, presión, movimiento o gases, y la transmiten de forma inalámbrica a una estación central de procesamiento [16].

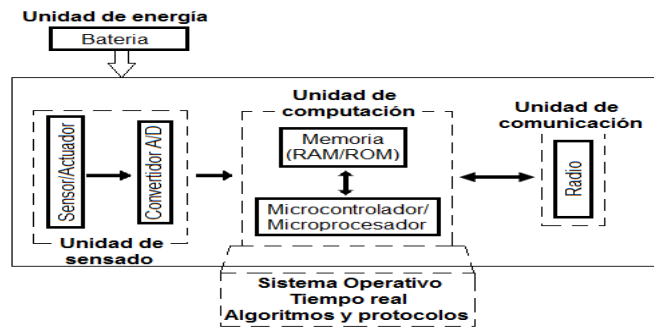
Las ventajas adicionales de las redes inalámbricas de sensores son la posibilidad de monitorear estas aplicaciones desde lugares remotos y tener un sistema que puede proporcionar grandes cantidades de datos por largos periodos de tiempo [10].

Para el establecimiento y arquitectura de la red es necesario disponer los sensores usando una topología de red, siempre y cuando uno de los dispositivos sea el que actúa como coordinador. Entre las topologías se encuentran: malla, árbol y estrella.

2.2.1. Arquitectura de un dispositivo sensor inalámbrico

Un dispositivo sensor inalámbrico se compone de las siguientes unidades mostradas en la figura 1.

Figura 1.Arquitectura general de un dispositivo sensor inalámbrico



Fuente: Handbook of Sensors Networks: Compacts Wireless and Wired Sensing Systems [13]

- *Unidad de sensado:* Compuesta por el sensor que es el encargado de transformar la medida física en impulsos eléctricos analógicos y el convertidor analógico digital que es quien convierte en bits dicha medida para entregársela a la unidad de cálculo y procesado.
- *Unidad de computación:* Reúne a todas las demás unidades y algunos periféricos adicionales. Su principal objetivo es procesar (ejecutar) las instrucciones relativas al sensado, la comunicación y auto-organización.
- *Unidad de comunicación:* Se trata de un dispositivo vía radio que permite enviar y recibir datos para comunicarse con otros dispositivos dentro de su rango de transmisión. Los medios a elegir para realizar una comunicación inalámbrica son varios, entre los cuales se encuentran: radio frecuencia y comunicación óptica mediante laser e infrarrojos.
- *Unidad de energía:* Es la encargada de proveer la potencia necesaria para hacer funcionar los demás elementos.
- *Sistema Operativo:* Encargado de operar y controlar estas unidades.

2.3. ESTÁNDAR IEEE 802.15.4

El estándar IEEE 802.15.4 define las especificaciones de la capa física (PHY) y la subcapa de control de acceso al medio (MAC) para redes inalámbricas de área personal con bajas tasas de transmisión de datos. Las características más importantes son la flexibilidad de la red, bajo coste y bajo consumo de energía [17].

Define dos diferentes tipos de dispositivos que pueden participar en una red:

- *Dispositivo de funcionalidad completa (FFD)*: Puede funcionar en tres modos, actuando como un coordinador de red de área personal (PAN), un coordinador o un dispositivo. Se comunica con otros dispositivos de funcionalidad completa o dispositivos de funcionalidad reducida.
- *Dispositivo de funcionalidad reducida (RFD)*: Está diseñado para aplicaciones que son muy simples y no tienen la necesidad de enviar grandes cantidades de datos; sólo puede asociarse con un único dispositivo de funcionalidad completa a la vez y se puede implementar utilizando un mínimo de recursos y capacidad de memoria.

Usa dos modos de direccionamiento:

- *Direccionamiento extendido de 64 bits*: Significa que el máximo número de dispositivos en la red puede ser 2^{64} o aproximadamente 1.8×10^{19} . Así una red inalámbrica no tiene prácticamente límite en el número de dispositivos que pueden unirse a la red.
- *Direccionamiento corto de 16 bits*: Permite comunicaciones dentro de una sola red, una reducción en la longitud de los mensajes y ahorrar espacio de memoria requerido que es asignado para almacenamiento de direcciones.

Emplea varios mecanismos para mejorar la probabilidad de éxito de la transmisión de datos. Estos mecanismos son:

- *Mecanismo de acceso múltiple sensible a la portadora con prevención de colisiones (CSMA-CA):* Está basado en la naturaleza compartida del canal de radio frecuencia. Siempre que dos o más emisores están activos en un canal simultáneamente, la probabilidad de que cualquiera de ellos tenga éxito en el envío de un mensaje disminuye debido a las colisiones y su interferencia mutua. Una forma de evitar colisiones en el canal es escuchar primero y transmitir sólo si el canal está libre. El mecanismo asegura el acceso al canal libre y reduce la posibilidad de una colisión. De esta manera, la capacidad del canal está más plenamente utilizado.
- *Trama de acuse de recibo:* Si el coordinador no recibe un acuse de recibo después de un cierto período, asume que la transmisión no tuvo éxito y vuelve a intentar la transmisión de tramas. Si después de varios intentos aún no se recibió un acuse de recibo, puede optar por cancelar la transacción o volver a intentarlo. Cuando el acuse de recibo no se requiere, asume que la transmisión tuvo éxito.
- *Verificación de datos:* Se usa una comprobación de redundancia cíclica (CRC) para detectar errores en cada unidad de datos de servicio de la capa física.

Define tres tipos de transferencia de datos:

- De un dispositivo hacia un coordinador.
- De un coordinador hacia un dispositivo.
- Entre dos dispositivos iguales o semejantes.

Los mecanismos para cada tipo de transferencia dependen del soporte de la red en la transmisión de difusores de señal de radio.

2.4. ZIGBEE

ZigBee es un estándar que define las especificaciones de la capa de red (NWK), seguridad y capa de aplicación (APL) para redes de área personal inalámbricas de bajo costo, baja velocidad de datos y bajo consumo de energía. El estándar ZigBee tiene adoptado la capa física y subcapa de control de acceso al medio del estándar IEEE 802.15.4 como parte del protocolo de redes ZigBee [5].

Define tres tipos diferentes de dispositivos según el papel que cumplen en la red:

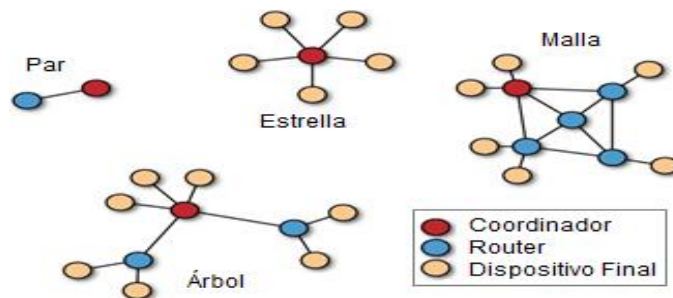
- *Coordinador*: Es un dispositivo de funcionalidad completa (coordinador de red de área personal IEEE 802.15.4) y es responsable de la gestión total de la red. Cada red tiene exactamente un coordinador. El coordinador realiza las siguientes funciones:
 - Selecciona el canal a ser utilizado por la red.
 - Inicia la red.
 - Determina la manera cómo se asignan las direcciones a los nodos o enrutadores.
 - Permite a otros dispositivos unirse o abandonar la red.
 - Contiene una lista de vecinos y routers.
 - Transfiere los paquetes de programas de aplicación.
- *Router*: Es un dispositivo de funcionalidad completa (coordinador IEEE 802.15.4), usado en una topología árbol o malla para ampliar la cobertura de la red. La función de un router es encontrar la mejor ruta hacia el destino

sobre el cual transfiere un mensaje. Un router realiza todas las funciones similares a un coordinador, excepto el establecimiento de una red.

- *Dispositivo final*: Es un dispositivo de funcionalidad reducida. Se puede conectar a un router o coordinador. Sólo consume energía durante la transmisión de la información. Por lo tanto, la arquitectura ZigBee está diseñada de manera que el tiempo de transmisión de un dispositivo final sea corto. Tiene mínimo tamaño de memoria y poca capacidad de procesamiento y características. El dispositivo final realiza las siguientes funciones:
 - Se une o abandona la red
 - Transfiere los paquetes de programas de aplicación.

Las redes ZigBee soportan cuatro topologías de red, las cuales son par, estrella, malla y árbol. Ver figura 2.

Figura 2. Topologías ZigBee



Fuente: Building Wireless Sensor Network [6]

En cualquiera de estas topologías siempre existe un dispositivo de red que asume el papel de coordinador y que siempre está listo para comenzar a trabajar en el

momento en que cualquier otro dispositivo salga del estado de suspensión para optimizar el consumo.

ZigBee maneja dos tipos de transmisión de datos, unicast y broadcast. En las transmisiones unicast los datos son enviados de un dispositivo a otro, mientras que en la broadcast se envían a muchos o todos los dispositivos de la red.

Las transmisiones unicast requieren algún tipo de enrutamiento. ZigBee incluye diferentes caminos con sus ventajas y desventajas las cuales se muestran en la tabla 1.

Tabla 1. Enrutamientos ZigBee

Enrutamiento	Descripción	Cuando usar
Enrutamiento malla vector distancia sobre demanda ad hoc (AODV)	Caminos de enrutamiento son creados entre origen y destino, posiblemente atravesando múltiples nodos (“saltos”). Cada dispositivo conoce a cual dispositivo siguiente enviar los datos para llegar a tiempo al destino.	Usada en redes de una escala no mayor de 40 dispositivos finales.
Enrutamiento muchos a uno	Una sola transmisión broadcast configura rutas inversas en todos los dispositivos dentro del dispositivo que envía el broadcast.	Es útil cuando muchos dispositivos remotos envían datos a una única puerta de enlace o dispositivo colector.
Enrutamiento de origen	Los paquetes de datos incluyen toda la ruta que el paquete debe atravesar para llegar desde el origen al destino.	Mejora la eficiencia de encaminamiento en redes de gran tamaño (más de 40 dispositivos remotos).

Fuente: Adaptada del Manual módulos RF ZigBeeXBee/XBeePRO [24]

2.5. MÓDULOS DE TRANSMISIÓN INALÁMBRICA XBEE

Los módulos XBee están diseñados para operar dentro del protocolo ZigBee, requieren un mínimo de energía, proporcionan la entrega fiable de datos entre los dispositivos remotos que conforman la red y operan dentro de la banda de frecuencia ISM de 2,4 GHz.

Permiten 2 tipos de direccionamiento, de 16 y 64 bits. A través de estas direcciones se referencian o direccionan entre sí. La dirección de 64 bits es una dirección única para cada dispositivo asignada durante la fabricación y la dirección de 16 bits la recibe el dispositivo cuando se une a una red, por esta razón es llamada dirección de red.

En la Figura 3 se puede observar el diseño de un módulo XBee, el cual tiene integrado una antena por donde transmite inalámbricamente la información.

Figura 3. Módulo XBee

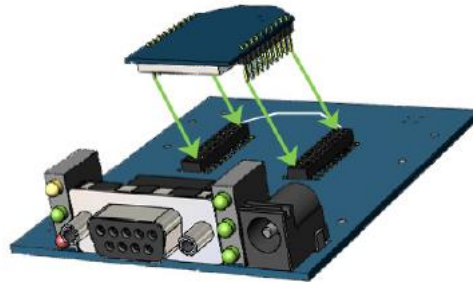


Fuente: Manual módulos RF ZigBeeXBee/XBeePRO [24]

Los módulos XBee fueron diseñados para montarse en una tarjeta integrada y por lo tanto no requiere ningún tipo de soldadura. La tarjeta integrada contiene conectores RS-232 o USB y utiliza dos receptáculos³ de 10 pines cada uno para recibir los módulos como se muestra en la figura 4.

³Cavidad en que se contiene o puede contenerse el módulo.

Figura 4. Módulo montado sobre una tarjeta de interfaz RS-232



Fuente: Manual módulos RF ZigBeeXBee/XBeePRO [24]

Estas tarjetas integradas pueden ser incorporadas dentro de un adaptador para mayor seguridad y protección.

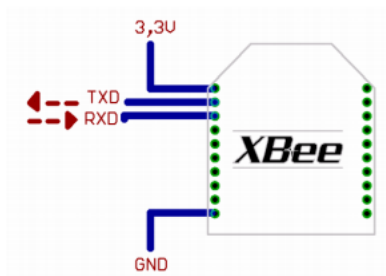
Figura 5. Adaptador Sensor XBee



Fuente: XBeeDrop-in Networking Accessories User's Guide[25]

En la Figura6 se observan las conexiones mínimas que necesita el módulo XBee para poder ser utilizado. Luego de esto, se debe configurar según el modo de operación que se desea para la aplicación requerida por el usuario.

Figura 6. Conexiones mínimas requeridas para el XBee



Fuente: Manual módulos RF ZigBeeXBee/XBeePRO [24]

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del Transmisor-Receptor Asíncrono Universal (UARTTXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún convertidor adecuado para los niveles de voltaje.

Los módulos soportan dos protocolos de operación interfaz serial, transparente e interfaz de programación de aplicaciones (API). En la Tabla 2 se muestran las características de los protocolos.

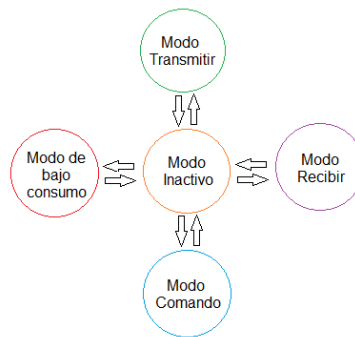
Tabla 2. Características de los protocolos de operación de interfaz serial

Operación Transparente	
Interfaz simple	Todos los datos serial recibidos son transmitidos a menos que el módulo este en modo comando.
Fácil soporte	Es más fácil para una aplicación soportar el funcionamiento transparente y modo comando.
Operación de Interfaz de programación de aplicaciones	
Fácil gestión de las transmisiones de datos a múltiples destinos	La transmisión de datos de radio frecuencia a múltiples dispositivos remotos sólo requiere el cambio de la dirección en la trama API. Este proceso es mucho más rápido que en la operación transparente donde la aplicación debe entrar en modo comando AT, cambiar la dirección, salir del modo de comando y entonces transmitir los datos. Cada transmisión API puede devolver una trama de estado de transmisión indicando el éxito o el motivo del fallo.
Las tramas de datos recibidas indican la dirección del remitente	Todas las tramas API de datos de radio frecuencia recibidas indican la dirección de origen.
Avanzado soporte de direccionamiento ZigBee	Las tramas API transmitidas y recibidas pueden exponer campos de direccionamiento ZigBee incluyendo puntos finales de origen y destino, identificador de grupo y perfil. Esto hace que sea fácil el soporte de comandos ZDO y tráfico de perfil público.
Avanzados diagnósticos de red	Las tramas API pueden proporcionar una indicación de muestras de entrada-salida desde dispositivos remotos, y mensajes de identificación de nodos.
Configuración remota	Establecer/leer la configuración de los dispositivos por medio de comandos, las cuales pueden ser enviadas a dispositivos remotos para configurarlos según sea necesario utilizando la API.

Fuente: Adaptada del Manual módulos RF ZigBeeXBee/XBeePRO [24]

Los módulos XBee usan cinco modos de operación, como se ilustra a continuación.

Figura 7. Modos de operación del módulo XBee



Fuente: Manual módulos RF ZigBeeXBee/XBeePRO [24]

- *Modo inactivo (Idle mode):* Cuando no está recibiendo o transmitiendo datos, el módulo de radio frecuencia se encuentra en modo inactivo. El módulo se desplaza a los otros modos de operación en las siguientes condiciones:
 - *Modo transmitir:* El dato serial en el buffer de recepción está listo para ser empaquetado. El módulo de radiofrecuencia saldrá del modo de reposo y tratará de transmitir los datos, asegurando que se han establecido la dirección de 16 bits y la ruta al nodo destino.
 - *Modo recibir:* El dato de radiofrecuencia valido es recibido a través de la antena y transferido al buffer de transmisión serial.
 - *Modo de bajo consumo:* Solo dispositivos finales, hace posible que el módulo de radio frecuencia entre en modo de bajo consumo de energía cuando no se encuentra en uso.
 - *Modo comando:* Para modificar o leer los parámetros del módulo de radio frecuencia, el módulo primero tiene que entrar en modo de

comando; un estado en el cual los caracteres entrantes son interpretados como comandos.

2.5.1. Operación API

Este modo de operación es más complejo, pero permite el uso de tramas con cabeceras que aseguran la entrega de los datos, al estilo protocolo de control de transmisión (TCP).

Una trama de transmisión de datos incluye:

- Trama de datos de radio frecuencia transmitida.
- Trama de comandos (equivalente a comandos transparentes).

Mientras que una trama de recepción de datos incluye:

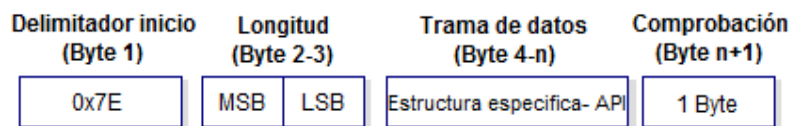
- Trama de información de radio frecuencia recibida.
- Comando de respuesta.
- Notificaciones de eventos como Reset, Associate, Disassociate, etc.

La interfaz de programación de aplicaciones, provee alternativas para la configuración del módulo y enrutamiento de los datos en la capa de aplicación host. Una capa de aplicación host puede enviar información al módulo XBee, estos datos serán contenidos en una trama cuya cabecera tendrá información útil referente al módulo. Esta información además se podrá configurar, es decir, en vez de estar usando el modo de comandos para modificar las direcciones, la interfaz de programación de aplicaciones lo realiza automáticamente. El módulo enviará tramas de datos a la aplicación, las cuales contienen los paquetes de estado; así como tramas que contienen el origen, el indicador de intensidad de

señal recibida (RSSI) y la información de carga útil de los paquetes de datos recibidos.

La estructura de una trama de datos API está definida como se muestra en la figura 8.

Figura 8. Estructura de trama de datos API



Fuente: Adaptada del Manual módulos RF ZigBeeXBee/XBeePRO [24]

- Delimitador de inicio: Indica el comienzo de la trama.
- Longitud: Especifica el número de bytes que están contenidos en los campos de la trama de datos. No incluye el campo de comprobación.
- Trama de datos: Contiene el identificador y los campos correspondientes a la trama.
- Comprobación: Su contenido es la suma de todos los bytes de la trama de datos, manteniendo solo los 8 bits menos significativos del resultado y substraer el resultado desde 0xFF.

2.6. **SENSORES DE GAS**

Un sensor de gas es un transductor⁴ que detecta moléculas de gas y el cual produce una señal eléctrica con una magnitud proporcional a la concentración del

⁴ Dispositivo capaz de transformar o convertir un determinado tipo de energía de entrada, en otra diferente a la salida.

gas. A diferencia de otros tipos de medición los gases son mucho más complicados de tratar, porque existen literalmente cientos de diferentes gases, y hay una amplia gama de aplicaciones diversas en la cual estos gases están presentes, cada aplicación debe implementar un conjunto único de requisitos [3].

Los sensores, pueden detectar una gran gama de gases en rangos de ppm o combustibles. Se pueden conseguir diferentes características de respuesta variando los materiales, las técnicas del proceso y la temperatura de trabajo.

Los sensores se clasifican en electroquímicos, catalíticos, estado sólido, infrarrojo y de fotoionización.

- *Sensores electroquímicos:* Operan por reacción con el gas de interés y producen una señal eléctrica proporcional a la concentración del gas. Son afectados mínimamente por cambios en la presión y son muy sensibles a la temperatura, por lo tanto, están internamente con compensación de temperatura. Requieren baja energía para operar en comparación con los demás tipos de sensores.
- *Sensores catalíticos:* Son sensores en los cuales las mezclas de gas combustible no se queman hasta que alcancen una temperatura de ignición. Sin embargo, en la presencia de ciertos medios químicos, el gas empieza a quemar o encender a temperaturas más bajas. En presencia de gases combustibles, las moléculas de gas se queman sobre la superficie del sensor, lo cual causa que la temperatura del sensor se incremente.
- *Sensor de estado sólido:* Se compone de uno o más óxidos metálicos de metales de transición. En presencia de un gas, el óxido metálico causa una disociación del gas en iones cargados o complejos, resultando la transferencia de electrones. Un par de electrodos apropiados se insertan en

el óxido metálico, para medir sus cambios de conductividad en forma de señal.

- *Sensor infrarrojo:* Mide la interacción de la radiación infrarroja con las moléculas de gas. Esto es diferente a la mayoría de las otras tecnologías en las cuales los sensores están directamente en contacto con el gas objetivo.
- *Sensor de fotoionización:* Utilizan luz ultravioleta para ionizar las moléculas de gas y se emplean comúnmente en la detección de compuestos orgánicos volátiles (VOCs).

3. LENGUAJE DE PROGRAMACIÓN

3.1. PYTHON

Python es un lenguaje de programación de propósito general y código abierto. Está optimizado para la calidad del software, la productividad del desarrollador, la portabilidad del programa, y la integración de componentes [12].

Es utilizado por muchos desarrolladores en todo el mundo en áreas tales como Internet scripting, programación de sistemas, interfaces de usuario, personalización de productos y programación numérica.

Las principales características del lenguaje de alto nivel Python son: sintaxis muy simple, legible y fácil de mantener; integración con componentes externos codificados en otros idiomas; diseño multiparadigma con programación orientada a objetos, funcional y estructuras modulares; y amplia colección de interfaces pre-codificadas y utilidades. Su conjunto de herramientas hace que sea un lenguaje ágil y flexible, ideal para tareas tácticas rápidas y en el desarrollo de aplicaciones estratégicas de largo alcance [2].

Python es un lenguaje:

- Orientado a objetos, su modelo de clase soporta nociones avanzadas tales como el polimorfismo, la sobrecarga de operadores, y la herencia múltiple.
- Tipado dinámicamente, lo que significa que no es necesario declarar explícitamente la clase de los objetos en los scripts.

- Fuertemente tipado, lo que significa que no se puede hacer algo con un objeto que no está permitido por la misma clase.
- Multiplataforma, la implementación estándar de Python está escrita en ANSI C portable, este se compila y ejecuta en casi todas las principales plataformas actualmente en uso.

3.2. BASES DE DATOS

Una base de datos para redes de sensores, es una base distribuida que recoge mediciones físicas del ambiente, las indexa y sirve a las preguntas de los usuarios y otras aplicaciones externas con origen o destino en la red.

3.2.1. MySQL

MySQL es un sistema gestor de bases de datos relacionales (RDBMS) rápido, robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente/servidor. Se proporciona con muchas herramientas y es compatible con muchos lenguajes de programación [15].

Uno de los módulos de base de datos de Python es DB-API; como su nombre lo indica, ofrece una interfaz para programación de aplicaciones de base de datos, el modulo DB-API proporciona acceso a bases de datos mediante el controlador de MySQLdb [11].

La librería MySQL-Python, es la que se encarga de hacer la conexión con cualquier base de datos que se encuentre alojada en MySQL para su respectiva manipulación con Python.

Las secuencias de comandos de acceso de MySQL a través de DB-API por medio de MySQLdb en general, llevan a cabo los siguientes pasos:

- Importar el módulo MySQLdb.
- Crear el objeto de la conexión a la base de datos.
- Crear el objeto cursor para permitir el acceso a la base de datos.
- Interactuar con la base de datos.
- Cerrar la conexión.

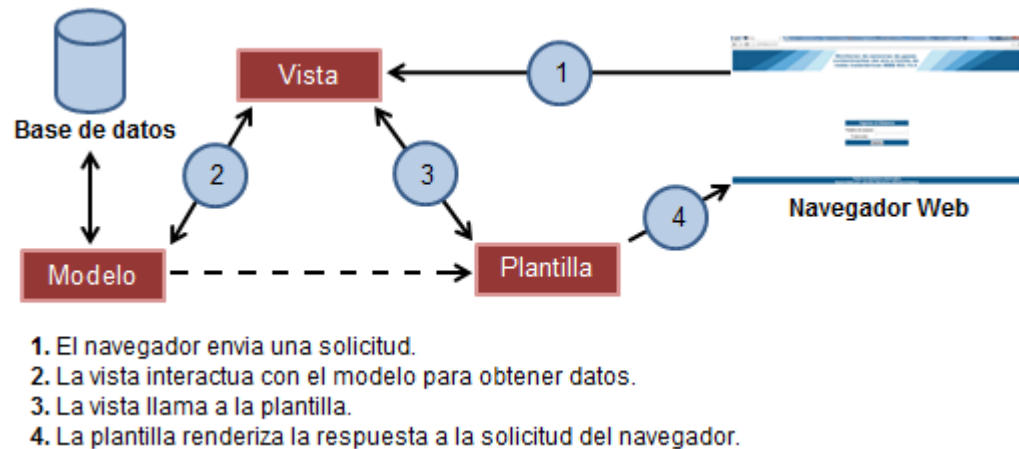
3.3. DJANGO

Django es un framework de desarrollo web escrito en Python, que permite construir aplicaciones de alta calidad más rápidamente y con menos código. Su objetivo principal es facilitar la creación de páginas web apoyadas en bases de datos [29].

Django usa el patrón Modelo-Plantilla-Vista (MTV), el cual permite separar las aplicaciones en datos, interfaz de usuario y lógica de control.

El Funcionamiento del Modelo-Plantilla-Vista de Django se ilustra en la figura 9.

Figura 9. Funcionamiento patrón Modelo-Plantilla-Vista de Django



Fuente: Adaptada por los autores de Django para perfeccionistas con deadlines [26]

- *Modelo:* Define la estructura de la base de datos, mediante clases de Python. Cada modelo es una subclase de Model que se encuentra en el paquete `django.db.models` y corresponde a una tabla.
- *Vista:* Contiene funciones de Python, que se encargan de diversas tareas como gestionar los datos a mostrar desde la base de datos, la autenticación de usuarios, validación de formularios, entre otras. El Mapeo-Objeto-Relacional (ORM), permite escribir código Python para las consultas a la base de datos en lugar de código SQL.
- *Plantilla:* Está compuesta por código HTML, puede generar contenido XML, CSS, Javascript, CVS, entre otros. Usa etiquetas para el manejo de las estructuras de control (sentencias `if`, `else`, `while`, `for`) y se encargan de recibir los datos consultados por la vista y organizarlos para ser mostrados en el navegador web.

Django contiene:

- Un mapeador objeto-relacional que media entre los modelos de datos (definidos como clases de Python) y una base de datos.
- Un gestor de peticiones web, basado en expresiones regulares que procesan las URLs entrantes.
- Un sistema de vistas para procesar las peticiones entrantes.
- Un sistema de plantillas que asiste en la visualización de respuestas a las peticiones entrantes.

4. METODOLOGÍA

La metodología de desarrollo de software que se va a aplicar está enmarcada dentro de la metodología de programación extrema (XP), la cual es una metodología ágil que maneja un diseño incremental durante el desarrollo.

Se escogió esta metodología debido a la comunicación continua que se mantiene entre el cliente y el equipo de desarrollo; y la constante realización de pruebas que se realizan por cada iteración, pues de esta manera se mantiene un código claro y sencillo.

Los roles que se manejan dentro de la metodología son:

- *Programador*: Escribe las pruebas unitarias y produce el código del sistema. Define las tareas que conlleva cada historia de usuario, y estima el tiempo que requerirá cada una.
- *Cliente*: Escribe las historias de usuario y las pruebas funcionales para validar su implementación. Asigna la prioridad a las historias de usuario y decide cuales se implementan en cada iteración.
- *Encargado de pruebas*: Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- *Encargado de seguimiento*: Verifica las estimaciones realizadas, evalúa el progreso de cada iteración, así como la factibilidad de los objetivos con las restricciones de tiempo y recursos presentes. Mantiene contacto directo con

el equipo de desarrollo, realizando cambios para lograr los objetivos de cada iteración.

- *Consultor*: Miembro externo del equipo quien posee conocimiento en algún tema necesario para el proyecto.
- *Gestor*: Es el vínculo entre clientes y programadores, su función principal es la coordinación.

La Programación Extrema es un conjunto coherente de valores, principios y prácticas para abordar problemas de desarrollo de software [9]. Los doce principios básicos son:

- *El juego de la planificación*: Determina rápidamente el alcance de la próxima versión mediante la combinación de las prioridades del negocio y las estimaciones técnicas. Sobre las tareas, se actualiza el plan.
- *Entregas cortas*: Poner un sistema simple en producción rápidamente, a continuación, liberar nuevas versiones en un ciclo muy corto.
- *Metáfora*: Oriente todo el desarrollo con una historia compartida que describe cómo funciona el sistema.
- *Diseño simple*: El sistema debe ser diseñado de la manera más sencilla posible en algún momento determinado. La complejidad extra es removida tan pronto como se descubra.
- *Pruebas*: Los programadores continuamente escriben pruebas unitarias, las cuales deben ejecutarse sin problemas para el continuo desarrollo. Los

clientes escriben pruebas demostrando que características se han finalizado.

- *Refactorización*: Los programadores reestructuran el sistema sin cambiar su comportamiento para eliminar la duplicación, mejorar la comunicación, simplificar o añadir flexibilidad.
- *Programación en pares*: Todo el código es escrito con dos programadores en una sola máquina.
- *Propiedad colectiva*: Cualquiera puede modificar el código en cualquier parte del sistema en cualquier momento.
- *Integración continua*: Integrar y desarrollar el sistema varias veces al día, cada vez que se complete una tarea.
- *Semanas de 40 horas*: No trabajar más de 40 horas a la semana.
- *El cliente en su sitio*: Incluir un usuario real en el equipo, disponible tiempo completo.
- *Normas de codificación*: Los programadores escriben el código en conformidad con las reglas haciendo hincapié en la comunicación a través del código.

Las fases que maneja la metodología son:

- *Fase de exploración*: Los clientes plantean las historias de usuario que son de interés para la entrega del producto, al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que

se utilizaran en el proyecto. La fase de exploración toma semanas o meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- *Fase de planificación:* El cliente establece la prioridad de cada historia de usuario, los programadores descomponen las historias en tareas y realizan una estimación del esfuerzo necesario de cada una de ellas. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación.
- *Fase de iteraciones:* Incluye varias iteraciones sobre el sistema antes de ser entregado y el cliente decide que historias se implementarán en cada iteración. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.

Una iteración consta de los siguientes pasos:

- Historias de usuario: Escogidas por el cliente para ser trabajadas durante la iteración.
- Expectativas: Después de recibir las historias de parte del cliente, se interpreta qué es lo que el cliente espera del software.
- Especificaciones: Se descompone la historia en especificaciones organizadas en conjuntos de tareas.
- Definición de las tareas: Asignación a los programadores de las tareas a realizar.

- Diseño y elaboración de las pruebas con costeo: Se diseñan tipos de pruebas que permitan evaluar el funcionamiento de las tareas.
- Diseño y costeo de los módulos software: Se agrupan las tareas a desarrollar para cada fase, de igual modo se estiman y asignan recursos para su ejecución.
- Codificación de los módulos software: Generación de código fuente.
- Aplicación de prueba: Se ejecutan las pruebas a las tareas ya elaboradas.
- Evaluación de los resultados de las pruebas: Se analizan los resultados obtenidos de las pruebas, y junto con el cliente se evalúa el cumplimiento en el desarrollo de la tarea.

Después de terminada la iteración si las pruebas no cumplen las expectativas, se vuelve a comenzar una nueva iteración a partir del numeral 6.

- Fase de producción: Requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- Fase de mantenimiento: Mientras la primera versión se encuentra en producción, se debe mantener el sistema en funcionamiento al mismo tiempo que se desarrolla documentación de soporte para el cliente.

- Fase final del proyecto: El cliente no tiene más historias para ser incluidas en el sistema. Se genera la documentación final del sistema y no se realizan cambios en la arquitectura.

5. DESARROLLO DE LA HERRAMIENTA SOFTWARE

Este capítulo tiene como objetivo la aplicación de la metodología ágil programación extrema en el desarrollo de la herramienta software, la cual fue descrita anteriormente.

Para el desarrollo se establecieron inicialmente los roles de acuerdo a la metodología, los cuales fueron designados de la siguiente manera para cada uno de los participantes del equipo de desarrollo:

Cliente: Pedro Javier Trujillo Tarazona

Gestor: Jennifer Mendoza

Encargado de seguimiento y pruebas: Mildred Arroyave y Jennifer Mendoza

Programador: Mildred Arroyave y Jennifer Mendoza

5.1. FASE DE EXPLORACIÓN

El primer paso para el desarrollo de la herramienta software es la interacción con el cliente, para lo cual, según las características de la metodología son necesarias las historias de usuario; estas historias dan un aspecto general de las operaciones que deberá realizar la herramienta y que poco a poco, al transcurrir las iteraciones, serán pulidas y perfeccionadas para lograr un sistema robusto que sea capaz de satisfacer las necesidades del cliente.

En la primera entrevista con el cliente, en este caso el profesor Pedro Javier Trujillo Tarazona, se determinaron las especificaciones de la herramienta software:

- Crear una interfaz web, para la consulta de los datos sensados y depurados de acuerdo a la normatividad colombiana de calidad del aire, además de la visualización del despliegue de la red.
- Realizar operaciones como: agregar y eliminar los registros de los módulos inalámbricos de dicha base de datos.
- Crear una interfaz de administrador donde se puedan realizar las operaciones anteriormente mencionadas y se ejecuten los programas para la adquisición y depuración de los datos.
- Crear la Base de datos y realizar la respectiva conexión con el aplicativo web y la interfaz del administrador para la manipulación de la información.
- Realizar la codificación para la adquisición de los datos tomados de los sensores que luego serán almacenados en la base de datos.
- Crear un servidor de alojamiento web.

5.1.1. Historias de Usuario

De los puntos tratados anteriormente, se desglosan las siguientes historias de usuario.

- Diseño Interfaz web
- Diseño Interfaz de administrador
- Creación y administración de base de datos
- Construcción de rutinas para adquisición de datos
- Implementación de procesos para la depuración de datos
- Creación de un servidor de alojamiento web
- Definición de procedimientos para el despliegue de la red

Estas historias de usuario son manejadas por medio de tarjetas donde el cliente realiza una descripción en forma clara y sencilla de la historia.

Tabla 3. Historia de usuario No 1. Creación y administración de base de datos

Historia de Usuario No 1
Nombre: Creación y administración de base de datos
Programador responsable: Jennifer Mendoza, Mildred Arroyave
Descripción: Se debe crear una base de datos con las tablas correspondientes para el almacenamiento de los datos tomados de la lectura de los sensores, su tratamiento y depuración, y la información de identificación de los módulos.
Observaciones: El lenguaje de programación utilizado para la creación es MySQL.

Fuente: Autores

Tabla 4. Historia de usuario No 2. Construcción de rutinas para adquisición de datos

Historia de Usuario No 2
Nombre: Construcción de rutinas para adquisición de datos
Programador responsable: Jennifer Mendoza, Mildred Arroyave
Descripción: Se debe codificar las rutinas que son necesarias para la adquisición de los datos tomados por medio de las lecturas de los sensores y acceder a la base de datos donde se almacenará la información de acuerdo a los campos en la tabla correspondiente.
Observaciones: El lenguaje de programación utilizado para la codificación es Python.

Fuente: Autores

Tabla 5. Historia de usuario No 3. Implementación de procesos para la depuración de datos

Historia de Usuario No 3	
Nombre:	Implementación de procesos para la depuración de datos
Programador responsable:	Jennifer Mendoza, Mildred Arroyave
Descripción:	Los datos almacenados previamente en la base de datos originados de las lecturas de los sensores, deben ser depurados de acuerdo al índice nacional de la calidad del aire y almacenados para ser consultados posteriormente desde la interfaz web.
Observaciones:	El lenguaje de programación utilizado para la codificación es Python.

Fuente: Autores

Tabla 6. Historia de usuario No 4. Diseño interfaz web

Historia de Usuario No 4	
Nombre:	Diseño interfaz web
Programador responsable:	Jennifer Mendoza, Mildred Arroyave
Descripción:	Se debe desarrollar una interfaz web que permita: <ul style="list-style-type: none"> – Realizar un nuevo registro de usuario. – Usuario registrado: consultar por medio de formularios los datos depurados almacenados en la base de datos y visualizar donde y como se encuentra desplegada la red. – Administrador: consultar y gestionar las tablas definidas en los modelos de Django, y control de los usuarios registrados.
Observaciones:	El framework utilizado para el desarrollo de la aplicación es Django.

Fuente: Autores

Tabla 7. Historia de usuario No 5. Diseño interfaz de administrador

Historia de Usuario No 5	
Nombre: Diseño interfaz de administrador	
Programador responsable: Jennifer Mendoza, Mildred Arroyave	
Descripción:	Se debe desarrollar una interfaz sencilla donde el administrador luego de ser autenticado, pueda agregar o eliminar dispositivos de la red inalámbrica y ejecutar los programas correspondientes a la rutina y depuración.
Observaciones:	El lenguaje de programación utilizado para la codificación es Python. Se usa la librería de interfaz gráfica wxPython para el desarrollo.

Fuente: Autores

Tabla 8. Historia de usuario No 6. Creación de un servidor de alojamiento web

Historia de Usuario No 6	
Nombre: Creación de un servidor de alojamiento web	
Programador responsable: Jennifer Mendoza, Mildred Arroyave	
Descripción:	Se debe configurar un servidor web para la respectiva visualización de la página desde cualquier dispositivo con acceso a internet.
Observaciones:	El sistema operativo utilizado es Ubuntu y el servidor web Apache.

Fuente: Autores

Tabla 9. Historia de usuario No 7. Definición de procedimientos para el despliegue de la red

Historia de Usuario No 7	
Nombre:	Definición de procedimientos para el despliegue de la red
Programador responsable:	Jennifer Mendoza, Mildred Arroyave
Descripción:	Se debe definir cómo y donde serán ubicados los dispositivos para el respectivo despliegue de la red.
Observaciones:	

Fuente: Autores

5.1.2. Lenguajes de programación y tecnologías

El segundo paso de esta fase es la familiarización del equipo de desarrollo con las herramientas y tecnologías usadas para la implementación del software.

Los lenguajes de programación utilizados para el desarrollo de la herramienta, son MySQL, Python y Django, de los cuales solo se tenía un previo conocimiento de MySQL, por lo tanto, se realizó una indagación sobre el manejo del lenguaje Python y el framework Django.

En esta indagación se consultaron libros de diferentes autores que exponen de manera clara los tipos de variables, sintaxis, paradigmas y estructuras del lenguaje Python [2], [4], [12]; además se adquirió conocimiento de wxPython para la creación de aplicaciones gráficas orientadas a eventos [14]. Para el desarrollo de la interfaz web con el framework Django, se consultó documentación que describe el funcionamiento y estructura del patrón Modelo-Plantilla-Vista (MTV) [29], [26], [1], [7].

Las tecnologías usadas en el desarrollo comprenden los módulos inalámbricos XBee y los sensores de gas. Para los módulos XBee se realizó la búsqueda de información directamente en la documentación del fabricante⁵, donde se encontró un manual con las especificaciones del módulo y el manejo de las tramas, y un programa llamado X-CTU para la respectiva configuración de los módulos, además se realizaron conversaciones con los asesores de Digi⁶ para despejar inquietudes procedentes del manejo de las tramas en el entorno de programación Python e internamente en los módulos.

Se realizó un análisis sobre la normatividad colombiana de la calidad del aire [23], para conocer los rangos de medición y determinar las condiciones en que debían trabajar los sensores, se consultaron las páginas web de fabricantes o distribuidores de sensores para identificar los que más se ajustarán a las condiciones requeridas [27], [28], [31], [32], [33] y [34], con esta información se realizó la adquisición con el fabricante Alphasense de los sensores de dióxido de nitrógeno y ozono, cada uno con su respectiva tarjeta integrada, los cuales se pueden observar en el Anexo A. Los sensores vienen calibrados de fábrica garantizando el ajuste del cero del medidor.

Se hizo un estudio sobre el funcionamiento de los sensores para entender cómo debían ser adaptados al dispositivo módulo inalámbrico XBee.

Por último, se consultó documentación sobre la configuración, instalación y requerimientos para Apache, para poner en ejecución la página web desarrollada en Django [30].

⁵ <http://www.digi.com/support/productdetail?pid=3430&type=documentation>

⁶ Fabricante de los módulos de transmisión inalámbrica XBee. Ver <http://www.digi.com/es/aboutus/>

5.2. FASE DE PLANIFICACIÓN

Realizada la valoración de las historias de usuario, se estima el número de iteraciones necesarias para el desarrollo de la herramienta y la cantidad de historias a ejecutar en cada iteración.

Tabla 10. Especificación del número de iteraciones

No Iteración	Cantidad de historias	Nombre historias de usuario
1	2	Construcción de rutinas para adquisición de datos, y creación y administración de base de datos.
2	1	Implementación de procesos para la depuración de datos
3	1	Diseño Interfaz web
4	2	Diseño Interfaz de administrador y creación de servidor de alojamiento
5	1	Definición de procedimientos para el despliegue de la red

Fuente: Autores

Se descomponen las historias en tareas, las cuales se muestran en la siguiente fase.

5.3. FASE DE ITERACIONES

5.3.1. Primera iteración

En la primera iteración se procede al desarrollo de las historias de usuario 1 y 2, llamadas *creación y administración de base de datos*, y *construcción de rutinas para adquisición de datos*. La historia de usuario 2 se descompone en tareas debido a que su desarrollo tiene mayor complejidad.

- Historia de usuario No 1.

Se creó la base de datos en MySQL y la tabla de información de los dispositivos; la estructura de las tablas para el almacenamiento de las lecturas de los sensores, su tratamiento y depuración se crearon en el modelo de Django y se realizó la sincronización desde Django para incluir las tablas en la base de datos en MySQL, además cuando se realiza esta sincronización Django crea por defecto las tablas de usuarios.

A continuación se muestra a detalle cada campo de las tablas, estas tablas son no relacionales, debido a que no existen llaves foráneas que las relacione.

La primera tabla es la denominada Inf_modulo, en ella se encuentra almacenada la información sobre los módulos desplegados en la red.

Tabla 11. Estructura de la tabla Inf_modulo

Nombre	Tipo de campo	Tipo de datos	Tamaño
serial	Llave primaria	VarChar	8
tipo_dispositivo	-	Char	20

Fuente: Autores

La segunda tabla es la denominada principal_datosensor, donde se encuentra almacenada información correspondiente a las lecturas tomadas de los sensores por medio de las rutinas de envío y transmisión de datos.

Tabla 12. Estructura de la tabla principal_datosensor

Nombre	Tipo de campo	Tipo de datos	Tamaño
muestra	Llave primaria	AutoField	10
tipo_gas	-	CharField	30
Concentrac	-	DecimalField	5
fecha	-	DateField	-
hora	-	TimeField	-

Fuente: Autores

Las siguientes tablas se denominan principal_dato_analizado_1, principal_dato_analizado_8 y principal_dato_analizado_24, en ellas se encuentran los datos en unidades de concentración ppm registrados de acuerdo al tiempo de exposición según lo estipulado en la resolución 610 del 24 de Marzo de 2010.

Tabla 13. Estructura de las tablas principal_dato_analizado

Nombre	Tipo de campo	Tipo de datos	Tamaño
muestra	Llave primaria	AutoField	10
tipo_gas	-	CharField	30
concentrac	-	DecimalField	5
fecha	-	DateField	-
hora	-	TimeField	-

Fuente: Autores

Las tablas siguientes se denominan principal_dato_ica_1, principal_dato_ica_8 y principal_dato_ica_24, en ellas se almacenan los datos que se encuentran registrados en la tabla anterior y han sido depurados de acuerdo al índice nacional de la calidad del aire.

Tabla 14. Estructura de las tablas principal_dato_ICA

Nombre	Tipo de campo	Tipo de datos	Tamaño
muestra	Llave primaria	AutoField	10
tipo_gas	-	CharField	30
concentrac	-	PositiveIntegerField	3
fecha	-	DateField	-
hora	-	TimeField	-

Fuente: Autores

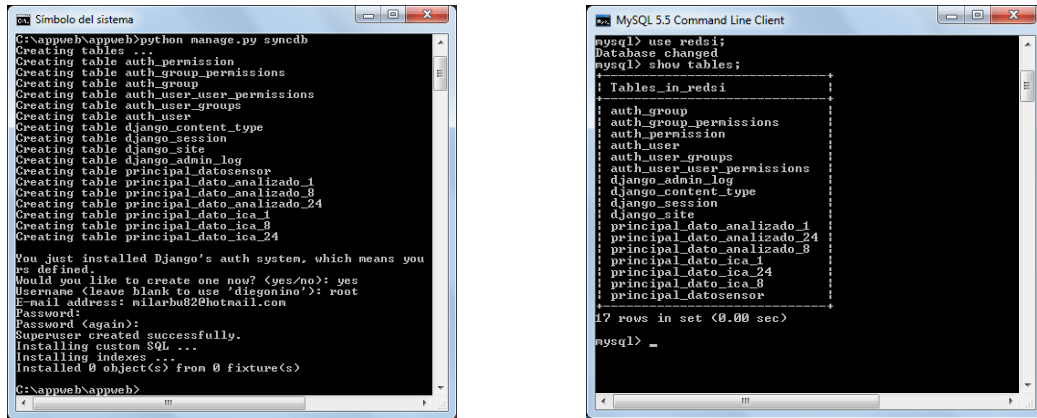
A continuación se muestran las pruebas realizadas para la creación de las tablas.

Tabla 15. Prueba 1 para la historia de usuario No 1

Prueba 1	
Descripción:	La prueba consiste en comprobar la correcta sincronización entre el modelo de Django y la base de datos de MySQL.
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 10. Sincronización de tablas entre Django y MySQL



Fuente: Autores

Al realizar la sincronización de las tablas, se crea un superusuario para gestionar el administrador de Django.

- Historia de usuario No 2

Tabla 16. Tarea 1 para la historia de usuario 2

Tarea de Ingeniería	
Número Tarea: 1	Número Historia: 2
Nombre: Configuración de módulos XBee	
Programador Responsable: Mildred Arroyave	
Descripción: Realizar la configuración de los módulos con el programa X-CTU.	

Fuente: Autores

Se descargó y ejecutó el programa X-CTU de la página de Digi, y haciendo uso del programa, se procede a realizar la configuración necesaria para el funcionamiento correcto de los módulos en la red, la cual consiste en:

- Cambiar el firmware de los módulos a ZigBee debido a que estaban configurados con Znet.
- Habilitar la API dentro de los módulos para acceder al manejo de las tramas.
- Configurar el firmware de los módulos como Coordinador ZigBee, Router ZigBee o Dispositivo Final ZigBee.
- Configurar la identificación de red de área personal (PAN ID) y el canal común a todos los módulos, para que exista comunicación entre ellos.
- Habilitar el enrutamiento muchos a uno, para que los módulos puedan enviar las tramas sin que exista pérdida de datos.
- Habilitar en los routers la verificación del canal para confirmar que un coordinador se encuentra en el mismo canal y con igual identificador de red de área personal.
- Habilitar la llave de seguridad para bloquear el acceso a la red de módulos no autorizados.
- Habilitar las entradas-salidas digitales 1 y 2 en los módulos de los dispositivos finales como conversor Análogo-Digital.

Tabla 17. Tarea 2 para la historia de usuario 2

Tarea de Ingeniería	
Número Tarea: 2	Número Historia: 2
Nombre: Codificar rutinas para el envío y recepción de datos desde los sensores.	
Programador Responsable: Jennifer Mendoza	
Descripción: Teniendo identificados los tipos de tramas de recepción y transmisión, y el tipo de enrutamiento necesarios, realizar la codificación de las rutinas y abrir la conexión con la base de datos para almacenar los datos recibidos de los sensores.	

Fuente: Autores

Para realizar la codificación de las rutinas fue necesario descargar la librería Pyserial⁷ para habilitar el acceso a los datos por el puerto serial. La codificación de las rutinas se realiza de la siguiente manera:

Se envía una trama de descubrimiento de nodos desde el dispositivo coordinador, para reconocer los dispositivos finales o router que forman parte de la red, identificados por tener la misma llave de seguridad, identificador de red de área personal y encontrarse en el mismo canal; cada dispositivo de la red devuelve al coordinador dos tramas de respuesta en código ASCII.

- Una trama de recepción de respuesta al descubrimiento de nodos, que consta de la dirección de 64 bits con la cual se identifica el dispositivo final o router.

⁷ Ver <https://pypi.python.org/pypi/pyserial>

- Una trama indicadora de registro de ruta que contiene las direcciones de 16 bits correspondientes a los dispositivos router por los cuales debe pasar el mensaje para llegar al coordinador.

Después de conocer el registro de ruta de cada dispositivo router, se empaqueta cada ruta en una trama de creación de ruta de origen para definir por donde debe pasar el mensaje, disminuyendo así el riesgo de pérdida de datos.

Una vez creada la ruta de origen, se envían tramas de solicitud de muestra cada 15 minutos a los dispositivos finales, los cuales devuelven en una trama de recepción las lecturas del sensor, correspondientes a las dos entradas de conversión análoga-digital habilitadas.

Estas tramas son desempaquetadas para disponer de las lecturas e introducidas en una fórmula especificada en el manual de usuario del sensor, esta fórmula se usa para la conversión de las lecturas a ppm.

$ppm = (OP_1 - OP_2) / Sensibilidad$ donde:

- OP_1 y OP_2 son las lecturas de las entradas de conversión análoga-digital
- La sensibilidad viene especificada para cada sensor desde fábrica.

Y por último, se almacena en la base de datos la información de las lecturas para cada sensor con la hora y fecha en que fueron tomadas.

Si la trama recibida no está completa se retransmite la trama.

Al ser terminadas las tareas de la historia se realizan las pruebas correspondientes para la evaluación del buen funcionamiento de la configuración

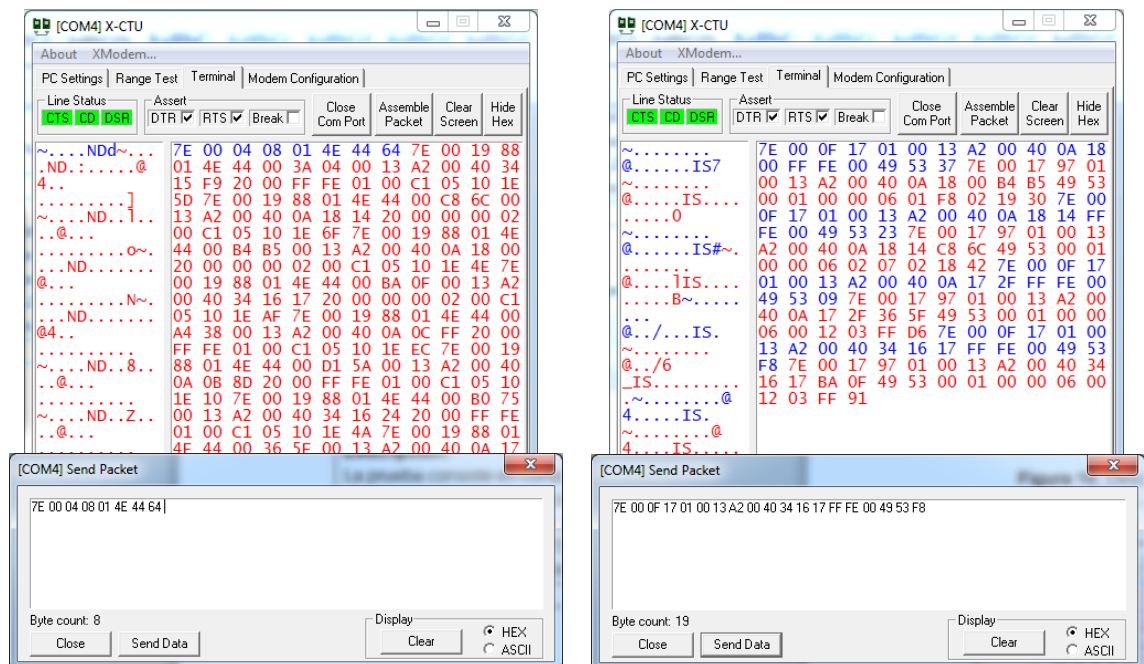
de los módulos y codificación de las rutinas. A continuación se muestran las pruebas realizadas para evaluar el desarrollo de las tareas.

Tabla 18. Prueba 1 para la historia de usuario 2

Prueba 1	
Descripción:	La prueba consiste en comprobar el adecuado funcionamiento de los dispositivos mediante el envío y recepción de tramas por medio del programa X-CTU, se hace la conexión del dispositivo coordinador de la red a un portátil por medio del puerto serial utilizando un cable RS-232 y se despliegan los demás dispositivos de la red (routers y dispositivos finales).
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 11. Descubrimiento de nodos y, envío y recepción de trama.



Fuente: Autores

Pruebas similares se realizaron para las diferentes tramas manejadas en el desarrollo de las rutinas.

Para la realización de las siguientes pruebas se realiza la misma conexión y despliegue de los dispositivos descrita en la prueba 1.

Tabla 19. Prueba 2 para la historia de usuario 2

Prueba 2	
Descripción:	Con esta prueba se pretende verificar que el código de las rutinas para el descubrimiento de nodos y rutas funcione correctamente.
Evaluación de la prueba:	100% completado

Fuente: Autores

Tabla 20. Prueba 3 para la historia de usuario 2

Prueba 3	
Descripción:	El fin de la prueba es comprobar que la creación de la ruta de origen está bien definida al recibir la trama de adquisición de datos por el coordinador de la red.
Evaluación de la prueba:	100% completado

Fuente: Autores

Tabla 21. Prueba 4 para la historia de usuario 2

Prueba 4	
Descripción:	
La prueba consiste en verificar el almacenamiento de la información en la tabla Dato_sensor y el total funcionamiento de las rutinas.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 12. Envío de tramas a través de rutinas y almacenamiento de los datos en MySQL

```

Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MS
C v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for mo
re information.
>>> ----- RESTART -----
>>>
97010013a200400a1800b4b54953000100006020f0223
97010013a200400a1814c86c49530001000060223021d
97010013a200400a172f365f4953000100006001201df
97010013a200400a1617ba0f4953000100006001202ca
97010013a200400a1800b4b54953000100006020a021c
97010013a200400a1814c86c4953000100006020a021e
97010013a200400a172f365f4953000100006001203ff
97010013a200400a1617ba0f4953000100006001203ff
97010013a200400a1800b4b5495300010000601fe0214
97010013a200400a1814c86c4953000100006022202e2
97010013a200400a172f365f4953000100006001202e4
97010013a200400a1617ba0f4953000100006001203ff
97010013a200400a1800b4b54953000100006021a0230
97010013a200400a1814c86c49530001000060211021b
97010013a200400a172f365f4953000100006001300ff
97010013a200400a1617ba0f4953000100006001203ff
97010013a200400a1800b4b54953000100006020a020a
  
```

```

358 | ozono | 0.00380 | 2013-01-08 | 00:38:38 |
359 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:38:38 |
360 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:38:38 |
361 | dióxidoNitrogeno | 0.07110 | 2013-01-08 | 00:39:31 |
362 | ozono | 0.00240 | 2013-01-08 | 00:39:31 |
363 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:39:31 |
364 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:39:31 |
365 | dióxidoNitrogeno | 0.00170 | 2013-01-08 | 00:40:32 |
366 | ozono | 0.00250 | 2013-01-08 | 00:40:32 |
367 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:40:32 |
368 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:40:32 |
369 | dióxidoNitrogeno | 0.00880 | 2013-01-08 | 00:41:32 |
370 | ozono | 0.02400 | 2013-01-08 | 00:41:32 |
371 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:41:32 |
372 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:41:32 |
373 | dióxidoNitrogeno | 0.09430 | 2013-01-08 | 00:42:34 |
374 | ozono | 0.78140 | 2013-01-08 | 00:42:34 |
375 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:42:34 |
376 | dióxidoSulfuro | 0.00880 | 2013-01-08 | 00:43:35 |
377 | dióxidoNitrogeno | 0.02950 | 2013-01-08 | 00:43:35 |
378 | ozono | 0.78140 | 2013-01-08 | 00:43:35 |
379 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:43:35 |
380 | dióxidoSulfuro | 0.00170 | 2013-01-08 | 00:44:35 |
381 | dióxidoNitrogeno | 0.00460 | 2013-01-08 | 00:44:35 |
382 | ozono | 0.78140 | 2013-01-08 | 00:44:35 |
383 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:44:35 |
384 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:44:35 |
385 | dióxidoNitrogeno | 0.07540 | 2013-01-08 | 00:45:36 |
386 | ozono | 0.00090 | 2013-01-08 | 00:45:36 |
387 | monoxidoCarbono | 0.78140 | 2013-01-08 | 00:45:36 |
388 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:45:36 |
389 | dióxidoNitrogeno | 0.07110 | 2013-01-08 | 00:46:37 |
390 | ozono | 0.00540 | 2013-01-08 | 00:46:37 |
391 | monoxidoCarbono | 0.78070 | 2013-01-08 | 00:46:37 |
392 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:46:37 |
393 | dióxidoNitrogeno | 0.10370 | 2013-01-08 | 00:47:37 |
394 | ozono | 0.00750 | 2013-01-08 | 00:47:37 |
395 | monoxidoCarbono | 0.78070 | 2013-01-08 | 00:47:37 |
396 | dióxidoSulfuro | 0.78140 | 2013-01-08 | 00:47:37 |
397 | dióxidoNitrogeno | 0.07540 | 2013-01-08 | 00:48:38 |
398 | ozono | 0.02720 | 2013-01-08 | 00:48:38 |
399 | monoxidoCarbono | 0.78070 | 2013-01-08 | 00:48:39 |
  
```

Fuente: Autores

5.3.2. Segunda iteración

En la segunda iteración se procede al desarrollo de la tercera historia de usuario, llamada *implementación de procesos para el tratamiento y depuración de datos*. Se descompone en las siguientes tareas:

Tabla 22. Tarea 1 para la historia de usuario 3

Tarea de Ingeniería	
Número Tarea: 1	Número Historia: 3
Nombre: Tratamiento de los datos según resolución 610 del 24 de Marzo de 2010.	
Programador Responsable: Mildred Arroyave	
Descripción: Tratar los datos previamente almacenados, según el tiempo de exposición del gas especificado en la resolución y almacenarlos en nuevas tablas en la base de datos.	

Fuente: Autores

La clasificación de los datos se realiza de acuerdo al tiempo de exposición a condiciones de referencia para contaminantes criterio establecidos en la resolución 610 del 24 de Marzo de 2010, este tiempo depende del tipo de gas.

Tabla 23. Tiempos de exposición para contaminantes criterio

Contaminante	Tiempo de exposición
SO ₂	24 horas
NO ₂	1 hora
	24 horas
O ₃	1 hora
	8 horas
CO	1 hora
	8 horas

Fuente: Adaptada por los autores de la Resolución 610 del 24 de Marzo de 2010

De acuerdo a este tiempo de exposición los datos que están previamente almacenados en la tabla principal_datosensor se analizan de acuerdo al tiempo de exposición y se almacenan en la base de datos en las tablas principal_dato_analizado_1, principal_dato_analizado_8 y principal_dato_analizado_24.

Tabla 24.Tarea 2 para la historia de usuario 3

Tarea de Ingeniería	
Número Tarea: 2	Número Historia: 3
Nombre: Depuración de los datos de acuerdo al índice de calidad del aire	
Programador Responsable: Jennifer Mendoza	
Descripción: Teniendo los datos tratados según la resolución, realizar la depuración de los datos de acuerdo al índice de calidad del aire y almacenarlos en la base de datos.	

Fuente: Autores

El ICA será calculado a partir de la siguiente ecuación, que corresponde a la metodología utilizada por la US-EPA para el cálculo del Índice de la Calidad del aire y será reportado el mayor valor que se obtenga de las diferentes ecuaciones individuales para cada gas.

$$I_p = \frac{I_{Hi} - I_{Lo}}{BP_{Hi} - BP_{Lo}} (C_p - BP_{Lo}) + I_{Lo}$$

Donde:

I_p = Índice para el contaminante p

C_p = Concentración medida para el contaminante p

BP_{Hi} = Punto de corte mayor o igual a C_p

BP_{Lo} = Punto de corte menor o igual a C_p

I_{Hi} = Valor del índice de calidad del aire correspondiente al BP_{Hi}

I_{Lo} = Valor del índice de calidad del aire correspondiente al BP_{Lo}

Los valores para BP_{Hi} , BP_{Lo} , I_{Hi} e I_{Lo} son tomados de la siguiente tabla.

Tabla 25. Puntos de corte del ICA

ICA	Descripción	O ₃ 8h (ppm)	O ₃ 1h (ppm)	CO 8h (ppm)	SO ₂ 24h (ppm)	NO ₂ 1h (ppm)
0-50	Buena	0,000-0,059	-	0,0-4,4	0,000-0,034	-
51-100	Moderada	0,060-0,075	-	4,5-9,4	0,035-0,144	-
101-150	Dañina a la salud para grupos sensibles	0,076-0,095	0,125-0,164	9,5-12,4	0,145-0,224	-
151-200	Dañina a la salud	0,096 - 0,115	0,165- 0,204	12,5- 15,4	0,225- 0,304	-
201-300	Muy dañina a la salud	0,116- 0,374	0,205- 0,404	15,5- 30,4	0,305- 0,604	0,65- 1,24
301-400	Peligrosa	-	0,405- 0,504	30,5- 40,4	0,605- 0,804	1,25- 1,64
401.500	Peligrosa	-	0,505- 0,604	40,5- 50,4	0,805- 1,004	1,65- 2,04

Fuente: Adaptada de Sisaire⁸

Se toma el mayor valor del día para cada gas de las tablas principal_dato_analizado_1, principal_dato_analizado_8 y principal_dato_analizado_24, se calcula el índice para el contaminante por medio de la ecuación de acuerdo a lo especificado en la tabla y se procede a

⁸ Ver www.sisaire.gov.co

almacenarlo en las tablas principal_dato_ica_1, principal_dato_ica_8 y principal_dato_ica_24.

A continuación se muestran las pruebas realizadas para evaluar el desarrollo de las tareas.

Tabla 26. Prueba 1 para la historia de usuario 3

Prueba 1	
Descripción:	
Realizar la verificación del almacenamiento de los datos tratados según la resolución en las tablas principal_dato_analizado_1, principal_dato _analizado_8 y principal_dato _analizado_24.	
Evaluación de la prueba:	100% completado

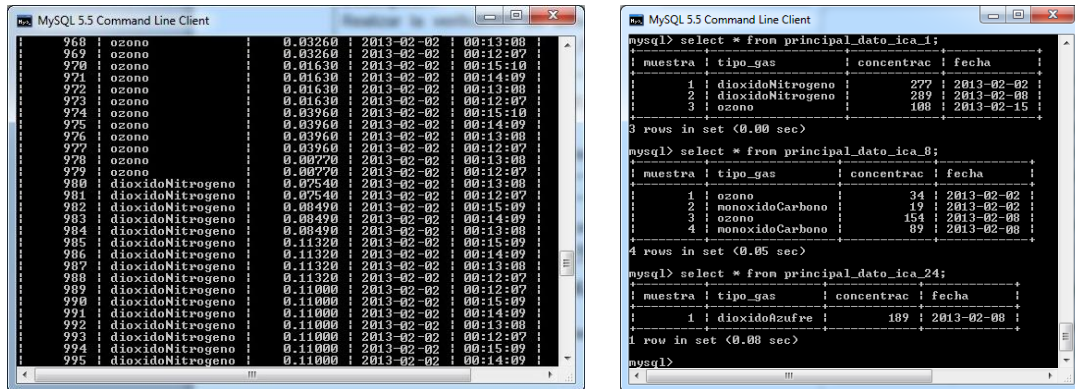
Fuente: Autores

Tabla 27. Prueba 2 para la historia de usuario 3

Prueba 2	
Descripción:	
Realizar la verificación del almacenamiento de la depuración de los datos en las tablas principal_dato_ica_1, principal_dato_ica _8 y principal_dato_ica _24.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 13. Almacenamiento de los datos tratados y depurados



Fuente: Autores

5.3.3. Tercera iteración

Para la tercera iteración se desarrolla la historia de usuario No 4 llamada *diseño Interfaz web*, se empleó el framework Django que permite generar un código más práctico al crear los componentes de la interfaz de manera independiente, modelo (bases de datos), vista (formularios) y plantilla (html, etiquetas, Javascript, css, cvs).

Para el desarrollo de la historia, se realizó la instalación de librerías de Python, que son útiles para tareas específicas como *staticfiles*⁹ para gestión de archivos estáticos, *gmap*¹⁰ para la administración del mapa de georeferenciamiento, entre otras, estas librerías fueron descargadas del sitio oficial de Python.

Esta iteración se descompone en las siguientes tareas:

⁹ Ver <https://pypi.python.org/pypi/django-staticfiles>

¹⁰ Ver <https://pypi.python.org/pypi/django-gmap>

Tabla 28. Tarea 1 para la historia de usuario 4

Tarea de Ingeniería	
Número Tarea: 1	Número Historia: 4
Nombre: Diseñar página de inicio para ingreso y registro de usuarios	
Programador Responsable: Jennifer Mendoza	
Descripción: Diseñar un formulario de ingreso, donde el usuario debe digitar su nombre y contraseña para acceder a los datos; si el usuario no se encuentra registrado debe diligenciar el formulario de nuevo usuario.	

Fuente: Autores

Tabla 29. Tarea 2 para la historia de usuario 4

Tarea de Ingeniería	
Número Tarea: 2	Número Historia: 4
Nombre: Diseñar menú para facilitar el acceso a la consulta de datos y visualización de la red.	
Programador Responsable: Mildred Arroyave	
Descripción: Diseñar un menú interactivo donde el usuario podrá acceder a la diferente información mostrada mediante la página web.	

Fuente: Autores

Tabla 30.Tarea 3 para la historia de usuario 4

Tarea de Ingeniería	
Número Tarea: 3	Número Historia: 4
Nombre: Diseñar formulario para la consulta de datos	
Programador Responsable: Mildred Arroyave	
Descripción: Diseñar un formulario donde el usuario podrá seleccionar entre: el tiempo de exposición, el tipo de gas y la fecha de medición inicial y final, para visualizar los datos depurados de acuerdo al ICA, estos datos deben ser mostrados por medio de una tabla y una gráfica. Además mostrar alertas si los campos no han sido seleccionados correctamente o si algún campo se encuentra vacío.	

Fuente: Autores

Tabla 31.Tarea 4 para la historia de usuario 4

Tarea de Ingeniería	
Número Tarea: 4	Número Historia: 4
Nombre: Visualización de la ubicación de los dispositivos de la red	
Programador Responsable: Jennifer Mendoza	
Descripción: Diseñar un esquema web donde se mostrará la ubicación de los dispositivos por medio de un mapa georeferenciado, además de un grafo que muestre la forma en que se encuentra desplegada la red.	

Fuente: Autores

Tabla 32.Tarea 5 para la historia de usuario 4

Tarea de Ingeniería	
Número Tarea: 5	Número Historia: 4
Nombre: Diseñar formulario para consulta de reporte diario de los datos	
Programador Responsable: Jennifer Mendoza	
Descripción: Diseñar un formulario donde se seleccione una fecha y posteriormente se despliegue una tabla donde se muestren los datos según el tipo de gas, el tiempo de exposición, mayor concentración y los daños que puede causar esta concentración dependiendo del rango en que se encuentre.	

Fuente: Autores

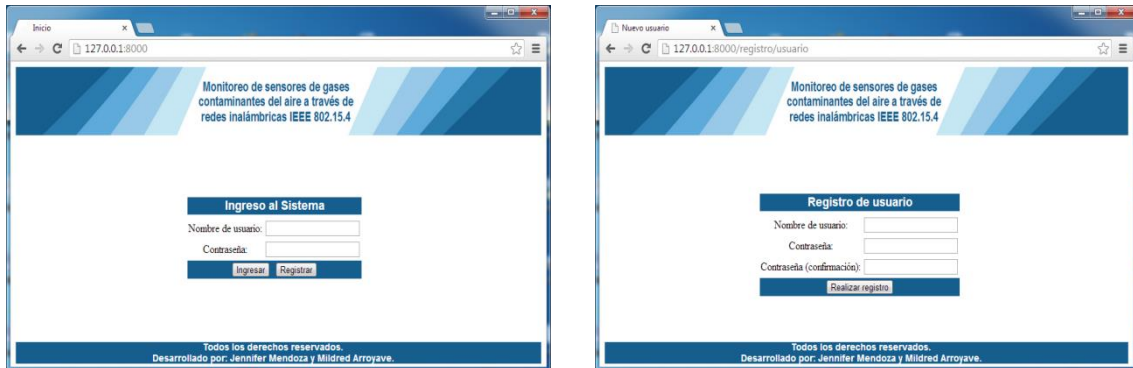
A continuación se muestran las pruebas realizadas para evaluar el desarrollo de las tareas.

Tabla 33.Prueba 1 para la historia de usuario 4

Prueba 1	
Descripción: Realizar la revisión y control del ingreso del usuario y su posterior registro en caso de no encontrarse en la base de datos.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 14. Interfaz de ingreso y registro de usuario



Fuente: Autores

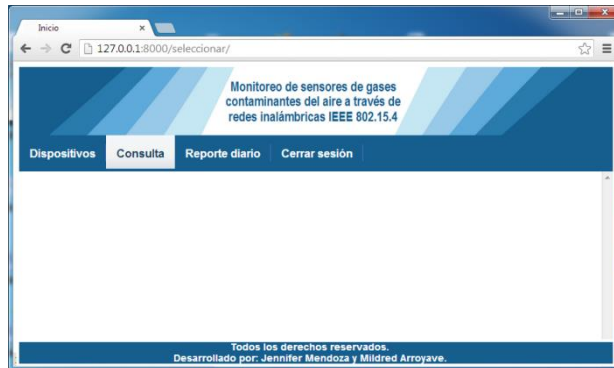
El ingreso al sistema consta de un formulario que solicita el nombre de usuario y la contraseña, si el usuario no está registrado, se muestra el formulario respectivo para realizar el nuevo registro.

Tabla 34. Prueba 1 para la historia de usuario 4

Prueba 2	
Descripción:	
Verificar la funcionalidad del menú.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 15. Menú interfaz web



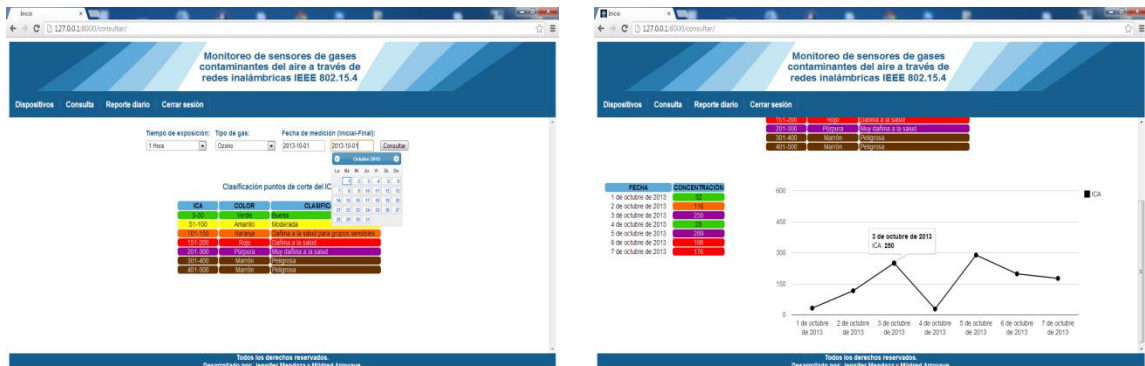
Fuente: Autores

Tabla 35. Prueba 3 para la historia de usuario 3

Prueba 3	
Descripción:	
Comprobar que efectivamente los datos seleccionados en el formulario sean correctamente consultados en la base de datos y visualizados en la página web.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 16. Formulario de consulta de datos depurados



Fuente: Autores

Tabla 36. Prueba 3 para la historia de usuario 3

Prueba 4	
Descripción:	
Confirmar la correcta visualización de la ubicación de los dispositivos por medio del mapa y la respectiva estructuración de la red.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 17. Visualización del despliegue de la red



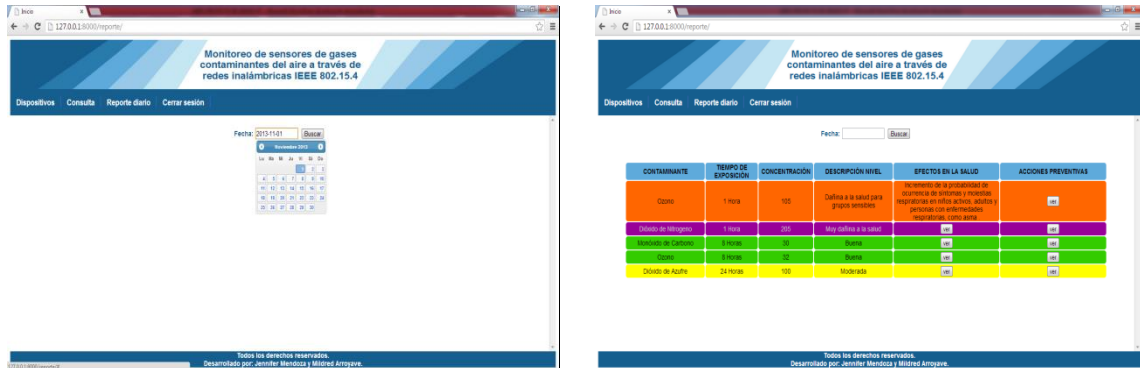
Fuente: Autores

Tabla 37. Prueba 5 para la historia de usuario 4

Prueba 5	
Descripción:	
Comprobar que la fecha seleccionada en el formulario de reporte diario sea correctamente consultada en la base de datos y posteriormente se visualicen los datos en la página web.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 18.Formulario de consulta de reporte diario



Fuente: Autores

5.3.4. Cuarta iteración

En la cuarta iteración se procede al desarrollo de la quinta y sexta historias de usuario, llamadas *diseño interfaz de administrador y creación de un servidor de alojamiento web*.

Estas historias se descomponen en las siguientes tareas.

Tabla 38.Tarea 1 para la historia de usuario 5

Tarea de Ingeniería	
Número Tarea: 1	Número Historia: 5
Nombre: Diseñar ventana de ingreso de administrador	
Programador Responsable: Jennifer Mendoza	
Descripción: Diseñar un formulario donde se digite el nombre de usuario y la contraseña para acceder al entorno de administración.	

Fuente: Autores

Tabla 39.Tarea 2 para la historia de usuario 5

Tarea de Ingeniería	
Número Tarea: 2	Número Historia: 5
Nombre: Diseñar ventana de procesos	
Programador Responsable: Jennifer Mendoza	
Descripción: Diseñar una ventana emergente que permita por medio de un menú seleccionar la operación que se realiza a los dispositivos y la ejecución de las rutinas.	

Fuente: Autores

Tabla 40. Tarea 1 para la historia de usuario 6

Tarea de Ingeniería	
Número Tarea: 1	Número Historia: 6
Nombre: Instalación y configuración de un servidor web con apache	
Programador Responsable: Mildred Arroyave	
Descripción: Realizar la instalación completa del sistema operativo Ubuntu Linux y la configuración necesaria para montarlo como un servidor web usando apache, proceder a instalar los programas necesarios para el funcionamiento de la página web realizada en Django y así ser mostrada por medio del servidor.	

Fuente: Autores

Se realizó la instalación del sistema operativo Ubuntu en un servidor PowerEdge T105, ubicado en la oficina 159 de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander; posteriormente se instala Xampp para Linux el cual contiene Apache 2.4.4, MySQL 5.5 y mod_python, donde mod_python se encarga de embeber Python dentro de Apache.

Después de concluida la instalación se corren los comandos correspondientes para la instalación de Python, el cual viene por defecto en el sistema operativo Linux, luego se procede a instalar Django y las librerías de Python necesarias para el correcto funcionamiento de la página web.

Para terminar la configuración de Apache, se crea un script en Python que es guardado dentro de la carpeta del proyecto realizado en Django, el cual permite reconocer este proyecto como una página web.

La IP manejada en el servidor es una IP privada 192.168.65.163 que solo puede ser accedida dentro del campus universitario.

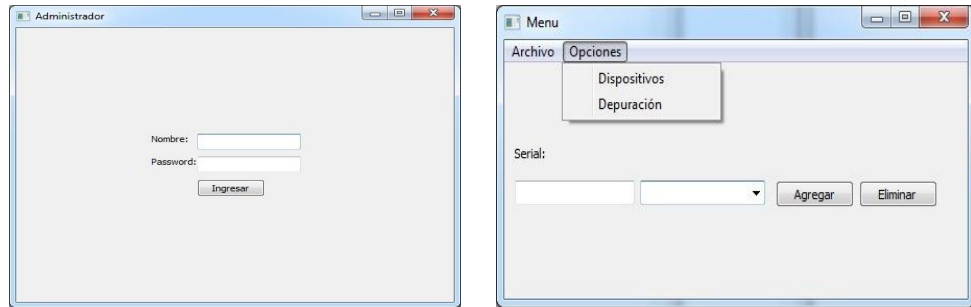
A continuación se muestran las pruebas realizadas para evaluar el desarrollo de las tareas.

Tabla 41. Prueba 1 para la historia de usuario 5

Prueba 1	
Descripción:	Verificar la funcionalidad total de la interfaz. Evaluando el ingreso del administrador y las acciones que puede realizar.
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 19. Interfaz de gestión del administrador



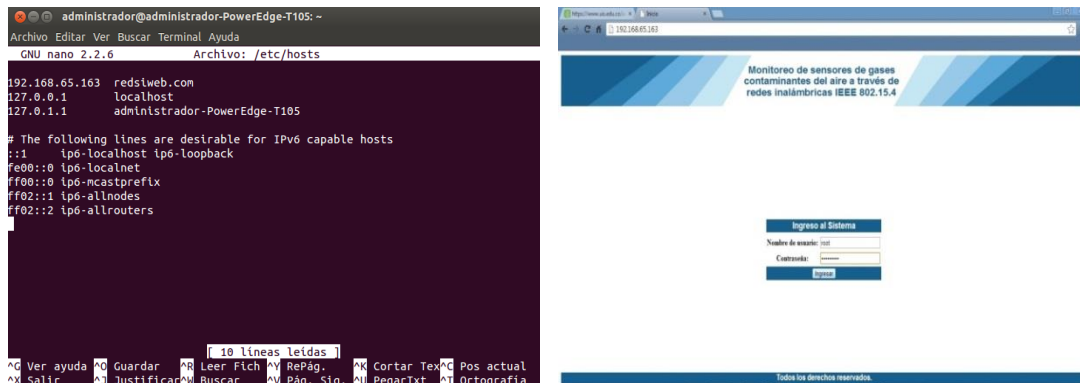
Fuente: Autores

Tabla 42. Prueba 1 para la historia de usuario 6

Prueba 1	
Descripción:	
Comprobar que el servidor web está correctamente configurado al realizar consultas a la página web desde cualquier dispositivo que cuente con internet dentro del campus universitario.	
Evaluación de la prueba:	100% completado

Fuente: Autores

Figura 20. Asignación de IP y visualización de la página web desde el campus universitario



Fuente: Autores

5.3.5. Quinta iteración

En la quinta iteración se procede al desarrollo de la séptima historia de usuario, llamada *definición de procedimientos para el despliegue de la red*.

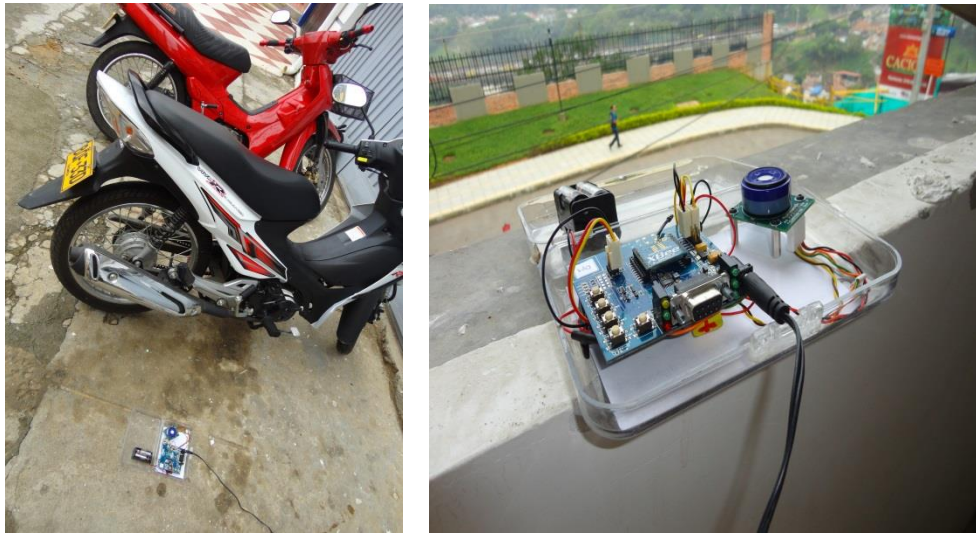
El despliegue de la red se realiza dentro del campus universitario debido a que el dispositivo coordinador debe estar conectado al servidor web por medio del puerto serial utilizando un cable RS-232; por la adquisición de solo dos sensores, se simulan los datos correspondientes a los otros dos gases que se visualizan en el despliegue de la red.

5.4. FASE DE PRODUCCIÓN

Se realizaron pruebas adicionales antes de desplegar la red dentro del campus universitario.

Se desplego la red en la casa de una de las alumnas para comprobar el funcionamiento de los sensores. Las lecturas eran enviadas al coordinador el cual se encontraba conectado a un portátil por medio de un cable RS-232 donde eran almacenadas en la base de datos y visualizadas en la página web en el localhost.

Figura 21. Despliegue de los sensores de dióxido de nitrógeno y ozono



Fuente: Autores

Figura 22. Despliegue de coordinador y router



Fuente: Autores

Se comprobó que los sensores median correctamente, se encontraban dentro de los rangos especificados y con emisiones de gas altas se observaban los cambios en las concentraciones.

5.5. FASE DE MANTENIMIENTO

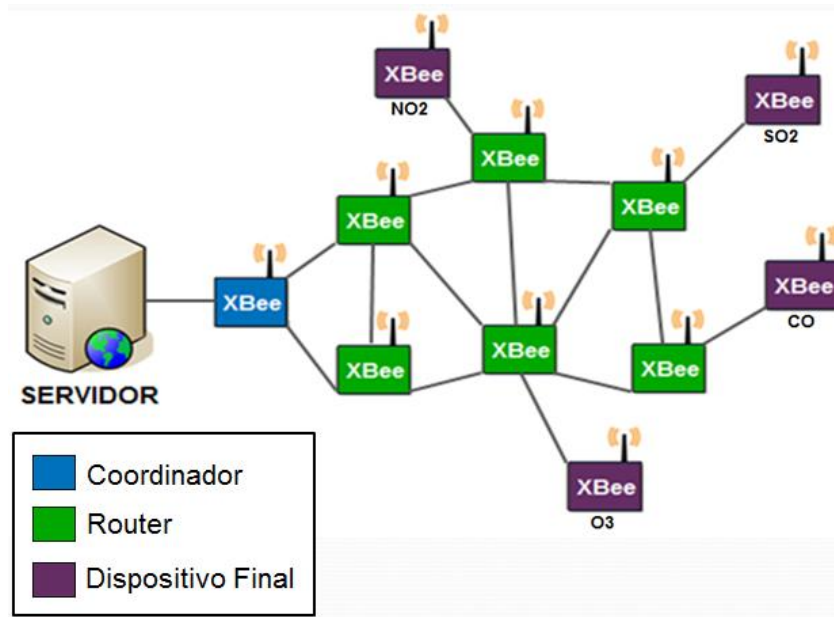
Se realiza el despliegue de la red dentro del campus universitario. La ubicación de los dispositivos inalámbricos queda distribuida de la siguiente manera:

Tabla 43. Ubicación de los dispositivos inalámbricos dentro del campus universitario

DISPOSITIVO	SERIAL	UBICACIÓN
Coordinador	0013A200400A1257	Oficina 159 de la escuela de Ingeniería de Sistemas e Informática
Router1	0013A200403415F9	Edificio de la escuela de Ingeniería de Sistemas e Informática
Router2	0013A200400A0B8D	Edificio de la escuela de Ingeniería de Sistemas e Informática.
Router3	0013A20040341624	Edificio de la escuela de Ingeniería de Petróleos.
Router4	0013A200400A0CFF	Edificio de la escuela de Ingeniería de Petróleos.
Router 5	0013A20040016049	Edificio de la escuela de Ingeniería Civil.
Router 6	0013A200400A0F9F	Edificio de la escuela de Ingeniería de Petróleos.
Dispositivo Final 1 (Dióxido de Nitrógeno)	0013A200400A1800	Oficina 160. Posgrados de petróleo. Escuela de Ingeniería de Sistemas e Informática
Dispositivo Final 2 (Ozono)	0013A200400A1814	Oficina 131. Grupo de investigación de recobro mejorado. Escuela de ingeniería de petróleo.
Dispositivo Final 3 (Monóxido de Carbono)	0013A200400A172F	Oficina 130. Escuela de ingeniería Civil
Dispositivo Final 4 (Dióxido de Azufre)	0013A20040341617	Escuela de Metalúrgica.

Fuente: Autores

Figura 23. Despliegue de la red inalámbrica de sensores



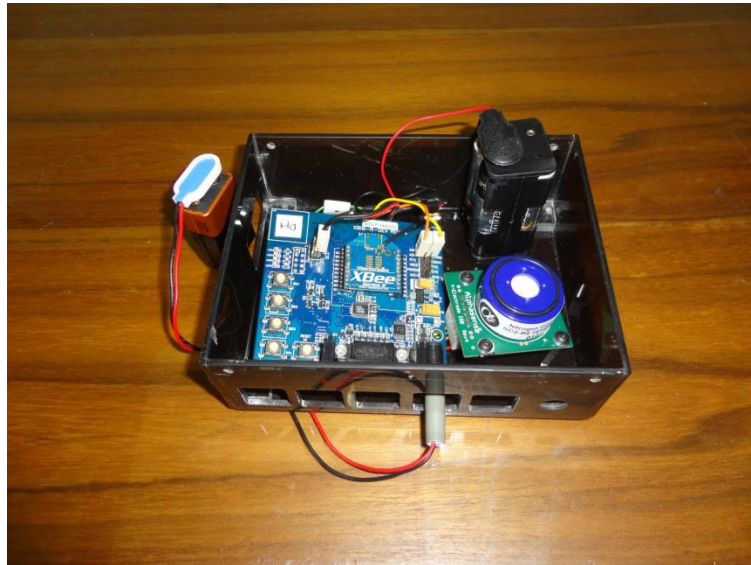
Fuente: Autores

Figura 24. Ubicación Coordinador. Oficina 159.



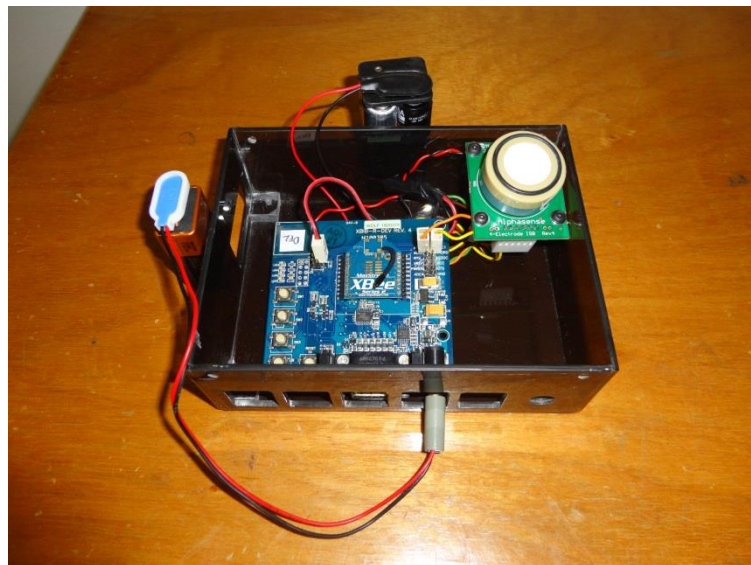
Fuente: Autores

Figura 25. Ubicación sensor de dióxido de Nitrógeno. Oficina 160



Fuente: Autores

Figura 26. Ubicación sensor de ozono. Oficina 131.



Fuente: Autores

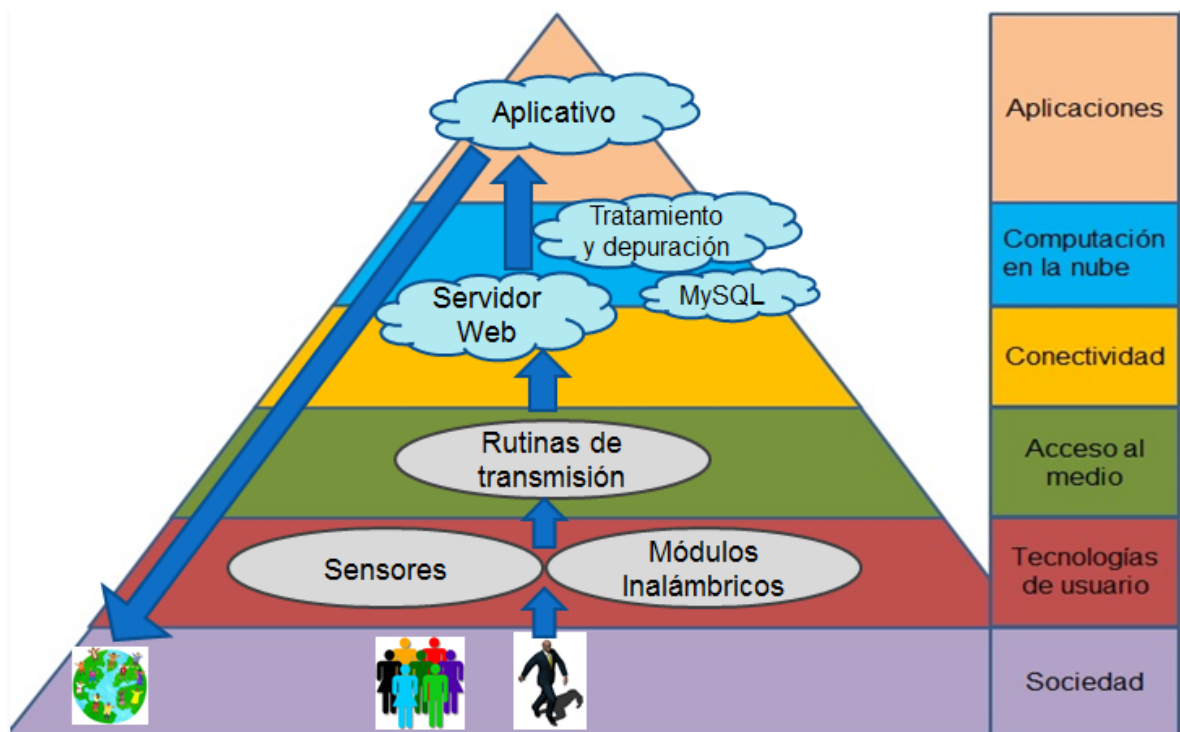
Los routers se distribuyeron de forma que la señal alcanzara a transmitirse hacia los dispositivos finales.

5.6. FASE FINAL DEL PROYECTO

Realizada la valoración del cumplimiento de las historias de usuario, por medio de las pruebas y el funcionamiento total del sistema, se concluye la finalización del proyecto.

Tomando como referencia el modelo de integración del centro TIC de la Universidad Industrial de Santander se realizó el diseño del sistema, el cual fue distribuido en las capas del modelo como se muestra a continuación.

Figura 27. Diseño del sistema



Fuente: Autores

6. CONCLUSIONES

- El desarrollo de ésta herramienta software permite al usuario tener a disposición información en tiempo real, útil y completa sobre los niveles de contaminación en las zonas monitoreadas.
- Las pruebas del sistema se realizan de forma constante y son parte fundamental de la aplicación de la metodología ágil programación extrema, cuya finalidad es incrementar la calidad del producto reduciendo el porcentaje de error y garantizando que la aplicación cumple con los requerimientos establecidos.
- Los sensores adquiridos cuentan con la precisión necesaria para que sus medidas estén en el rango especificado por la normatividad.
- Se comprendió el funcionamiento de los dispositivos de transmisión inalámbrica XBee y su modo de configuración.
- La participación en el programa de Apps.co permitió evaluar en un contexto real si el proyecto desarrollado era una posible solución a la contaminación del aire.
- El desarrollo del proyecto fue una excelente oportunidad para adquirir conocimientos sobre nuevos lenguajes de programación, la legislación colombiana referente a la calidad del aire, condiciones y/o problemas ambientales actuales y el uso de tecnologías que pueden servir como apoyo en el seguimiento y control de la contaminación.

7. RECOMENDACIONES

- Dentro de la diversidad del conocimiento, se recomienda explorar las facilidades de desarrollo que ofrece Python.
- Se recomienda el desarrollo de proyectos en el área de redes, el cual tiene campos de trabajo amplios y sin explorar.
- Desarrollar un aplicativo móvil para que el usuario pueda consultar la información desde cualquier lugar.
- Rediseñar el almacenamiento de las lecturas, usando compuertas celulares que permitan el envío de las lecturas a la base de datos por medio de internet.
- Implementar un sistema de seguimiento por GPS automático para el rastreo de los sensores, de tal manera que en la página se encuentren actualizadas las ubicaciones de los mismos.

BIBLIOGRAFÍA

- [1] ALCHIN, Marty. *Pro Django*. 2 ed. Apress. 2009. 273 p.
- [2] BEAZLEY, David M. *Python Essential Reference*. 4 ed. Addison-Wesley Professional. 2009. 744 p.
- [3] CHOU, Jack. *Hazardous Gas Monitors: A practical Guide to Selection, Operation and Applications*. MacGraw-Hill. 256 p.
- [4] CHUN, Wesley J. *Core Python Programming*. 2 ed. Prentice Hall. 2006. 1136 p.
- [5] ELAHI, Ata y GSHWENDER, Adam. *ZigBee Wireless Sensor and Control Network*. Prentice Hall.. 2009. 288 p.
- [6] FALUDI, Robert. *Building Wireless Sensor Network*. O'Reilly Media, Inc. 2010. 320 p.
- [7] FORGER, Jeff; BISSEX, Paul y CHUN Wesley. *Python Web Development with Django*. Addison-Wesley. 2009. 405 p.
- [8] KARL, Holger y WILLIG, Andreas. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & sons Ltd. 2007. 481 p.
- [9] KENT, Beck. *Extremme Programming Explained*. 2da edición. Addison-Wesley. 2004. 175 p.
- [10] LABRADOR, Miguel A. y WIGHTMAN, Pedro M. *Topology Control in Wireless Sensor Networks with a Companion Simulation Tool for Teaching and Research*. Springer. 2009. 207 p.

[11] LUKASZEWSKI, Albert. *MySQL for Python*. Packt Publishing. 2010. 440p

[12] LUTZ, Mark. *Programming Python*. 4ed. O'Reilly Media, Inc. 2011. 1157 p.

[13] MOHAMMAD, Ilyas y MAHGOUB, Imad. *Handbook of Sensor Networks Compact Wireless and Wired Sensing Systems*. CRC Press. 2004. 776 p.

[14] RAPPIN, Noel y DUNN Robbin. *wxPython in Action*. 2006. Manning Publications Co. 2006. 583 p.

[15] THIBAUD, Cyril. *MySQL 5: Instalación, Implementación, Administración, Programación*. Ediciones ENI. 2006. 464 p.

[16] WALTENEGUS, Dargie y POELLABAUER, Christian. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & sons Ltd. 2010. 303 p.

Recursos web

[17] IEEE Std 802.15.4-2006. "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," (*Revision of IEEE Std 802.15.4-2003*). 2006. 305 p. Disponible en:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1700009&isnumber=35824>>

[18] Colombia. *Política de Prevención y Control de la Contaminación del Aire*. Ministerio de Ambiente, Vivienda y Desarrollo Territorial. Bogotá D.C. 2010. 48 p. <http://www.minambiente.gov.co/documentos/normativa/ambiente/politica/polit_calidad_aire.pdf>

[19] *Manual de Fundamentos y Planeación de Inventarios de Emisiones*. Ministerio de Ambiente, Vivienda y Desarrollo Territorial. K2 Ingeniería Ltda. 52 p. Disponible en: <http://www.minambiente.gov.co/documentos/3110_1_Planeacion.pdf>

[20] *Guías de calidad del aire de la OMS relativas al material particulado, el ozono, el dióxido de nitrógeno y el dióxido de azufre*. Organización Mundial de La salud. 2006. 21 p. Disponible en: <http://whqlibdoc.who.int/hq/2006/WHO_SDE_PHE_OEH_06.02_spa.pdf>

[21] CASTRO, Henry. *Informe Anual de Calidad del Aire de Bucaramanga 2011*. Sistema de vigilancia de calidad del aire del área metropolitana de Bucaramanga. Coordinación de información e investigación ambiental. 2012. Disponible en: <http://www.cdmb.gov.co/web/index.php/descargas/doc_download/688-informe-anual-calidad-del-aire-2011.html>

[22] *Resolución 689*. Ministerio de Tecnología de la Información y las Comunicaciones. 2004. Disponible en: <http://www.cntv.org.co/cntv_bop/basedoc/resolucion/mincomunicaciones/resolucion_mincomunicaciones_0689_2004.html>

[23] *Resolución 610*. Ministerio de Ambiente, Vivienda y Desarrollo Territorial. 2010. [Consulta 20 de enero de 2012]. Disponible en: <http://www.minambiente.gov.co/documentos/normativa/ambiente/resolucion/res_0610_240310.pdf>

[24] DIGI. *XBee/XBee-PRO ZigBee RF Modules*. Digi International, Inc. 2012. 68 p. [Consultado 5 de febrero de 2012]. Disponible en:
<http://ftp1.digi.com/support/documentation/90000982_F.pdf>

[25] DIGI. *XBee Drop-in Networking Accessories User's Guide*. Digi International, Inc. 2010. 78 p. Disponible en:
<http://ftp1.digi.com/support/documentation/90000891_E.pdf>

[26] *Curso Django para perfeccionistas con deadlines*, 2012. 103 p. Disponible en:
<<http://www.maestrosdelweb.com/guias/#guias-django>>

[27] Alphasense the sensor technology company. Disponible en:
<<http://www.alphasense.com/industrial-sensors/index.html>>

[28] Membrapor. Disponible en:
<<http://www.membrapor.ch/>>

[29] Página oficial de Django. Disponible:
<<https://www.djangoproject.com/>>

[30] Guía para configurar un servidor para Django en Ubuntu. Disponible en:
<<http://www.ventanazul.com/tutoriales/guia-configurar-servidor-django-nginx-apache-ubuntu>>

[31] PARALLAX INC. Store. Sensors. Gas. Disponible en :
<<http://www.parallax.com/Store/Sensors/GasSensors/tabid/843/List/0/CategoryID/91/Level/a/SortField/0/Default.aspx>>

[32] WINSEN Professional gas sensing solution. Disponible en:
<http://www.winsensor.com/en/ProList45_1.html>


[33] NEMOTO SENSOR ENGINEERING CO., LTD. Disponible en:
<<http://www.nemoto.co.jp/en/products/sensor/industrial/index.html>>

[34] FIGARO. World leader in gas sensing innovation. Disponible en:
<<http://www.figaro.co.jp/en/product/index.php?mode=search&kbn=1&type=20&id=212015>>

ANEXOS

ANEXO A. Especificación técnica de los sensores de gas Alphasense

Technical Specification



O3-B4 Ozone Sensor 4-Electrode


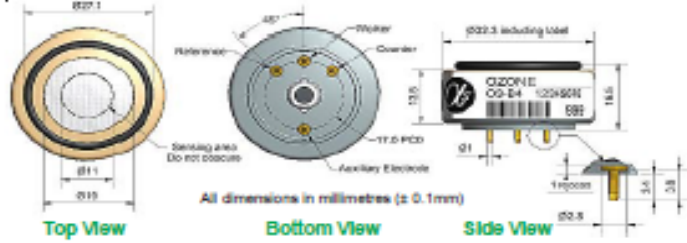


Figure 1 O3-B4 Schematic Diagram PATENTED



All dimensions in millimetres (± 0.1mm)

PERFORMANCE		
Sensitivity	nA/ppm in 100ppb O ₃	-600 to -1000
Response time	t ₉₀ (s) from zero to 100ppb	< 30
Zero current	nA in zero air at 20°C	0 to 220
Noise*	RMS noise (ppb equivalent)	< 3
Limit of detection*	ppb equivalent	< 5
Range	ppm O ₃ limit of performance warranty	< 2
Linearity	ppm error at full scale, linear at zero and 1ppm O ₃	< 5%
Overgas limit	maximum ppm for stable response to gas pulse	nd
* Requires a low noise potentiostat circuit for lowest noise and best resolution		
LIFETIME		
Zero drift	ppb equivalent change/year in lab air	nd
Sensitivity drift	% change/year in lab air, monthly test	< 10
Operating life	months until 60% original signal (12 month warranted)	> 24
ENVIRONMENTAL		
Sensitivity @ -20°C	% (output @ -20°C/output @ 20°C) @ 500ppb O ₃	nd
Sensitivity @ 50°C	% (output @ 50°C/output @ 20°C) @ 500ppb O ₃	nd
Zero @ -20°C	ppm equivalent change from 20°C	< ±0.05
Zero @ 50°C	ppm equivalent change from 20°C	< ±0.2
CROSS SENSITIVITY		
H ₂ S sensitivity	% measured gas @ 5ppm H ₂ S	< -250
NO ₂ sensitivity	% measured gas @ 0.35ppm NO ₂	45 to 60
Cl ₂ sensitivity	% measured gas @ 10ppm Cl ₂	< 50
NO sensitivity	% measured gas @ 1ppm NO	< -2
SO ₂ sensitivity	% measured gas @ 0.2ppm SO ₂	< -3
CO sensitivity	% measured gas @ 10 ppm CO	< 0.1
H ₂ sensitivity	% measured gas @ 400ppm H ₂	< 1
C ₂ H ₄ sensitivity	% measured gas @ 400ppm C ₂ H ₄	< 0.1
NH ₃ sensitivity	% measured gas @ 20ppm NH ₃	< 1
CO ₂ sensitivity	% measured gas @ 5% CO ₂	< 0.1
KEY SPECIFICATIONS		
Temperature range	°C	-20 to +50
Pressure range	kPa	60 to 120
Humidity range	% rh non-condensing	15 to 85
Flow rate	minimum sccm during calibration	500 (0.5L/m)
Bias voltage	V	0
Storage period	months @ 3 to 20°C (stored in sealed pot)	6
Load resistor	Ω (recommended)	33 to 100
Weight	g	< 13

NOTE: all sensors are tested at ambient environmental conditions, with 10 ohm load resistor, unless otherwise stated. As applications of use are outside our control, the information provided is given without legal responsibility. Customers should test under their own conditions, to ensure that the sensors are suitable for their own requirements.

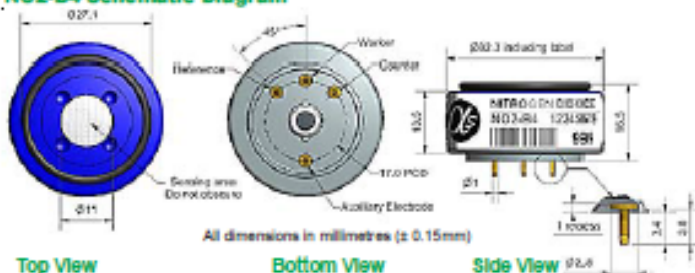
Alphasense Ltd, Sensor Technology House, 306 Avenue West, Skyline 126, Great Notley, CM77 7AA, UK
 Telephone: +44 (0) 1376 888 700 Fax: +44 (0) 1376 335 899 E-mail: sensors@alphasense.com Website: www.alphasense.com



NO2-B4 Nitrogen Dioxide Sensor 4-Electrode



Figure 1 NO2-B4 Schematic Diagram



Technical Specification

PERFORMANCE

Sensitivity	nA/ppm in 1ppm NO ₂	-350 to -750
Response time	t ₉₀ (s) from zero to 1ppm NO ₂	< 35
Zero current	nA in zero air at 20°C	0 to 50
Noise*	RMS noise (ppb equivalent)	< 5
Limit of detection*	ppb	< 5
Range	ppm NO ₂ , limit of performance warranty	20
Linearity	ppm error at full scale, linear at zero and 5ppm NO ₂	< ±1
Overgas limit	maximum ppm for stable response to gas pulse	50

* Requires a low noise potentiostat circuit for lowest noise and best resolution

LIFETIME

Zero drift	ppb equivalent change/year in lab air	< 20
Sensitivity drift	% change/year in lab air, monthly test	< 50
Operating life	months until 80% original signal (12 month warranted)	> 8

ENVIRONMENTAL

Sensitivity @ -20°C	% (output @ -20°C/output @ 20°C) @ 2ppm NO ₂	35 to 60
Sensitivity @ 50°C	% (output @ 50°C/output @ 20°C) @ 2ppm NO ₂	120 to 140
Zero @ -20°C	ppm equivalent change from 20°C	< 0 to 0.05
Zero @ 50°C	ppm equivalent change from 20°C	< 0 to 0.01

CROSS

H ₂ S	sensitivity % measured gas @ 20ppm	H ₂ S	< -250
NO	sensitivity % measured gas @ 10ppm	NO	< 0.5
Cl ₂	sensitivity % measured gas @ 10ppm	Cl ₂	< 100
SO ₂	sensitivity % measured gas @ 20ppm	SO ₂	< -2
CO	sensitivity % measured gas @ 10ppm	CO	< 0.1
C ₂ H ₄	sensitivity % measured gas @ 400ppm	C ₂ H ₄	< 0.1
NH ₃	sensitivity % measured gas @ 20ppm	NH ₃	< 0.1
CO ₂	sensitivity % measured gas @ 5% Vol	CO ₂	< 0.1
O ₃	sensitivity % measured gas @ 100ppb	O ₃	< 70
Halothane	sensitivity % measured gas @ 100ppm	Halothane	< 0.1

KEY SPECIFICATIONS

Temperature range	°C	-30 to 50
Pressure range	kPa	80 to 120
Humidity range	% rh continuous	15 to 85
Storage period	months @ 3 to 20°C (stored in sealed pot)	6
Load resistor	Ω (recommended)	33 to 100
Weight	g	< 13



At the end of the product's life, do not dispose of any electronic sensor, component or instrument in the domestic waste, but contact the instrument manufacturer, AlphaSense or its distributor for disposal instructions.

NOTE: all sensors are tested at ambient environmental conditions, with 47 ohm load resistor, unless otherwise stated. As applications of use are outside our control, the information provided is given without legal responsibility. Customers should test under their own conditions, to ensure that the sensors are suitable for their own requirements.

AlphaSense Ltd, Sensor Technology House, 300 Avenue West, Skyline 120, Great Northway, CM27 7AA, UK
Telephone: +44 (0) 1378 558 700 Fax: +44 (0) 1378 335 898 E-mail: sensors@alphasense.com Website: www.alphasense.com