DESARROLLO DE UN PROTOTIPO DE APLICACIÓN WEB EN COMBINACIÓN CON LA PLATAFORMA ARDUINO PARA CONTROLAR LA CALIDAD DE AIRE DE LA CIUDAD DE QUITO.

Diego Carrera[1], Walter Fuertes[2], César Villacís[3], Theofilos Toulkeridis[4]

1 Universidad de las Fuerzas Armadas ESPE, Ecuador, diegomcarrera@gmail.com

2 Universidad de las Fuerzas Armadas ESPE, Ecuador, wmfuertes@espe.edu.ec

3 Universidad de las Fuerzas Armadas ESPE, Ecuador, cjvillacis@espe.edu.ec

4. Universidad de las Fuerzas Armadas ESPE, Ecuador, theousfq@yahoo.com

RESUMEN

Este proyecto presenta el proceso de conceptualización, análisis y desarrollo, de un prototipo tecnológico de bajo costo, compuesto por una aplicación Web, que en combinación con la plataforma Arduino permite medir parámetros referenciales de calidad de aire en la ciudad de Quito. Se utilizó una combinación entre las metodologías Scrum y Extreme Programming (XP) con el objetivo de abordar todas las tareas y requerimientos necesarios de una manera eficiente y eficaz. El dispositivo electrónico está equipado con tres sensores y con la capacidad de conectarse al Internet para la transmisión, en tiempo real, de la información recolectada utilizando un servicio Web de tipo RESTful. La prueba de concepto se la aplicó en la ciudad de Quito con resultados positivos, tanto en el funcionamiento del software y del hardware que componen el prototipo, como en la medición referencial de la concentración de los contaminantes.

Palabras Clave: Calidad del Aire, Arduino, JEE, EJB , RESTful, Sensores

ABSTRACT

This technical paper provides and insight of the process of conceptualization, analysis and development of an affordable technology prototype, which consists of a Web application in combination with the Arduino platform which measures air quality parameters in the city of Quito. A combination between Scrum and Extreme Programming (XP) methodologies was used in order to address all necessary tasks and requirements efficiently and effectively to develop the prototype. The electronic device is equipped with three air quality sensors and the ability to connect to the Internet to transmit in real-time the information collected through a RESTful Web Service. The proof of concept of the solution was applied in the city of Quito with positive results both, in the operation of software and hardware in the prototype, and in the referential measurement of the concentration of pollutants

KeyWords: Air Quality, Arduino, JEE, EJB, RESTful, Sensors

1. INTRODUCCIÓN

La contaminación atmosférica y la falta de puntos de monitoreo de calidad del aire es uno de los principales retos ambientales y tecnológicos de las ciudades del mundo y del Ecuador. Actualmente, solo Quito, Guayaquil y Cuenca monitorean la calidad del aire de sus urbes mientras otras ciudades apenas están en proceso de desarrollar programas de medición

Quito posee una Red de Monitoreo Atmosférica convencional que mide calidad del aire únicamente en 9 puntos de la ciudad. La información captada por este sistema se procesa y publica a través del portal Web de la Secretaría de Ambiente del Municipio de Quito, pero no es lo suficientemente amplia ni representativa de la calidad del aire de la ciudad (Secretaría de Ambiente Quito, 2014). Por esto, es importante encontrar una alternativa tecnológica versátil que permita mejorar el proceso de medición de la calidad del aire en la ciudad de Quito y brindar valores referenciales en sitios que las redes de monitoreo convencionales no llegan a cubrir.

Es por esto que este proyecto propone una solución tecnológica de bajo costo capaz de medir la concentración de tres contaminantes atmosféricos, transmitir la información recolectada por medio de Internet en tiempo real, almacenarlos en un sistema de base de datos y presentarlos en una aplicación Web de manera amigable y moderna. Esto implica la construcción de hardware y software capaz de interactuar entre sí. Existen trabajos similares que se enfocan en uno de los dos elementos que componen este proyecto.

El presente artículo se ha organizado de la siguiente manera: en la Sección Metodología se describe las herramientas de Ingeniería de Software utilizadas a lo largo del ciclo de vida del proyecto; en Hardware y Software se hace mención a los recursos tecnológicos utilizados para alcanzar los objetivos planteados para este proyecto; en Diseño e Implementación se detalla el diseño de la aplicación, servicio Web y la Arquitectura de Software empleada; en Resultados se presenta el análisis aplicado a las mediciones de los valores referenciales sobre la concentración de contaminantes en la ciudad de Quito.

2. METODOLOGÍA

2.1 Scrum

Scrum se define como un marco referencial de trabajo que propone un conjunto de prácticas, principios y valores para organizar y gestionar las tareas en el desarrollo de productos o servicios. Consta de dos etapas en su proceso de organización del trabajo; la primera denominada *Sprint 0* o etapa inicial; y la segunda de iteraciones sucesivas, también llamadas *Sprints* (Álvarez García, 2012).

La etapa inicial o Sprint 0, de duración variable pero no indefinida, se encarga de preparar toda la logística, mecánica y metodología a seguir a lo largo del proyecto. También se realiza la primera versión del listado de requerimientos generales del software, listado que se convertirá en el Product Backlog, mismo que es un insumo del proyecto y será refinado en la etapa de iteraciones sucesivas.

La segunda etapa de iteraciones sucesivas o Sprints, se divide en cuatro actividades las cuales se componen de una secuencia de reuniones que buscan garantizar el cumplimiento de los compromisos del Equipo Scrum. Estas cuatro actividades ejecutadas en cada Sprint son: Planificación, Scrum diario, revisión, retrospectiva.

Con el objetivo de mantener el control y hacer el seguimiento del trabajo, que el equipo de desarrollo va a realizar, Scrum define una serie de artefactos y herramientas: Product Backlog, Sprint Backlog, Product Increment o Demo

2.2 Extreme Programming (XP)

Kent Beck (1999) define a Extreme Programming (XP) como un método ágil para el desarrollo de software muy útil si tenemos que abordar proyectos con requerimientos cambiantes o vagamente definidos. XP se basa en un conjunto de valores que son los que guían el desarrollo, entre los que destacan: la comunicación, la simplicidad, la retroalimentación, el coraje y el respeto.

El proceso XP está compuesto por cuatro actividades estructurales: planeación, diseño, codificación y pruebas (Pressman, 2010). Al igual que en la metodología Scrum, Extreme Programming (XP) cuenta también con una serie de actividades que son: planeación, diseño, codificación, pruebas

Las prácticas determinan la manera en la que el equipo de desarrollo trabaja día a día. La adopción de una u otra práctica puede variar en función del contexto en el que se esté trabajando (Álvarez García, 2012). Las practicas utilizadas en el proyecto son: diseño incremental, integración continua, pruebas antes de programar.

2.3 Especificación de Requerimientos

Para especificar los requerimientos, se utilizó el estándar IEEE830-1998, propuesto por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), con el objetivo de definir con claridad el ámbito del software y hardware que compone el proyecto. Es importante tomar en cuenta que el análisis de los requerimientos constituye una etapa importante en el ciclo del desarrollo del software.

3. HARDWARE Y SOFTWARE

Para la ejecución del proyecto se emplearon herramientas de hardware y software de libre distribución y código abierto. Estas herramientas se detallan a continuación:

3.1 Java Enterprise Edition (JEE)

La plataforma Java Enterprise Edition o Java EE, por sus siglas en Inglés, es un conjunto de componentes conocidas como API's, cuyo objetivo principal es simplificar el desarrollo de software empresarial y fue diseñada para soportar aplicaciones que implementan servicios empresariales para consumidores, empleados, proveedores y socios estratégicos de la empresa. Este tipo de aplicaciones generalmente son complejas, acceden a varias fuentes de información y son consumidas por varios tipos clientes (Oracle, 2014).

La plataforma Java EE implementa un modelo distribuido multicapa para las aplicaciones empresariales, lo que significa que los elementos que componen el software están divididos de acuerdo a una función y que suelen estar instalados en diferentes computadores.

3.2 Plataforma Arduino

Arduino es un plataforma abierta de computación física compuesta por un microcontrolador montado en una placa electrónica, que puede ser usada para desarrollar objetos interactivos capaces de recolectar datos por medio de sensores y controlar luces, motores u otros elementos físicos (Banzi, 2011). La plataforma Arduino está formado por dos elementos fundamentales: (1) la placa electrónica Arduino que vendría a ser el hardware y es con lo que se construye los objetos interactivos, (2) el software o Ambiente de Desarrollo Integrado (Arduino IDE) que se usa para programar el microcontrolador que está montado en la placa electrónica.

3.2.1 Sensores

Massimo Banzi (2011), define a los sensores como elementos electrónicos capaces de interactuar con el ambiente que los rodea. Así como los ojos convierten la luz en una señal que el cerebro transforma en imágenes, los sensores transforman los eventos físicos en señales eléctricas que pueden ser interpretadas por el microcontrolador de la placa Arduino. Los sensores utilizados en este proyecto son: sensor MG-811 para medir la concentración de Dióxido de Carbono (CO2), sensor óptico Sharp GP2Y1010AU0F para medir la densidad de polvo, sensor MQ-7 para medir la concentración de Monóxido de Carbono (CO).

4. DISEÑO E IMPLEMENTACIÓN

4.1 Modelo de Datos

Un modelo de datos es un conjunto de herramientas que describen la estructura de una base de datos, las

relaciones, las restricciones y en ocasiones incluye semántica o significado de los datos. (Hansen & Hansen, 1997). La Figura 1 ilustra el diseño físico de la base de datos utilizado en e el proyecto para almacenar la información recolectada por los sensores electrónicos.

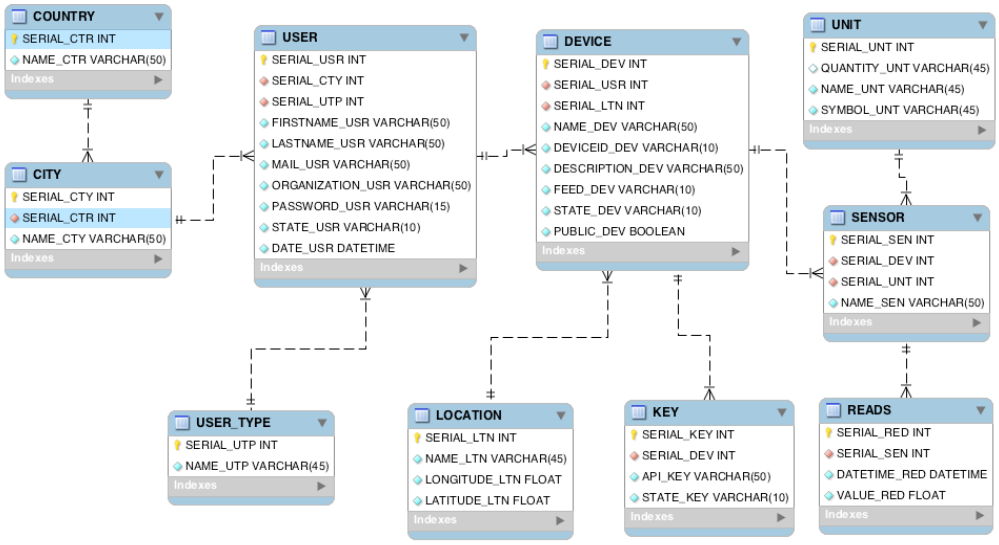

Figura 1 Diseño de base de datos

4.2 Diagramas UML

Para el modelado de la aplicación Web y el firmware utilizado por el dispositivo electrónico basado en la plataforma Arduino, se utilizaron Diagramas de Casos de Uso, Diagramas de Secuencia ordenados por escenarios, Diagramas de Clases para cada una de las capas y un Diagrama de Paquetes. La Figura 2 ilustra el diagrama de Casos de Uso que presenta al ámbito del prototipo.

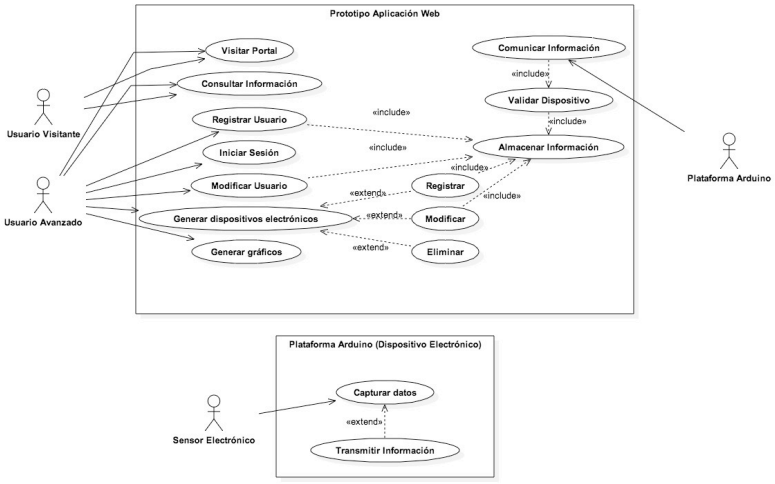

Figura 2 Diagrama general de Casos de Uso

La Figura 7 muestra el diagrama de la arquitectura del software donde se detallan las distintas capas que componen la aplicación, el dispositivo electrónico, el servicio Web RESTful y el sistema de base de datos donde se almacenarán los valores captados por los sensores.

4.3 Pruebas

Las pruebas unitarias aplicadas al software de la aplicación, constituyen una de las prácticas fundamentales que propone la metodología Extreme Programming (XP). La Figura 4 muestra el resultado de las pruebas apli-

cadas a un conjunto de clases del software, utilizando la herramienta JUnit y Arquillian. Adicionalmente, se ha seguido el tracking que presenta el servidor de aplicaciones para comprobar la transmisión de datos desde el dispositivo electrónico por medio del servicio Web. La Figura 5 muestra el dispositivo electrónico construido.

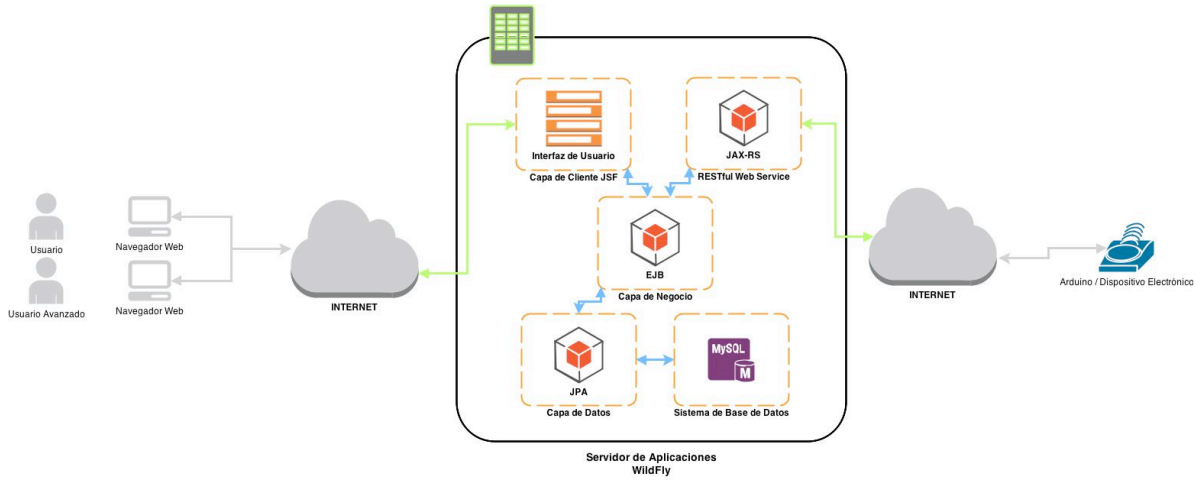

Figura 3 Diagrama de Arquitectura.

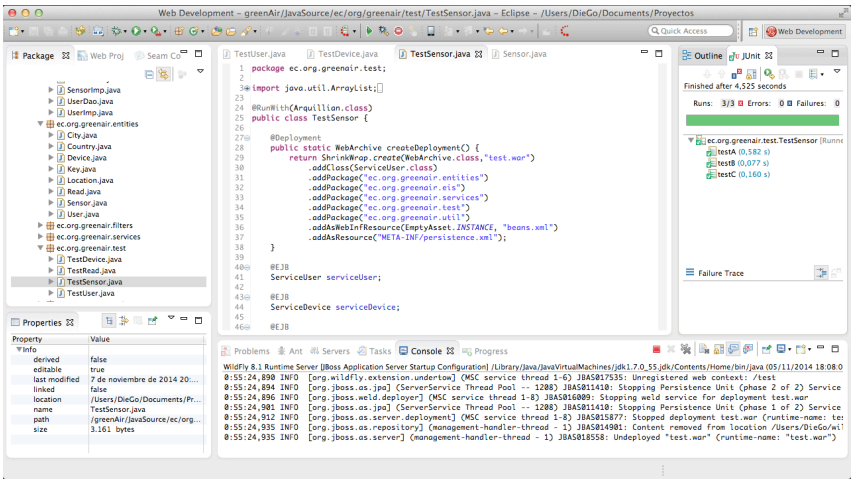

Figura 4 Pruebas Unitarias

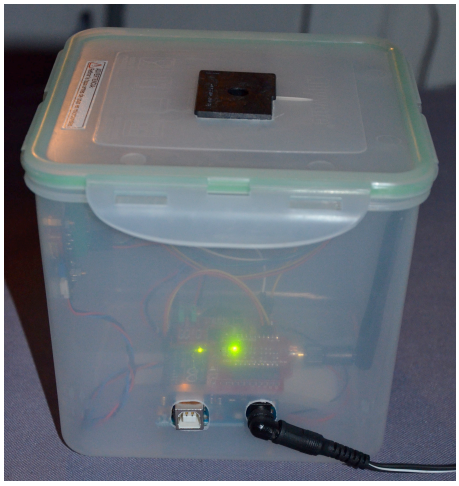

Figura 5 Dispositivo Electrónico

5. RESULTADOS

En la elaboración de la prueba de concepto, se obtuvieron muestras representativas de la concentración de los tres contaminantes que los sensores del dispositivo electrónico son capaces de medir. Las mediciones se realizaron en la Parroquia Jipijapa de la ciudad de Quito; y en las cuevas de investigación científica en la ciudad del Tena.

5.1 Parámetros de la ciudad de Quito

La Figura 4 muestra el resultado del análisis estadístico de la datos obtenidos de la medición de la concentración de Monóxido de Carbono (CO) en la ciudad de Quito. El conjunto de datos obtenidos suman 169 lecturas cuyo valor mínimo de concentración es 12 ppm y el máximo de 18 ppm. Se puede observar una mayor concentración de lecturas entre los valores de 13 ppm y 14 ppm, lo que se ratifica con una media para esta muestra de 13,68 ppm en una hora de medición. Este promedio está por debajo de los valores permisibles en períodos de exposición de una hora según la USAEPA.

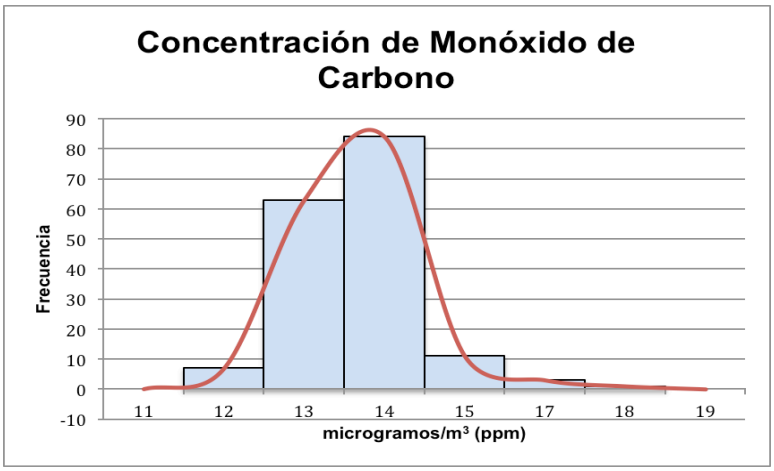

Figura 6 Concentración de Monóxido de Carbono Quito

El recuento de las lecturas, del sensor de Densidad de Polvo, hacienden a 169 datos cuyo valor mínimo es de 0,00 mg/m^3 y su máximo es de 0,23 mg/m^3. Se evidencia una mayor concentración de los datos entre los valores de 0,06 mg/m^3 y 0,08 mg/m^3, obteniéndose de esta muestra representativa una media de 0,075 mg/m^3, promedio que se encuentra por debajo del límite permisible para esta variable según el estándar ASHRAE-62. La Figura 5 ilustra el análisis estadístico aplicado.

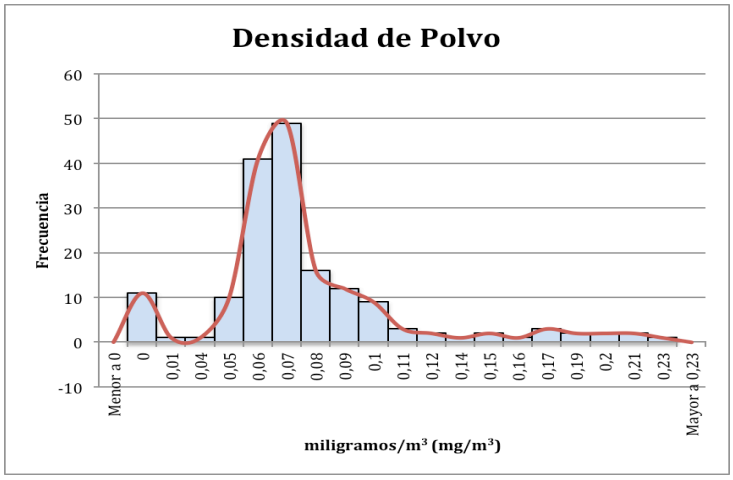

Figura 7 Densidad de Polvo Quito

5.2 Parámetros de las cuevas de investigación científica en el Tena

Las mediciones en las cuevas de investigación ubicadas en las cercanías de la ciudad del Tena se las realizó bajo la dirección del Dr. Theofilos Toulkeridis. Se utilizaron los sensores de Dióxido de Carbono (CO2) y Densidad de Polvo para este ejercicio.

La Figura 6 ilustra el análisis estadístico aplicado a la primera cueva visitada, la cual presenta una media de 0,035 mg/m^3 en la variable de Densidad de Polvo de un recuento total de 269 lecturas efectuadas, que van desde 0,0 mg/m^3 hasta 0,10 mg/m^3, donde las mayor concentración se evidencia entre los 0,02 mg/m^3 y 0,05 mg/m^3.

La Figura 7 muestra el análisis estadístico aplicado para la variable de Dióxido de Carbono (CO2) para los datos recolectados de la cueva Castillo, con una media de 4.424,24 partes por millón (ppm) en una muestra de 154 lecturas. Se distingue que la mayor concentración de los datos están entre los 2.200,50 ppm y los 6.939,50 ppm.

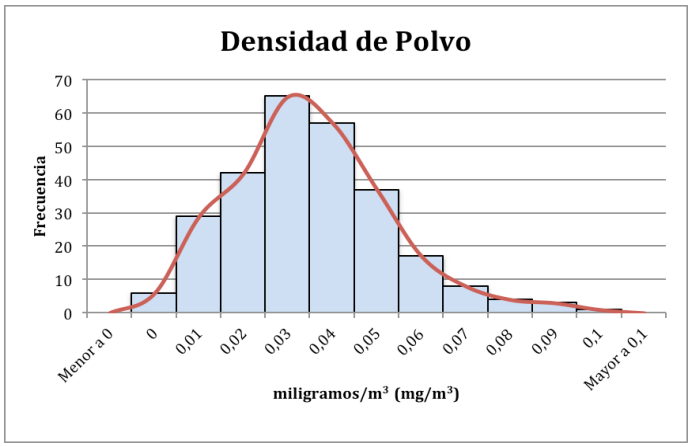

Figura 8 Densidad de Polvo – Cueva Castillo

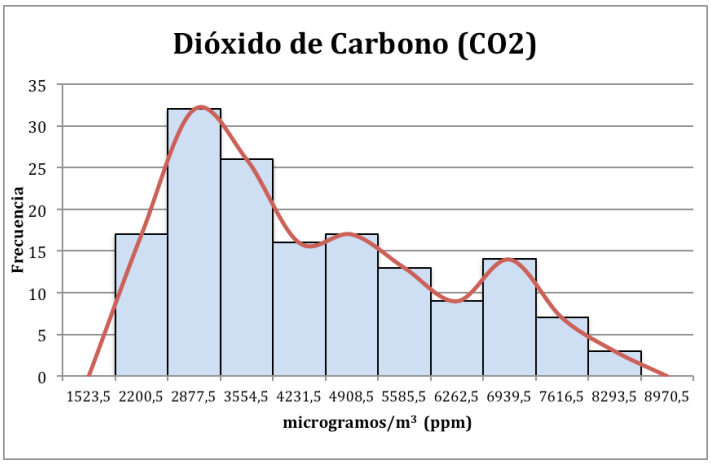

Figura 9 Dióxido de Carbono – Cueva Castillo

En la segunda cueva visitada se consiguieron 313 lecturas de la variable de Densidad de Polvo, con una media de 0,03 miligramos por metro cúbico (mg/m^3), ubicando la mayor cantidad de datos entre los 0,02 mg/m^3 y los 0,05 mg/m^3 . La Figura 8 ilustra los resultados del análisis estadístico aplicado a esta variable.

La concentración de Dióxido de Carbono, ilustrada en la Figura 9, tiene una media de 4.879,81 partes por millón (ppm) en un conjunto de datos de 318 lecturas, cuyo valor mínimo es de 1790,63 ppm y su máximo de 9.033, ppm. La mayor concentración de las lecturas se encuentran entre los 2.877,50 ppm y los 7227,50 ppm.

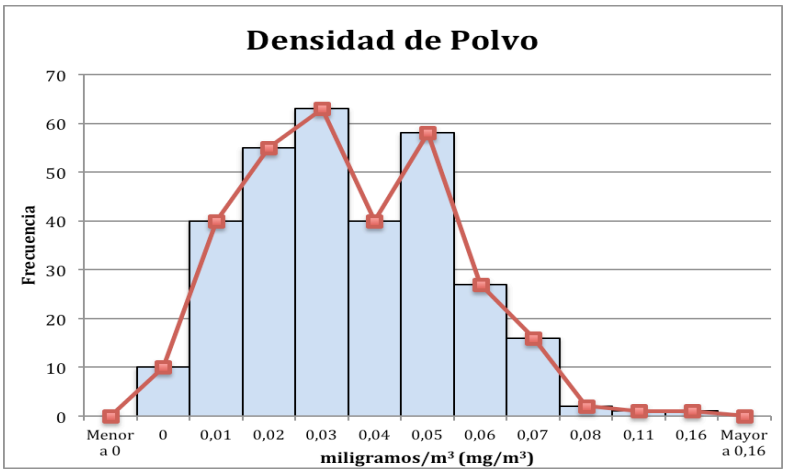

Figura 10 Densidad de Polvo – Gruta de la Virgen

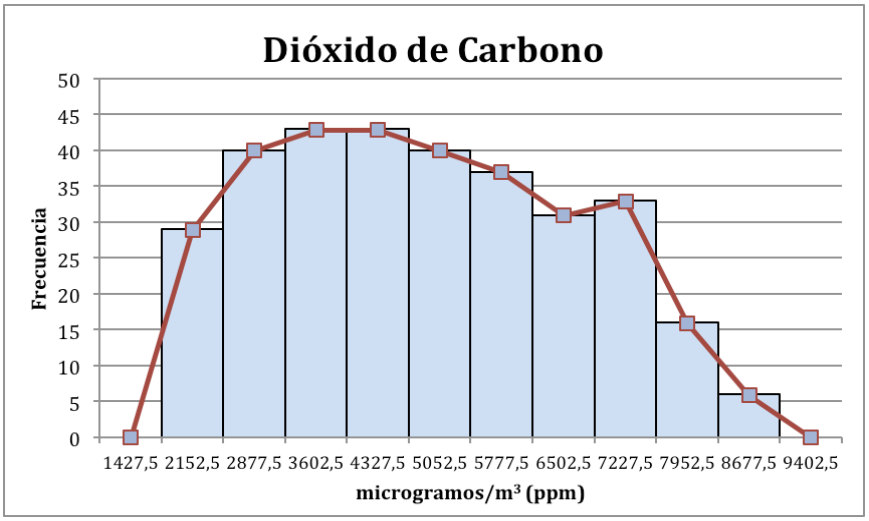

Figura 11 Concentración de Dióxido de Carbono – Gruta de la Virgen

6. CONCLUSIONES Y TRABAJO FUTURO

6.1 Conclusiones

Para llevar a cabo el presente proyecto de investigación se empleó la combinación entre las metodologías Scrum y Extreme Programming (X) a lo largo del ciclo de vida del proyecto, obteniendo un producto que cubre todas las necesidades planteadas y dentro de los plazos previstos. El resultado del análisis estadístico de las mediciones obtenidas, utilizando el prototipo construido, en la ciudad de Quito, en el sector de la Jipijapa, nos indican que la calidad del aire se encuentra bajo los límites permisibles establecidos por las normas internacionales publicadas por US-EPA. En el caso de las cuevas de investigación científica, encontramos que la concentración de CO_2 supera el valor referencial de la media atmosférica de 400 partes por millón. La Densidad de Polvo, sin embargo es menor, a la media que presentó la ciudad de Quito. El uso de soluciones tecnológicas capaces de medir, transmitir en tiempo real y procesar la información recolectada; haciendo uso de herramientas como el Internet y sistemas de información modernos, brindan un nuevo enfoque en el trabajo de control de la calidad de aire en las ciudades su difusión. Este prototipo está diseñado principalmente para que, los responsables de la gestión de la calidad del aire en la ciudades, tomen mejores decisiones y puedan optimizar los recursos disponibles, es importante recalcar que los usuarios finales de la información recogida por el sistema son los ciudadanos.

6.2 Trabajo Futuro

Un futuro crecimiento del prototipo incluiría dos iniciativas. La primera, efectuar las mediciones en un período de tiempo mayor y en más puntos de referencia de la ciudad. La segunda se enfoca en la elección de sensores más precisos que puedan medir otros parámetros de la calidad del aire y que nos permitan comparar los resultados con los datos de medición especializada.

7. REFERENCIAS BIBLIOGRÁFICAS

1. MAE, 2. (2014). Programa Calidad del Aire Ecuador Fase III. Obtenido de http://www.ambiente.gob.ec/programa-calidad-del-aire-fase-iii/
2. Secretaría de Ambiente Quito. (2014). *Red Metropolitana Ambiental*. Recuperado de Red Metropolitana Ambiental: http://190.152.144.74/paginas/web/menu_t1.html
3. Álvarez García. (2012). *Métodos Ágiles y Scrum.* Madrid: Anaya Multimedia.
4. Pressman, R. (2010). *Ingeniería de Software.* México: McGraw-Hill.
5. Chromatic. (2003). *Extreme Programming Pocket Guide.*
6. Banzi, M. (2011). *Getting Started with Arduino.* Sebastopol: O'Reilly
7. Arduino. (2014). Arduino Website. Obtenido de http://www.arduino.cc
8. ArduinoXbeeShield. (2014). http://arduino.cc/en/Main/ArduinoXbeeShield
9. DFRobot. (2014). DFRobot Drive the Future. Obtenido de http://www.dfrobot.com/wiki/index.php/CO2_Sensor_SKU:SEN0159: www.dfrobot.com
10. Sparkfun-Sharp. (2006). https://www.sparkfun.com/datasheets/Sensors/gp2y1010au_e.pdf.
11. Gary W. Hansen, J. V. (1997). *Diseño y Administración de Base de Datos.* México: Prentice-Hall.
12. Kimel, P. (2006). Manual de UML. México: Mc Graw Hill.
13. US-EPA. (20 de 11 de 2014). Environmental Protection Agency. Obtenido de Concentración Global de CO2 en la atmósfera en el tiempo.: http://www.epa.gov/climatechange/images/indicator_downloads/ghg-concentrations-download1-2014.png