



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TRABAJO DE TITULACIÓN, PREVIO LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: DESARROLLO DE UNA APLICACIÓN IoT PARA EL
MONITOREO Y PREDICCIÓN DEL TRÁFICO EN LA
AVENIDA SIMÓN BOLÍVAR DE QUITO, UTILIZANDO UNA
RED DE SENSORES BASADOS EN LA PLATAFORMA
ARDUINO.**

**AUTORES: ABAD SOJOS, MARIA ALEJANDRA
NUGRA MADERO, HUGO ERNESTO**

DIRECTOR: ING. FUERTES, WALTER PhD.

SANGOLQUÍ,

2016



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, “DESARROLLO DE UNA APLICACIÓN IoT PARA EL MONITOREO Y PREDICCIÓN DEL TRÁFICO EN LA AVENIDA SIMÓN BOLÍVAR DE QUITO, UTILIZANDO UNA RED DE SENSORES BASADOS EN LA PLATAFORMA ARDUINO” realizado por el Sr. Hugo Ernesto Nugra Madero y la Srta. María Alejandra Abad Sojos, ha sido guiada y revisada periódicamente y cumple las normas estatutarias establecidas por la ESPE, en el Reglamento Estudiantes de la Universidad de las Fuerzas Armadas-ESPE, por lo tanto me permito acreditarlo y autorizar al Sr. Hugo Ernesto Nugra Madero y la Srta. María Alejandra Abad Sojos para que lo sustenten públicamente.

Este trabajo tiene un excelente contenido científico, que coadyuva a la aplicación de conocimientos y su desarrollo profesional.

Sangolquí, Junio de 2016

Ing. Walter Fuertes PhD

DIRECTOR DE PROYECTO DE GRADO



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA
AUTORÍA DE RESPONSABILIDAD

Nosotros, María Alejandra Abad Sojos con CI:1721645255 y Hugo Ernesto Nugra Madero con CI: 1400739486 declaro que este trabajo de titulación “DESARROLLO DE UNA APLICACIÓN IoT PARA EL MONITOREO Y PREDICCIÓN DEL TRÁFICO EN LA AVENIDA SIMÓN BOLÍVAR DE QUITO, UTILIZANDO UNA RED DE SENSORES BASADOS EN LA PLATAFORMA ARDUINO”, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos declaramos responsables del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, Julio de 2016.

María Alejandra Abad Sojos

CC: 1721645255

Hugo Ernesto Nugra Madero

CC:1400739486



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA
AUTORIZACIÓN

Nosotros, María Alejandra Abad Sojos y Hugo Ernesto Nugra Madero, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “DESARROLLO DE UNA APLICACIÓN IoT PARA EL MONITOREO Y PREDICCIÓN DEL TRÁFICO EN LA AVENIDA SIMÓN BOLÍVAR DE QUITO, UTILIZANDO UNA RED DE SENSORES BASADOS EN LA PLATAFORMA ARDUINO”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría

Sangolquí, Julio de 2016.

María Alejandra Abad Sojos

CC: 1721645255

Hugo Ernesto Nugra Madero

CC: 1400739486

DEDICATORIA

El presente proyecto está dedicado a todas las personas que nos apoyaron y guiaron a lo largo de carrera de Ingeniería en Sistemas e Informática.

A mi padre, Iván Patricio Abad Suarez que desde el cielo me ha cuidado, guiado y me enseñó a entender el porqué de las cosas y buscar soluciones a los problemas, sin olvidar sonreír antes las adversidades. Lo hicimos Papi.

A mi madre, María Fernanda Sojos Dávalos que supo guiar mi camino, ser ejemplo a seguir con sus valores, fortaleza y amor, superando las adversidades y fortaleciendo la unión Familiar.

A mis Hermanas, Andrea Abad y Sofía Abad que me han brindado su amor, paciencia, consejo y apoyo enseñándome a ser valiente y salir adelante juntas.

A mi Gran Familia, Bis Abuelos, Abuelos, Tías/os, Primos, Tías/os-Abuelos, tías/os políticos y todos aquellos que han estado siempre pendientes, han sido mi guía y soporte a lo largo del camino, enseñándome valores, la importancia de la familia unida y la fortaleza para salir adelante pese a las adversidades.

Alejandra Abad

Olmedo Lara decía "El éxito de uno, es el éxito para toda la familia", palabras de mi bis abuelo que las comparto y a las cuales me adhiero y dedico completamente este objetivo cumplido a mi familia y de manera especial a mis padres Anita y Hugo, que han sido mi ejemplo en cada momento de mi vida y siempre faltarán palabras para decirles cuanto los amo y los respeto.

Hugo Nugra

AGRADECIMIENTO

Mediante el presente expreso mi sincero agradecimiento a todas las personas que han sido parte del desarrollo de este documento. A mis padres que me han brindado todo el apoyo en cada aspecto y en todo momento. A mi hermano por su paciencia y cariño. A Madelyn por su interés, cariño y aporte en diseño. De igual manera a toda mi familia que siempre se ha encontrado interesada en mi avance durante cada etapa de mi vida.

Agradezco a nuestro director de proyecto, Dr. Walter Fuertes y mi compañera Alejandra Abad. De manera adicional extendiendo mi gratitud hacia Francisca Sosa y Juan Pombosa por el apoyo brindado.

Hugo Nugra

Agradezco a mi Familia por su apoyo y dedicación sin ellos no hubiera sido posible. Al Ing. Marco Flores por cuidarme, escucharme, aconsejarme y estar siempre presente a lo largo de este camino.

Al equipo de Cheerleader de la ESPE que me dio la oportunidad de representar a esta hermosa universidad y ser parte por varios años de este hermoso deporte.

A la UBE y residencia de la ESPE por brindarme la oportunidad de conocer tantas personas maravillosas y compartir tantos momentos hermosos.

A mi compañero de tesis Hugo Nugra por todos estos años de amistad, dedicación, esfuerzo y aventuras que vivimos en juntos en la Universidad.

Alejandra Abad

Índice

CERTIFICACIÓN	ii
AUTORÍA DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
Índice.....	vii
Índice de Cuados	xi
Índice de Figuras	xii
RESUMEN.....	xiv
ABSTRACT	xv
CAPÍTULO I.....	1
INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Problemática.....	2
1.3 Análisis del estado del arte	2
1.4 Justificación.....	4
1.5 Objetivos	4
1.6 Alcance	5
CAPÍTULO II	7
MARCO TEÓRICO.....	7
2.1 Cloud Computing	7
2.2 Arduino.....	9
2.3 WSN.....	11
2.4 Minería de datos	12

2.5	Laravel.....	13
2.6	Bootstrap	15
2.7	SCRUM.....	16
CAPÍTULO III.....		18
ESPECIFICACIÓN DE REQUERIMIENTOS		18
3.1	Propósito.....	18
3.2	Alcance	18
3.3	Limitaciones del Prototipo	20
3.4	Personal Involucrado	20
3.5	Definiciones.....	21
3.6	Acrónimos	21
3.7	Visión General del Documento	22
3.8	Descripción General	22
3.8.1	Perspectiva del Producto	22
3.8.2	Funciones del Producto	23
3.8.3	Características de los Usuarios.....	25
3.8.4	Restricciones	26
3.8.5	Suposiciones y Dependencias	26
3.9	Requisitos Específicos.....	27
3.9.1	Interfaces Externas	27
3.9.2	Funciones	29
3.9.3	Requisitos No Funcionales.....	30
CAPÍTULO IV.....		32
DISEÑO DEL PROTOTIPO		32
4.1	Diseño de la base de datos.....	32
4.2	Diagramas de casos de uso.....	33

4.3	Diagramas de secuencia	38
4.4	Diagrama de clases	42
4.5	Diagrama de paquetes	44
4.6	Diagrama de arquitectura.	44
4.7	Dispositivo electrónico.....	45
4.8	Minería de datos e inteligencia de negocios.....	47
CAPÍTULO V		48
PLANIFICACIÓN, DESARROLLO Y PRUEBAS.....		48
5.1	Planificación del proyecto utilizando la metodología Scrum.....	48
5.2	Estándares de programación.....	50
5.3	Desarrollo y pruebas de la Iteración 1	51
5.3.1	Sprint Backlog de la iteración 1	52
5.3.2	Revisión y seguimiento de la iteración 1	53
5.3.3	Pruebas de la iteración 1	53
5.3.4	Demo Iteración 1	53
5.4	Desarrollo y pruebas de la Iteración 2	54
5.4.1	Sprint Backlog iteración 2.....	55
5.4.2	Revisión y seguimiento de la iteración 2	57
5.4.3	Pruebas de la iteración 2	57
5.4.4	Demo iteración 2	59
5.5	Desarrollo y pruebas de la Iteración 3	60
5.5.1	Sprint Backlog iteración 3.....	61
5.5.2	Revisión y seguimiento iteración 3	62
5.5.3	Pruebas de la iteración 3	62
5.5.4	Demo iteración 3	63
5.6	Desarrollo y evaluación de la Iteración 4.....	65

5.6.1	Sprint Backlog iteración 4.....	65
5.6.2	Revisión y seguimiento iteración 4.....	66
5.6.3	Pruebas de la iteración 4.....	66
5.6.4	Demo iteración 4.....	67
5.7	Desarrollo y pruebas de la Iteración 5.....	67
5.7.1	Sprint Backlog de la iteración 5.....	67
5.7.2	Revisión y seguimiento de la iteración 5.....	69
5.7.3	Pruebas de la iteración 5.....	69
5.7.4	Demo Iteración 5.....	71
CAPÍTULO VI.....		72
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....		72
6.1	Análisis de resultados con minería de datos e inteligencia de negocios.	72
6.1.1	Análisis con minería de datos.....	75
6.2	Correspondencia de los datos con variables aleatorias.....	76
6.3	Generación del modelo.....	77
CAPÍTULO 7.....		79
CONCLUSIONES Y RECOMENDACIONES.....		79
7.1	Conclusiones.....	79
7.2	Recomendaciones.....	80
7.3	Trabajo futuro.....	80
Referencias.....		82

Índice de Cuados

Cuadro 1 Procesos del Prototipo	19
Cuadro 2 Personal Involucrado	20
Cuadro 3 Funciones generales del prototipo.....	24
Cuadro 4 Características de los usuarios.....	25
Cuadro 5 Descripción de las restricciones del proyecto	26
Cuadro 6 Requisitos Específicos.....	27
Cuadro 7 Requerimientos Funcionales	29
Cuadro 8 Pila de producto o ProductBacklog	49
Cuadro 9. Reglas de codificación.....	51
Cuadro 10. Historias de usuario seleccionadas para la iteración 1	52
Cuadro 11. Sprint Backlog de la iteración 1	52
Cuadro 12. Tareas completadas de la iteración 1	53
Cuadro 13 Historias de usuario seleccionadas para la iteración 2	55
Cuadro 14. Sprint Backlog de la iteración 2	56
Cuadro 15. Tareas completadas de la iteración 2.....	57
Cuadro 16. Historia de usuario seleccionada para la iteración 3	60
Cuadro 17. Sprint Backlog iteración 3	61
Cuadro 18. Tareas completadas de la iteración 3.....	62
Cuadro 19. Historias de usuario seleccionadas para la iteración 4	65
Cuadro 20. Sprint Backlog iteración 4.....	65
Cuadro 21. Tareas completadas de la iteración 4.....	66
Cuadro 22. Historias de usuario seleccionadas para la iteración 5	68
Cuadro 23. Sprint Backlog de la iteración 5	68
Cuadro 24. Tareas completadas de la iteración 5.....	69

Índice de Figuras

Figura 1. Cloud Computing	8
Figura 2. Elementos de la plataforma Arduino	9
Figura 3. Placa Arduino UNO r3	10
Figura 4. Red de sensores inalámbricos	12
Figura 5. Proceso minería de datos	13
Figura 6. Componentes MVC Laravel	14
Figura 7. Bootstrap.....	15
Figura 8. GridSystem	16
Figura 9. Scrum.....	17
Figura 10. Casos de uso	23
Figura 11. Diseño de la base de datos.	32
Figura 12. Casos de uso	34
Figura 13. Caso de uso: Visitar el sitio web.....	34
Figura 14. Caso de uso: Observar estadísticas en tiempo real	35
Figura 15. Caso de uso: Observar el flujo de tráfico.....	35
Figura 16. Caso de uso: Análisis de información	36
Figura 17. Caso de uso: Captura de datos	37
Figura 18. Caso de uso: Transmisión de datos.....	38
Figura 19. Escenario: Visita al sitio y observación de estadísticas y flujo	39
Figura 20. Escenario: Análisis de información	40
Figura 21. Escenario: Captura de datos.....	41
Figura 22. Escenario: Transmisión de datos	42
Figura 23. Diagrama de clases	43
Figura 24. Diagrama de paquetes.....	44
Figura 25. Diagrama de arquitectura.....	45
Figura 26. Diseño del dispositivo.....	46
Figura 27. Modelo en estrella	47
Figura 28. Prototipo del dispositivo de monitoreo.....	54
Figura 29. Diseño de la base de datos.	58
Figura 30. Evidencia de pruebas realizadas sobre la 12 de Octubre	58
Figura 31. Dispositivo de monitoreo pruebas	58

Figura 32. Evidencia de pruebas realizadas avenida 12 de Octubre, lugar donde se posicionó el dispositivo de monitoreo.....	59
Figura 33. Base de datos implementada en la nube	59
Figura 34. Datos recolectados por el dispositivo de monitoreo y almacenados por la aplicación web.....	60
Figura 35. Prueba unitaria del sistema.	63
Figura 36. Sistema de monitoreo en web	63
Figura 37. Sistema de monitoreo en móviles Parte A.....	64
Figura 38. Sistema de monitoreo en móviles Parte B.	64
Figura 39. Dispositivos de monitoreo	67
Figura 40. Modelo en estrella	69
Figura 41. Conexión de prueba a la base de datos	70
Figura 42. Medidas y dimensiones del cubo	70
Figura 43. Pantalla de ingreso a Pentaho BI Server.....	71
Figura 44. Consola de usuario de SaikuAnalytics	71
Figura 45. Flujo de tráfico dividido en días.	72
Figura 46. Flujo de tráfico dividido en el sentido.	73
Figura 47. Flujo de tráfico dividido en el sentido en función del tiempo.	73
Figura 48. Comparación de sentidos: izquierda norte - sur, derecha sur – norte.....	74
Figura 49. Flujo de tráfico dividido en clima.....	74
Figura 50. Comparación entre martes y viernes.....	75
Figura 51 Modelo del carril 1.....	77
Figura 52 Modelo del carril 2.....	78

RESUMEN

La congestión del tráfico vehicular en las ciudades con mayor densidad de población es un serio problema que disminuye la calidad de vida de sus ciudadanos. Ante este escenario, este estudio presenta una propuesta de bajo costo, con el fin de monitorear, analizar y plantear soluciones. Se propone el diseño y desarrollo de un sistema de monitoreo inalámbrico, basado en un modelo distribuido multicapa, que combinado con la tecnología de minería de datos, permite medir el tráfico en las carreteras y estudiar sus resultados. En este intento, se combinó la metodología “Extreme Programming” con “Scrum” con el propósito de asegurar la calidad del software. El dispositivo electrónico cuenta con un sensor de distancia basado en láser para la detección de vehículos, y tiene la capacidad de conectarse a Internet para la transmisión en tiempo real de la información tomada por peticiones HTTP sobre la red celular. Sobre los datos recolectados se monta un motor de minería de datos que permite la obtención de información en forma detallada. La prueba de concepto se aplicó en la ciudad de Quito, en la avenida Simón Bolívar sobre sus dos carriles centrales con resultados positivos, tanto en el funcionamiento del software y del hardware que componen el prototipo, como también en la medición referencial del tráfico.

Palabras clave

- **MONITOREO DE VEHÍCULOS**
- **SENSORES INALÁMBRICOS**
- **ARDUINO**
- **RED CELULAR**
- **IOT**

ABSTRACT

Congestion of vehicular traffic in the cities with the highest population density results into a diminution of life quality of its citizens. Facing this problem, our study intends to implement a low-cost Internet of Thing proposal, in order to monitor and analyze traffic circulation and provide solutions to reduce the above mentioned negative effects. To fulfill such proposal, a system of low-cost wireless monitoring has been designed and developed, which is supported on a distributed multilayer model. This system interacts with a device consisting of two electronic Arduino platforms, which themselves interact in master-slave mode having a distance sensor based on laser for detection of vehicles. The latter has the ability to connect to the Internet transmitting in real-time data taken by HTTP requests over the cellular network. On the data collected an engine for data mining is mounted in order to receive information about the traffic on highways and subsequently proposing potential solutions. The proof of concept has been applied in the city of Quito. There, on over their two central lanes of the Simon Bolivar Avenue such kind of solutions have been applied with positive results, both in the operation of the software and hardware that compose the prototype, as well as in reference measurements of traffic. Based on our findings, the system manages to suggest extrapolations, which enables to reduce traffic congestion, fuel waste, and also air pollution.

Keywords

- **MONITORING OF VEHICLES**
- **WIRELESS SENSORS**
- **ARDUINO**
- **CELLULAR NETWORK**
- **IO**

CAPÍTULO I

INTRODUCCIÓN

Para determinar la importancia del proyecto planteado a continuación se definirá los antecedentes, problemática, estado del arte, justificación, objetivos y alcance para el presente trabajo.

1.1 Antecedentes

La avenida Simón Bolívar, inaugurada en el año 2008, es una de las principales arterias viales de la ciudad de Quito. Tiene una longitud de 15 kilómetros, conectando el sur con el norte de una manera directa. A su vez, cuenta con seis carriles, tres en sentido sur y tres en sentido norte haciendo de ella una de las vías con más espacio para tránsito de vehículos en el país. Sin embargo, es una de las vías de mayor congestión por accidentes que se suscitan a diario, evidenciándose que no existe un sistema capaz de mostrar en tiempo real el flujo del tránsito y la comparativa con semanas anteriores.

Para este proyecto, como un componente importante, se ha recurrido a la plataforma Arduino. Esta plataforma, desde su lanzamiento en el año 2005, ha permitido que la electrónica se convierta en una herramienta abierta de la que muchos han sacado provecho con tan solo conocer fundamentos básicos, lo que la posiciona como una de las tendencias de tecnología y proyectos de nuestro tiempo abriéndose paso en numerosos trabajos vinculados al sector vehicular en las áreas de automatización y de comunicación, en la que algunas empresas públicas y privadas han puesto especial interés.

Para el componente de Minería de datos, se utilizará Pentaho. Este es un software de código abierto que permite descubrir patrones en grandes conjuntos de datos, convirtiéndolos en información útil.

Para las aplicaciones web, se usará PHP con el framework Laravel. PHP es un lenguaje de programación de código abierto del lado del servidor, originalmente

diseñado para el desarrollo web de contenido dinámico. Laravel es un framework basado en PHP que permite desarrollar aplicaciones de forma escalable y segura.

1.2 Problemática

La avenida Simón Bolívar es una de las vías más transitadas en Quito, debido a que conecta el norte, sur y los valles. En muchas ocasiones ocurren accidentes de tránsito que obstaculizan el flujo vehicular durante varias horas, lo que da como resultado grandes embotellamientos. Debido a los pocos puntos de retorno, los conductores se ven obligados a esperar, causando mucho malestar ya que no pueden llegar a sus destinos en el tiempo esperado.

El conductor promedio del Distrito Metropolitano de Quito, utiliza servicios digitales y equipos de telecomunicaciones capaces de acceder a páginas web. Es un hecho que estos conductores a pesar de poseer la potencia de la tecnología en sus manos, no tienen la información en tiempo real sobre la situación del tráfico en las vías principales como es el caso de la avenida Simón Bolívar.

1.3 Análisis del estado del arte

En el análisis de la literatura se han encontrado varios esfuerzos por parte de la comunidad científica, mismos que fueron clasificadas por el tipo de recolección de datos, aplicación, topología, método de análisis. En relación al tipo de recolección de datos a su vez existen dos variantes: euleriana que se basan en estaciones estáticas y lagrangiana que utilizan dispositivos en movimiento.

Los trabajos de tipo euleriano, como el de (Magrini, y otros, 2015) y (Paul, 2014) utilizan el método de análisis de imágenes captadas por cámaras. Por su parte, (Wang, Zheng, Xu, & Huang, 2015) y (Dessai, 2014) utilizan sensores magnéticos para la recolección de datos. Así mismo, (Perttunen, Kostakos, Riekkilä, & Ojala, 2015) y (Pascale, Nicoli, Deflorio, Dalla Chiara, & Spagnolini, 2012) aplican el método de loop inductivo. Las investigaciones presentadas por (Fernández-Lozano, Martín-Guzmán, Martín-Ávila, & García-Cerezo, 2016) y (Lee, Zhong, Du, Gutesa, & Kim, 2015) se basan en sensores Bluetooth, que captan las señales de los dispositivos que emiten estas ondas. Los trabajos a continuación utilizan sensores de

proximidad de distinto tipo. Los trabajos propuestos por (Jo, Choi, & Jung, 2014) y (Fernández-Lozano, Martín-Guzmán, Martín-Ávila, & García-Cerezo, 2016) usan sensores ultrasónicos. (Swathi, Sivanagaraju, Manikanta, & Kumar, 2016) utiliza sensores infrarrojos y por último (Fernández-Lozano, Martín-Guzmán, Martín-Ávila, & García-Cerezo, 2016) incluye una combinación de sensores ultrasónicos, bluetooth y laser. En comparación con nuestra propuesta al igual que Fernández-Lozano, J usamos sensores laser por su exactitud, para la lectura del flujo del tráfico, ubicándonos en la categoría euleriana y de sensores de proximidad.

Las propuestas de tipo lagrangiana usan celulares son: (Canepa, y otros, 2014), (Mousa, Abdulaal, Boyles, & Claudel, 2015) y (Costanzo, 2013) captan datos a través de nodos ubicados en los vehículos. Mientras que (Jiang & Claudel, 2013) y (Singha & Kalita, 2013) en sus dos publicaciones utilizan celulares para la lectura del tráfico.

En referencia a los estudios que fusionan las dos categorías de extracción de datos, tanto euleriana como lagrangiana, (Jiang & Claudel, 2013) y (Elmotelb, Shabana, & Tolba, 2016) lo consiguen al usar una red de sensores wireless WSN y el GPS de celulares.

En función de la aplicación obtenida, los trabajos se encuentran ordenados de la siguiente manera: Una aplicación para observar el flujo en tiempo real presentadas por (Al-Sakran, 2015) , (Jo, Choi, & Jung, 2014), (Singha & Kalita, Using Mobile Phone Network for Urban Traffic Management, 2013) y (Costanzo, 2013). El caso de (Zhou, Chen, Chen, & Zhao, 2013) avanza más al ajustar la aplicación a los requerimientos reales del usuario. Los trabajos de (Zhou, Chen, Chen, & Zhao, 2013), (Pascale, Nicoli, Deflorio, Dalla Chiara, & Spagnolini, 2012) y (Paul, 2014) se enfocan en encontrar la tendencia del tráfico. Por otra parte (Eren, Pakka, AlGhamdi, & Yue, 2013) , (Canepa, y otros, 2014) , y (Al-Sakran, 2015) buscan realizar una simulación del tráfico. A su vez (Singha & Kalita, Mapping Mobile Phone Network onto Urban Traffic Network, 2013) en sus dos publicaciones citadas y (Liang, 2013) se enfocan en realizar una descripción del tráfico. Nuestra propuesta se enfoca en una aplicación en tiempo real y la descripción del comportamiento del tráfico.

El uso de una topología de red que centralice las señales emitidas por los nodos que detectan el flujo vehicular, en uno central, encargado de conectarse con los servicios en la nube, que incorpora dispositivos ZigBee se encuentra en (Eren, Pakka, AlGhamdi, & Yue, 2013), (Lee, Zhong, Du, Gutesa, & Kim, 2015), (Dessai, 2014) y (Swathi, Sivanagaraju, Manikanta, & Kumar, 2016). Este aspecto se contrasta con nuestro trabajo debido que a que cada nodo medidor de tráfico, es emisor de datos hacia el servidor sin contar con un concentrador de señales que centralice la conexión.

En función del análisis de los datos recopilados, (Zhang, 2013) utiliza un modelo matemático, mientras que (Al-Sakran, 2015) incorpora agentes de software y por último (Zhou, Chen, Chen, & Zhao, 2013) y (Singha & Kalita, Using Mobile Phone Network for Urban Traffic Management, 2013) aplican matrices origen – destino. A diferencia de todas las propuestas anteriores presentamos un análisis basado en minería de datos como una respuesta a este objetivo.

1.4 Justificación

El proyecto presentado tiene la capacidad de obtener y mostrar una medición de la cantidad de vehículos que transitan por la avenida Simón Bolívar, aportando de esta forma al mejoramiento de la movilidad en la ciudad.

Dada la falta de información en tiempo real del estado del tráfico, los usuarios continúan experimentando congestión vehicular y desinformación; además, sin ejecutar un proyecto como este, es posible que no se pueda tener registros del número de vehículos que transitan en esta vía dados los índices de crecimiento del parque automotor en el Distrito Metropolitano de Quito.

Con el presente proyecto los usuarios podrán tomar decisiones sobre la mejor ruta a pueden escoger, de modo que lleguen a su destino en el tiempo esperado y en general mejorar el tránsito vehicular de una de las principales vías del Distrito Metropolitano de Quito.

1.5 Objetivos

- Objetivo General

Desarrollar una aplicación IoT para el monitoreo y predicción del tráfico en la avenida Simón Bolívar de Quito, utilizando una red de sensores basados en la plataforma Arduino que brindarán información al usuario para tomar decisiones y mejorar el tránsito en la ciudad.

- **Objetivos Específicos**
 1. Realizar un análisis secuencial de la literatura y el marco teórico de IoT con tentativa para medir el tráfico.
 2. Diseñar una red de sensores basados en la plataforma Arduino para la medición del tráfico.
 3. Aplicar la metodología SCRUM y XP para el desarrollo de la aplicación web.
 4. Pruebas y evaluación del tráfico en tiempo real y visualización de los resultados en una interfaz amigable.
 5. Proponer un mecanismo de predicción para el tráfico vehicular

1.6 Alcance

Este proyecto consiste en el desarrollo de una aplicación funcional para la medición del número de vehículos en tiempo real y predicción del tráfico en la avenida Simón Bolívar. Esta aplicación se desarrollará con recursos propios y contará con el desarrollo tanto del hardware como del Software que interactuarán entre sí.

Hardware:

Dispositivo de monitoreo: Mediante es uso de la plataforma Arduino para realizar el conteo de los vehículos por medio de un sensor láser y reportarlo por medio del módulo GSM.

Software:

Aplicaciones web:

1. Se desarrollará una aplicación web capaz de recibir los datos enviados por el dispositivo de monitoreo.

2. Sistema de Monitoreo: Se desarrollará una Aplicación Web para visualizar el estado del tráfico en tiempo real.
3. La aplicación será desarrollada con el patrón de arquitectura MVC (Modelo Vista Controlador), y en lenguaje PHP.
4. Se utilizará el framework Laravel.
5. La base de datos será implementada sobre el motor Mysql.

Recolección de Datos:

Se recolectaran datos en la Avenida Simón Bolívar de Quito usando el dispositivo de monitoreo por un periodo de dos semanas.

Estadísticas y Predicción:

1. La aplicación contará con estadísticas de tráfico mediante la herramienta Pentaho en base a los datos recolectados.
2. Se propondrá un mecanismo de predicción del tráfico en base a los datos recolectados

CAPÍTULO II

MARCO TEÓRICO

Para el desarrollo del presente estudio se realizará la medición del número de vehículos en dos carriles, se utilizará dos dispositivos de monitoreo: uno en sentido Norte-Sur y otro en sentido Sur-Norte de la avenida Simón Bolívar de la ciudad de Quito, Ecuador. Los datos obtenidos alimentarán una base de datos ubicada en la nube que se integrará con una interfaz amigable que permitirá al usuario obtener datos en tiempo real y predicciones del tráfico.

El presente proyecto ha sido posible gracias a las siguientes tecnologías de software y hardware: Cloud Computing, Arduino, WSN, Laravel, Bootstrap, Minería de datos, XP y Scrum.

2.1 Cloud Computing

Según Eduardo Martín a través de TicBeat.com, Cloud Computing se refiere al término utilizado para definir la prestación de servicios computacionales a través de Internet, aumentando la accesibilidad a cualquier prestación que funcione bajo este nuevo paradigma además de volverlos flexibles y personalizables en función de la demanda requerida. A la característica de extender y contraer funciones y capacidades, se la denomina elasticidad, y se refiere a que el usuario podrá utilizar los recursos que necesita y no más, a no ser que él lo desee; resultando en una optimización de prestaciones en comparación con el modelo tradicional que requiere la compra de un servidor con prestaciones definidas con capacidad de crecer pero no de contracción. Estos servicios están divididos en tres categorías: infraestructura como servicio (IAAS), plataforma como servicio (PAAS) y software como servicio (SAAS) (Martín, 2014). Figura 1. A continuación se detallaran cada uno de los servicios antes mencionados.

Infraestructura como servicio.- Es la denominación brindada para especificar la prestación de recursos informáticos en su más bajo nivel es decir definir la cantidad de memoria, procesamiento, almacenamiento, red, entre otros, que podrán ser accedidos por el usuario y brindar el uso que éste desee. Es similar su

funcionamiento al de una máquina virtual y sobre ella se podrá elegir el sistema operativo y todas las aplicaciones a ser instaladas. Al ser la capa más baja en las categorías de servicios de Cloud se puede tener un mayor control sobre el servicio solicitado, pero a su vez, requiere de mayores configuraciones lo que se traduce en mayor dificultad. Por lo tanto, existe una categoría que cuenta con configuraciones preestablecidas (Martín, 2014).

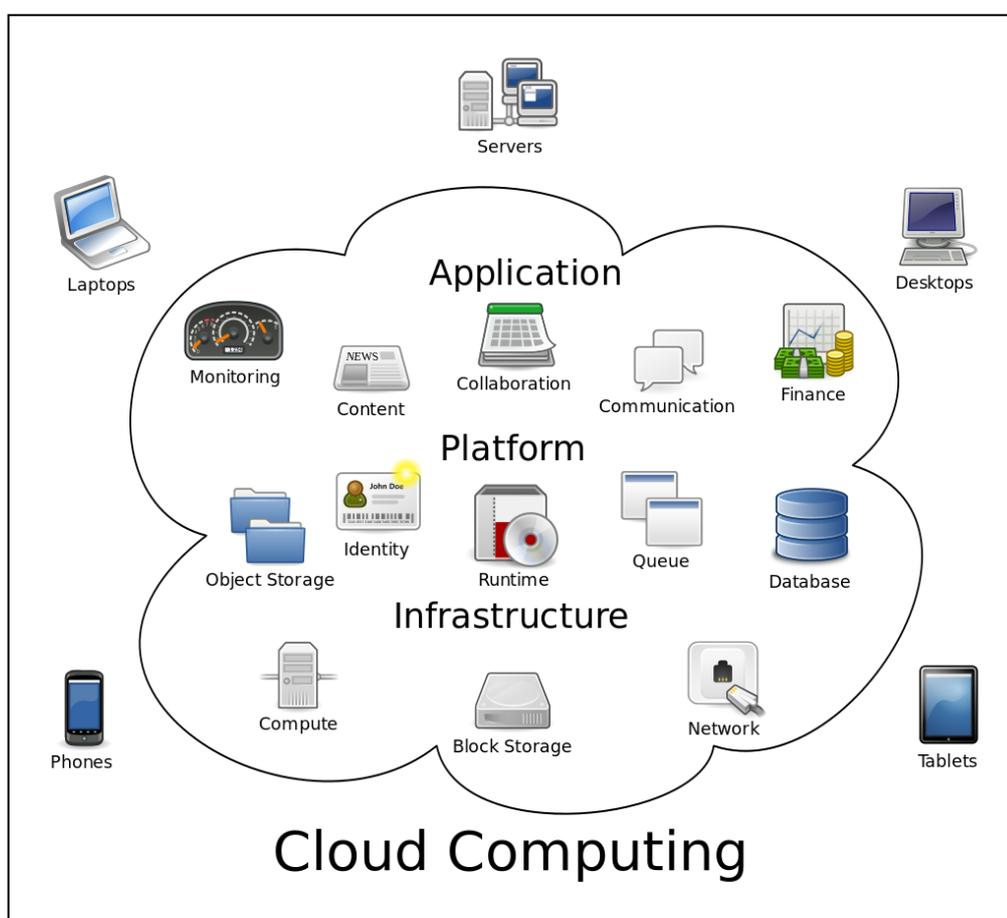


Figura 1. Cloud Computing

Fuente: (Johnston, 2009)

Plataforma como servicio, se trata de la siguiente capa de prestación de recursos de Cloud y en función de requerimientos en común por parte de los usuarios; se juntan un conjunto de prestaciones como por ejemplo: un servidor web, un motor de administración de bases de datos, un administrador de archivos entre otras funcionalidades necesarias, que comúnmente son solicitadas, únicamente tienen que pasar a ser configuradas por el cliente y no instaladas, representando un ahorro en

precio y dificultad con respecto a IAAS, pero se pierde control sobre el entorno. Sin embargo, el usuario final usualmente requiere utilizar una aplicación en específico (Martín, 2014).

Software como servicio.- Es la capa final de prestaciones en la que el usuario accede a un software con toda la configuración necesaria para que éste solo utilice la aplicación sin preocuparse del alojamiento, beneficiando a empresas y usuarios finales que no requieren de una plataforma informática para alojar (Martín, 2014).

2.2 Arduino

Arduino se define como el proyecto compuesto por la placa de “hardware libre” con un chip microcontrolador, un lenguaje de programación e IDE desarrollados en el instituto IVREA, Italia, con la finalidad de proveer una plataforma electrónica que permita desarrollar proyectos tecnológicos (Artero, 2013). La categorización de Arduino se ilustra en la figura 2.

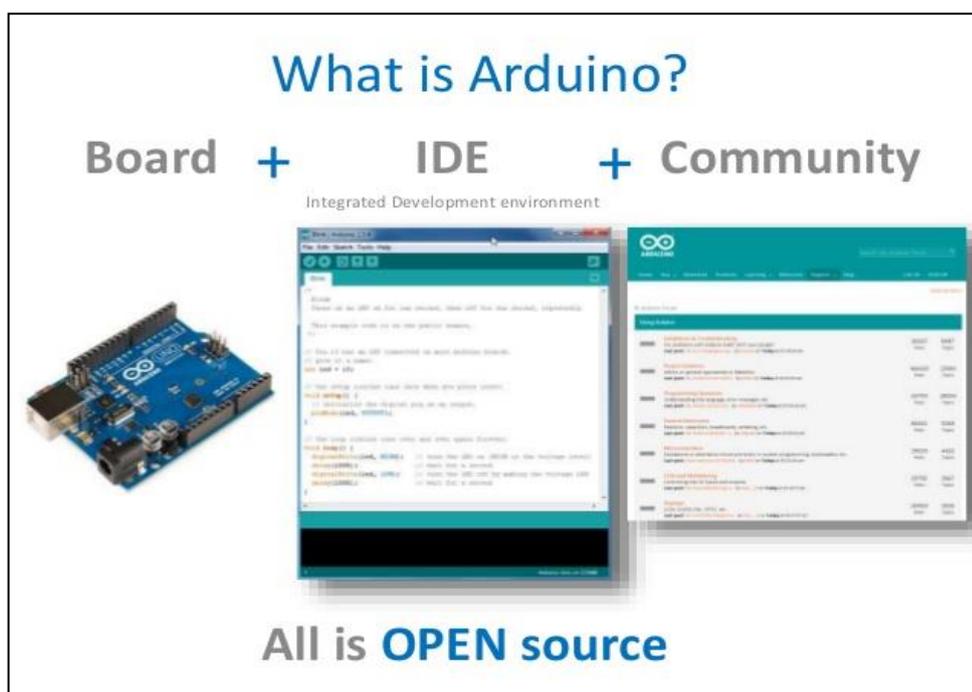


Figura 2. Elementos de la plataforma Arduino

Fuente: (Riachi, 2014)

La placa Arduino se encuentra disponible en 21 modelos distintos con capacidades y diseños que se adaptan a cualquier necesidad del usuario. Funciona

con un adaptador de siete a doce voltios y dependiendo del modelo también puede recibir energía a través de un puerto micro-USB o USB-b, utiliza un microcontrolador Atmel con distintas velocidades y tamaño de memoria. Sus dimensiones varían entre dispositivos de 4.8 x 1.8 cm de área en su versión micro hasta 10.1 x 53.3 cm en la versión mega. Cuenta con distinta cantidad de pines los cuales se encuentran distribuidos entre los básicos de alimentación, los de entrada y salida digital y los que incorporan una salida PWM (Pulse Width Modulation). Forman parte elementos básicos tales como el reloj y leds informativos y un botón de reseteo como se evidencia en la Figura 3. Una de las funciones que presenta es la capacidad de transformar señales analógicas en señales digitales sin código de programación alguno, es decir de manera automática, motivo por el cual se han desarrollado sensores especializados para las prestaciones de Arduino (Artero, 2013).

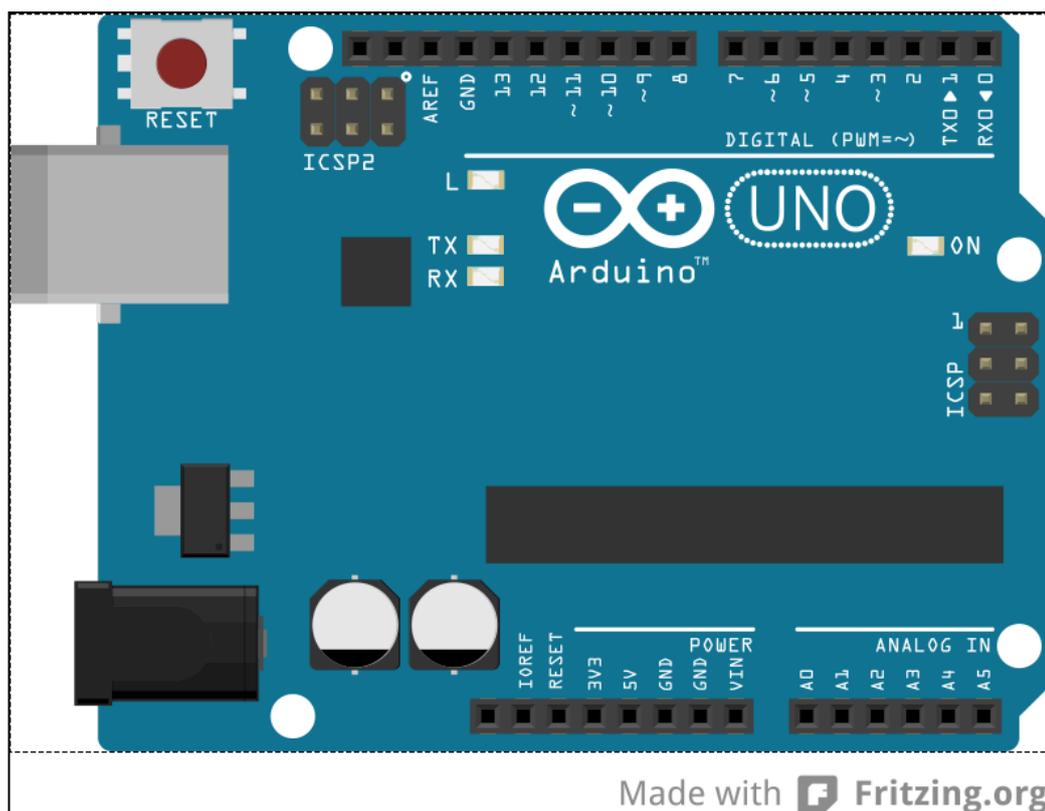


Figura 3. Placa Arduino UNO r3

Fuente: (Riachi, 2014)

Una de las características de la plataforma Arduino y de la cual depende en gran parte el éxito del proyecto, es la capacidad de expansión de la placa a través de módulos extensibles uno sobre otro lo cual permite añadir múltiples funcionalidades. Se puede añadir la capacidad de acceder a una red GSM de telefonía celular únicamente añadiendo el componente necesario entre más opciones. Los precios y características y calidad varían en el mercado haciendo flexible el desarrollo de proyectos con esta plataforma (Artero, 2013). Para controlar el hardware, se cuenta con el lenguaje con el mismo nombre.

El lenguaje de programación Arduino está basado en el paradigma orientado a objetos, es una derivación del lenguaje Processing, el cual su vez, es derivado de Java, diseñado para trabajar con objetos desde sus fundamentos. El lenguaje Arduino además es fuertemente tipado, y su estructura en todos los programas, es orientada a eventos (Artero, 2013). Hoy el lenguaje puede ser escrito con una mayor cantidad de IDE's aparte de su original, a través de módulos que pueden ser instalados sobre plataformas como Sublime 3 o Visual Studio.

El entorno de desarrollo cuenta con control de sintaxis del lenguaje, configuración de la placa a la cual se va a enviar el código compilado por el mismo IDE y la ejecución del código en la placa configurada. No cuenta con auto-completación de código ni manejo de ficheros. La sintaxis del lenguaje es bastante similar a C por lo que los bloques de código se delimitan con llaves y al final de cada sentencia se debe utilizar el signo punto y coma (Artero, 2013).

2.3 WSN

Wireless sensor network se refiere a un conjunto de nodos (sensores) dispersos en un área geográfica, con un mismo fin o parecido e interconectados por un medio inalámbrico hacia un servidor central que controla los diferentes puntos (Figura 4). Alcanzar una WSN es de los mayores estados que se pueden alcanzar en IoT (Internet de las cosas) dado el gran alcance que éstas tienen.

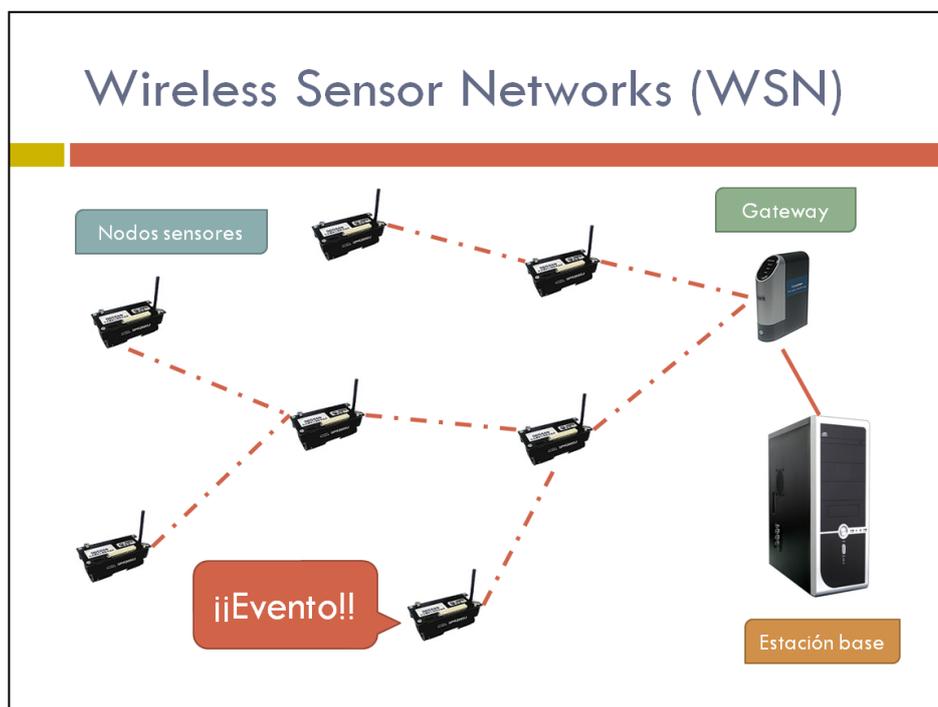


Figura 4. Red de sensores inalámbricos

Fuente: (BLOGSPOT, 2016)

2.4 Minería de datos

Consiste en un conjunto de técnicas y métodos que permiten obtener patrones, tendencias o reglas mediante el análisis de grandes bases de datos. Se busca obtener patrones o reglas en los datos al analizarlos en base a una o varias variables que intervienen en el entorno del proceso de recolección de datos.

Para encontrar estas coincidencias es necesario usar algoritmos de búsqueda, estadística o modelos matemáticos que permitan obtener conocimiento de los datos y las variables que influyen en su modificación.

El análisis termina cuando se han obtenido las reglas y las variables que afectan el entorno, estas permiten crear nuevos datos predictivos que ayudan a la toma de decisiones, se debe tomar en cuenta que siempre existe un margen de error.



Figura 5. Proceso minería de datos

La figura 5 explica el proceso de minería de datos, en la primera imagen se observa la base de datos que tiene todos los datos recolectados, en la segunda imagen se observa la selección de datos, esto quiere decir que se descartaron datos como por ejemplo ruido o estática que no son datos válidos para el análisis, luego se continúa con el evaluación de los datos con el objetivo de encontrar patrones, tendencias o reglas que expliquen el comportamiento de los datos bajo variables preestablecidas. Finalmente, se obtiene el conocimiento que son los patrones o reglas que permiten realizar predicciones sobre el comportamiento de los datos en el entorno.

2.5 Laravel

Es un framework de desarrollo de aplicaciones web de código abierto, basado en PHP 5. Fue desarrollado en el año 2011 con el objetivo de crear código de forma elegante y rápida, aprovechando las últimas características que proporciona PHP 5 (Nunez, 2014).

Laravel utiliza Composer, que es una herramienta para la gestión de dependencias, es decir, permite obtener todas las dependencias, librerías actualizadas y en las versiones necesarias para el funcionamiento de la aplicación web. Laravel tiene las siguientes características: a. Sistema de ruteo, también RESTful; b. Blade, Motor de plantillas; c. Peticiones Fluent; d. Eloquent ORM; e. Basado en Composer; f. Soporte para el caché; g. Soporte para MVC; h. Usa componentes de Symfony; i. Bootstrap (ECURED, 2014).

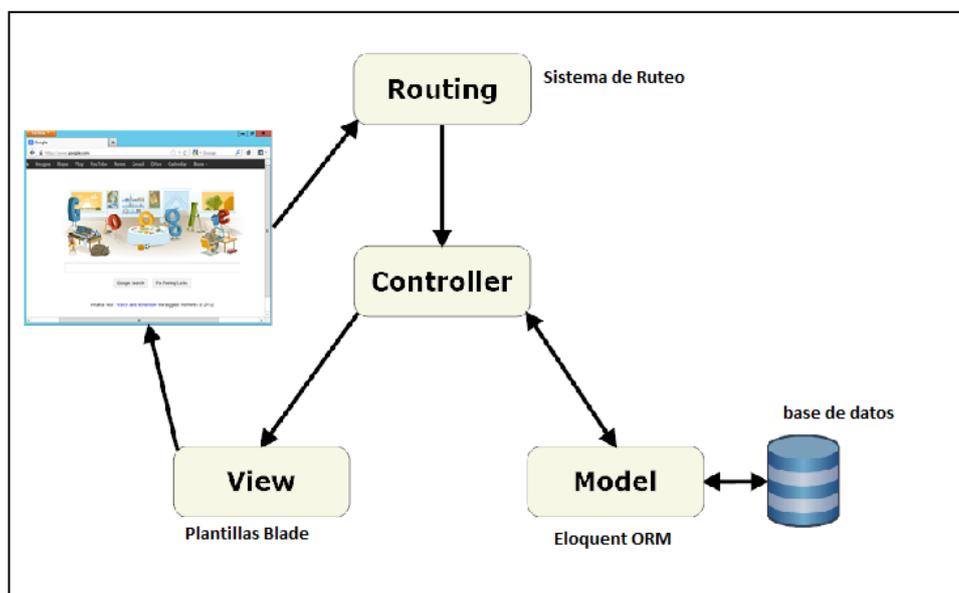


Figura 6. Componentes MVC Laravel

Fuente: (ECURED, 2014).

La figura 6 ilustra cómo los componentes que se detallarán a continuación interactúan entre sí, para permitir el funcionamiento del framework Laravel. El sistema de ruteo consiste en un URI el cual permite definir en `app/routes.php` todas las rutas del sistema, haciendo mención a la función del controlador que se direcciona y el retorno. Ese sistema permite tener URLs amigables y permiten organizar el código y la navegación de una forma simple.

El sistema de plantillas Blade proporcionado por Laravel es potente, la diferencia con otros sistemas de plantillas radica en que no restringe el uso de código PHP en las vistas. Todas las vistas Blade son compiladas con el código PHP y almacenadas en el cache hasta que sean modificadas. Blade evita la sobrecarga en su aplicación y usa las vistas con extensión `blade.php` que estén localizadas en `app/views`.

Eloquent ORM incluido en Laravel ofrece una implementación simple de Active Record para trabajar con la base de datos. Cada tabla de base de datos tiene un "Modelo" correspondiente que se utiliza para interactuar con esa tabla. Eloquent ORM maneja automáticamente el borrado lógico y la auditoria de cada registro en la tabla actualizando automáticamente la fecha de creación o modificación y eliminación, sin necesidad de poner filtros o enviar las fechas en las consultas.

2.6 Bootstrap

Fue creado por Mark Otto y Jacob Thornton para mejorar las herramientas internas en Twitter. En agosto del 2011 Twitter liberó Bootstrap como código abierto. Es un framework, que permite crear interfaces web con CSS y JavaScript, permitiendo crear la interfaz web del sitio una sola vez y que este se adapte automáticamente al tamaño del dispositivo en que se visualice, es decir, el sitio web se adapta automáticamente en un teléfono, tableta, PC, etc. Esta técnica se denomina “responsive design” o diseño adaptativo (ARWEB, 2014)



Figura 7. Bootstrap

Fuente: (ARWEB, 2014)

GridSystem, permite crear una rejilla con filas y columnas que se ajustan automáticamente al tamaño de la pantalla. Al crear el diseño se puede determinar que partes del código se visualizarán en los diferentes dispositivos en el caso que se requiera usando las siguientes etiquetas: 1) xs: teléfonos menores a 768px; 2) sm: tablets mayor o igual a 768px; 3) md: computadores de escritorio mayor o igual a 992px; 4) lg: computadores de escritorio grandes mayor o igual a 1200px. (ARWEB, 2014).

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Figura 8. GridSystem

Fuente: (ARWEB, 2014)

En la figura 8 se puede observar el GridSystem que divide la pantalla en 12 columnas y permite agruparlas para crear columnas más grandes según se requiera, dependiendo del tamaño de la pantalla estas 12 columnas se re organizan para que el contenido se visualice correctamente en cualquier pantalla (ARWEB, 2014).

Bootstrap es compatible con la mayoría de los navegadores web: Google Chrome, Firefox, Internet Explorer, Opera, and Safari, versiones actuales y antiguas como IE 8.

2.7 SCRUM

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación (SOFTENG, 2014).

La estructura de Scrum basa todas sus prácticas en un proceso iterativo e incremental como se muestra en la figura 9, primero se crea el requerimiento del proyecto y se definen las tareas de la iteración, el círculo más grande representa las iteraciones que contienen actividades de desarrollo, el entregable de esta iteración es un producto incremental. El círculo de la parte superior representa la inspección diaria que ocurre durante la iteración, donde cada miembro de equipo inspecciona el trabajo de otro y hace las adaptaciones apropiadas. El entregable final es un incremento de la funcionalidad o un entregable listo para salir a producción.

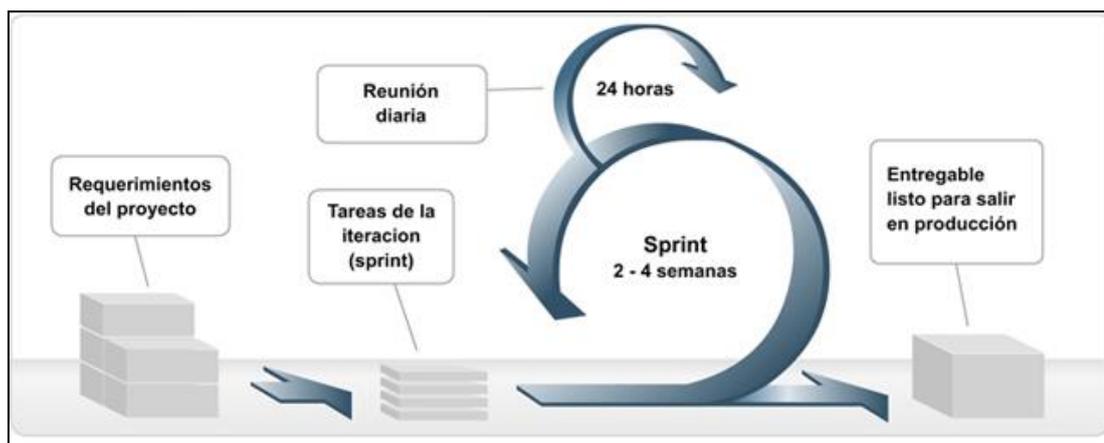


Figura 9. Scrum

Fuente: (RESERV, 2014)

CAPÍTULO III

ESPECIFICACIÓN DE REQUERIMIENTOS

La Ingeniería de Requerimientos es un amplio conjunto de tareas y técnicas que proporcionan el mecanismo apropiado para analizar, entender y especificar las necesidades del cliente (Pressman, 2005).

Para el proyecto planteado se elaborará a continuación una Especificación de Requerimientos de Software (ERS) que permitirá definir y analizar las necesidades tanto de hardware como de software.

3.1 Propósito

La presente Especificación de Requerimientos de Software pretende identificar el contexto del proyecto y determinará los requerimientos funcionales y no funcionales.

Esta especificación se basará en el estándar del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE por sus siglas en Inglés), Practicas Recomendadas para los Requerimientos de Software IEEE/ANSI 830-1998 (IEEE, 2008)

3.2 Alcance

El presente proyecto, atenderá los procesos de medición, transmisión, tratamiento, almacenamiento, presentación y análisis de la información referente al número de vehículos que circulan por dos carriles de la avenida Simón bolívar de Quito. El Cuadro 1 detalla estos procesos.

Cuadro 1

Procesos del Prototipo

Proceso	Descripción
Medición	Este proceso se encarga del conteo de vehículos que pasan por un carril específico de la avenida Simón Bolívar mediante el uso de sensores láser, debidamente montados, configurados y calibrados en la placa electrónica Arduino.
Transmisión	La primera placa Arduino procesa la información obtenida del sensor láser y la transmite a la segunda placa Arduino, la cual mediante el módulo GSM transmite los datos dentro de la URL usando el protocolo HTTP.
Tratamiento	Se encarga de recibir y procesar la información enviada por la placa electrónica Arduino. Con este fin se ha creado una Aplicación Web que recibe la información enviada por la URL y consulta el estado del clima desde un API.
Almacenamiento	Una vez verificada la información, este proceso la guardará en una base de datos, para su posterior análisis. La estructura del sistema de base de datos, garantizará la integridad de la información recibida.
Presentación	De los datos almacenados anteriormente, la aplicación Web consulta la información y la presenta en forma de gráficos de línea en tiempo real, adicionalmente, se presenta un mapa de ubicación geográfica de los dispositivos.
Análisis	Mediante la herramienta Pentaho se realiza el análisis de los resultados obtenidos en la medición, los cuales permiten obtener una visión global del estado del tráfico en un periodo de tiempo establecido.

3.3 Limitaciones del Prototipo

El proyecto implementará las funciones principales de un sistema completo de este tipo, por lo que se han determinado algunas limitaciones que se detallan a continuación:

- No encriptará la información que se transmite de la plataforma Arduino a la aplicación Web.
- No implementará el protocolo HTTPS para la transmisión de datos segura.
- No se realizará la medición de más de dos carriles.
- No se realizara mediciones por un periodo mayor a 2 semanas.
- No consultará información de otros sistemas de medición de tráfico.
- No implementará ningún tipo de alerta en base a los parámetros medidos, ya que la información que se espera recolectar es referencial y no exacta.

Los beneficios de usar una aplicación web en el presente proyecto, que sea capaz de interactuar con el dispositivo de monitoreo para procesar información sobre la medición de tráfico son:

- Escalabilidad en la medición de más de dos carriles.
- Conocer el estado del tráfico en tiempo real.
- Compartir la información sobre el estado del tráfico.
- Almacenar la información para estudios posteriores.

3.4 Personal Involucrado

El equipo de personas que llevarán a cabo este proyecto está compuesto por profesionales en distintos campos del conocimiento. El cuadro 2 detalla los roles que cumplirán cada uno de los miembros del equipo de trabajo.

Cuadro 2

Personal Involucrado

Nombre	Campo Profesional	Rol	Información
Walter Fuertes	Ingeniero en Sistemas	Director del proyecto	wmfuertes@espe.edu.ec
Alejandra Abad	Egresados de Ingeniería en Sistemas	Analista y programador	maabad2@espe.edu.ec
Hugo Nugra			henugra@espe.edu.ec

3.5 Definiciones

Prototipo: Es una primera versión o modelo del software en el que se ha incorporado algunas características del producto final.

Tráfico: Se refiere al tránsito o desplazamiento de medios de transporte, seres humanos u objetos por algún tipo de camino o vía.

Arduino: Placa electrónica que facilita el desarrollo de prototipos, existen varias versiones de placas con diferentes características.

Sensores: Elementos electrónicos capaces de interactuar con el ambiente que los rodea.

Dispositivo de monitoreo: Elemento hardware compuesto por 2 placas Arduino, un sensor láser y un módulo GSM, capaz de tomar datos y enviarlos por medio del Internet.

Base de Datos: Conjunto de datos relacionados entre sí.

Sistemas de Bases de Datos: Software que facilita la gestión de una base de datos.

3.6 Acrónimos

IR: Identificación de requerimiento.

RE: Requerimiento específico.

NR: Nombre del requerimiento.

ERS: Especificación de Requerimientos de Software.

TCP/IP: (Transfer Control Protocol - Internet Protocol), hace referencia al conjunto de protocolos en los que se basa el Internet.

HTTP: (HyperText Transfer Protocol), protocolo de transferencia de hipertexto usado en cada transferencia sobre Internet.

HTTPS: (HyperText Transfer ProtocolSecure), protocolo seguro de transferencia de hipertexto.

URL: del inglés UniformResourceLocator, es un identificador de recursos uniforme.

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.

IEEE 802.11 b/g: Norma de funcionamiento para redes inalámbricas.

GSM: Sistema global para las comunicaciones móviles.

GPRS: servicios generales de paquetes por radio

WNS: red de sensores Wireless.

3G: Tercera generación de transmisión de voz y datos a través de telefonía móvil.

3.7 Visión General del Documento

La presente Especificación de Requerimientos de Software (ERS) se ha dividido en tres secciones. La primera presenta una visión general del proyecto a desarrollarse y la introducción a la Especificación de Requerimientos de Software. La segunda describirá de forma general las funciones que realizara aplicación Web junto con el dispositivo de monitoreo, los datos recolectados, los tipos de usuarios, restricciones, y dependencias que existirán en la aplicación. La última sección especificará los requerimientos funcionales y no funcionales.

3.8 Descripción General

En esta sección de la Especificación de Requerimientos de Software (ERS) se describirá el producto que se pretende construir, el cual brindará una solución al problema planteado anteriormente.

3.8.1 Perspectiva del Producto

El prototipo del sistema de monitoreo y del dispositivo de monitoreo usarán un conjunto de tecnologías apropiadas, que deberán interactuar entre sí para contar el número de vehículos en la Avenida Simón Bolívar. Estos datos serán recolectados, procesados, transmitidos, almacenados, consultados y analizados, con el propósito de proporcionar información en tiempo real del tráfico.

3.8.2 Funciones del Producto

La figura 10 identifica los Casos de Uso que se determinaron en el análisis del proyecto, interactuando de la siguiente forma: el conteo de vehículo se obtiene de la placa Arduino A, que interpreta la distancia enviada por el sensor láser mediante el algoritmo desarrollado en la plataforma Arduino, y determina el número de vehículos que transitan por el carril monitoreado. La placa Arduino B solicita el conteo a la placa Arduino A y cada 5 minutos envía esta información mediante la red GSM/GPRS. El dispositivo de monitoreo, mediante la red GSM/GPRS se conecta con la aplicación web que registra el conteo en la base de datos, junto con el estado del clima; ésta a su vez, es consultada por en el sistema de monitoreo que permite al usuario obtener las estadísticas en tiempo real. Además, mediante Pentaho se obtiene el análisis de los datos recolectados, que es útil para tomar decisiones. Estas funciones están relatadas en el Cuadro 3, en el cual se observa los procesos que se ejecutarán con el objetivo de cumplir con la perspectiva del producto.

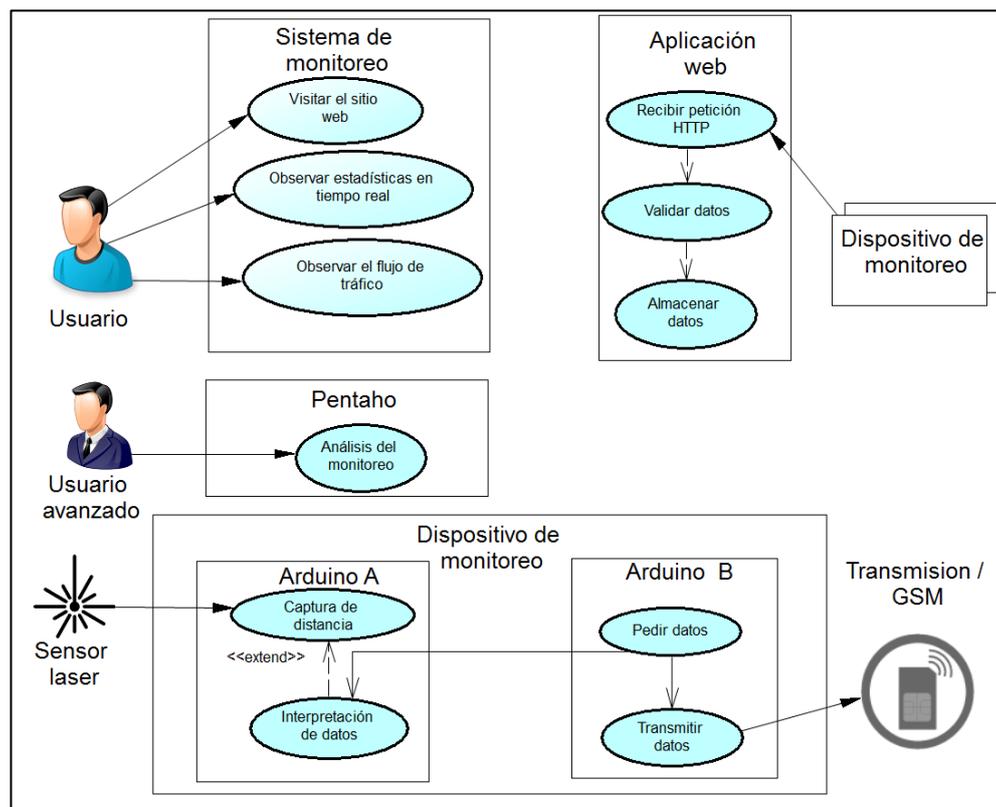


Figura 10. Casos de uso

Cuadro 3

Funciones generales del prototipo.

Proceso	Descripción
Medición	La función principal en este proceso será la de recolectar los valores obtenidos del sensor láser. Cada lectura generada será almacenada en una variable temporal definida en el software que ejecuta la placa Arduino.
Transmisión	Una vez capturada la información, se la procesará para ser transmitida a la aplicación Web. Esta transmisión de datos se realiza de esta forma: la primera placa Arduino procesa la información obtenida del sensor láser y la trasmite a la segunda placa Arduino, la cual mediante el módulo GSM trasmite los datos dentro de la URL por el protocolo HTTP.
Tratamiento	La aplicación Web se encarga de recibir y procesar la información enviada por la placa electrónica Arduino mediante la URL y adicionalmente consulta el clima de un api para su posterior almacenamiento.
Almacenamiento	Una vez verificada la información, este proceso la almacenará, en la base de datos, para su posterior análisis. La estructura de la base de datos deberá garantizar la integridad de los datos y se incluirá información como hora y fecha de la lectura
Presentación	De los datos almacenados anteriormente, la aplicación Web consulta la información y la presenta en forma de gráficos de línea en tiempo real, adicionalmente, se presenta un mapa de ubicación geográfica de los dispositivos.
Análisis	Mediante la herramienta Pentaho se realiza el análisis de los resultados obtenidos en la medición, los permiten acceder a una visión global del estado del tráfico en un periodo de tiempo.

3.8.3 Características de los Usuarios

El prototipo de la aplicación de monitoreo identifica tres tipos de usuarios, de los cuales uno es el dispositivo de monitoreo capaz de captar la información requerida, el otro es el usuario visitante que puede acceder a la aplicación y obtener los valores del monitoreo. En el presente prototipo cabe recalcar que existen dos dispositivos de monitoreo, cada dispositivo contiene un sensor, el cual es usuario del software que lo controla. Este pequeño programa, que se construirá usando el entorno de desarrollo que provee Arduino, procesará la información captada por cada sensor y permitirá su transmisión hacia la aplicación Web.

Existe un tercer usuario que es capaz de acceder a la aplicación Pentaho, este usuario avanzado puede acceder a las estadísticas generales que la herramienta permite. El Cuadro 4 describe las características generales de cada tipo de usuario.

Cuadro 4

Características de los usuarios

Usuario	Descripción	Capacidad Técnica
Sensor Electrónico	Captará la variación de la distancia y entregará esta información al dispositivo electrónico para que sea procesada y enviada.	Deberá reaccionar a los cambios de distancia.
Dispositivo de Monitoreo	Procesará la información entregada por los sensores electrónicos y podrá enviarla, haciendo uso del módulo GSM usando el protocolo HTTP, hacia la aplicación Web.	Deberá tener la capacidad suficiente para captar información, procesarla y transmitirla usando los protocolos de comunicación de datos HTTP.
Visitante	Podrá consultar la información sobre el estado del tráfico desde el sistema de monitoreo, sin necesidad de registrarse como usuarios	No necesitará una capacidad técnica especializada y solo deberá estar familiarizado con el uso de un navegador Web.
Avanzado	Podrá acceder a la aplicación Pentaho, visualizar y analizar las estadísticas en base a los datos recolectados.	Debe estar familiarizado con la Plataforma Pentaho.

3.8.4 Restricciones

Las restricciones detalladas a continuación, permitirán conocer las limitaciones del hardware que se empleará en el desarrollo del proyecto, las interfaces con otras aplicaciones, los protocolos de comunicación de datos y los lenguajes de programación que se utilizarán. El Cuadro 5 describe cada una de las restricciones.

Cuadro 5

Descripción de las restricciones del proyecto

Restricción	Descripciones
Limitaciones del Hardware	Para el envío del conteo de una Arduino a otro el protocolo de comunicación solo permite enviar 16 bytes por lo que el Arduino A adicionalmente separa el numero en arreglos de 8 bits para que el Arduino 2 los recepte y genere el entero nuevamente.
Protocolos de comunicación	Los protocolos de comunicación de datos serán HTTP y por medio de una petición tipo GET.
Lenguajes de Programación	La placa Arduino está programada con el lenguaje Arduino en la plataforma del mismo nombre. La aplicación web está desarrollada en PHP

3.8.5 Suposiciones y Dependencias

El equipo donde se ejecutará la aplicación Web, deberá cumplir con los requisitos necesarios para soportar el servidor de aplicaciones y el sistema de gestión de base de datos. Deberá también, estar conectado adecuadamente al Internet para permitir la transmisión de datos desde los dispositivos electrónicos.

Cada dispositivo de monitoreo tendrá una fuente de alimentación de energía adecuada y deberá tener señal, sin estas características no se podrá transmitir la información recolectada hacia la aplicación Web.

3.9 Requisitos Específicos

El Cuadro 6 recoge los requisitos que permitirán planificar, diseñar, desarrollar y probar el prototipo de aplicación Web y el software que ejecutará cada dispositivo electrónico.

Cuadro 6

Requisitos Específicos

RE	NR	Características
RE01	Consultar información	Los usuarios podrán acceder al monitoreo en tiempo real.
RE02	Capturar y Procesar las lecturas	El dispositivo electrónico podrá capturar las lecturas de los sensores y procesarlas
RE03	Transmitir la información	El dispositivo de monitoreo deberá transmitir, hacia la aplicación Web, la información captada por los sensores electrónicos.
RE04	Almacenar información	Permite almacenar la información recolectada y transmitida por el dispositivo de monitoreo.
RE05	Iniciar sesión	El usuario avanzado deberá identificarse para acceder a Pentaho
RE06	Análisis de información con minería de datos	Permite generar gráficos de los datos del tráfico vehicular recolectados en función del tiempo.

3.9.1 Interfaces Externas

A continuación, se describirán los requisitos que afectan a las interfaces de usuario, hardware, software y comunicaciones. También se incluye el detalle de la interfaz que se ejecutará en el dispositivo electrónico Arduino.

3.9.1.1 Interfaces de Usuario

La interfaz, con el usuario visitante, será construida en base a un mapa de la ciudad de Quito, donde estarán localizados, cada uno de los dispositivos de monitoreo, y presentará la información del tránsito en tiempo real en gráficos de línea, Adicionalmente se mostrará el número de vehículos.

La interfaz para el usuario avanzado estará dada por la plataforma Pentaho la cual permite obtener gráficos a partir de los datos recolectados, en función de tiempo.

3.9.1.2 Interfaces de Hardware

La aplicación Web se ejecutará en un servidor que cubra las características necesarias para soportar la ejecución del servidor de aplicaciones y el sistema de base de datos. Las características recomendadas son:

- Procesador 2.7Ghz
- Memoria RAM de 2Gb
- Disco duro 500gb
- Conexión a Internet
- Conexión FTP

Será necesario contar con una placa electrónica Arduino equipada con un módulo de expansión para GSM que permita la conexión a Internet. También es necesario el sensor láser.

3.9.1.3 Interfaces de Software

Tanto el servidor de aplicaciones, como el sistema de base de datos, deberán ser independientes del sistema operativo para asegurar su portabilidad. En este proyecto en particular se utilizará Linux como sistema operativo y el acceso a la aplicación Web será posible mediante el uso de un navegador de Internet, que soporte el estándar HTML.

3.9.1.4 Interfaces de Comunicación

La comunicación entre los usuarios, la aplicación Web y los dispositivos de monitoreo se la hará en mediante el protocolo HTTP. El dispositivo de monitoreo requiere tener datos móviles para la conexión a Internet.

3.9.2 Funciones

En el Cuadro 7 se detallan las funciones que tendrán, tanto la aplicación Web como el dispositivo de monitoreo. Se han tomado en cuenta los requerimientos específicos anteriormente expuestos en el Cuadro 6.

Cuadro 7

Requerimientos Funcionales

RE	Nombre	Descripción
RE01	Consultar información	La aplicación Web presentará en un mapa la ubicación geográfica de cada dispositivo de monitoreo junto con la información sobre el estado del tráfico de cada dispositivo en forma de grafico de línea en tiempo rea
RE02	Capturar y Procesar las lecturas	El software que se ejecuta en el dispositivo de monitoreo programado en la plataforma Arduino, capturará y procesará la información que genera el sensor laser. Cada placa Arduino tendrá su propio código para lograr este objetivo.
RE03	Transmitir la información	El dispositivo de monitoreo mediante el módulo GSM transmitirá la información, captada por el sensor láser, hacia la aplicación Web haciendo uso de los protocolos de comunicación de datos móviles.
RE04	Almacenar información	Permite almacenar la información recolectada y transmitida por el dispositivo de monitoreo.
RE05	Iniciar sesión	El usuario avanzado deberá identificarse para acceder a Pentaho
RE06	Análisis de información con minería de datos	Permite generar gráficos de los datos del tráfico recolectados en función de distintos factores.

3.9.3 Requisitos No Funcionales

La siguiente sección tratará sobre los atributos no funcionales que deberán cumplir el prototipo Web y el software que controla los dispositivos de monitoreo. Los factores que se detallaran a continuación permitirán definir el rendimiento, seguridad, fiabilidad y disponibilidad del prototipo que se creara.

3.9.3.1 Requisitos de Rendimiento

Tanto la aplicación Web, como el software que se ejecuta en el dispositivo de monitoreo, se desarrollaran de forma óptima garantizando un correcto desempeño. Las interacciones que se realizarán con la base de datos deberán estar diseñados para que la respuesta sea rápida y no comprometa el rendimiento del software. Es importante recalcar que al usar el módulo GSM para el envío de datos se depende mucho de la eficacia del proveedor de Internet.

3.9.3.2 Seguridad

Los visitantes sólo podrán consultar la información pública que estará disponible en el portal de la aplicación de monitoreo, información que deberá estar dispuesta en una interfaz intuitiva.

En cuanto a la seguridad en la transmisión de información desde los dispositivos de monitoreo hacia la aplicación web, se ha creado una URL específica que permitirá identificar el origen de los datos y su composición.

3.9.3.3 Fiabilidad

Todas las funciones identificadas en este proyecto deberán operar de manera fiable. El almacenamiento de la información en la base de datos deberá garantizar la integridad y correcta relación entre los datos, esto facilitará su consulta posterior.

3.9.3.4 Disponibilidad

La aplicación web será diseñada para estar siempre disponible a los usuarios visitantes y dispositivos de monitoreo, Pentaho deberá estar disponible para usuarios

avanzados. Esta condición dependerá del servidor Web donde se encuentre alojada y del servicio de Internet.

CAPÍTULO IV

DISEÑO DEL PROTOTIPO

En función de los requerimientos identificados en el capítulo 3 para el presente proyecto, se diseñará a continuación los diagramas de base de datos, casos de uso, secuencia, paquetes, arquitectura y dispositivo electrónico que permitirá crear el prototipo del sistema y dispositivo de monitoreo.

4.1 Diseño de la base de datos

Con el objetivo de optimizar el rendimiento y en base de los requerimientos identificados, se ha diseñado una base de datos conformada por una tabla en la que se representen los atributos necesarios. Al contener una gran cantidad de datos esta solución se convierte en una forma rápida de cumplir con la velocidad de procesamiento sin dejar de lado las propiedades inherentes de la realidad.

La figura 11 ilustra la tabla utilizada en la que el dispositivo de monitoreo depositará sus datos a través de la aplicación web de inserción de datos y también será consumida por la aplicación web de monitoreo en tiempo real del tráfico.

<u>id</u>	<pi>	Integer	<M>
Estacion		Integer	
Fecha		Date & Time	
Estado_Clima		Characters (32)	
Temperatura		Float	
Medicion		Integer	
Carril		Integer	
Identifier_1	<pi>		

Figura 11. Diseño de la base de datos.

4.2 Diagramas de casos de uso

Un caso de uso es una historia en la que se describe el modo en el que un usuario (que puede participar con distintos roles), interactúa con el sistema ante circunstancias específicas. La historia puede ser descrita de distintas formas pero sin distinción un caso de uso muestra el software desde el punto de vista del usuario final (Pressman, 2005).

La Figura 12 identifica los Casos de Uso que se determinaron en el análisis del proyecto, interactuando de la siguiente forma: el conteo de vehículos se obtiene de la placa Arduino A, que interpreta la distancia enviada por el sensor láser mediante el algoritmo desarrollado en la plataforma Arduino, y determina el número de vehículos que transitan por el carril monitoreado. La placa Arduino B solicita el conteo a la placa Arduino A. Cada 5 minutos envía esta información mediante la red GSM/GPRS. El dispositivo de monitoreo, mediante la red GSM/GPRS se conecta con la aplicación web que registra el conteo y el clima en la base de datos, y esta a su vez, es consultada por en el sistema de monitoreo que permite al usuario obtener las estadísticas en tiempo real.

Mediante Pentaho se obtiene el análisis de los datos recolectados, el cual es útil en la tomar decisiones. Para poder explicar de mejor manera cada caso de uso se los ha subdividido y presentado con mayor detalle.

Para que la información pueda ser observada primero se requiere del ingreso al sitio web. El usuario accede al sitio web a través de un navegador ingresando la URL y el sistema se encarga de desplegar los elementos necesarios en la interfaz. En la Figura 13 se observa el caso de uso Visitar el sitio web.

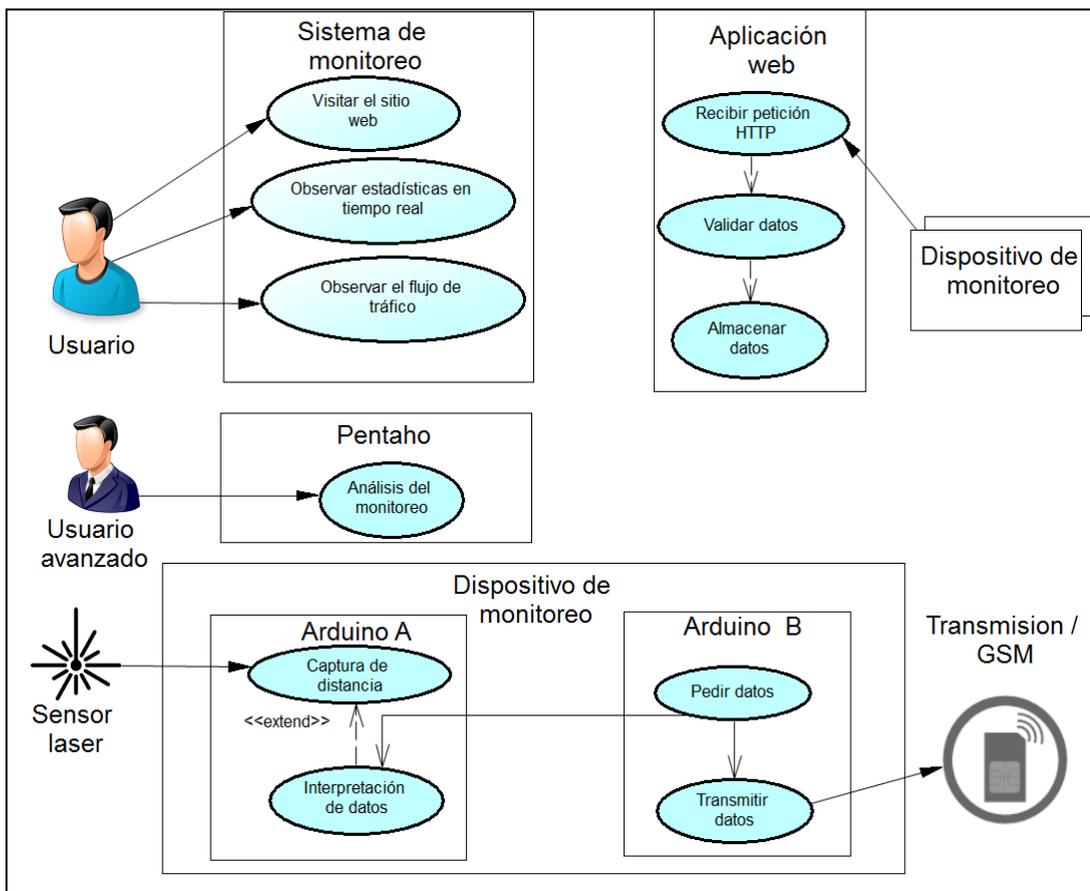


Figura 12. Casos de uso

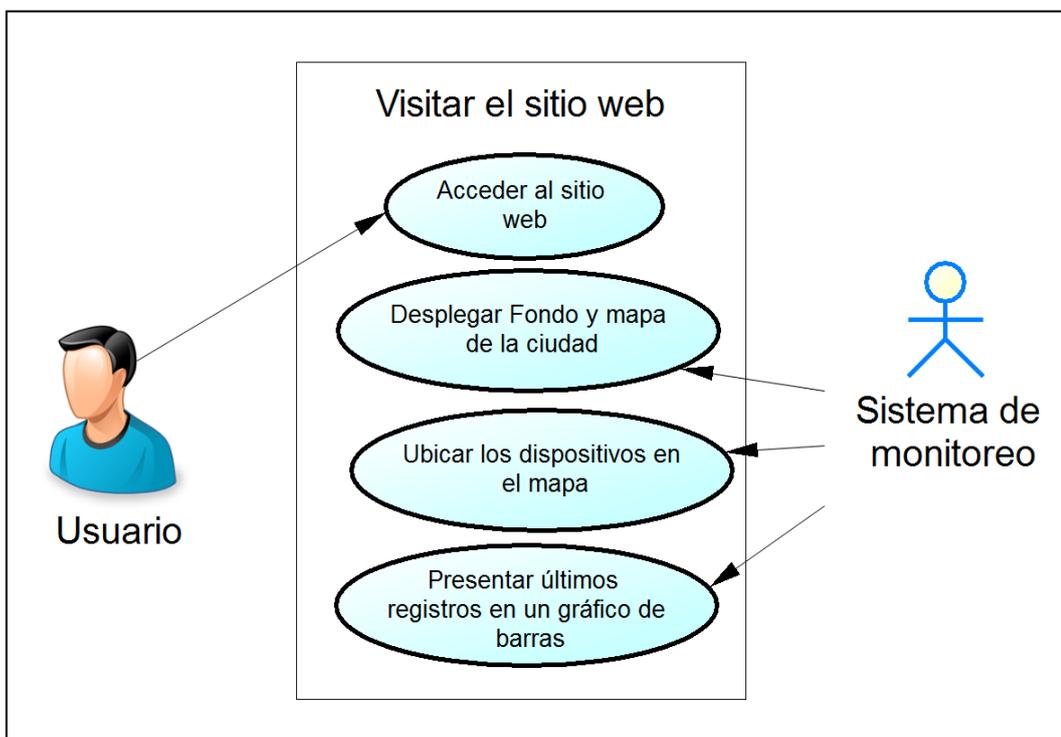


Figura 13. Caso de uso: Visitar el sitio web

En la figura 14 se ilustra la observación de estadísticas. El usuario se dispone a visualizar los datos presentados en los gráficos, mientras que el sistema actualizará los datos automáticamente cada 5 minutos.



Figura 14. Caso de uso: Observar estadísticas en tiempo real

El usuario para observar el último registro de flujo de tráfico captado por cada dispositivo se dirige al mapa y selecciona el dispositivo que prefiera. El sistema se encarga de actualizar los datos en el mapa cada 5 minutos, caso de uso ilustrado en la figura 15.



Figura 15. Caso de uso: Observar el flujo de tráfico

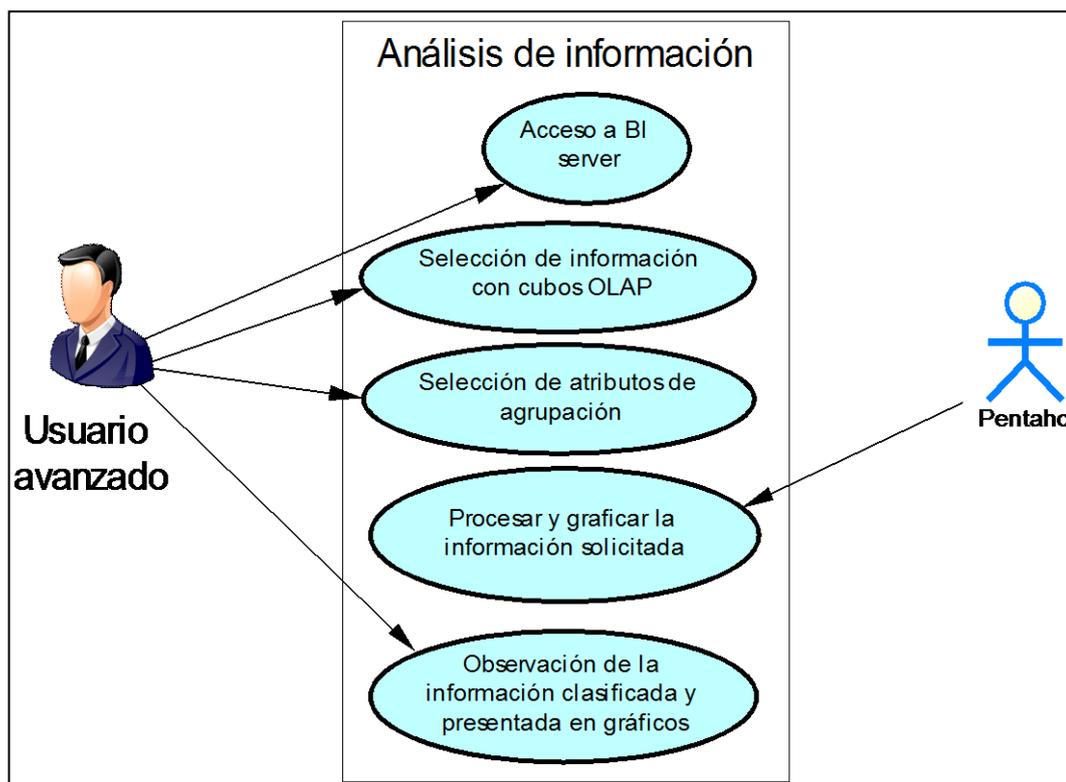


Figura 16. Caso de uso: Análisis de información

Un tipo especial de usuario que es el avanzado; tendrá la capacidad de acceder a la información adquirida a través del proceso minería de datos e inteligencia de negocios con la herramienta BI server de Pentaho.

El usuario deberá ingresar a BI server y seleccionar la opción de presentar información mediante cubos OLAP. A continuación el usuario tendrá la capacidad de agrupar, filtrar y ordenar los datos en función de los atributos de los elementos. Pentaho se encargará de procesar la solicitud y presentarla al usuario. Finalmente el usuario podrá observar el resultado de la consulta. Este caso de uso de ilustra en la figura 16.

Los siguientes casos de uso son diferentes debido a que los actores que participan son elementos electrónicos o de software por lo que no hay una acción humana directa implicada.

En la figura 17 se ilustra el caso de uso de captura de datos por parte del dispositivo de monitoreo, desde la medición de la distancia adquirida con el sensor láser hasta el envío de datos por la red 3G/GPRS.

Del mismo modo en la figura 18 se muestra la manera en que interactúan el dispositivo de monitoreo, la aplicación web de captura de datos y la api de clima. La aplicación web se encarga de juntar esta información y almacenarla en una base de datos.

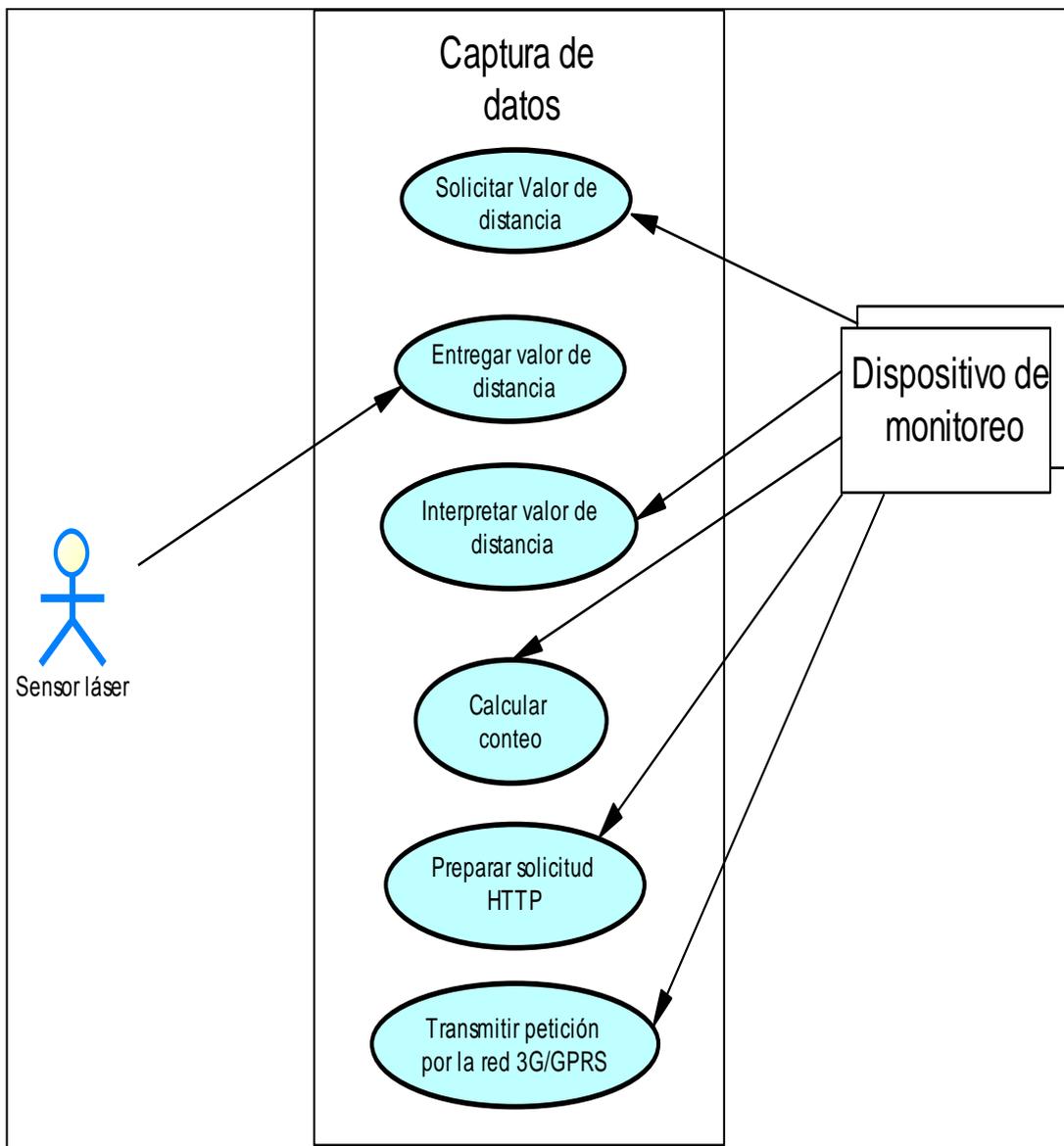


Figura 17. Caso de uso: Captura de datos

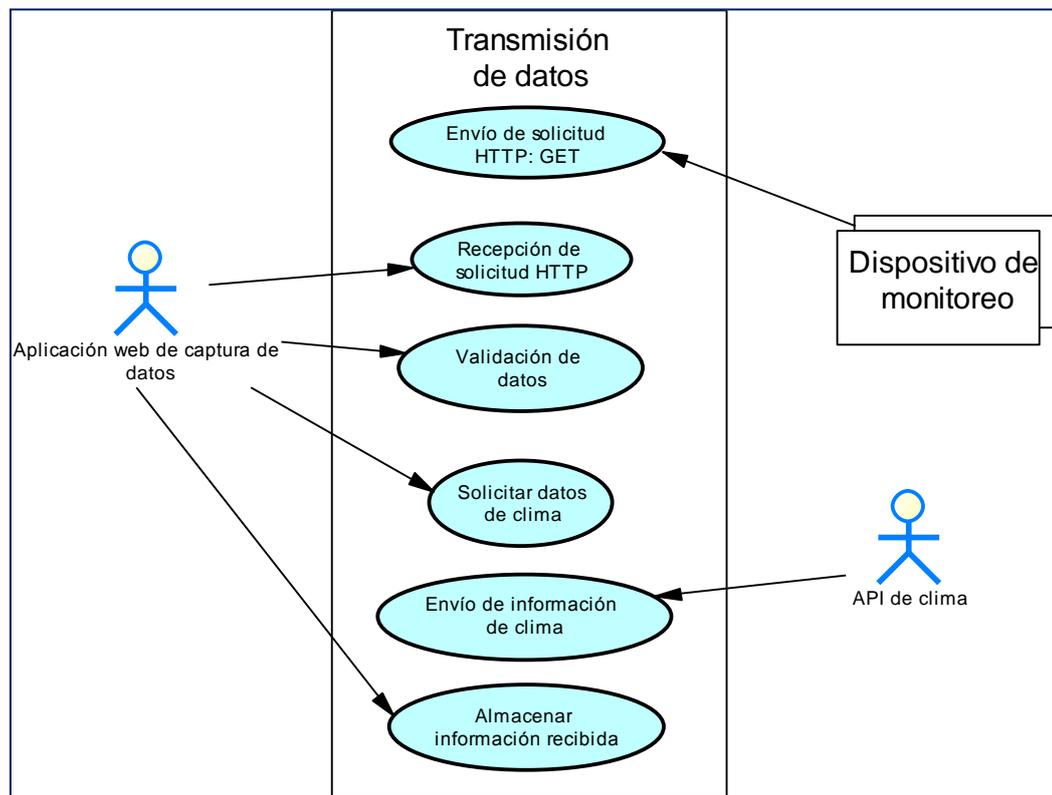


Figura 18. Caso de uso: Transmisión de datos

4.3 Diagramas de secuencia

Los diagramas de secuencia resaltan el orden temporal que tienen los mensajes que se intercambian entre los distintos elementos del sistema (Booch, 1999). Se pueden distinguir 8 actores que serán los que participan en los siguientes diagramas que representarán distintos escenarios de uno o más casos de uso.

En la figura 19 podemos observar la secuencia de mensajes existente en los casos de uso generados para que el usuario pueda observar la información. En este se juntan los casos de uso de visita al portal, observación de estadísticas en tiempo real y flujo del tráfico. En la figura 20 se ilustra el caso de uso análisis de información en el que interactúan el usuario avanzado y Pentaho.

En la figura 21 se muestra de manera detallada el modo en que el dispositivo de monitoreo interactúa internamente entre sus placas Arduino, junto con su módulo de expansión GSM y el sensor de distancia láser incorporado.

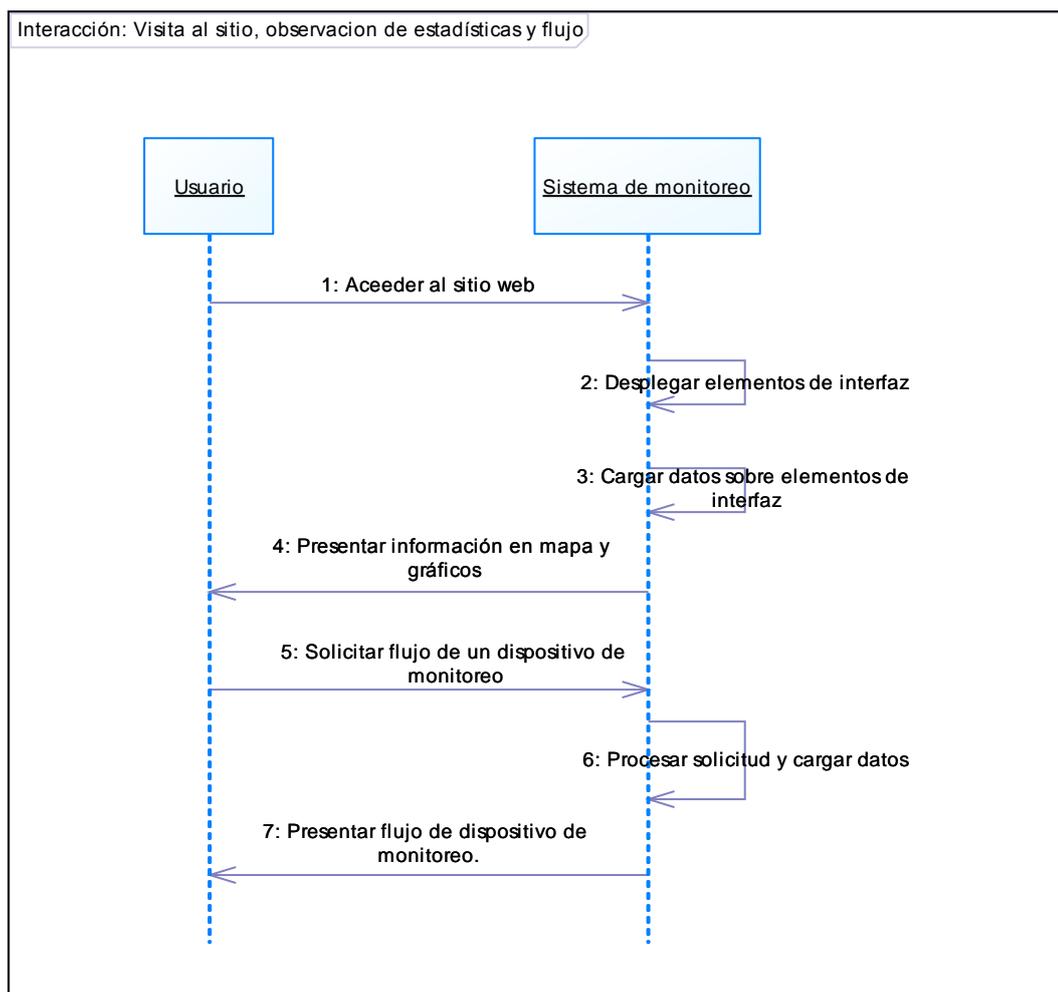


Figura 19. Escenario: Visita al sitio y observación de estadísticas y flujo

El modo en que interactúan las partes que conforman el dispositivo, se puede presentar de la siguiente manera: 1. Inicio de la aplicación; 2. Instanciación de variables y objetos requeridos; 3. Conexión a la red celular; 4. Solicitud de distancia de la placa A al sensor láser; 5. Envío de la distancia del sensor hacia la placa A; 6. Interpretación del resultado para determinar si ha pasado un vehículo y añadirlo a una variable de conteo; 7. Cumplido un intervalo de cinco minutos la placa B solicita a la placa A el número de vehículos almacenado; 8. Envío de la cantidad de vehículos hacia la placa A; 9. Conformación de la solicitud HTTP hacia el servidor con los datos necesarios; 10. Envío la solicitud HTTP al servidor web Figura 21.

La figura 22 ilustra la interacción que se realiza en el proceso de transmisión de datos. Los actores que participan son el dispositivo de monitoreo, la aplicación web y la api del clima utilizada.

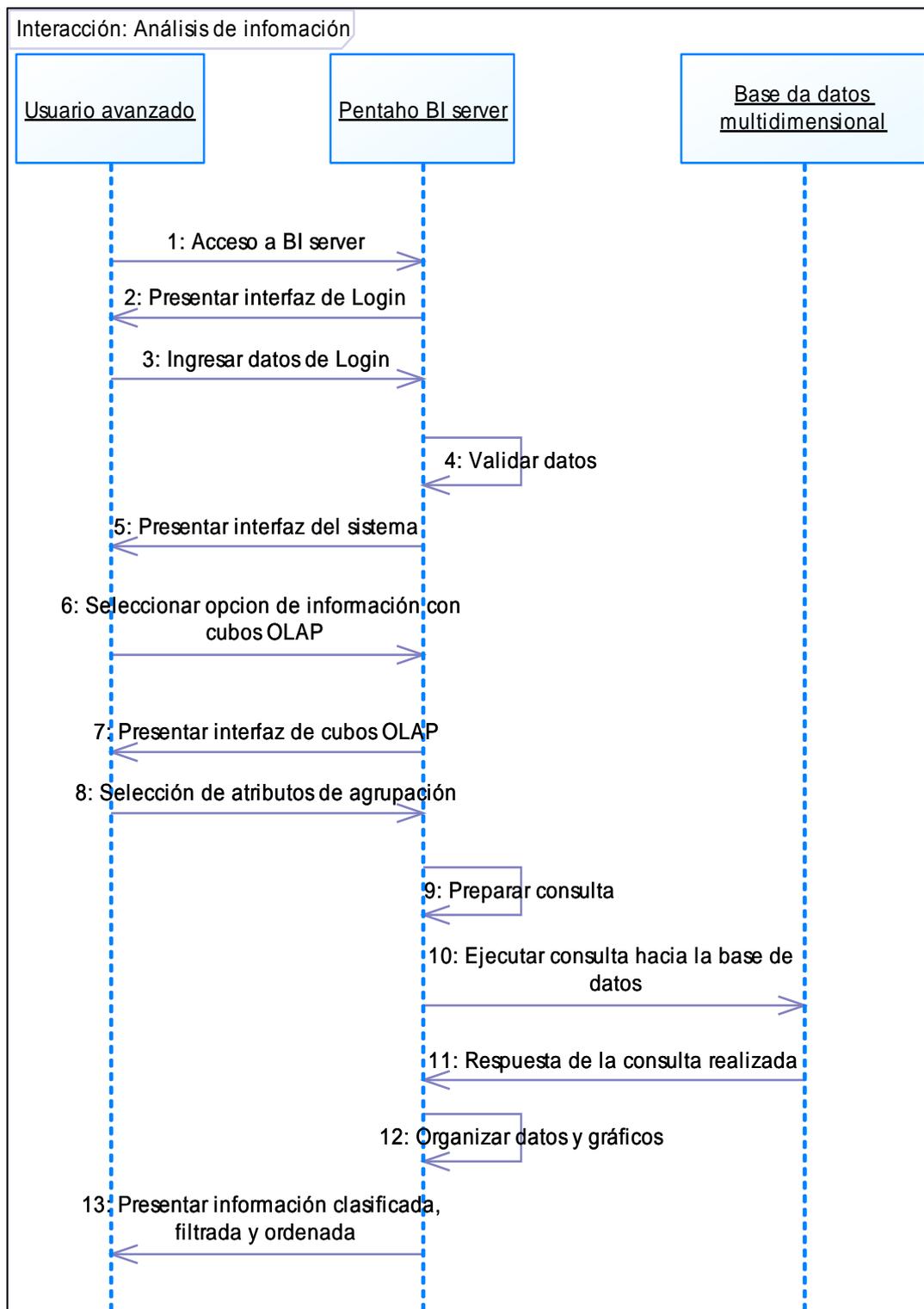


Figura 20. Escenario: Análisis de información

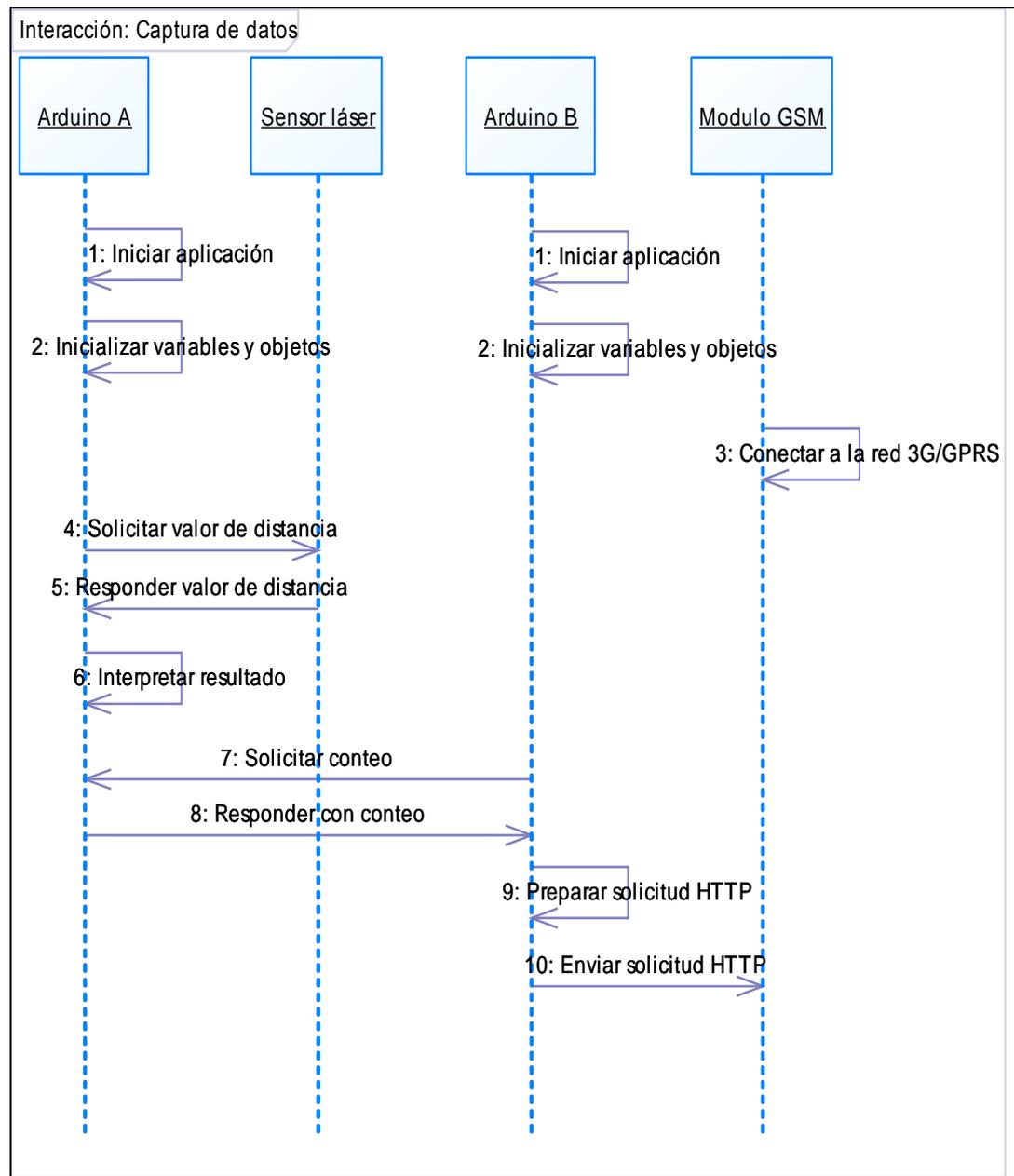


Figura 21. Escenario: Captura de datos

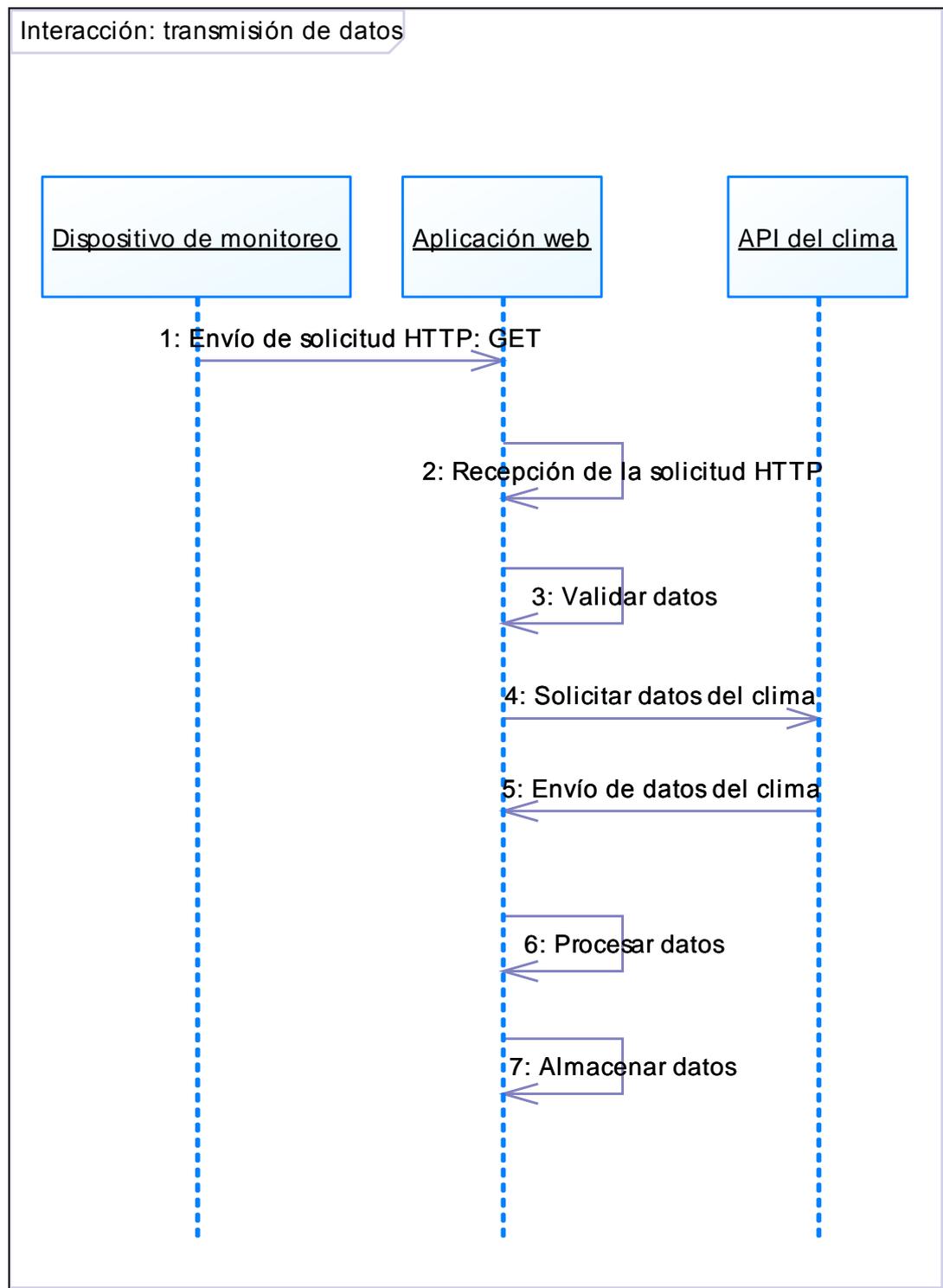


Figura 22. Escenario: Transmisión de datos

4.4 Diagrama de clases

El modelo de clases muestra las clases que conforman el sistema y sus relaciones implícitas (Booch, 1999). La organización en capas utilizada ha sido proporcionada

por el framework Laravel, el cual implementa el estándar modelo vista controlador (MVC) dentro de su estructura. El sistema tras haber optimizado la base de datos en una sola tabla, permite que la aplicación cuente con una sola clase en la capa modeloConteo.php que interactúa con la tabla del mismo nombre y se ubicada en el paquete Model. El controlador es la clase BusquedaController.php que implementa el método buscar() que se encarga de interactuar con la capa modelo y obtener datos de la base de datos. La vista index.php hereda su estructura la clase Master.php y es la que se presenta ante los usuarios para que obtengan la información del monitoreo. Otro elemento dividido es la clase Routes que se encarga de direccionar las peticiones que llegan al servidor conectando las vistas con sus respectivos controladores.

En la figura 23 se han presentado las clases de nuestra autoría, puesto que existen otras clases generadas automáticamente por el framework, sin embargo su función es genérica y no impacta en el comportamiento de la aplicación, razón por la cual no han sido incluidas.

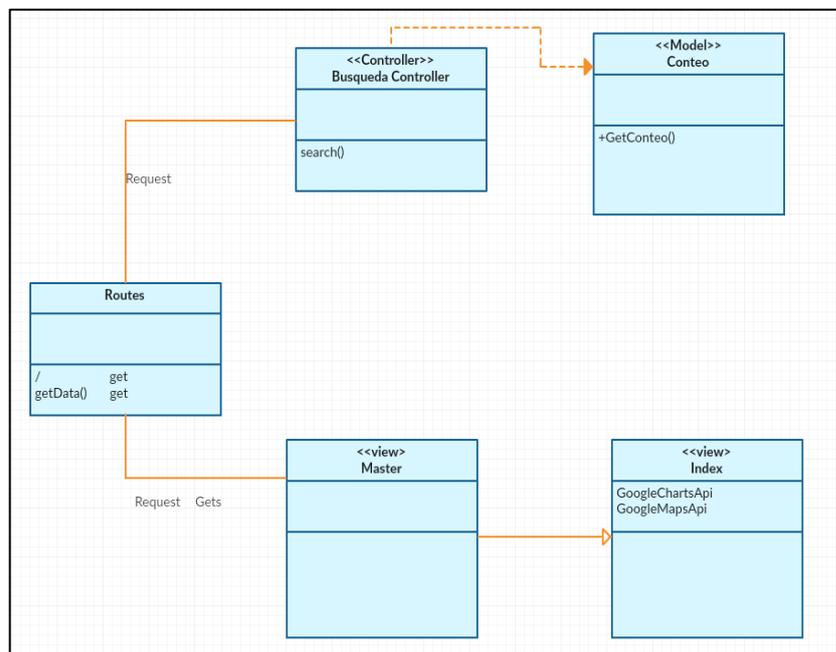


Figura 23. Diagrama de clases

4.5 Diagrama de paquetes

Los paquetes ofrecen un mecanismo general para la organización de los modelos o subsistemas agrupando elementos de modelado. De igual manera se enuncia que cada paquete representa un subsistema y que una clase tiene la capacidad de ser parte de dos paquetes por la importación a través de la relación de dependencias entre paquetes (Torres, 2002). En el diagrama de paquetes presentado forman parte los paquetes más representativos del framework Laravel, además, cabe mencionar que esta organización y muchas de las clases que son parte, han sido generadas de manera automática por Laravel, siendo Model, Controllers y Views los paquetes en los que se desarrolla la mayor parte de la aplicación, quedando los restantes como elementos complementarios para el funcionamiento y configuración de la aplicación Figura 24.

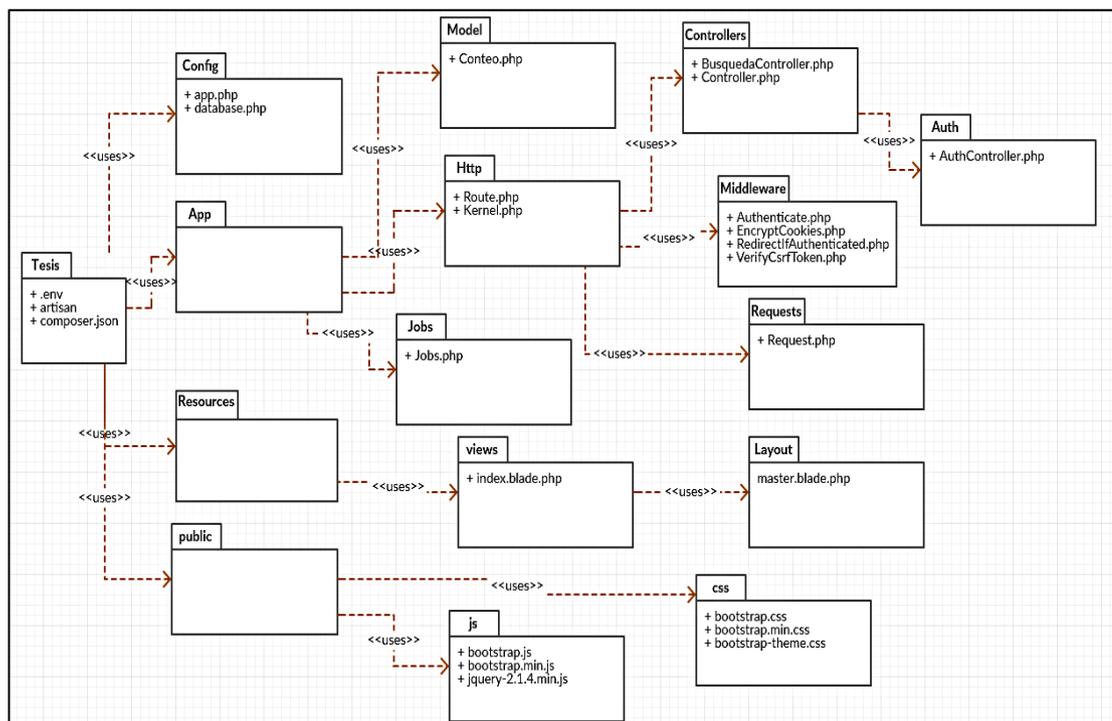


Figura 24. Diagrama de paquetes

4.6 Diagrama de arquitectura.

Un diagrama de arquitectura permite tener una visión completa del sistema que va a ser construido. Ilustra la estructura y organización de los componentes de software,

sus conexiones y propiedades (Pressman, 2005). En la figura 25, se ilustra la arquitectura del sistema desde una vista general de los componentes.

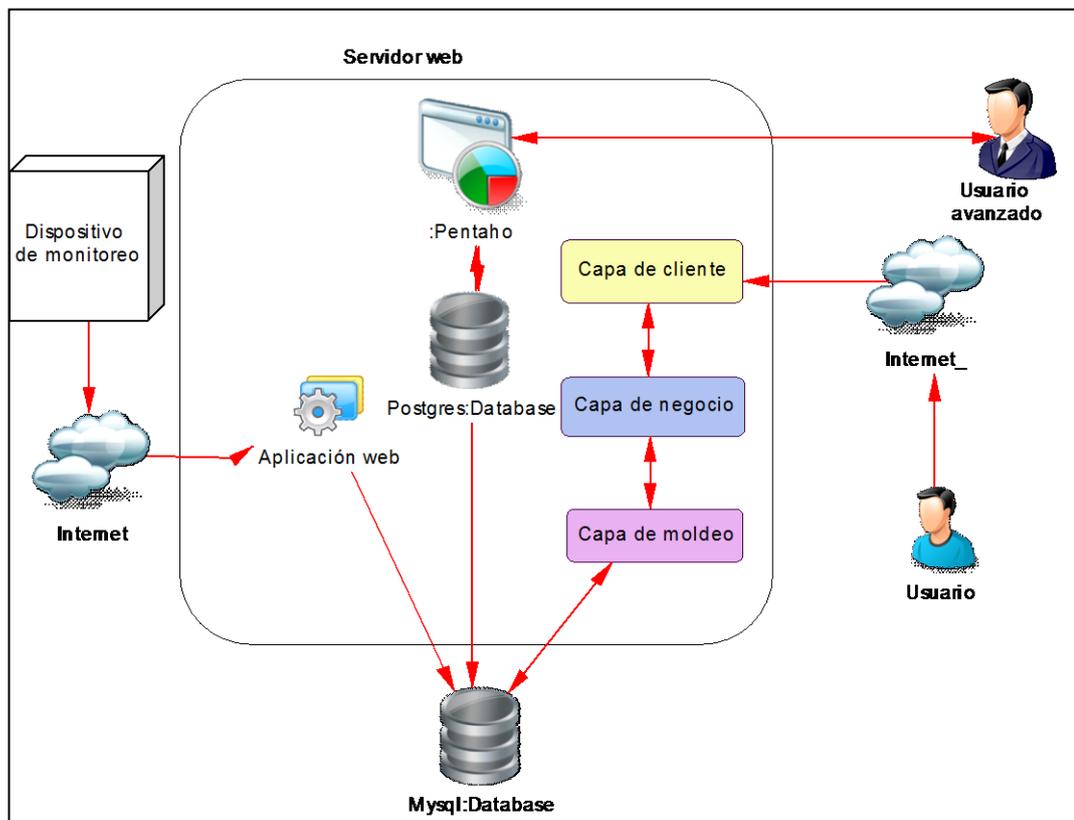


Figura 25. Diagrama de arquitectura

4.7 Dispositivo electrónico

El dispositivo desarrollado ha sido montado sobre dos placas Arduino UNO R3 que interactúan entre sí a modo Maestro - Esclavo y cada una con una función específica. La primera placa cuenta con el sensor de distancia basado en láser, Lidar Lite 2, que es controlado mediante la librería ofrecida por la empresa desarrolladora, y junto con un algoritmo inédito creado para la detección de vehículos que se encarga del conteo del tráfico. La segunda placa cuenta con el módulo de expansión GSM que a su vez, solicita la cantidad de autos que ha detectado la primera placa Arduino a través del protocolo I2C, durante los últimos cinco minutos. Luego envía esta medición junto con la información descriptiva propia de la estación a través de la red 3G/GPRS al servidor Web.

Para la alimentación de energía se utilizó una batería de 20000 mA repartida a través de dos puertos USB que mantendrán encendidas las dos placas Arduino incorporadas por un tiempo prolongado.

La figura 26 ilustra las partes se encuentran ensambladas. La primera placa Arduino cuenta con el módulo de expansión GSM y funciona con el chip SIM 900. A ésta, se la debe añadir un chip con tecnología GSM para que se conecte al proveedor de señal 3G/GPRS. La segunda placa Arduino, cuenta con el sensor de distancia láser que utiliza los pines SDA y SCL. Ambos se usan comunicación serial I2C que para su uso requiere que se interconecten los pines analógicos A4 y A5, así como que un pin de tierra.

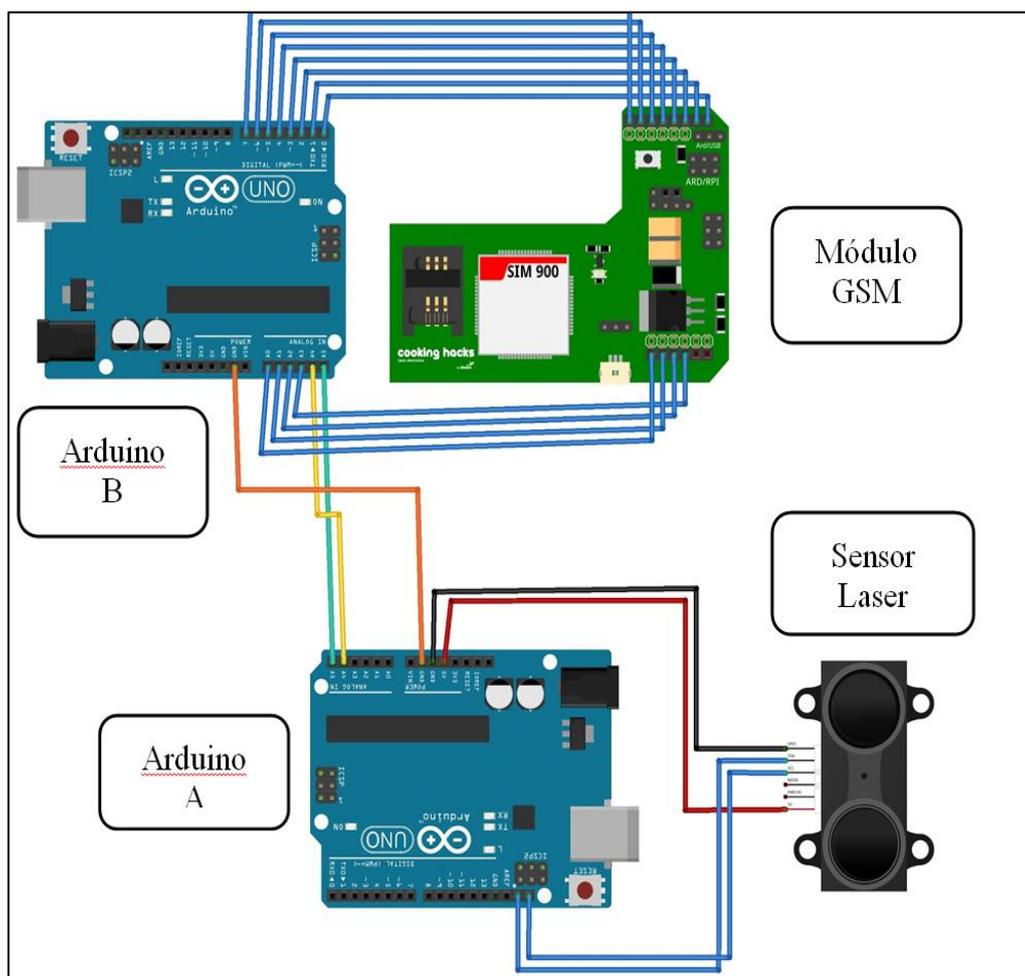


Figura 26. Diseño del dispositivo

4.8 Minería de datos e inteligencia de negocios.

Con el objetivo de extraer información detallada acerca del comportamiento del tráfico en el sitio de estudio, incorporando nuevas líneas de negocio o distintas soluciones al problema de la alta congestión vehicular. El módulo de inteligencia de negocios de Pentaho es BI server puesto que permite trabajar con cubos y bases de datos multidimensionales que a su vez incluye una variedad de herramientas de análisis que van desde reportes básicos hasta modelamiento predictivo.

La Figura 27 muestra el modelo en estrella de la base de datos multidimensional utilizada. Este modelo es un requerimiento para la formación de cubos que solicita BI server. Todos los datos capturados durante el período de dos semanas del estudio, se encuentran almacenados en el DBMS MySQL. Posterior a esto, los datos son copiados y transformados y cargados sobre la base de datos multidimensional que se encuentra en el DBMS Postgres.

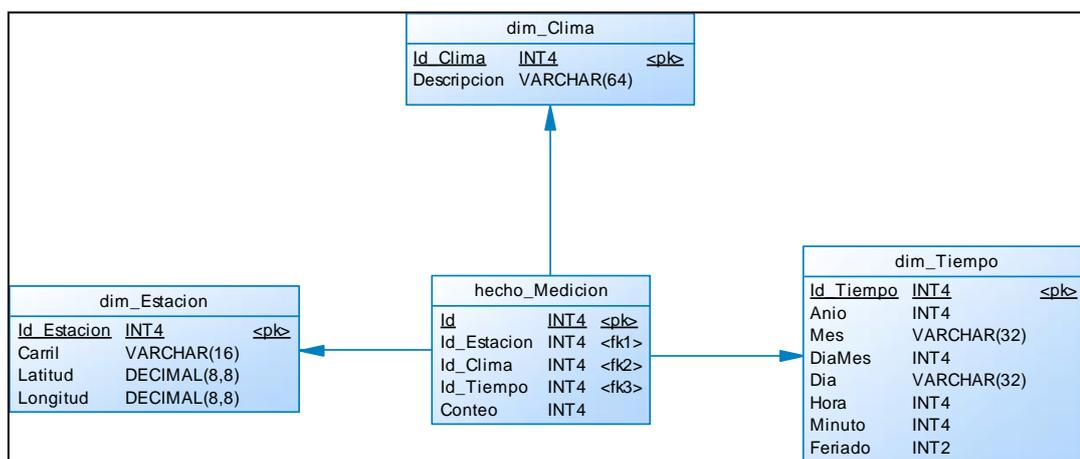


Figura 27. Modelo en estrella

CAPÍTULO V

PLANIFICACIÓN, DESARROLLO Y PRUEBAS

Con el objetivo de realizar la planificación, desarrollo y evaluación del proyecto, se ha utilizado una combinación entre dos metodologías ágiles que juntas aportan mayor valor: Scrum y Extreme programming (XP).

5.1 Planificación del proyecto utilizando la metodología Scrum

De acuerdo con la metodología Scrum, el punto de partida es la construcción del “ProductBacklog” o pila de producto, este se creó con base en los cinco requerimientos levantados en el tercer capítulo del presente documento y una actividad adicional que es la Toma de datos del tráfico en la avenida Simón Bolívar que se requiere para hacer el análisis de datos.

Se han tomado en cuenta los siguientes ítems para establecer un formato de documento que defina el proceso de construcción del ProductBacklog:

- **ID:** Identificador único.
- **Historia de usuario:** Descripción general del requerimiento
- **Estimación:** Tiempo necesario
- **Prioridad:** Es el grado de importancia que tendrá el ítem en relación al resto. Esto será definido por el propietario del producto o ProductOwner.
- **Criterio aceptación:** Descripción de la funcionalidad que se espera tenga el demo del producto.

El cuadro 8 detalla las seis primeras historias de usuario que fueron identificadas en el capítulo 3 del presente documento; cada historia engloba una serie de tareas que serán abordadas en la etapa de Sprints o iteraciones y se documentará en cada Spring Backlog. Cada tarea tendrá una priorización y una especificación de tiempo.

Cuadro 8

Pila de producto o ProductBacklog

ID	Historia de usuario	Estimación (semanas)	Prioridad	Criterio de aceptación
1	Capturar y procesar las lecturas	4	1	Ingresar a la plataforma de desarrollo de Arduino, conectar el dispositivo de monitoreo por medio del puerto USB y comprobar la información captada por el sensor láser.
2	Transmitir la información	1	2	Ingresar a la plataforma de desarrollo de Arduino, conectar el dispositivo electrónico con la computadora por medio del puerto USB y comprobar la conexión entre la placa Arduino y el módulo GSM. Verificar el código HTTP de respuesta del servidor de aplicaciones.
3	Almacenar Información	2	3	Ingresar a la aplicación PhpMyadmin MySql, comprobar la cantidad de lecturas almacenadas en la base de datos y que fueron procesadas por el Servicio Web del prototipo.
4	Consultar información	1	4	Ingresar al portal del prototipo de sistema de monitoreo y verificar la ubicación de los dispositivos de monitoreo en el mapa. Comprobar en el grafico la información de las lecturas registradas.
5	Toma de Datos	3	5	Se requiere realizar una toma de datos de dos semanas de la avenida Simón Bolívar para poder realizar un análisis de Datos.
6	Minería de datos e inteligencia de negocios.	1	6	Ingresar al módulo de Bi de Pentaho SaikuAnalytcs, iniciar sesión en la pantalla que se despliega; dar clic en new SaikuAnalytcs, elegir el cubo con el cual se trabaja y elegir los atributos de clasificación, orden o filtrado y finalmente generar los gráficos

Adicionalmente, la metodología también indica que el ProductBacklog o pila de producto deberá ser lo más específica posible para facilitar la planificación de cada iteración. Para esto se ha definido una escala tanto para el tiempo como la priorización de cada historia de usuario. Las escalas que se emplearán son:

- Escala de tiempo estimado: 1 a 4 semanas.
- Escala de prioridad: 1 a 6, siendo 6 el menos urgente.

Este proceso de refinamiento del ProductBacklog o pila de producto requiere la aplicación de varios criterios que ayudarán a priorizar y ordenar las historias de usuarios. Los criterios aplicables son:

- Cada ítem debe representar un incremento significativo a la funcionalidad del prototipo.
- El equipo de desarrollo debe ser capaz de construir cada ítem en un solo Sprint.
- Todos los participantes en el proyecto deben entender lo que se va a desarrollar para cada requerimiento.

Una vez organizado y priorizado el ProductBacklog, se inició la etapa de Sprints o iteraciones sucesivas. En esta etapa se llevó a cabo la planificación o Sprint Planning, donde los miembros del equipo de desarrollo escogieron una o un grupo de historias de usuario del ProductBacklog para desarrollarlo en una sola iteración. A este nuevo listado de tareas se lo conoce como pila de tareas o Sprint Backlog.

5.2 Estándares de programación.

Los estándares de código resultan importantes en cualquier proyecto de desarrollo, pero son especialmente importantes cuando muchos desarrolladores trabajan en el mismo proyecto. Los estándares de código ayudan a asegurar que el código tenga calidad y pueda ser mantenido. Además de que el uso de estándares nos permitirá ser objetivos a la hora de revisar código de otros, nos ayudará con la reusabilidad y compatibilidad del código con otros proyectos (ZEND FRAMEWORK, 2011).

La codificación del prototipo en el lado del servidor empleará las buenas prácticas de codificación del lenguaje PHP, publicadas por Zend en el 2011. El Cuadro 9 presenta las principales reglas de codificación que se utilizaron.

Cuadro 9.

Reglas de codificación

Regla	Ejemplo
Los nombres de las clases deben ser CamelCase, con la inicial en mayúscula	BusquedaController
Variables, propiedades, funciones y métodos. Estos nombres deberán ser camelCase, con la inicial en minúscula.	searchData() \$estacionNorteSur \$estacionSurNorte
Las sentencias de control if, elseif, foreach deben tener un espacio antes del paréntesis de apertura del condicional y un solo espacio después del paréntesis de cierre.	foreach(\$estacionNorteSur as \$estNs)

Fuente: adaptado de: (ZEND FRAMEWORK, 2011)

Para el desarrollo de interfaces de usuario del prototipo del sistema de monitoreo se utilizará las plantillas Blade de Laravel que están construidas en base a HTML5 y permiten crear con la ayuda de Bootstrap interfaces full responsive, es decir, se puede ver correctamente la aplicación en cualquier resolución, dispositivo móvil y en la web; también se utilizarán Javascript, el cual se conectará con la api de Google Maps y Google Charts, y permitirá tener la información actualizada cada 5 minutos.

5.3 Desarrollo y pruebas de la Iteración 1

Para la primera iteración se tomó la primera historia de usuario, Capturar y procesar las lecturas, esta tuvo una duración de 4 semanas como se especifica en el Cuadro 10.

Esta primera iteración contempló la creación del dispositivo de monitoreo y la programación de los algoritmos en las placas Arduino para lograr el objetivo del conteo de vehículos.

Cuadro 10.

Historias de usuario seleccionadas para la iteración 1

ID	Historia de usuario	Estimación (semanas)	Prioridad	Criterio de aceptación
1	Capturar y procesar las lecturas	4	1	Ingresar a la plataforma de desarrollo de Arduino, conectar el dispositivo de monitoreo por medio del puerto USB y comprobar la información captada por el sensor láser.

5.3.1 Sprint Backlog de la iteración 1

A continuación, se detallará las tareas que fueron necesarias para llegar a completar la historia de usuario seleccionada. El Cuadro 11 describe el conjunto de tareas requeridas, el ProductBacklog al que pertenece y la fecha de inicio y la de entrega de cada tarea.

Cuadro 11.

Sprint Backlog de la iteración 1

Sprint 1 Inicio 3-01-2016 Duración 4 semanas			3 - 9 Enero	10-16 Enero	17-23 Enero	24-30 Enero
ID	Tarea	ProductBacklog				
1	Investigación de sensores que midan distancia y envío de datos por GSM	1	X	X		
2	Diseño del dispositivo electrónico	1			X	
3	Codificación de las placas Arduino	1			X	
4	Implementación del Diseño	1				X
5	Pruebas del dispositivo	1				X

5.3.2 Revisión y seguimiento de la iteración 1

Para realizar la revisión del Sprint se requirió enlistar las tareas y mostrar el estado de cada una, esto con el objetivo de medir el avance de la historia de usuario seleccionada. El Cuadro 12 detalla las tareas en el Sprint.

Cuadro 12.

Tareas completadas de la iteración 1

Sprint 1 Numero de Tareas:4			
ID	Tarea	Iteración	Estado
1	Investigación de sensores que midan distancia y envío de datos por GSM	1	Completado
2	Diseño del dispositivo electrónico	1	Completado
3	Codificación de las placas Arduino	1	Completado
4	Implementación del Diseño	1	Completado
5	Pruebas del dispositivo	1	Completado

5.3.3 Pruebas de la iteración 1

En la figura 26 se puede visualizar el diseño del dispositivo de monitoreo. La figura 29 permite visualizar el prototipo del dispositivo de monitoreo terminado.

5.3.4 Demo Iteración 1

El demo de la primera iteración presenta el prototipo del dispositivo de monitoreo. La figura 26 muestra los componentes del prototipo y la figura 28 muestra el prototipo implementado.

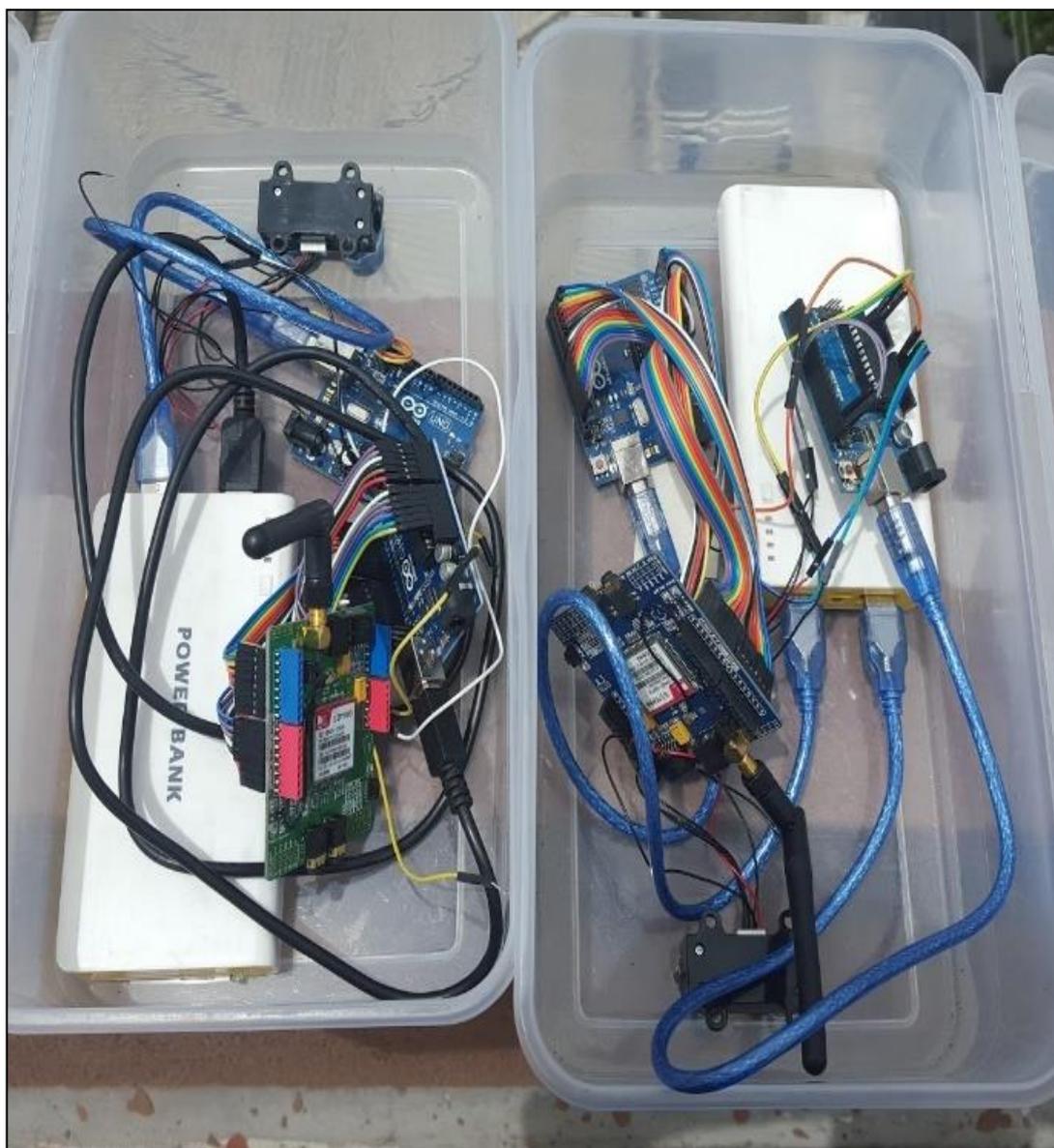


Figura 28. Prototipo del dispositivo de monitoreo

5.4 Desarrollo y pruebas de la Iteración 2

Para la segunda iteración se tomaron la segunda y tercera historia de usuario, las mismas que se encargarán de transmitir y almacenar la información; la estimación de tiempo para esta parte del proyecto fue de tres semanas. El Cuadro 13 presenta las historias de usuario elegidas que fueron contempladas en la ejecución de la segunda iteración.

Cuadro 13

Historias de usuario seleccionadas para la iteración 2

ID	Historia de usuario	Estimación (semanas)	Prioridad	Criterio de aceptación
2	Transmitir la información	1	2	Ingresar a la plataforma de desarrollo de Arduino, conectar el dispositivo electrónico con la computadora por medio del puerto USB y comprobar la conexión entre la placa Arduino y el módulo GSM. Verificar el código HTTP de respuesta del servidor de aplicaciones.
3	Almacenar Información	2	3	Ingresar a la aplicación PhpMyadmin de MySQL, comprobar la cantidad de lecturas almacenadas en la base de datos y que fueron procesadas por el Servicio Web del prototipo.

5.4.1 Sprint Backlog iteración 2

Para la segunda iteración al igual que en la primera, se requirió definir las tareas, definir las fechas en las que serán cumplidas e identificar el ProductBacklog al que pertenecen, adicionalmente se seleccionaron las historias de usuario que corresponden a esta iteración, las cuales fueron: transmitir y almacenar información. El Cuadro 14 detalla el Sprint Backlog de la iteración 2.

La tareas que se identificó en esta iteración son: consultar la trasmisión de datos por el módulo GSM, codificación de la petición por HTTP en la placa Arduino que controla el módulo GSM, pruebas de envío de la petición HTTP del Arduino, diseño de la base de datos, implantación de la base de datos, codificación de la aplicación web que recepta la información de la placa Arduino, integración con la api de clima

y finalmente codificación del almacenamiento en la base de datos de los datos receptados. Estas actividades fueron estimadas en un lapso de 3 semanas comprendidas desde el 31 de enero al 20 de febrero del 2016.

Cuadro 14.

Sprint Backlog de la iteración 2

Sprint 2			31 - 6 Febrero	7-13 Febrero	14-20 Febrero
Inicio 31-01-2016					
Duración 3 semanas					
ID	Tarea	ProductBacklog			
1	Consultar la transmisión de datos por el módulo GSM	2	X		
2	Codificación de la petición por HTTP en la placa Arduino que controla el módulo GSM	2	X		
3	Pruebas de Envío de la petición HTTP del Arduino	2		X	
4	Diseño de la base de datos	3		X	
5	Implantación de la base de datos	3		X	
6	Codificación de la aplicación web que recepta la información de la placa Arduino	3			X
7	Integración con la api de clima	3			X
8	Codificación del almacenamiento en la base de datos de los datos receptados	3			X

5.4.2 Revisión y seguimiento de la iteración 2

Para realizar la revisión del segundo Sprint se requirió enlistar las tareas y mostrar el estado de cada una y la iteración a la que pertenecen, esto con el objetivo de medir el avance de las historias de usuarios seleccionadas. El Cuadro 15 detalla las tareas en el Sprint.

Cuadro 15.

Tareas completadas de la iteración 2

Sprint 2 Numero de Tareas:8			
ID	Tarea	Iteración	Estado
1	Consultar la trasmisión de datos por el módulo GSM	2	Completado
2	Codificación de la petición por HTTP en la placa Arduino que controla el módulo GSM	2	Completado
3	Pruebas de Envío de la petición HTTP del Arduino	2	Completado
4	Diseño de la base de datos	2	Completado
5	Implantación de la base de datos	2	Completado
6	Codificación de la aplicación web que recepta la información de la placa Arduino	2	Completado
7	Integración con la api de clima	2	Completado
8	Codificación del almacenamiento en la base de datos de los datos receptados	2	Completado

5.4.3 Pruebas de la iteración 2

La figura 29 permite observar el modelo de base de datos diseñado e implementado en la nube y la figura 30 permite evidenciar las pruebas realizadas sobre la avenida 12 de octubre, la figura 31 permite conocer el prototipo del dispositivo de monitoreo funcionando y la figura 32 muestra el lugar donde se posicionó el prototipo del dispositivo de monitoreo, en dicha imagen se puede observar un vehículo pasando. Cabe recalcar que la prueba consistía en contar el

número de vehículos que pasaban bajo este punto de control y comparar con la lectura obtenida en la base de datos en la nube figura 34.

<u>id</u>	<pi>	Integer	<M>
Estacion		Integer	
Fecha		Date & Time	
Estado_Clima		Characters (32)	
Temperatura		Float	
Medicion		Integer	
Carril		Integer	
Identifier_1	<pi>		

Figura 29. Diseño de la base de datos.



Figura 30. Evidencia de pruebas realizadas sobre la 12 de Octubre

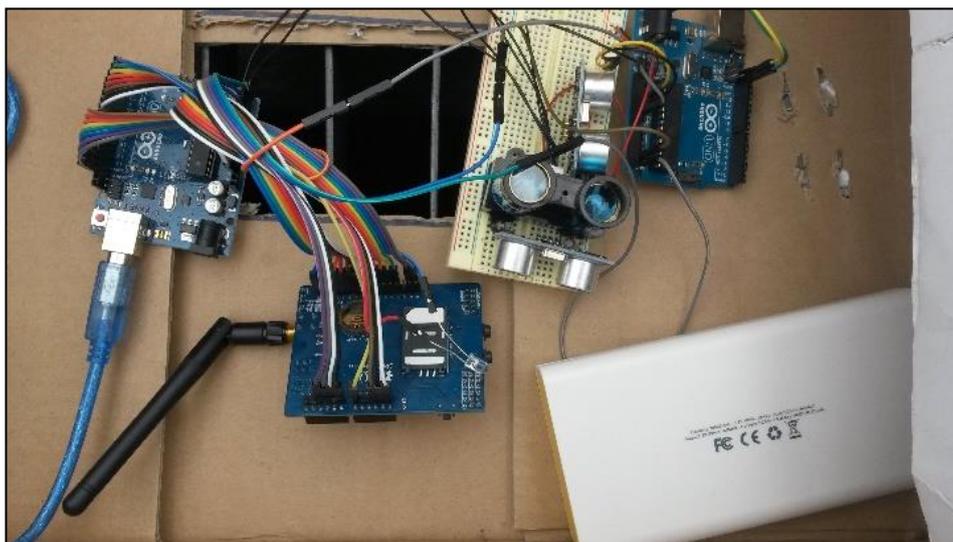


Figura 31. Dispositivo de monitoreo pruebas



Figura 32. Evidencia de pruebas realizadas avenida 12 de Octubre, lugar donde se posicionó el dispositivo de monitoreo

5.4.4 Demo iteración 2

La figura 33 permite observar la base de datos implementada en la nube. En la figura 34 se puede verificar los datos enviados por el dispositivo de monitoreo junto con los datos de la api del clima recolectados en la aplicación web.

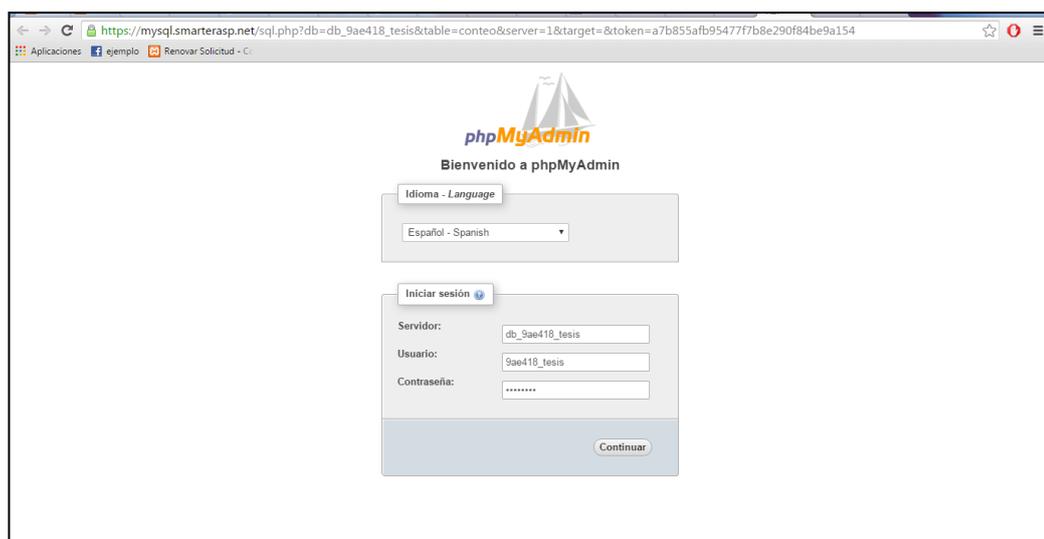


Figura 33. Base de datos implementada en la nube

Mostrando filas 0 - 24 (total de 3029. La consulta tardó 0.0091 segundos.)

```
SELECT * FROM `conteo`
```

Número de filas: 25 Filtrar filas: Buscar en esta tabla

Ordenar según la clave: Ninguna

	id	Estacion	Fecha	Estado_Clima	Temperatura	Medicion	Carril
<input type="checkbox"/>	260		2 2015-11-23 06:01:13	nubes dispersas	15.222	0	1
<input type="checkbox"/>	259		1 2015-11-23 06:01:03	nubes dispersas	15.222	2	2
<input type="checkbox"/>	258		1 2015-11-23 06:01:01	nubes dispersas	15.222	1	1
<input type="checkbox"/>	257		2 2015-11-23 06:00:16	nubes dispersas	15.222	14	2
<input type="checkbox"/>	256		2 2015-11-23 06:00:14	nubes dispersas	15.222	7	1
<input type="checkbox"/>	255		1 2015-11-23 06:00:01	nubes dispersas	15.222	9	2
<input type="checkbox"/>	254		1 2015-11-23 06:00:00	nubes dispersas	15.222	4	1
<input type="checkbox"/>	253		2 2015-11-23 05:59:15	llovuvia débil	8.616	7	2
<input type="checkbox"/>	252		2 2015-11-23 05:59:13	llovuvia débil	8.616	6	1
<input type="checkbox"/>	251		1 2015-11-23 05:59:03	llovuvia débil	8.616	3	2
<input type="checkbox"/>	250		1 2015-11-23 05:59:01	llovuvia débil	8.616	14	1
<input type="checkbox"/>	249		2 2015-11-23 05:58:16	llovuvia débil	8.616	4	2
<input type="checkbox"/>	248		2 2015-11-23 05:58:14	llovuvia débil	8.616	4	1

Figura 34. Datos recolectados por el dispositivo de monitoreo y almacenados por la aplicación web

5.5 Desarrollo y pruebas de la Iteración 3

Para abordar la tercera iteración se tomó la historia de usuario número 4 que corresponde a la consulta de información, esta historia de usuario se refiere a la aplicación web que se encargará de mostrar los datos obtenidos por el dispositivo de monitoreo en una aplicación full responsive para el usuario final. El cuadro 16 muestra la historia de usuario seleccionada para esta iteración.

Cuadro 16.

Historia de usuario seleccionada para la iteración 3

ID	Historia de usuario	Estimación (semanas)	Prioridad	Criterio de aceptación
4	Consultar información	1	4	Ingresar al portal del prototipo del sistema de monitoreo y verificar la ubicación de los dispositivos de monitoreo en el mapa. Comprobar en el grafico la información de las lecturas registradas.

5.5.1 Sprint Backlog iteración 3

A continuación se mostrará las tareas necesarias que se definieron para cumplir con esta iteración 3, la cual tiene un tiempo estimado de una semana, el cuadro 17 detalla las tareas necesarias.

Cuadro 17.

Sprint Backlog iteración 3

Sprint 3			
Inicio 21-02-2016			
Duración 1 semanas			
			21-27 Febrero
ID	Tarea	ProductBacklog	
1	Diseño de la interfaz para la visualización de información	4	x
2	Codificación del modelo de datos	4	x
3	Codificación del Controlador para la obtención de datos de cada estación	4	x
4	Integración con la api de Google Maps para el posicionamiento en el mapa de los dispositivos de monitoreo	4	x
5	Integración con la api de Google Analytics para generar gráficos de línea con las 10 últimas lecturas del dispositivo de monitoreo	4	x
6	Creación de la interfaz gráfica full responsive usando Bootstrap	4	X
7	Auto-actualización de los gráficos para tener información en tiempo real con JavaScript	4	X

5.5.2 Revisión y seguimiento iteración 3

Para verificar el avance de la iteración 3 se listarán a continuación las tareas propuestas y se especificará como se realizó el avance de cada cuadro 18. Así se evaluará el nivel de avance de la iteración 3.

Cuadro 18.

Tareas completadas de la iteración 3

Sprint 3 Numero de Tareas:7			
ID	Tarea	Iteración	Estado
1	Diseño de la interfaz para la visualización de información	3	Completado
2	Codificación del modelo de datos	3	Completado
3	Codificación del Controlador para la obtención de datos de cada estación	3	Completado
4	Integración con la api de Google Maps para el posicionamiento en el mapa de los dispositivos de monitoreo	3	Completado
5	Integración con la api de Google Analytics para generar gráficos de línea con las 10 últimas lecturas del dispositivo de monitoreo	3	Completado
6	Creación de la interfaz gráfica full responsive usando Bootstrap	3	Completado
7	Auto-actualización de los gráficos para tener información en tiempo real con JavaScript	3	Completado

5.5.3 Pruebas de la iteración 3

Para esta iteración se aplicaron las pruebas unitarias para verificar el correcto funcionamiento del sistema de monitoreo, la figura 35 permite evidenciar la programación de la prueba unitaria y el resultado obtenido.

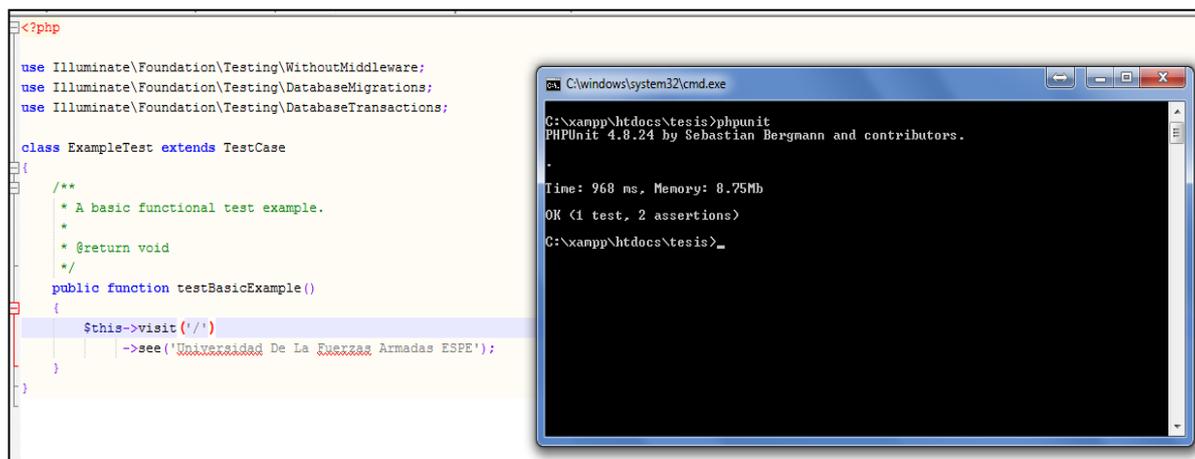


Figura 35. Prueba unitaria del sistema.

5.5.4 Demo iteración 3

El demo de esta iteración obtuvo como resultado el sistema de monitoreo que permite visualizar en tiempo real el estado del tráfico mediante gráficos de línea. En la figura 38 se puede observar el aplicativo en un dispositivo móvil, y en la figura 36 para la web.



Figura 36. Sistema de monitoreo en web



Figura 37. Sistema de monitoreo en móviles Parte A.

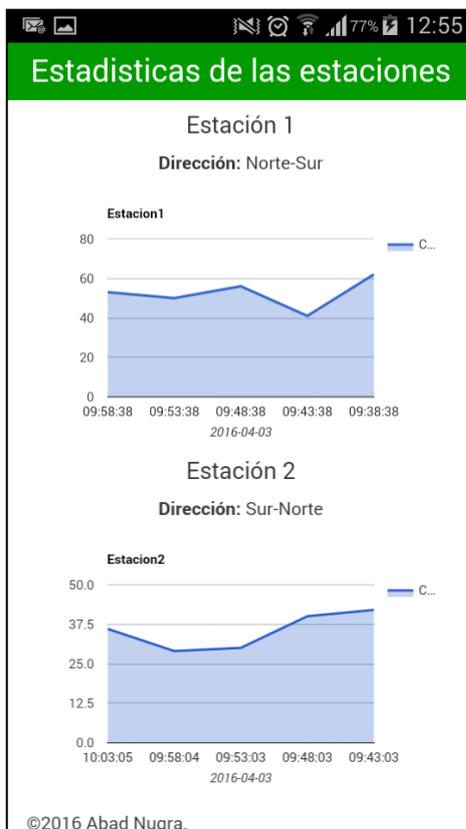


Figura 38. Sistema de monitoreo en móviles Parte B.

5.6 Desarrollo y evaluación de la Iteración 4

Para la cuarta iteración se tomó únicamente la historia de usuario número 5 que corresponde a la toma de datos, que se realizó en un tiempo de dos semanas. Cabe recalcar que se requirió una semana más para realizar pruebas, esta historia de usuario consistió en la toma de datos en la avenida Simón Bolívar de Quito, en el horario de 7am a 12am. El cuadro 19 muestra la historia de usuario seleccionada para esta iteración.

Cuadro 19.

Historias de usuario seleccionadas para la iteración 4

ID	Historia de usuario	Estimación (semanas)	Prioridad	Criterio de aceptación
5	Toma de Datos	3	5	Se requiere realizar una toma de datos de dos semanas de la avenida Simón Bolívar para poder realizar una análisis de datos

5.6.1 Sprint Backlog iteración 4

A continuación se presentan las tareas necesarias para cumplir con esta iteración, que tuvo un tiempo estimado de tres semanas, el cuadro 20 detalla las tareas que fueron necesarias para su cumplimiento.

Cuadro 20.

Sprint Backlog iteración 4

Sprint 3 Inicio 21-02-2016 Duración 1 semanas		28-5 Marzo	6-12 Marzo	13-19 Marzo
Tarea	ProductBacklog			
Definición de puntos sobre la avenida Simón Bolívar	5	X		
Pruebas del monitoreo, ubicación de sensores	5	X		
Primera semana de monitoreo	5		X	
Segunda semana de monitoreo	5			X

5.6.2 Revisión y seguimiento iteración 4

Para verificar el avance de la iteración 4 se enlistaron las tareas propuestas anteriormente y se especificó el avance de cada una como se muestra en el cuadro 21. De esta forma se evaluó el nivel de avance de la iteración 4.

Cuadro 21.

Tareas completadas de la iteración 4

Sprint 4 Numero de Tareas:4			
ID	Tarea	Iteración	Estado
1	Definición de puntos sobre la avenida Simón Bolívar	4	Completado
2	Pruebas del monitoreo, ubicación de sensores	4	Completado
3	Primera semana de monitoreo	4	Completado
4	Segunda semana de monitoreo	4	Completado

5.6.3 Pruebas de la iteración 4

Para esta iteración se aplicó pruebas en campo donde se verificó el aumento de los registros a medida que el tiempo avanzaba. Las mediciones se realizaron de 7am a 12am, usando dos estaciones una en sentido norte sur y otra en sentido sur norte. La figura 39 permite evidenciar los dispositivos de monitoreo en la avenida Simón Bolívar.



Figura 39. Dispositivos de monitoreo

5.6.4 Demo iteración 4

El demo de esta iteración obtuvo como resultado datos reales de 2 semanas de toma de datos de la avenida Simón Bolívar de Quito, datos tomados en horario de 7am a 12am. Estos datos tienen como objetivo ser utilizados en un análisis posterior.

5.7 Desarrollo y pruebas de la Iteración 5

Para la quinta iteración se tomó la cuarta y quinta historia de usuario, iniciar sesión y análisis de información con minería de datos, esta tuvo una duración de una semana. Se representa en el Cuadro 22. Esta quinta iteración contempla todo el proceso de migración de datos e implementación de Pentaho BI server para analizar la información con minería de datos e inteligencia de negocios.

5.7.1 Sprint Backlog de la iteración 5

Se detallaron las tareas necesarias para llegar a completar las historias de usuario seleccionadas que son: iniciar sesión y análisis de información con minería de datos. El Cuadro 23 describe el conjunto de tareas requeridas, el ProductBacklog al que pertenece y la fecha de inicio y entrega de cada tarea de usuario.

Cuadro 22.

Historias de usuario seleccionadas para la iteración 5

ID	Historia de usuario	Estimación (semanas)	Prioridad	Criterio de aceptación
1	Iniciar sesión	1	5	Presentar interfaz inicial de Pentaho BI Server que permite el ingreso al sistema
2	Análisis de información con minería de datos	1	6	Presentar información a través de gráficos y tablas usando Pentaho BI Server con el módulo SaikuAnalytics

Cuadro 23.

Sprint Backlog de la iteración 5

Sprint 5						
Inicio 3-01-2016						
Duración 4 semanas						
			3 - 9 Enero	10-16 Enero	17-23 Enero	24-30 Enero
ID	Tarea	ProductBacklog				
1	Extracción de datos desde el DBMS MySql	6	X			
2	Transformación y carga de datos hacia el modelo en estrella ubicado en el DBMS Postgres	6	X			
3	Instalación de Pentaho BI Server	6		X		
4	Configuración de conexión a la base de datos y formación del cubo OLAP	6			X	
5	Instalación del módulo SaikuAnalytics y observación de información.	6				X

5.7.2 Revisión y seguimiento de la iteración 5

Para realizar la revisión del Sprint se requiere enlistar las tareas y mostrar el estado de cada una, esto con el objetivo de medir el avance en la historia de usuario seleccionada. El Cuadro 24 detalla las tareas en el Sprint.

Cuadro 24.

Tareas completadas de la iteración 5

Sprint 1 Numero de Tareas:4			
ID	Tarea	Iteración	Estado
1	Extracción de datos desde el DBMS MySql	5	Completado
2	Transformación y carga de datos hacia el modelo en estrella ubicado en el DBMS Postgres	5	Completado
3	Instalación de Pentaho BI Server	5	Completado
4	Configuración de conexión a la base de datos y formación del cubo OLAP	5	Completado
5	Instalación del módulo SaikuAnalytics y observación de información.	5	Completado

5.7.3 Pruebas de la iteración 5

En la figura 40 se puede visualizar el diagrama en estrella implementado en Postgres, de igual manera en la figura 41 se ilustra la conexión exitosa con la base de datos Postgres desde Pentaho BI Server. Por último en la figura 42 se observa las medias con sus correspondientes dimensiones que conforman el cubo OLAP.

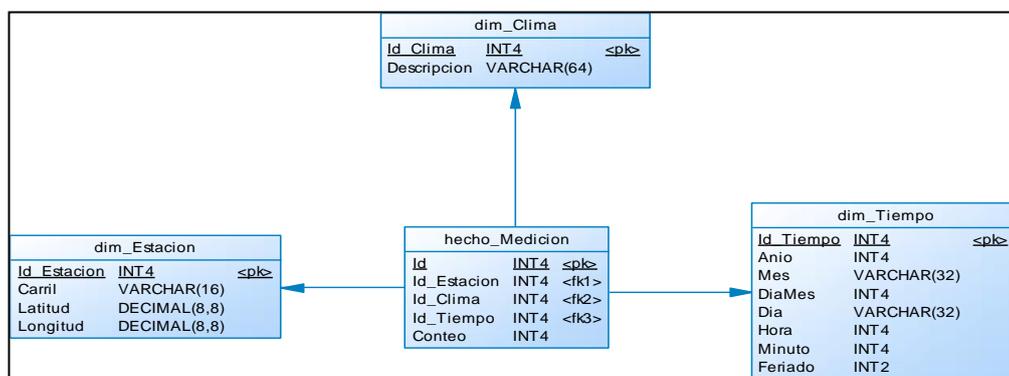


Figura 40. Modelo en estrella

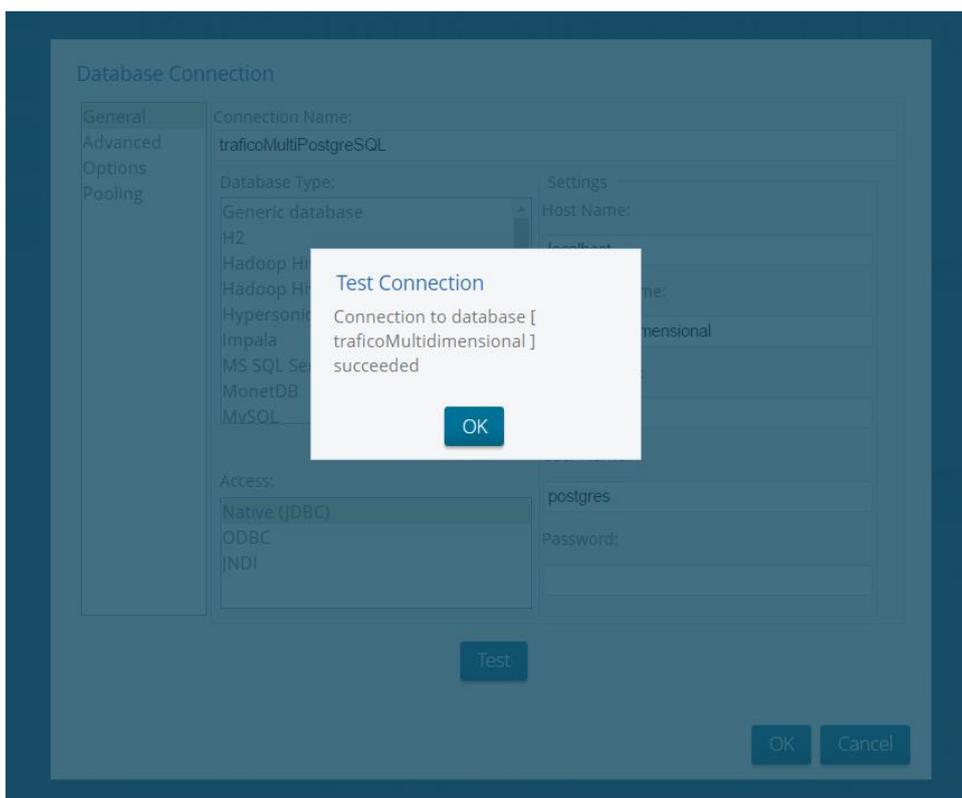


Figura 41. Conexión de prueba a la base de datos

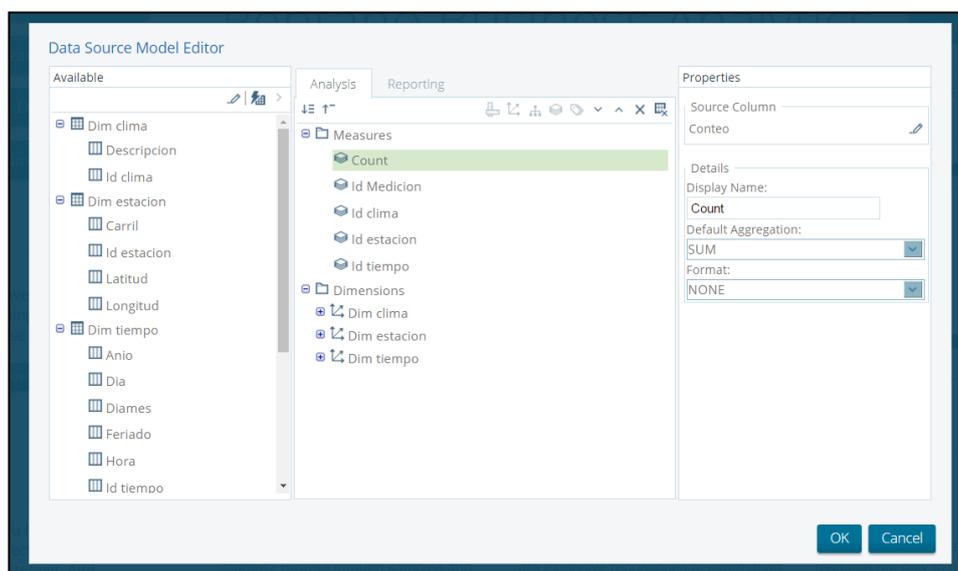


Figura 42. Medidas y dimensiones del cubo

5.7.4 Demo Iteración 5

El demo de la quinta iteración consiste en presentar la pantalla de ingreso a Pentaho BI Server que se muestra en la figura 43, así como la consola de análisis del módulo SaikuAnalytics presentada en la figura 44.



Figura 43. Pantalla de ingreso a Pentaho BI Server

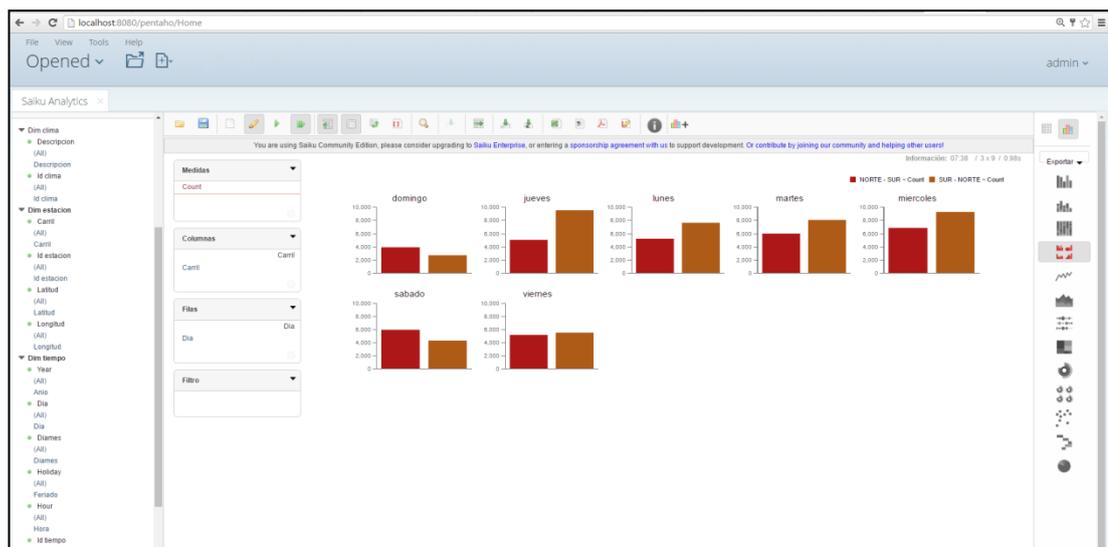


Figura 44. Consola de usuario de SaikuAnalytics

CAPÍTULO VI

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.

Con el objetivo de poder comprobar la prueba de concepto se realizó el proceso de medición del flujo de tráfico en la avenida Simón Bolívar en su cruce con la avenida interoceánica, analizando los dos carriles centrales. Este proceso se realizó en el periodo comprendido entre el lunes 21 de marzo y domingo 3 de abril desde las 7 am hasta las 12 am.

Se seleccionó la avenida Simón Bolívar debido a su comportamiento de avenida periférica de alto tráfico que en el sector seleccionado que cuenta con un puente, lo que permitió la ubicación de los dispositivos de monitoreo sobre los carriles centrales.

6.1 Análisis de resultados con minería de datos e inteligencia de negocios.

Con el apoyo de la herramienta BI server de Pentaho junto a su módulo SaikuAnalytics se puede observar gráficamente la información obtenida. De este modo, el análisis ha sido realizado en función de parámetros que al ser seleccionados y combinados dan respuestas a interrogantes acerca del tráfico. Este análisis está enfocado en los gráficos obtenidos por el proceso mencionado con el objetivo de respaldar de manera efectiva los resultados, y que su observación sea de manera clara, resaltando únicamente estadísticas con impacto.

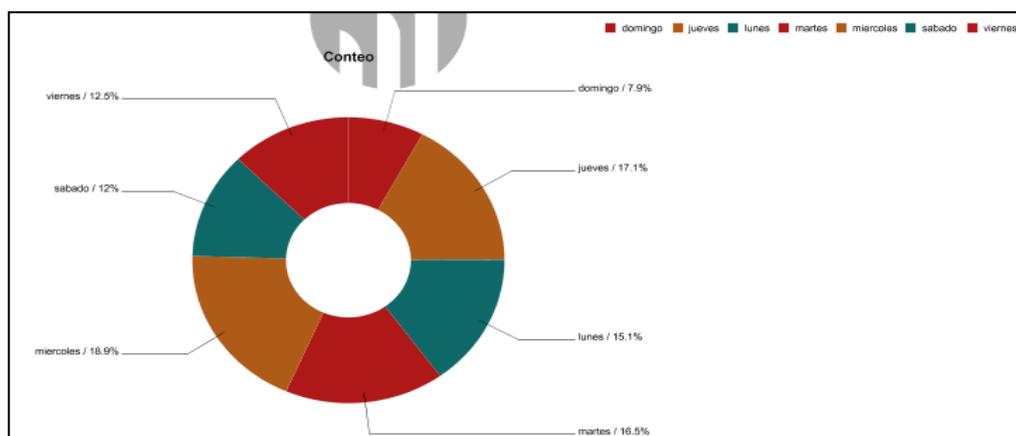


Figura 45. Flujo de tráfico dividido en días.

En la figura 45 se observa el resultado del estudio estadístico de los datos recolectados en función de los días de la semana. Se puede evidenciar que el día con mayor cantidad de vehículos es el miércoles y representa un 18,9% del total. De manera contraria el día con un menor flujo es el domingo y representa un 7,9% del total de datos. Ésta información es de suma importancia para el control de pico y placa aplicado en la ciudad puesto que serviría para comprobar que la distribución de vehículos es homogénea y en caso de no serlo poder aplicar acciones correctivas por parte de las autoridades.

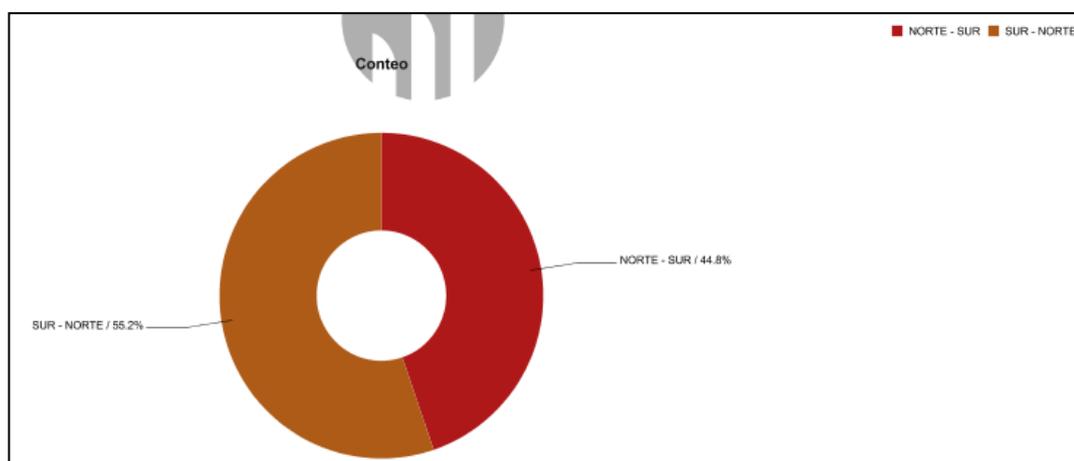


Figura 46. Flujo de tráfico dividido en el sentido.

En la figura 46 se ilustra la distribución del flujo de vehículos en función del sentido en el que se dirigen. El sentido con mayor ocupación es el sur –norte con un 55,2% mientras que el restante 44,8% corresponde al sentido norte – sur. Se puede mencionar que no existe una diferencia acentuada entre ambos lados del análisis sin tomar en cuenta en tiempo.

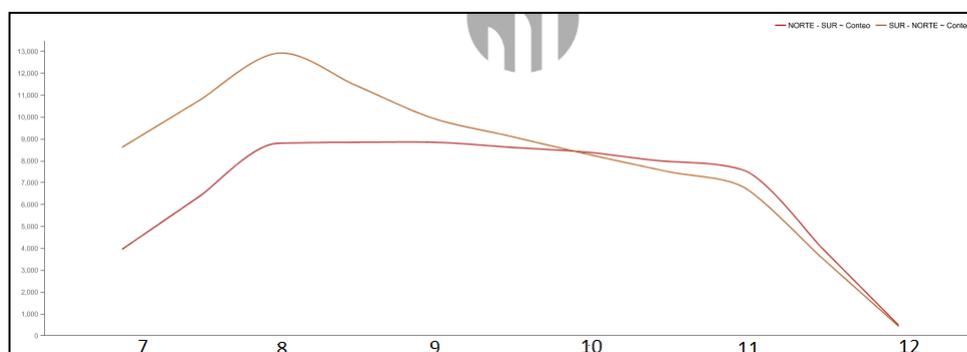


Figura 47. Flujo de tráfico dividido en el sentido en función del tiempo.

La figura 47 es más detallada en comparación a la anterior de carácter general, se puede apreciar que el sentido sur – norte es más transitado durante las primeras horas del día siendo muy notable esta información en la figura 48 correspondiente al viernes 1 de abril a las 8:10 am. El carril norte – sur pasa a ser más ocupado a partir de las 10 am de manera no muy pronunciada. Es de utilidad esta información al ser sustento para posibles soluciones como contra flujo en las horas en las que el flujo entre sentidos no está siendo transitado de manera equilibrada.



Figura 48. Comparación de sentidos: izquierda norte - sur, derecha sur – norte

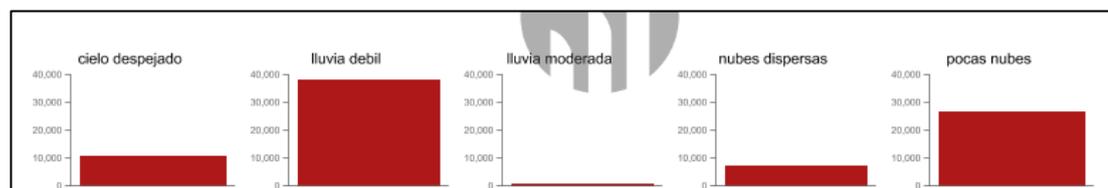


Figura 49. Flujo de tráfico dividido en clima.

En la figura 49 se observa cómo se encuentra disperso el tráfico en función del clima, se distingue que ante una lluvia débil el flujo es mayor, sin embargo durante el periodo de almacenamiento de datos, este fue el estado del clima predominante junto con nubes dispersas.

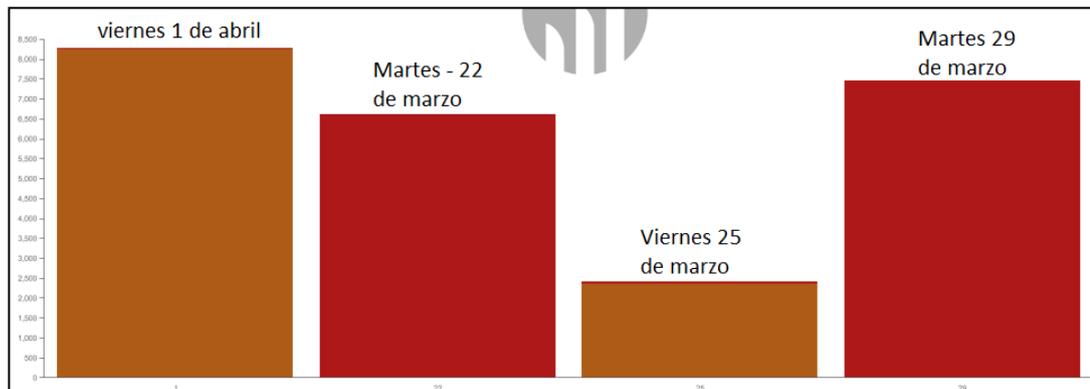


Figura 50. Comparación entre martes y viernes.

Durante el periodo de extracción de datos hubo un feriado dando un comportamiento distinto al común, hecho evidenciado en la figura 50 en el día viernes en el que se puede ver una variación fuerte entre las dos semanas. Por este motivo es que entre las variables del modelo también se incluyó si un día es feriado por el fuerte impacto causado por este fenómeno. El día martes, por otro lado, presenta una variación menor entre las dos semanas lo que conduce a una mayor predicción del flujo.

6.1.1 Análisis con minería de datos

A través de Weka que permite hacer una exploración detallada de los datos. Tras la búsqueda de los principales atributos que influyen en el flujo del tráfico se realizó el proceso CfsSubsetEval, que evalúa el valor de un subconjunto de atributos considerando la capacidad de predicción individual de cada uno, junto con el grado de redundancia entre ellos. Esto en conjunto con el método GreedyStepwise que Realiza una búsqueda codiciosa adelante o hacia atrás a través del espacio de subconjuntos de atributos. Los atributos seleccionados fueron:

- 1 Día,
- 2 Hora
- 3 Carriles

6.2 Correspondencia de los datos con variables aleatorias

Para este propósito, se utilizó una aproximación estadística para el número de vehículos que pasan por un carril específico en la Avenida Simón Bolívar de Quito, en condiciones normales, es decir, de lunes a viernes. Este análisis, se busca determinar si los datos obtenidos corresponden a una distribución Poisson, distribución normal o ninguna de ellas, se ha seleccionado estas candidatas debido a que la teoría sugiere que los datos de tráfico vehicular podrían corresponderse a Poisson. Adicionalmente, según la graficación inicial de los datos se tenía la sospecha de que estos podrían ser ajustados por una distribución normal discretizada.

Para determinar la correspondencia de datos se realizó pruebas de bondad de ajuste, las cuales permiten verificar que la población de la cual proviene una muestra tiene una distribución especificada o supuesta. En el caso de que no se ajuste a ninguna distribución se procede a generar un modelo propio para los datos existentes (Kohan, 2014).

La prueba de ji-cuadrado es una prueba de hipótesis que compara la distribución observada de los datos con una distribución esperada de los datos. Prueba de ji-cuadrado de bondad de ajuste se utiliza para probar qué tan bien una muestra de datos categóricos se ajusta a una distribución teórica (Minitab, 2016).

A continuación se realizó las pruebas ji cuadrado para los diferentes casos:

- Al aplicar la prueba ji cuadrado para una posible distribución de Poisson en el carril 1, dio como resultado el estadístico práctico 1270,153378 por lo cual se rechaza la hipótesis de que los datos se ajusten a una distribución Poisson, al igual que en el carril 2 debido a que el resultado de la prueba de ji-cuadrado de bondad de ajuste es: 527,5572468, el estadístico teórico para ambos casos con una significancia de 5% es respectivamente 113 y 67. Ver Apéndice- A
- Para el caso de distribución normal, la prueba ji cuadrado de bondad de ajuste para el carril 1, dio como resultado el estadístico práctico 427, por lo cual se rechaza la hipótesis de que los datos se ajusten a una distribución normal, al igual que en el carril 2 debido a que el resultado práctico de la prueba de ji-

cuadrado de bondad de ajuste es: 474, el estadístico teórico para ambos casos con una significancia de 5% es 7.81. Ver Apéndice- B

Debido a que los datos recolectados no se ajustaron a las distribuciones propuestas se recurrió a generar un modelo propio en base los datos recolectados.

6.3 Generación del modelo

Para la generación del modelo se tomó todos los datos del número de carros que pasaron cada 5 minutos representado por la variable (x) del carril 1 y posteriormente del carril 2 en condiciones normales, es decir, de dos semanas de lunes a viernes.

En base a estos datos se determinó la ocurrencia de cada valor de (x), y en base a este se calculó la probabilidad, dando como resultado los gráficos 51 y 52. En Anexo C se puede evidenciar la tabla donde constan los datos del conteo, ocurrencia, probabilidad, media y esperanza.

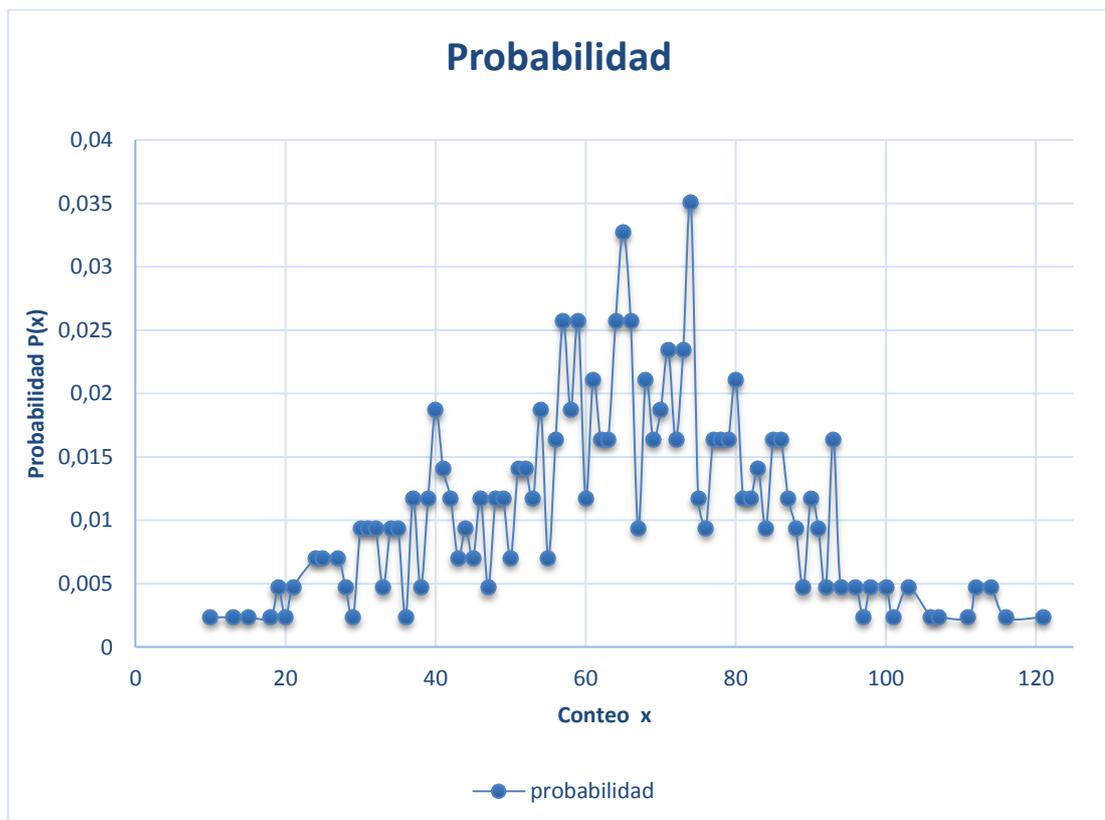


Figura 51 Modelo del carril 1

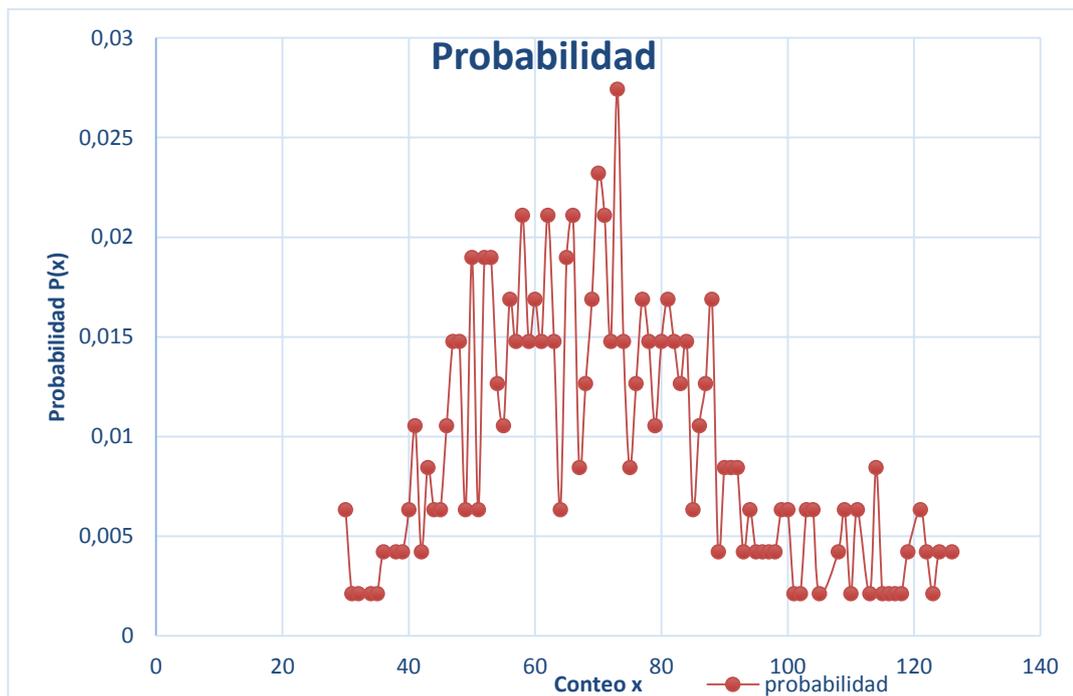


Figura 52 Modelo del carril 2

Para determinar la esperanza matemática de la variable aleatoria "x = count", que es un número real único, se sigue la siguiente fórmula:

$$E[x] = \sum_{k=1}^n P_k X_k$$

Donde P_k representa la probabilidad de la variable aleatoria definida como "count". Mientras X_k representa las frecuencias de la variable mencionada.

En el cálculo de la esperanza matemática y media, se obtuvo como resultado 63,6838407 para el carril1 y 81,47679325 para el carril 2, cada 5 minutos

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones

- a) El empleo de una red de sensores inalámbricos que utilice la red 3G/GPRS aporta con la portabilidad de los dispositivos.
- b) El uso combinado entre las metodologías SCRUM y XP permitió cumplir de manera organizada con los requerimientos que se planteó para este proyecto.
- c) El uso de aplicaciones distribuidas es la base de una red de sensores inalámbricos para centralizar la recepción y procesamiento de datos.
- d) El sensor de distancia láser cuenta la mayor precisión entre los de su tipo para el uso de detección de vehículos, puesto que tanto los de tipo ultrasónico como infrarrojo generan un cono para la captura del valor de la distancia, mientras que el sensor utilizado genera una línea recta que brinda precisión.
- e) El empleo de dos placas Arduino distribuyendo tareas siendo uno para la detección de vehículos y otro para la transmisión de datos sirve para que el conteo permanezca constante e independiente del envío de información por la red.
- f) Usar el protocolo HTTP para que el dispositivo realice la petición y el servidor PHP la consuma, aumenta la flexibilidad de implementación de una red de sensores.
- g) El empleo de una aplicación de Internet de las cosas para el monitoreo en tiempo real y que esté aplicado al tráfico en las ciudades brinda información de ayuda para la toma de decisiones.
- h) El uso de minería de datos es fundamental para brindarle utilidad a la información obtenida al encontrar propiedades o patrones del comportamiento del tráfico que no son visibles de otro modo.
- i) La capacidad que de la aplicación de monitoreo tiene para adaptarse a dispositivos móviles amplía el rango de personas que puedan observar la página desde sus vehículos.

- j) El uso de herramientas libres tanto de hardware como de software han permitido desarrollar una solución económica que cierra el círculo desde la recolección de datos hasta dar utilidad a los mismos.

7.2 Recomendaciones

- a) Aumentar el tiempo de recolección de datos, nodos en la red y variables que permitirían configurar un modelo matemático para aumentar la precisión de la predicción y determinación de patrones del comportamiento del tráfico.
- b) Se sugiere el uso de una topología centralizada con un elemento que se encargue de la transmisión por la red 3G/GPRS y que recibe los datos de nodos por cada carril disminuiría el precio y optimizaría el proceso que cumple la solución.
- c) Es aconsejable compartir con las autoridades pertinentes los resultados obtenidos en el proyecto con el fin de contribuir al mejoramiento del flujo vehicular y la seguridad en esta avenida.
- d) Se propone ubicar los dispositivos de una red de sensores inalámbricos en sitios donde se encuentren seguros de posibles daños producidos por malas costumbres de las personas que transiten por el lugar.
- e) Es recomendable diseñar los equipos con protección externa, de modo que tengan resistencia a los distintos estados del clima, principalmente de la lluvia.

7.3 Trabajo futuro

Como trabajo futuro se propone que en la toma de datos se implementen cuatro cambios: el primero es mejorar el algoritmo de detección programado para que se pueda diferenciar el tipo de vehículo que ha pasado por el sensor, sabiendo de esta manera si es pesado, liviano o motocicleta. El segundo cambio es aplicar una topología en estrella que optimice el acceso a la red de los distintos nodos, siendo una central que se encargue de transmitir información por la red celular e internamente se trabaje con sensores de radio más económicos que los anteriores. Como tercer cambio a implementarse es el añadir paneles solares a los nodos para

aumentar la autonomía de los mismos y, por último, se diseñará y aplicará una estructura a la medida para ubicar los circuitos y que se encuentren protegidos.

Desde la aplicación se harán dos mejoras: la primera es aumentar las prestaciones de la aplicación de recepción de datos para que administre seguridades y una adecuada gestión de los nodos sensores que conforman la red de sensores inalámbricos; como segunda mejora es la adición de una aplicación móvil nativa para optimizar el rendimiento en dispositivos móviles y aumentar el acceso a la misma.

Desde el punto de vista del uso que se les da a los datos se implementará una automatización al proceso de extracción, transformación y carga de datos (ETL) utilizando la herramienta Spoon de Pentaho para este fin. El aumento final que se aplicará es el desarrollo un modelo de simulación basado en datos recolectados, en diferentes condiciones y horarios, para mejorar las predicciones referentes al tráfico vehicular en esta zona.

Referencias

- Al-Sakran, H. O. (2015). Intelligent Traffic Information System Based on Integration of Internet of Things and Agent Technology. *International Journal of Advanced Computer Science and Applications*, 37-43.
- Artero, Ó. T. (2013). *Arduino Curso Práctico de formación*. Obtenido de https://books.google.com.ec/books?hl=es&lr=&id=6cZhDmf7suQC&oi=fnd&pg=PR15&dq=que+es+arduino&ots=AZdAk0IvFK&sig=NZZ2tqNXjdp0dss9lzApfeT_MeU&redir_esc=y#v=onepage&q=que%20es%20arduino&f=false
- ARWEB. (2014, 09 26). *¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB?* Recuperado el 06 20, 2015, de *¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB?:* <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>
- BLOGSPOT. (2016). <http://2.bp.blogspot.com/>. Obtenido de <http://2.bp.blogspot.com/-IZxuE1VMSJc/UAXVb7reUQI/AAAAAAAAABM/7yAvXSkPCLY/s1600/WSN.png>
- Booch, G. R. (1999). *El lenguaje unificado de modelado (Vol. 1)*. Madrid: Addison-Wesley.
- Canepa, E., Odat, E., Dehwah, A., Mousa, M., Jiang, J., & Claudel, C. (2014). A sensor network architecture for urban traffic state estimation with mixed eulerian/lagrangian sensing based on distributed computing. *n Architecture of Computing Systems–ARCS 2014*.
- Costanzo, A. (2013). An arduino based system provided with GPS/GPRS shield for real time monitoring of traffic flows. *Application of Information and Communication Technologies*, 1-5.

- Dessai, S. S. (2014). Development of Wireless Sensor Network for Traffic Monitoring Systems. *International Journal of Reconfigurable and Embedded Systems (IJRES)*. *IJRES*.
- ECURED. (2014). *Ecured*. Obtenido de <http://www.ecured.cu/Laravel>
- Elmotelb, A. S., Shabana, B. T., & Tolba, A. S. (2016). A Simulated System for Traffic Signal Management Based on Integrating GIS & WSN Techniques.
- Eren, H., Pakka, H. M., AlGhamdi, A. S., & Yue, Y. (2013). Instrumentation for safe vehicular flow in intelligent traffic control systems using wireless networks. In *Instrumentation and Measurement Technology Conference. IEEE International*, 1301-1305.
- Fernández-Lozano, J. J., Martín-Guzmán, M., Martín-Ávila, J., & García-Cerezo, A. (2016). A wireless sensor network for urban traffic characterization and trend monitoring. *Sensors*, 26143-26169.
- IEEE. (2008). *Especificación de Requisitos según el estándar*.
- Jiang, J., & Claudel, C. (2013). A wireless computational platform for distributed computing based traffic monitoring involving mixed eulerian-lagrangian sensing. In *Industrial Embedded Systems*, 232-239.
- Jo, Y., Choi, J., & Jung, I. (2014). Traffic information acquisition system with ultrasonic sensors in wireless sensor networks. *International Journal of Distributed Sensor Networks*.
- Johnston, S. (2009). *Cloud Computing-wikipedia*. Obtenido de https://en.wikipedia.org/wiki/Cloud_computing#/media/File:Cloud_computing.svg
- Kohan, D. (2014, 06 24). *Facultad e Ingeniería*. Obtenido de PRUEBAS DE BONDAD DE AJUSTE : http://www.bioingenieria.edu.ar/academica/catedras/metestad/PRUEBAS_DE_BONDAD_DE_AJUSTE-1.pdf

- Lee, J., Zhong, Z., Du, B., Gutesa, S., & Kim, K. (2015). Low-Cost and Energy-Saving Wireless Sensor Network for Real-Time Urban Mobility Monitoring System. *Journal of Sensors*.
- Liang, B. J. (2013). Traffic flow detection based on wireless sensor network. *Journal of Networks*, 1959-1865.
- Magrini, M., Moroni, D., Palazzese, G., Pieri, G., Leone, G., & Salvetti, O. (2015). Computer Vision on Embedded Sensors for Traffic Flow Monitoring. In Intelligent Transportation Systems (ITSC). *IEEE 18th International Conference*, pp. 161-166.
- Martín, E. (2014, Diciembre 2). *TicBeat.com*. Obtenido de <http://www.ticbeat.com/cloud/que-es-cloud-computing-definicion-concepto-para-neofitos/>
- Minitab. (2016). *Soporte de Minitab*. Obtenido de ¿Qué es una prueba de chi-cuadrado?: <http://support.minitab.com/es-mx/minitab/17/topic-library/basic-statistics-and-graphs/tables/chi-square/what-is-a-chi-square-test/>
- Mousa, M., Abdulaal, M., Boyles, S., & Claudel, C. (2015). Wireless sensor network-based urban traffic monitoring using inertial reference data. *In Distributed Computing in Sensor Systems (DCOSS)*, 206-207.
- Nunez, J. C. (2014). *Nubelo*. Obtenido de <http://www.nubelo.com/blog/ventajas-del-framework-moda-laravel/>
- Pascale, A., Nicoli, M., Deflorio, F., Dalla Chiara, B., & Spagnolini, U. (2012). Wireless sensor networks for traffic management and road safety. *Intelligent Transport Systems*, 67-77.
- Paul, P. S. (2014). Traffic density estimation and flow control for video surveillance system (Doctoral dissertation, BRAC University).
- Pentaho. (2015). *pentaho*. Obtenido de [pentaho: http://community.pentaho.com/projects/data-mining/](http://community.pentaho.com/projects/data-mining/)

- Perttunen, M., Kostakos, V., Riekkki, J., & Ojala, T. (2015). Urban traffic analysis through multi-modal sensing. *Personal and Ubiquitous Computing*, 709-721.
- Pressman. (2005). *Ingeniería del software*. New York: Mc Graw Hill.
- RESERV. (2014). *Metodologia*. Obtenido de <http://www.reserv.com.ar/metodologia.php>
- Riachi, F. A. (2014, mayo 29). *Slideshare*. Obtenido de <http://www.slideshare.net/FahedAlRiachi/designers-please-mind-the-gap-35278771>
- Singha, M. R., & Kalita, B. (2013). Mapping Mobile Phone Network onto Urban Traffic Network. *In to appear Proceeding of International Multi conference of Engineers and Computer Scientists*.
- Singha, M. R., & Kalita, B. (2013). Using Mobile Phone Network for Urban Traffic Management. *International Journal of Computer Applications*.
- SOFTENG. (2014). *SOFTENG*. Obtenido de SOFTENG: <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>
- Swathi, K., Sivanagaraju, V., Manikanta, A. K., & Kumar, S. D. (2016). Traffic Density Control and Accident Indicator Using WSN. *Traffic*.
- Torres, P. L. (2002). *Desarrollo de Software orientado a objeto utilizando UML*. Valencia: Universidad Politécnica de Valencia (UPV) - España.
- Wang, Q., Zheng, J., Xu, B., & Huang, Y. (2015). Analysis and experiments of vehicle detection with magnetic sensors in urban environments. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. *IEEE International Conference*, 71-75.
- ZEND FRAMEWORK. (2011). *Estándares de codificación de Zend Framework para PHP*.
- Zhang, W. (2013). Traffic Information Detection Based on Scattered Sensor Data: Model and Algorithms. *Sensor Wireless Networks*, 18.

Zhou, J., Chen, C. P., Chen, L., & Zhao, W. (2013). A user-customizable urban traffic information collection method based on wireless sensor networks. *Intelligent Transportation Systems, IEEE Transactions* , 1119-1128.