



**ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE ELECTRÓNICA EN REDES Y  
COMUNICACIÓN DE DATOS**

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA EN REDES Y  
COMUNICACIÓN DE DATOS**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE  
TELE-MONITORIZACIÓN DE INFORMACIÓN FÍSICA A  
TRAVÉS DEL SENSAMIENTO POR MULTITUDES MÓVILES**

**AUTOR: TORRES MINCHALA DAVID ANTONIO**

**DIRECTOR: PHD. VINICIO CARRERA**

**CODIRECTOR: MSC. DANIEL TRIVIÑO**

**SANGOLQUÍ**

**2015**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**  
**DEPARTAMENTO DE ELECTRICA Y ELECTRONICA**  
**INGENIERÍA ELECTRONICA EN REDES Y**  
**COMUNICACIÓN DE DATOS**

**CERTIFICADO**

PhD. Vinicio Carrera

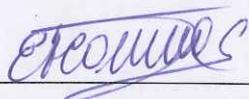
MSc. Daniel Triviño

**CERTIFICAN**

Que la tesis titulada “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELE-MONITORIZACIÓN DE INFORMACIÓN FÍSICA A TRAVÉS DEL SENSAMIENTO POR MULTITUDES MÓVILES”, ha sido dirigida y revisada periódicamente y cumple con las normas estatutarias establecidas en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas - ESPE.

Debido a que no existen modificaciones adicionales que se haya recomendado al alumno, se sugiere su publicación. El mencionado trabajo consta de un documento empastado y un CD, los cuales contienen los archivos en formato portátil de Adobe (PDF), Word y en presentación de PowerPoint. Autorizan al señor David Antonio Torres Minchala para que lo entregue al Sr. Nikolai Espinosa PhD, en su calidad de Director de la Carrera.

Sangolquí, 31 de Agosto del 2015



PhD. Enrique Vinicio Carrera  
DIRECTOR



MSc. Roberto Daniel Triviño  
CODIRECTOR

*Declaración de Responsabilidad*

**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE  
DEPARTAMENTO DE ELECTRICA Y ELECTRÓNICA**

**INGENERÍA ELECTRONICA EN REDES Y  
COMUNICACIÓN DE DATOS**

**DECLARACIÓN DE RESPONSABILIDAD**

David Antonio Torres Minchala

DECLARO QUE:

El proyecto de grado denominado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELE-MONITORIZACIÓN DE INFORMACIÓN FÍSICA A TRAVÉS DEL SENSAMIENTO POR MULTITUDES MÓVILES.” ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, consecuentemente este trabajo es de mi autoría. En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de la tesis de grado en mención.

Sangolquí, 31 de Agosto del 2015



David Antonio Torres Minchala

*Autorización de Publicación*

**UNIVERSIDAD DE LAS FUERZAS ARMADAS- ESPE**  
**INGENIERÍA ELECTRÓNICA EN REDES Y**  
**COMUNICACIÓN DE DATOS**

**AUTORIZACIÓN**

Yo, David Antonio Torres Minchala

Autorizó a la Universidad de las Fuerzas Armadas-ESPE la publicación, en la biblioteca virtual de la Institución la tesis “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELE-MONITORIZACIÓN DE INFORMACIÓN FÍSICA A TRAVÉS DEL SENSAMIENTO POR MULTITUDES MÓVILES”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, 31 de Agosto del 2015



---

David Antonio Torres Minchala

## DEDICATORIA

*Dedico este trabajo a las personas más importantes en mi vida, mis padres Marlene y Luis, que siempre estuvieron listos para brindarme incondicionalmente su ayuda y cariño. Dedico esta tesis también a mi hermano Pablo y Roberto gracias por su cariño y ánimo que brindaron para llegar hasta el día de hoy con mis estudios. Ahora es mi turno de devolver todo el inmenso apoyo que me han otorgado.*

*David A. Torres Minchala*

## **AGRADECIMIENTO**

A mis padres Marlene y Luis, y también a mi abuelitos. Millones de gracias por haberme dedicado su tiempo, esfuerzo y confianza para poder lograr mis sueños a pesar de lo distantes que se veían, gracias por su motivación y por darme la mano para caminar conmigo cuando sentía que el camino se terminaba, espero no de fraudarlos y continuar cosechando triunfos junto a ustedes.

A mis maestros, quienes con sus lecciones y experiencias me enseñaron que en un aula no solamente se aprende lo escrito en un libro, sino también a prepararse para los retos que impone la vida. Gracias Ing. Altamirano por brindarme la oportunidad de trabajar con Ud. en este proyecto, igualmente Ing. Triviño e Ing. Carrera sobre todo muchas gracias por su paciencia y tiempo.

Finalmente, a todos mis amigos, con quienes compartí alegrías, tristezas y malas noches. A ti Carla en conjunto estuvimos para brindarnos apoyo durante todo el tiempo que pasamos dentro y fuera de estas aulas, gracias en verdad.

*David A. Torres Minchala*

## ÍNDICE GENERAL

CERTIFICADO.....	ii
DECLARACIÓN DE RESPONSABILIDAD .....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
ÍNDICE GENERAL.....	vii
ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS .....	xi
RESUMEN.....	xiv
ABSTRACT .....	xv
1 INTRODUCCIÓN.....	1
1.1 Antecedentes.....	1
1.2 Justificación e Importancia.....	3
1.3 Alcance del Proyecto.....	5
1.4 Objetivos.....	6
1.4.1 General.....	6
1.4.2 Específicos .....	6
2 MARCO TEÓRICO .....	7
2.1 Sensado Colectivo por Móviles (MCS).....	7
2.2 Internet de las Cosas .....	10
2.3 Dispositivos Móviles Inteligentes .....	13
2.3.1 Arquitectura Android .....	14
2.3.2 Arquitectura Tizen .....	19

2.4 Dispositivos Electrónicos Vestibles .....	21
2.4.1 Relojes Inteligentes .....	22
2.5 Arquitectura Cliente - Servidor .....	23
2.5.1 Front-End .....	24
2.5.2 Back-End.....	30
3 DESCRIPCIÓN DE HARDWARE Y SOFTWARE .....	32
3.1 Descripción de Hardware .....	32
3.1.1 Gear S.....	32
3.1.2 Sensordrone.....	37
3.2 Descripción de Software.....	38
3.2.1 Entornos de Desarrollo IDE.....	38
3.2.2 Bibliotecas y Software de Terceros.....	40
3.2.2.1 API Google Maps.....	40
3.2.2.2 CanvasJS .....	41
3.2.2.3 Biblioteca Sensordrone .....	42
3.1.2 Servidor XAMPP .....	43
4 IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO .....	44
4.1 Arquitectura del Sistema .....	44
4.1.1 Escenario para el Sistema de Monitoreo con MCS.....	44
4.2 Desarrollo de la Plataforma Web.....	45
4.2.1 Front-End .....	45
4.2.2 Back-End.....	51
4.3 Desarrollo de Aplicaciones Móviles.....	56
4.3.1 Aplicación para Android .....	56
4.3.2 Widget para Tizen.....	58
5 PRUEBAS DE FUNCIONAMIENTO.....	61

5.1 Sistema de Monitoreo .....	61
5.1.1 Envío de información con ayuda de Sensordrone y Gear S.....	62
5.1.2 Recolección de Información Sensada .....	63
5.1.3 Análisis de la información de variables Climáticas .....	64
5.2 Transporte de la Información .....	70
5.2.1 Análisis de la cantidad de información .....	74
6 CONCLUSIONES Y RECOMENDACIONES .....	76
6.1 Conclusiones.....	76
6.2 Recomendaciones .....	77
6.3 Trabajos Futuros .....	78
7 BIBLIOGRAFÍA .....	80

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Versiones de Apis y Kernel de Linux que están en el SO Android.....	15
<b>Tabla 2.</b> Principales motores JavaScript de los navegadores usados en la Web.....	30
<b>Tabla 3.</b> Especificaciones técnicas del Gear-S.....	34
<b>Tabla 4.</b> Clases que componen la librería Sensordrone. ....	42
<b>Tabla 5.</b> Datos de Temperatura extraídos del sistema implementado en un intervalo de tiempo de 3 horas, realizadas desde el Sensordrone. Cálculo del error estimado. ....	67
<b>Tabla 6.</b> Datos de la humedad extraídos del sistema en un intervalo de tiempo de tres horas, realizadas desde el Sensordrone. Cálculo del error estimado. ....	68
<b>Tabla 7.</b> Datos de la presión extraídos del sistema en un intervalo de tiempo de tres horas de 10 min/muestra, realizadas desde el Sensordrone. Cálculo del error.....	69
<b>Tabla 8.</b> Descripción de comandos para la creación del filtro .....	73

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Esquema de sensado oportuno en áreas urbanas. ....	9
<b>Figura 2.</b> Esquema de los componentes que hace posible al internet de las cosas. ..	11
<b>Figura 3.</b> Cuidado de la salud de manera remota con objetos o cosas como dispositivos móviles vestibles. ....	12
<b>Figura 4.</b> Arquitectura Sistema Operativo Android. ....	14
<b>Figura 5.</b> Arquitectura Sistema Operativo Tizen. ....	21
<b>Figura 6.</b> Ejemplos de dispositivos avanzados portables o vestibles en el mercado en el presente año. ....	22
<b>Figura 7.</b> Algunos de los relojes inteligentes más avanzados del 2014 según la revista Trends. Gear-S Segundo desde la izquierda. ....	23
<b>Figura 8.</b> Ejemplo de una matriz de datos estructurado en formato JSON. ....	29
<b>Figura 9.</b> Dispositivo Reloj Inteligente Modelo Gear-S. ....	33
<b>Figura 10.</b> Componentes del Reloj <i>Gear-S</i> . ....	33
<b>Figura 11.</b> Formas de Funcionamiento del Gear-S. ....	36
<b>Figura 12.</b> App Samsung Gear, Administración de widgets en el reloj desde el teléfono. ....	36
<b>Figura 13.</b> Descripción general de los sensores en el Sensordrone. ....	38
<b>Figura 14.</b> Herramientas para desarrollar el sistema de monitoreo con MCS. ....	39
<b>Figura 15.</b> Tipos de Graficas encontradas en CanvasJS. ....	41
<b>Figura 16.</b> Esquema MCS para proveedor de información a la Base de Datos. ....	44
<b>Figura 17.</b> Esquema general de tecnologías utilizadas para la elaboración del Sistema de Monitoreo. ....	46

<b>Figura 18.</b> Algoritmo ejecutado en el lado del cliente, una vez descargado la página .php en un navegador web desde el servidor. ....	47
<b>Figura 19.</b> Algoritmo proceso de recepción y envío de información a través AJAX.....	48
<b>Figura 20.</b> Algoritmo utilizado para cargar y utilizar la API de Google Maps en la plataforma Web. ....	50
<b>Figura 21.</b> Esquema entidad relación para la base de datos <i>Crowdsensing</i> .....	51
<b>Figura 22.</b> Esquema de tablas creadas para el almacenamiento y gestión de las variables tanto ambientales como de salud para MCS.....	52
<b>Figura 23.</b> Interfaz GUI “phpmyadmin”, base de datos “CrowdsensingBD”.....	53
<b>Figura 24.</b> Diagrama de flujo descripción del funcionamiento al recibir una petición a través del método “POST”. ....	54
<b>Figura 25.</b> Diagrama de flujo descripción para ingresar los datos a la Base de datos desde PHP. ....	55
<b>Figura 26.</b> Pasos en la conexión lógica y física del Sensordrone y Smartphone. ....	56
<b>Figura 27.</b> Interfaz GUI desarrollada para la comunicación entre el Sensordrone, teléfono y el servidor.....	57
<b>Figura 28.</b> Widget desarrollado para la medición de ritmo cardiaco. ....	58
<b>Figura 29.</b> Esquema de red lógica para el envío de información desde el reloj inteligente hacia la base de datos. ....	59
<b>Figura 30.</b> Procedimiento para el intercambio de información con ayuda de AJAX ( <i>Asynchronous JavaScript XML</i> ). ....	60
<b>Figura 31.</b> Esquema general del sistema de monitoreo en base a MCS diseñado para la ESPE (campus Sangolquí).....	62
<b>Figura 32.</b> Comunicación establecida con los dispositivos encargados de la recolección de información.....	63
<b>Figura 33.</b> Interfaz de usuario de la aplicación actualizando los datos extraídos desde el Sensordrone.....	64

<b>Figura 34.</b> Estación de monitoreo Los Chillos, altitud 2453 m sobre el nivel del Mar, coordenadas (78°27'36" W, 0°18'00" S), equipo PM10, MET.....	65
<b>Figura 35.</b> Sistema de monitoreo de la calidad de aire, secretaria del ambiente municipio de Quito.....	66
<b>Figura 36.</b> Red de monitoreo de la Secretaria de Ambiente, sección reporte, variables físicas ambientales, recuadro opciones de sensores medidos. ....	66
<b>Figura 37.</b> Aproximación de los valores de temperatura medidos con referencia a un valor externo al sistema. ....	68
<b>Figura 38.</b> Aproximación de los valores de humedad medidos con referencia a un valor externo al sistema.....	69
<b>Figura 39.</b> Aproximación de los valores de presión atmosférica comparados con referencia a un valor externo al sistema de monitoreo.....	70
<b>Figura 40.</b> Comunicación entre un <i>Smartphone</i> y el Servidor de monitoreo en el envío de variables climáticas, visto a través de un <i>sniffer</i> . ....	71
<b>Figura 41.</b> Gráfica del entrada y salida de información referida en el servidor encargado del sistema de monitoreo a través de WireShark.....	71
<b>Figura 42.</b> Diagrama de secuencia y transmisión de la información. ....	72
<b>Figura 43.</b> Cantidad de paquetes en función del tiempo a través de WireShark (Línea Azul) teléfono inteligente, (Línea Roja) paquetes de salida al servidor.....	73
<b>Figura 44.</b> Porcentaje de paquetes transmitidos hacia el sistema de sensado, observados desde WireShark. ....	74
<b>Figura 45.</b> Reloj inteligente de Samsung “Simband” diseñado específicamente para el cuidado de la salud humana.....	79

## RESUMEN

El presente proyecto contempla el diseño e implementación de un sistema de monitoreo en tiempo real para el sensado de variables ambientales y del cuerpo humano, a través de dispositivos móviles inteligentes. Este proyecto ha sido llevado a cabo bajo el paradigma del sensado colectivo por móviles también conocido por sus siglas en inglés MCS (*Mobile-Crowd-Sensing*). Este paradigma define un método particular para medir el estado de las variables físicas de una determinada región, área o ciudad, aprovechando el uso de aquellos sensores incluidos en un dispositivo móvil inteligente o aditamentos externos. De esta manera el proyecto ha sido desarrollado con los siguientes recursos: un servidor que aloja la plataforma web para monitoreo, y dispositivos inteligentes como teléfonos con sistema operativo Android, relojes inteligentes y sensores externos Sensordrone. En conjunto, con todos los dispositivos antes mencionados, se ha conseguido la transmisión y captura de datos de temperatura, presión, luminosidad y ritmo cardíaco. Con ello, se logra que la información de las variables medidas de múltiples usuarios desde los sensores dentro de sus teléfonos mejore la precisión de una lectura ambiental en función de múltiples mediciones. Para el caso del cuidado humano, la frecuencia cardíaca de un usuario en particular, se recolecta y aloja en la base de datos del sistema de monitoreo para su posterior análisis. Esto da como resultado un sistema de información que ayuda al estudio estadístico y predictivo del comportamiento climático de una región específica al igual como una estadística de ritmo cardíaco particular de cada usuario.

### PALABRAS CLAVE:

- MCS (*MOBILE-CROWD-SENSING*)
- IDC (*INTERNET DE LAS COSAS*)
- SENSORDRONE
- ANDROID
- SAMSUNG-GEAR
- TECNOLOGÍAS WEB

## ABSTRACT

This project involves the design and implementation of a monitoring system for real time environment and heart rate sensing where the smart mobile devices of each person in a particular area provide the sensing information. In addition, this project has been performed according to the MCS (Mobile Crowd Sensing) paradigm. MCS defines a particular method for the participatory sensing of information through smart devices deployed in an area. Therefore, the project has been developed with the following resources: a server which provides the service of a web monitors system; also, a Sensordrone, a Gear-S smartwatch and smartphones that included Android. These devices have achieved sensing and transmitting the values of temperature, luminosity, pressure and human heart rate in a remote server. In this way, the participation of multiple users in the system will allow that values obtained by the sensors will improve the accuracy of an environment variable and heart rate for every one of the users, living in a particular region. This project has resulted in monitoring systems that provides of a statistical and predictive framework for the weather behavior and human health care in a specific region.

### KEY WORDS:

- MCS (*MOBILE-CROWD-SENSING*),
- IOT (*INTERNET OF THINGS*),
- SENSORDRONE
- ANDROID
- SAMSUNG-GEAR
- WEB TECHNOLOGIES

# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1 Antecedentes

Con el avance de los circuitos integrados así como los semiconductores, se mejoró la integración de *hardware* y *software*. De esta forma, el desarrollo de los primeros microcomputadores en el año de 1971 (Ameneiro, 2002), conllevó a que usuarios en oficinas y hogares puedan disponer de mejores herramientas para el tratamiento de la información.

En los años setenta y ochenta nace la denominada red de redes también conocido como Internet, fruto de proyectos de investigación y colaboración entre universidades junto con un fuerte apoyo económico de empresas y entidades gubernamentales de los Estados Unidos (Delgado Guerrero, 2007).

Conforme la tecnología y comunicación se fue consolidando en esa época, iniciaría la mayor producción e intercambio de información existente hasta la actualidad. Con ello, el desarrollo de *software* de tipo cliente-servidor aumentó al igual que el consumo de recursos físicos necesarios para soportar aplicaciones que

utilizaban un alto espacio de almacenamiento, capacidad de procesamiento y consumo de energía (Zikopoulos, 2011).

Esto provocó que a partir de los noventa se afianzase el uso de sistemas descentralizados de computadoras, que además de estar distribuidos en distintos sitios y comunicados entre sí a través de una red de banda ancha, logren desarrollar aplicaciones que disponen de ventajas como tolerancia a fallos, concurrencia y mayor capacidad de procesamiento (George Coulouris, 2013). Todo esto, de forma que el usuario los perciba como si fuera un solo computador (Ricardo Barranco Fragoso, 2012).

Precisamente, hoy en día es posible hallar servicios distribuidos en Internet como navegadores web, correo electrónico, *video-streaming*, geo-posicionamiento y monitorización vehicular (Harish Doddala, 2014), que fueron desarrollados con todas las ventajas antes mencionadas y además mostrándose de forma amigable y transparente para el usuario (George Coulouris, 2013).

Así, el surgimiento de un nuevo paradigma, conocido como Sensado Colectivo por Móviles (en inglés MCS, "*Mobile-Crowd-Sensing*"), es conocido a finales del año 2010. MCS es posible gracias al desarrollo e integración de nuevas tendencias tecnológicas, como son dispositivos móviles inteligentes, redes inalámbricas, computación en la nube y sistemas de monitorización (Bin Guo, 2014).

El MCS aprovecha los recursos que incorporan sensores integrados o acoplados a un teléfono inteligente o *Smartphone*. A diferencia del funcionamiento de una red tradicional de sensores inalámbricos (WSN), MCS utiliza la participación humana colectiva con ayuda de un dispositivo inteligente, llegando a ser la principal herramienta en la adquisición de datos físicos de una determinada área (Bin Guo, 2014). A su vez, la información sensada es almacenada en tiempo real en servidores para su estudio y análisis posterior.

En el paradigma MCS se deriva dos clases de eventos conocidos como sensado participatorio y oportuno, que permiten detallar la forma en la que el usuario puede estar consiente o no, de la transmisión de la información relacionada con variables físicas (Bin Guo, 2014).

En el año 2013, El Instituto Tecnológico de Massachusetts (MIT), con la ayuda del gobierno de EEUU, llevó a cabo el proyecto llamado “Kila Mahali”. El proyecto antes mencionado basado en el MCS, fue implementado con el objetivo de analizar información climática de lugares de la tierra como la atmósfera, océanos y desiertos para la prevención de desastres naturales (Victor Pankratius, 2013) (Bin Guo, 2014). El sensado colectivo por móviles ha comenzado a lograr notables aportes con proyectos como el antes mencionado, que contribuyen al desarrollo de la tecnología en el ámbito de adquisición de datos; gracias a la tendencia mundial de masificación y uso de dispositivos móviles inteligentes entre usuarios que poseen acceso a Internet (INEC, 2011).

Al mismo tiempo MCS está alineado con la investigación de proyectos para la implementación de redes de sensores móviles. De esta forma, se planea que MCS sea la base para proyectos a largo plazo como por ejemplo, las propuestas de CISCO System (Cisco System, 2006), con el Internet de las Cosas (IdC), y para la intercomunicación entre distintos dispositivos (Ameneiro, 2002).

## **1.2 Justificación e Importancia**

La existencia de sistemas de tele-monitoreo con software amigable y de fácil interacción con el usuario son cada vez más requeridas por gobiernos y usuarios que necesitan conocer el estado físico actual de alguna región. Todo ello con el objetivo de llegar a tomar adecuadas decisiones para, políticas públicas, así como para la prevención de desastres naturales.

Por ello MCS propone una notable mejora a la solución de sistemas de tele-monitoreo a través de la adquisición de datos en tiempo real de variables físicas, gracias a la ayuda colectiva de dispositivos móviles inteligentes desde múltiples usuarios.

El uso de dispositivos móviles, de acuerdo al censo realizado en el 2011, por el Instituto Ecuatoriano de Censos (INEC), ha crecido exponencialmente en esta última década; el mismo censo indica que, el número de dispositivos móviles existentes en el país igualó la misma cantidad de habitantes en el 2012 con 14 millones de dispositivos (INEC, 2012).

De esta forma, conforme el grado de integración del *hardware* mejora; la integración de componentes electrónicos provistos en dispositivos móviles, como salida de video HDMI, puerto USB, *Bluetooth* y conexión *Wi-Fi* entre los más comunes; ha hecho posible hoy en día se llegue a obtener casi todas las características que puede brindar un computador en un mismo dispositivo móvil (Ronan Schwarz, 2013).

Es así como en el año 2013, de acuerdo a los datos de la Agencia de Regulación y Control de las Telecomunicaciones ARCOTEL el tráfico de Internet generado por dispositivos móviles semejantes a un computador fue del 58% para *smartphones*, 30% *tablets* y un 12% de otros dispositivos (ARCOTEL-Supertel, 2012). En el año 2014 el número de dispositivos móviles inteligentes en el país es alrededor de 1,6 millones de aparatos, de los cuales cerca de cien mil llegaron a estar registrados y utilizan tecnología 4G (ARCOTEL-Supertel, 2014).

La diversidad de dispositivos móviles inteligentes y la facilidad en el mercado para su adquisición, ha permitido a empresas desarrollar hardware para el sensado de variables físicas y el desarrollo de aplicaciones. Así encontramos dispositivos como el Samsung-Gear, Netatmo, Wimoto, Sensorbug, Sensordrone, entre otros más que operan en conjunto con dispositivos inteligentes para brindar mejores beneficios de ayuda y conectividad a los usuarios.

De esta forma aquellos periféricos como antenas y sensores incorporados a un dispositivo móvil inteligente, permite proporcionar mediciones de un gran número de variables, como son posicionamiento, movimiento, temperatura, vibración, humedad y hasta los cambios químicos que sufre el aire, sin que represente un alto costo de procesamiento para el dispositivo y sin que el usuario lo note como una molestia (Harish Doddala, 2014).

Así, considerar el uso del dispositivo móvil inteligente como un aparato de medición puede llegar ser de importancia; además de las razones antes mencionadas, estos desarrollos representa disminución en tiempo y costo de implementación de sensores, fuentes de energía, y mantenimiento de dispositivos necesarios dentro de una red de sensores tradicional.

## 1.3 Alcance del Proyecto

El proyecto pretende entregar como producto final un sistema de telemonitorización basada en el paradigma de *Mobile-Crowd-Sensing* para la recolección de datos en tiempo real, de las siguientes variables físicas: temperatura, ubicación, presión y humedad; para ser almacenadas posteriormente en una base de datos, donde podrán ser presentados a través de en un portal web.

Para la adquisición y visualización de datos en tiempo real de las variables físicas propuestas, se creará una aplicación móvil para el sistema operativo Android versiones 4.0 y 4.2, que permita la recolección de los datos provenientes de los sensores internos y acoplados externamente al dispositivo móvil, en particular, para los dispositivos Sensordrone y Samsung-Gear.

Además, esta aplicación dispondrá de un registro de almacenamiento temporal, para la organización y preparación de la información para la transmisión hacia la Base de Datos. Dicha aplicación debe tener una interfaz gráfica (GUI) para observar el estado de la transmisión y la adquisición de los datos.

Para la transmisión de las variables sensadas se usará cualquiera de las tecnologías inalámbricas que pueda disponer el dispositivo móvil inteligente.

Para la implementación de la base de datos, se utilizará el sistema gestión MySQL, bajo el esquema de tipo entidad-relación. Dicha base de datos estará conformada por la información física de las variables sensadas desde el dispositivo móvil inteligente para la implementación.

El sistema de monitorización finalmente incluirá un portal web de contenido dinámico, diseñado en lenguaje de código abierto, para la comunicación con la base de datos. La observación de la información se realizará en tiempo real a través de gráficas históricas y mapas físicos de la ubicación donde se realice el sensado.

Por último, se propondrá un ambiente de pruebas para verificar el funcionamiento y rendimiento de la transmisión de los sensores y los protocolos utilizados en la comunicación desde el dispositivo móvil hacia al sistema base de datos distribuido para analizar los datos adquiridos.

## **1.4 Objetivos**

### **1.4.1 General**

Implementar un servicio de monitorización en tiempo real para el sensado y almacenamiento de variables físicas de acuerdo al paradigma de sensamiento por multitudes móviles.

### **1.4.2 Específicos**

- Identificar los componentes electrónicos y diseñar la topología necesaria para la comunicación entre los dispositivos inteligentes y servidores que participan en el sensado colectivo.
- Diseñar una aplicación en SO Android con la ayuda de APIs y bibliotecas necesarias, para el envío de los datos monitorizados.
- Crear una base de datos de tipo relacional, que permita el almacenamiento de variables enviados por los sensores.
- Implementar un portal web para la observación de la información recolectada proveniente de usuarios participando en el sensado.
- Evaluar el ambiente implementado para determinar el rendimiento del sistema de monitorización y mecanismos de comunicación seleccionados.

## CAPÍTULO 2

# MARCO TEÓRICO

### 2.1 Sensado Colectivo por Móviles (MCS)

*Mobile Crowd Sensing* (MCS) surge como un paradigma en la detección de variables físicas ambientales y humanas basado en el poder que tienen las multitudes, gracias a las capacidades de detección desde sus dispositivos móviles, tales como teléfonos inteligentes o dispositivos vestibles (C Tanas, 2014).

Como se ha mencionado, la metodología para la adquisición de datos se asemeja al funcionamiento de una red de sensores a mediana y gran escala. Sin embargo MCS busca conseguir mediciones detalladas y precisas existentes en pequeñas regiones y micro-climas donde factores locales como la cobertura de vegetación, topografía, temperatura, presión y la mano del hombre inciden notables cambios en el clima de dicha región. Con el desarrollo de la tecnología el humano actualmente porta varios dispositivos electrónicos en su cuerpo, muchos de ellos son provistos de sensores y circuitos de comunicación. Aquellos actualmente están saliendo de los laboratorios de investigación en menor tiempo desde compañías dedicadas al diseño de tecnología para luego ser introducidos en el mercado (David Cuartielles, 2015). Además debido a su estética, ergonomía y prestaciones ahora están siendo comercializadas por grandes compañías de telecomunicaciones.

Así el sensado colectivo es posible gracias al avance de la portabilidad de dichos dispositivos. MCS es encargada de distribuir los recursos de procesamiento y conectividad, para ello se ha basado en comprender como ha evolucionado la computación ubicua o pervasiva que facilita el análisis en la integración de la informática con la persona (Fang-Jing, 2011).

MCS se basa en la ubicuidad la capacidad para estar en muchos sitios a la vez, en el campo computacional e informático este concepto ha ido evolucionado conforme las tecnologías de telecomunicaciones y dispositivos electrónicos así lo han permitido (Aransay, 2009), es así como en una cualquiera urbe como se observa en la figura 1 es posible encontrar un sensado oportuno en casi todas partes gracias a dispositivos cuyos medios y características permitan una conectividad constante. Además debido al tiempo necesario para el intercambio de información entre largas distancias ha disminuido de segundos a milisegundos, hoy en día es posible recolectar y concentrar grandes cantidades de información en poco tiempo, dando una sensación al usuario que fuera en tiempo real algo que en realidad no sucede (Fabricio Carvalho, 2014).

Un aspecto para el análisis de información que tiene MCS y que también proviene del estudio de la computación ubicua en dispositivos electrónicos es la cantidad de procesamiento que soporta un microprocesador en función del espacio físico que ocupa en uno de estos. Hasta hace pocas décadas atrás, esta ley concebida de acuerdo a Moore, ha marcado un hito en la elaboración de circuitos diseñados e integrados a gran escala cada vez más reducidos. Por esto es bien conocido que memorias con mayor capacidad y procesadores más potentes y pequeños salgan al mercado cada año (Aransay, 2009).

Si bien el avance en las técnicas de integración de circuitos en pequeños espacios ha favorecido la lógica computacional con la cantidad de procesos posibles para desempeñar un algoritmo, no ha sucedido lo mismo en la manera de almacenar energía como se encuentra para el circuito de una batería. Como se observa en la figura 1 hoy es cotidiano encontrarse con autos y teléfonos que dependen de una batería para disponer de servicios como GPS, WiFi, temperatura entre otros. En la actualidad esta dependencia antes mencionada también resulta ser particularmente el principal factor limitante de dispositivos móviles inteligentes, en consecuencia MCS

para el continuo estado de sensado de un micro-clima también es afectado con esta limitante; la energía con que energizará a los sensores (Bin Guo, 2014).

Un estudio realizado en redes de sensores inalámbricos también conocidos en inglés como WSN en particular en nodos conocidos como Motes, vincula la adquisición de datos y procesamiento con técnicas apropiados al uso de la batería y ancho de banda para las lecturas de sensores de determinada región, justamente estas pruebas han arrojado como resultado que el principal limite es la dotación de energía requerida para su funcionamiento y monitoreo continuo (G Anastasi, 2012). Así MCS y una arquitectura a base de Motes definen dos principios de funcionamiento diferente, pero cuyo objetivo es idéntico. Ambos marcan el mismo obstáculo; la fuente de energía (Alan Mainwairing, 2012).

Sin embargo MCS posee una gran ventaja a otros tipos de tecnología como los Motes antes mencionados, y es que con MCS aprovecha un hardware de gran demanda en el mercado los teléfonos inteligentes que se encuentran en constante mejora debido a la competencia tecnología de empresas como Samsung, Apple, Huawei entre otros (Wen Sun, 2014). Este vendría a que los dispositivos móviles inteligentes con MCS su principal función no sea únicamente proveer de datos sensados de un área en particular sino además proveer de información de manera que el usuario no perciba como una intrusión ni tampoco consuma todos los recursos del dispositivo (Ioannis Krontiris, 2014).

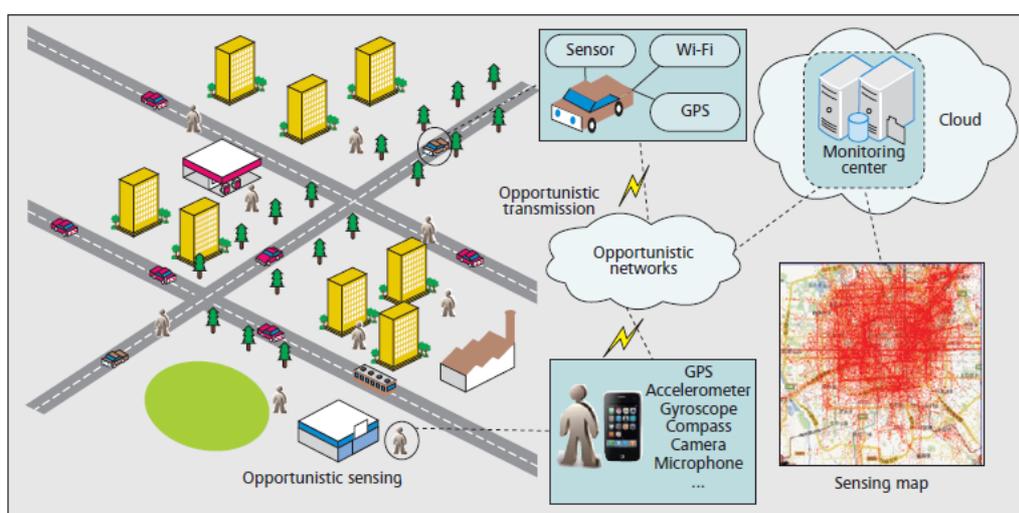


Figura 1. Esquema de sensado oportuno en áreas urbanas.

Fuente: (Huadong Ma, 2014)

Es así como muchos aspectos deben ser tomados en cuenta a la hora de implementar MCS como confiabilidad de la información del usuario, procesos en segundo plano, etc. Entre uno de los principales aspectos está contemplado la cantidad de procesamiento del lado del servidor *backend* que con lleva una implementación de base de datos y servidores web bajo este paradigma. Otro que también es de importancia es el algoritmo para la transmisión de los sensores de dispositivos móviles (Bin Guo, 2014). Hoy en día debido a MCS es una técnica emergente en la adquisición de información son pocas las aplicaciones que se basadas en este paradigma, todavía no existe una guía o un método para la transmisión a gran escala como urbes o ciudades de datos para este tipo de aplicaciones.

Por ello numerosos desafíos de investigación únicos surgen del paradigma de MCS, incluyendo la recolección de datos participativa sin la necesidad de un punto de acceso, la óptima selección de un nodo de detección y asignación de tareas, mecanismos de incentivos para el uso de MCS, la comunicación de red transitoria, procesamiento de datos entre el espacio, la psicología de dependencia que genera en el humano y dispositivo vestibles entre otros.

Finalmente, algunas aplicaciones basadas en MCS hasta la fecha han sido realizadas como se ha mencionado en la introducción, con ayuda de protocolos que aún no soportan servicios de tiempo real, o compatibilidad para cualquier sistema operativa incluyendo aquellos para dispositivos móviles. Con ello MCS desea especificar cambios notables en aspectos de la calidad y fiabilidad del dato que se producen en un clima determinado así como las técnicas de intercomunicación que utilizarían para transportar la información hasta su almacenamiento.

## **2.2 Internet de las Cosas**

El Internet de las cosas IdC es una idea que empezó a fines de la década de los noventa a partir del ingreso del concepto de computación ubicua que introdujo Mark Weiser, director científico de Xerox Palo Alto *Research Center* (PARC); este concepto viene dado por la integración de los elementos cotidianos mediante las comunicaciones (Ortiz, 2013). De este modo como se observa en la figura 2 el

concepto del Internet de las cosas está conformado a su vez por cuatro conceptos las cosas, servicios, dispositivos y datos que puede ser descrito de la siguiente forma:

- Cosas: Objetos físicos, que alguien o una empresa necesita conocer el estado, y funcionamiento del mismo en todo momento.
- Servicios: Proveer de acceso a la información de los objetos interesados a través de aplicaciones y sistemas web.
- Datos: Recolección de la información y capacidad de almacenamiento que un dispositivo u objeto dispone en una base de datos.
- Dispositivos: Circuitos eléctricos adheridos al objeto que permite la extracción de información y también la comunicación de un objeto.

Sin embargo existen otros roles y conceptos como el MCS que relacionan el aporte de personas a través de dispositivos como teléfonos inteligentes para un sistema donde se provee datos que posibilitan servicios como el de monitoreo, por ello MCS es la iniciación para el internet de las cosas ya que una vez relacionado los dispositivos, datos, servicios llegará a interconectar todo tipo de cosas.

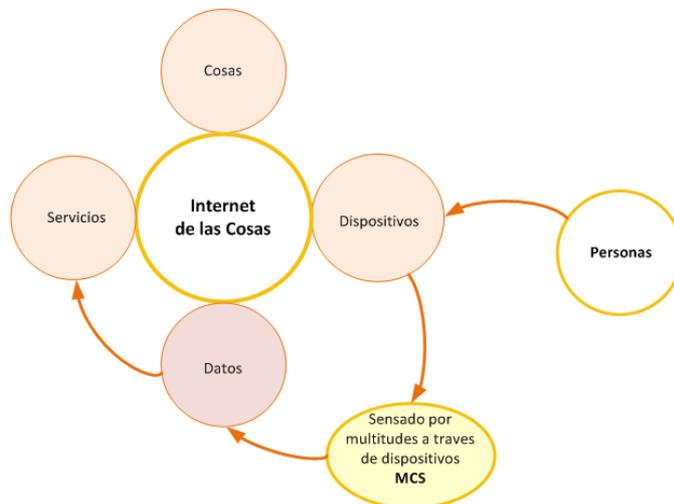


Figura 2. Esquema de los componentes que hace posible al internet de las cosas.

Fuente: (VDC Research Inc, 2014)

Como se dijo anteriormente Internet ha ido cambiando en forma y tamaño logrando la estandarización de algunos de sus componentes, sin embargo el propósito para el que fue diseñado se ha mantenido. Con ello este significado IdC adquiere importancia como la primera evolución real de Internet, donde IdC ha logrado que

Internet sea sensorial (temperatura, presión, humedad vibración de luz, humedad) y la manera cómo se extienda sea a través de MCS (Evans, 2011).

Por ello que el concepto de MCS se basa en muchas tecnologías que forman la columna vertebral para personas que pueden estar conociendo el estado físico del medio que lo rodea en todo momento; un concepto más bien abstracto. Uno de los principales promotores como es la compañía CISCO y ORACLE se han propuesto en el año 2020 tener 16 mil millones de dispositivos interconectados, que representará un promedio de seis dispositivos por persona en el planeta (Peter Middleton, 2014) (Cortes R, 2015).

Es así que el IdC principalmente se ayuda del MCS utilizando dispositivos que existen en varias áreas de nuestra vida, desde nuestros hogares, en el área de nuestro trabajo, en el transporte y en la industria, existen también dispositivos electrónicos que se utilizan en la salud. Por ello como se aprecia en la figura 3 una persona puede llegar estar sensada desde de varios dispositivos vestibles con ayuda de un teléfono inteligente para la detección y transmisión de signos vitales hacia un servidor para el cuidado de la salud.

Así uno de los campos donde el IdC se proyecta un futuro prometedor es el cuidado de la salud del cuerpo humano. A través de MCS algunas empresas que han empezado con este desafío son aquellas marcas principalmente de teléfonos inteligentes que incorporan sensores y biosensores para conocer el estado actual del usuario y mantener un registro de los signos vitales.

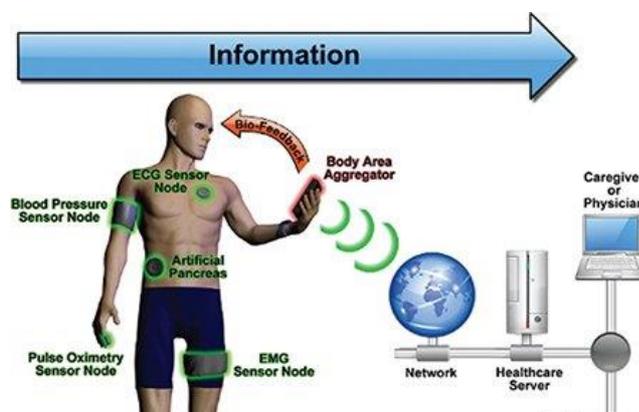


Figura 3. Cuidado de la salud de manera remota con objetos o cosas como dispositivos móviles vestibles.

Fuente: (Ianace, 2015)

De este modo para comprender mejor y facilitar el estudio de la práctica medicinal soportada desde dispositivos móviles inteligentes conocido en inglés también como mHealth (Albert Alonso, 2015) logre que la asistencia médica al igual como la lógica de negocio de la medicina cambie en estos próximos años. Al conseguir que el paciente se encuentre constantemente monitorizado es posible llevar un mejor control y seguimiento de un paciente.

El cuidado de salud remota o *remote Healthcare* es un área extensa donde su origen proviene del IdC, sin embargo debido a complejidad deber ser tratado en una rama de la investigación completamente nueva. Es por ello el que el IdC en la salud traerá ventajas como acortar los tiempos de espera y un flujo más ágil de la información. Además permitirá reducir el tiempo en que los médicos dedican a otras tareas como las administrativas, suponiendo esto una mejor seguridad y atención del usuario o paciente (Valencia, 2015).

## 2.3 Dispositivos Móviles Inteligentes

La existencia de estos dispositivos ha sido deseada desde mucho tiempo atrás. En estos últimos años, el usuario ahora ya es capaz de llevar un dispositivo portable que integre varias tecnologías de comunicación, alto procesamiento y sensores que no incomode al usuario. Por ello los dispositivos móviles inteligentes son conocidos como computadoras de mano y han sido estudiados a lo largo de su evolución dentro la computación ubicua (Aransay, 2009).

En general son aparatos de pequeño tamaño, con alta capacidad diseñados específicamente para una función básica de comunicación pero que con el paso del tiempo han ido evolucionando hasta llevar acabo funciones más complejas. Entre las principales características de estos aparatos encontramos:

- Son portátiles y pueden ser transportados en el bolsillo del usuario.
- Diseñados para soportar varios procesos gestionados bajo un sistema operativo.
- La capacidad de activar o desactivar algunos de sus recursos de hardware, con el objetivo de alargar la duración de la batería.

- Personalización de la interfaz visual, tanto en posesión como en operación, el cual es permite adaptarse al gusto del usuario.
- Capacidad de conexión permanente o temporal a una red para trabajar con aplicaciones instaladas.
- Conexión por USB o micro USB.

### 2.3.1 Arquitectura Android

En el desarrollo de las aplicaciones en Android, resulta importante conocer cómo está estructurado el sistema operativo. A esto se le llama arquitectura de Android, la cual está formada por varias capas que facilitan al desarrollador la creación de aplicaciones como se puede apreciar en la figura 4.



Figura 4. Arquitectura Sistema Operativo Android.

Fuente (Burton, 2012)

Esta distribución de la arquitectura en capas o pila, permite acceder a las capas más bajas mediante el uso de librerías, de esa manera el desarrollador no tenga que programar a bajo nivel las funcionalidades necesarias para que una aplicación haga uso de los componentes de *hardware* de los dispositivos. Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones (Burton, 2012). Para entender mejor, a continuación se describe cada una de las capas y componentes dentro de la arquitectura de Android.

### 2.3.1.1 Kernel

Es el núcleo del sistema operativo Android, se basa en el kernel de Linux, similar al que puede incluir en cualquier distribución de Linux en un computador, como por ejemplo Ubuntu, sin embargo esta variación de kernel como se observa en la tabla 1 está adaptado a las características del *hardware* en el que se ejecutará Android para dispositivos móviles. La API desarrollada más reciente es la 21 con una versión de kernel 3.10.61 actualmente disponibles para dispositivos con sistema operativo Lollipop.

Tabla 1.

Versiones de Apis y Kernel de Linux que están en el SO Android.

Android	Versión	Nivel de API	Versión de Linux Kernel
1.5	Cupcake	3	2.6.27
1.6	Donut	4	2.6.29
2.0.1	Eclair	5,7	2.6.29
2.2.x	Froyo	8	2.6.32
2.3.x	Gingerbread	9, 10	2.6.35
3.x.x	Honeycomb	11,13	2.6.36
4.0.x	Ice Cream Sandwich	14, 15	3.0.1
4.1.x	Jelly Bean	16	3.0.31
4.2.x	Jelly Bean	17	3.4.0
4.3	Jelly Bean	18	3.4.39
5.0	Lollipop	21	3.10.61
5.1	Lollipop	21	3.10.61

Fuente: (Chen Jing, 2015)

El núcleo actúa como una capa de abstracción entre el *hardware* y el resto de las capas de la arquitectura. Los desarrolladores no acceden directamente al núcleo, sino

que utiliza las bibliotecas disponibles en las capas superiores. Así también se evita el hecho de complicarnos para conocer las características precisas de cada teléfono.

Si necesitamos hacer uso de la cámara, el sistema operativo se encarga de utilizar el *software* incluido en el teléfono. Por ello, para cada elemento de *hardware* del teléfono hay un controlador conocido también como *driver*. El kernel además se encarga de gestionar los diferentes recursos del teléfono (energía, memoria, procesos, interfaces de comunicación etc.) además del sistema operativo en sí.

### **2.3.1.2 Bibliotecas**

Sobre el kernel están las bibliotecas nativas de Android, escritas en C o C++ y compiladas para la arquitectura *hardware* específico del teléfono. Normalmente están hechas por el fabricante, que además se encarga de instalarlas en el dispositivo antes de ponerlos al mercado. El propósito de las bibliotecas es dar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando eficiencia.

En esta sección habitualmente se encuentra: *OpenGL* (motor gráfico), bibliotecas multimedia (formatos de audio, imagen y video), *Webkit* (navegador), *SSL* (cifrado de comunicaciones), *FreeType* (fuentes de texto), *SQLite* (base de datos), entre otras (Burton, 2012).

### **2.3.1.3 Entorno de Ejecución**

El entorno de ejecución o *Runtime* de Android no se considera una capa en sí mismo, porque también está formado por librerías. Aquí hallamos las librerías con las funcionalidades habituales de Java y otras específicas de Android.

El principal componente del entorno de ejecución de Android es la máquina virtual Dalvik. Las aplicaciones son codificadas en Java y son compiladas en un formato específico para que esta máquina virtual las ejecute. La ventaja de esto es que las aplicaciones se compilan una única vez y de esta forma estarán listas para distribuirse con la total garantía de que podrán ejecutarse en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera la aplicación (Burton, 2012).

Java un lenguaje de programación que se usa únicamente como un herramienta de codificación de un algoritmo, son compilados y los ejecutables que se generan a través del SDK de Android tienen la extensión .dex que es específico para la máquina virtual Dalvik, y por ello no siempre podremos correr aplicaciones Java diseñadas para aplicaciones de escritorio en Android.

#### **2.3.1.4 Plataforma para Aplicaciones**

En la capa del *Framework* o plataforma de aplicaciones está formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik.

En el diagrama encontramos en la capa superior los siguientes:

1. Administrador de actividades. Se encarga de administrar la pila de actividades de nuestra aplicación así como su ciclo de vida.
2. Administrador de ventanas. Se encarga de organizar lo que se mostrará en pantalla. Básicamente crea las superficies en la pantalla que posteriormente pasarán a ser ocupadas por las actividades.
3. Proveedor de contenido. Esta librería es muy interesante porque crea una capa que encapsula los datos que se compartirán entre aplicaciones para tener control sobre cómo se accede a la información.
4. Vistas. En Android, las vistas son los elementos que ayuda a construir las interfaces de usuario: botones, cuadros de texto, listas y hasta elementos más avanzados como un navegador web o un visor de *Google Maps*.
5. Administrador de notificaciones. Engloba los servicios para notificar al usuario cuando algo requiera su atención mostrando alertas en la barra de estado. Un dato importante es que esta biblioteca también permite jugar con sonidos, activar el vibrador o utilizar los LEDs del teléfono en caso de tenerlos.
6. Administrador de paquetes. Esta biblioteca permite obtener información sobre los paquetes instalados en el dispositivo Android, además de gestionar la instalación de nuevos paquetes. Con ello se refiere a la forma en que se distribuyen las aplicaciones Android, estos contienen el archivo .apk, que a su

vez incluyen los archivos .dex con todos los recursos y archivos adicionales que necesite la aplicación, para facilitar su descarga e instalación.

7. Administrador de Telefonía. Con esta librería podremos realizar llamadas o enviar y recibir SMS/MMS, aunque no permite reemplazar o eliminar la actividad que se muestra cuando una llamada está en curso.
8. Administrador de Recursos. Con esta biblioteca podremos gestionar todos los elementos que forman parte de la aplicación y que están fuera del código, es decir, cadenas de texto traducidas a diferentes idiomas, imágenes, sonidos o *layouts*.
9. Administrador de localización. Permite determinar la posición geográfica del dispositivo Android mediante GPS o redes disponibles y trabajar con mapas.
10. Administrador de Sensores. Permite manipular los elementos de *hardware* del teléfono como el acelerómetro, giroscopio, sensor de luminosidad, sensor de campo magnético, brújula, sensor de presión, sensor de proximidad, sensor de temperatura, etc.
11. Cámara. Con esta biblioteca podemos hacer uso de las o la cámara del dispositivo para tomar fotografías o para grabar vídeo.
12. Multimedia. Permiten reproducir y visualizar audio, vídeo e imágenes en el dispositivo.

### **2.3.1.5 Aplicaciones**

En esta capa se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas (programadas en C o C++) y las administradas (programadas en Java), las que vienen preinstaladas en el dispositivo y aquellas que el usuario ha instalado.

Además en esta capa encontramos la aplicación principal del sistema: Inicio (*Home*) o lanzador (*launcher*), porque es la que permite ejecutar otras aplicaciones mediante una lista y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso widgets, que son también aplicaciones de esta capa.

Android proporciona un entorno sumamente poderoso para programar aplicaciones que hagan cualquier cosa con ayuda de Java. Todo de Android es

accesible y podemos jugar siempre con las aplicaciones de nuestro teléfono para optimizar cualquier tarea. El potencial de Android se basa en el control total que se le da al usuario para que haga de su teléfono un dispositivo a su medida.

### 2.3.2 Arquitectura Tizen

Tizen es una plataforma de *software* desarrollado con estándares de código abierto, apoyada por los principales operadores de telefonía móvil, proveedores de silicio y fabricantes de dispositivos como *Smartphones*, *tablets*, *netbooks*, *smartwatches* y dispositivos de información y entretenimiento como televisores inteligentes y más (Linux Foundation, 2014).

Además, esta arquitectura dispone de un sistema operativo basado en Linux, que provee de aplicaciones y servicios en el que el usuario lleva y traslada su contenido de un dispositivo a otro con facilidad. El proyecto Tizen fue creado en la *Linux Foundation* y es regido por un grupo de dirección técnica. Dicho grupo de dirección es el principal órgano de decisión para el proyecto donde se analizan el uso de metodologías relacionadas con la web 3.0 e Internet de las Cosas. Además al utilizar código abierto, facilita la ayuda de grupos de trabajo en todo el mundo.

Esta arquitectura proporciona un entorno robusto y flexible para los desarrolladores de aplicaciones, basadas en HTML5. Debido a la versatilidad que posee Tizen se está convirtiendo rápidamente en el entorno de desarrollo preferido para aplicaciones y servicios móviles. La API y el *SDK* de Tizen permiten que los desarrolladores utilicen tecnologías web para escribir aplicaciones que se ejecuten en múltiples segmentos del dispositivo (Tizen Org, 2014).

En el gráfico 5 es posible apreciar la arquitectura que se encuentra estructurada en el cual Tizen al igual que la arquitectura de Android se encuentra dividida en capas la mismas que facilitan el desarrollo de aplicaciones. Es así como se podrá describir de la siguiente manera.

1. El núcleo. Esta arquitectura y sistema operativo como se dijo anteriormente se encuentra basada en el kernel de Linux adaptado para un *hardware* de media capacidad, actualmente su uso es común entre dispositivos vestibles, Smart TV's y algunos teléfonos de la compañía Samsung.

2. Servicios de Núcleo. Esta capa encierra las drivers que gestiona el sistema como se observa en la figura 5 se puede encontrar drivers encargados de gestionar la posición, WiFi y teléfono, TV, reloj según sea el caso, además otro componente es la plataforma web que a su vez está conformada por:
  - Webkit. Es un motor de JavaScript semejante a los encontrados en los navegadores de escritorio, fue implementado en C++ y permite ejecutar código de lenguaje llamado con el mismo nombre JavaScript.
  - Runtime Core. Es aquel que permite generar eventos de tipo W3C DOM. Por ello esta capa es encargada de la administración del ciclo de vida de las aplicaciones web. Permite el acceso a los recursos vía la API de JavaScript y además se encarga de gestionar las tareas encargadas para la visualización de la interfaz de usuario (Jong Deok Choi, 2012).
  - Tizen Web API. Son aquellas bibliotecas que facilitan el acceso a algunos recursos del hardware llamadas al momento al momento de ejecutar una aplicación.
  - Núcleo instalador. Es encargado de gestionar la instalación y desinstalación de las aplicaciones Web o de aplicaciones nativas.
  - Núcleo de App de Seguridad. Es aquel que brinda soporte a la seguridad que se ejecutan en el dispositivo. Por ello es que toda aplicación o widget que se ejecuta en Tizen debe constar con un certificado de un autor o distribuidor registrado previamente en Samsung caso contrario se registrará e instalará como una aplicación no confiable. Los certificados de autenticidad y verificación de la aplicación se verifica en un archivo de tipo XML.
3. Aplicaciones para Tizen. Como se observa en la parte superior de la figura 5 es posible desarrollar dos tipos de aplicaciones, la misma que dependerá del grado de dependencia para el dispositivo.
  - Aplicaciones de tipo web
  - Aplicaciones nativas

Para el caso de aplicaciones de relojes inteligentes existen tres formas de diseñar aplicaciones como se observa en la figura 12, la misma que se verá más adelante, debido a esto por ahora no se hace mayor detalle en aquello.

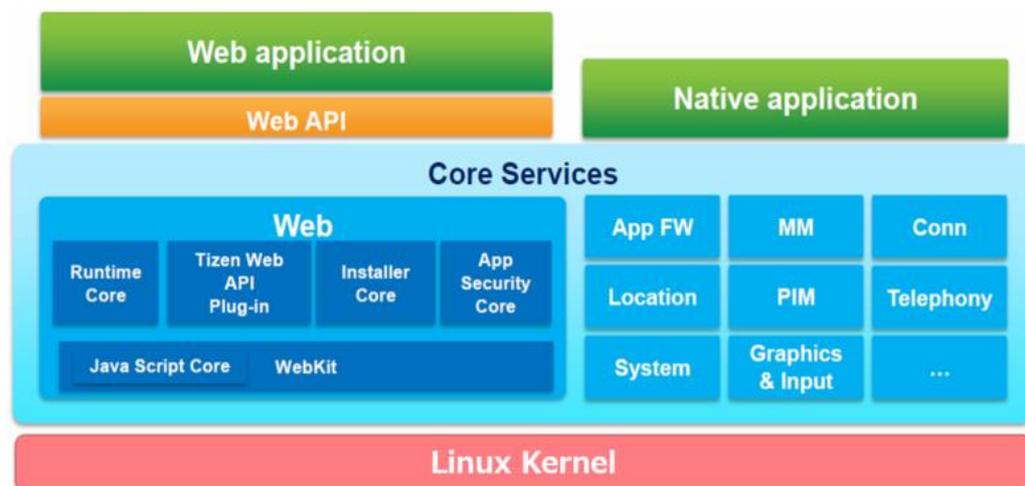


Figura 5. Arquitectura Sistema Operativo Tizen.

Fuente (Guangnan Ni, 2014)

Donde es posible albergar el desarrollo de aplicaciones de tipo web en cual se pueda llegar a interactuar con el usuario directamente o a través de un *Smartphone* como se puede observar en la figura 6 donde aquellos dispositivos debido a su capacidad. Para más detalle el proceso básico de interacción entre bloques es mostrado a continuación en el siguiente esquema.

## 2.4 Dispositivos Electrónicos Vestibles

Los dispositivos vestibles también conocidos en inglés como *wearables* han tenido un auge en esta última década. Estos dispositivos han sido destinados de acuerdo a su uso en campos como la medicina, entretenimiento, educativo y aunque no tan conocidos pero también para uso industrial y militar. Aquellos dispositivos también se caracterizan de ser dispositivos autónomos o semiautónomos (David Cuartielles, 2015). Entre los artículos más utilizados encontramos respectivamente los siguientes dispositivos:

- Relojes inteligentes.
- Bandas de deporte.
- Gafas.

Sin embargo hoy en día, se consta de una alta gama de accesorios como los mostrados a continuación.



(a) Banda de Salud



(b) Auriculares Samsung Circle



(c) Gafas inteligentes de Google



(d) Gafas inteligentes de Intel



(e) Reloj inteligente de Samsung



(f) Reloj inteligente autónomo de Samsung

Figura 6. Ejemplos de dispositivos avanzados portables o vestibles en el mercado en el presente año.

Fuente: (Paz Felicia, 2015)

## 2.4.1 Relojes Inteligentes

Los primeros modelos de relojes digitales eran capaces de realizar funciones básicas como cálculos en calculadoras básicas, traducciones a otros idiomas, y ejecutar juegos complejos, en la actualidad los relojes inteligentes son capaces de desempeñar funciones mejoradas, incluso ser un teléfono inteligente o ser un computador portátil. Muchos relojes inteligentes pueden ejecutar aplicaciones

móviles, algunos otros se ejecutan en un sistema operativo de teléfonos inteligentes para controlarlos, y unos pocos ya tienen las capacidades técnicas de un teléfono móvil (Loton, 2014).

Estos dispositivos pueden incluir características como un termómetro, altímetro, velocidad, barómetro, brújula, cronometro, calculadora, teléfono móvil, GPS, pantalla gráfica, manos libres, agenda, escuchar el tiempo. Donde en general cualquier de estos necesita de un procesador de datos, memoria, periféricos de entradas y salidas. Con ello se puede recoger información de los sensores internos o externos, controlar o recuperar datos de otros instrumentos o equipos usando únicamente una aplicación o widget. Además dichos widgets se apoya de tecnologías inalámbricas como *Wi-Fi* y *GPS*. Sin embargo, es posible que un reloj inteligente sólo sirva como interfaz para un sistema notificación remota proveniente de un teléfono inteligente del mismo usuario, como en el caso de los relojes que utilizan la tecnología *Bluetooth* y *Wi-Fi* para sincronizar los mensajes al igual como eventos que llagan al teléfono.



Figura 7. Algunos de los relojes inteligentes más avanzados del 2014 según la revista Trends. Gear-S Segundo desde la izquierda.

Fuente: (Trends, 2015)

## 2.5 Arquitectura Cliente - Servidor

Define aquella tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso de manera distribuida para un grupo de un trabajo en una red de datos (Cabello, 2014).

En el diseño de *software* de tipo web dicha arquitectura deberá funcionar conjuntamente con lenguajes de programación donde es posible asignar roles específicos hacia el cliente o al servidor. Así es posible gestionar los recursos de mejor forma ya sea en la parte física como por ejemplo el uso del procesamiento en una terminal de un usuario o su vez para el *software* con en la reutilización de código para un servidor.

Con ello hoy en día es posible diferenciar y catalogar a la lógica de negocio con la presentación y el procesamiento de la información. Dichas características están comprendidas en una clasificación conocida como *Front-End* y *Back-End* (Smith, 2013).

## **2.5.1 Front-End**

Este término es utilizado en el diseño web orientado al desarrollo de aplicaciones ricas en contenido multimedia, bajo el esquema de tipo cliente servidor se debe entender como el producto final o *software* que se ejecutara en el lado del usuario a través de una consola, motor web o máquina virtual una vez que se haya descargado del servidor al cual se lo hizo la solicitud (Smith, 2013).

### **2.5.1.1 HTML5**

Es una nueva versión del lenguaje de etiquetas que en conjunto con otras tecnologías web sirve para la publicación de contenido rico en multimedia para páginas dinámicas o estáticas a través de un navegador web.

Hasta estos últimos años el contenido desarrollado era para usuarios únicamente desde navegadores web o aplicaciones desde un computador, gracias a tecnologías basadas en la arquitectura cliente servidor y miniaturización se ha logrado embeber un motor web en dispositivos portables como relojes inteligentes (Aubry, 2014).

Por esta razón todo comenzó un tiempo atrás con su primera versión HTML, creada para la nueva formación básica de páginas web, con el limitado objetivo que se había planteado HTML conllevó a que otras empresas surjan para desarrollar nuevos lenguajes que sean más entretenidas agregando características a la web nunca antes implementadas. Esto al parecer parecía un juego para los desarrolladores con el

tiempo fue adquiriendo más sofisticadas aplicaciones, esto fue la causa de cambiar el concepto web.

De todas las opciones propuestas hubieron dos sobresalientes Java y Flash fueron las más exitosas que representó el futuro de Internet y que fueron masivamente adoptadas hasta hace pocos años atrás. Esto dio paso para que nuevas personas vean esto como un negocio por la interacción social que había, el inconveniente de estos dos lenguajes fue sus complementos los (*plugins*) ya que entre estas no existía comunicación e integración alguna entre aplicaciones y documentos.

HTML5 proporciona esencialmente con tres características: funcionalidad, estilo y estructura, también es considerado como la composición de HTML, CSS y JavaScript, todas estas tecnologías actúan casi por cuenta propia y también como una unidad independiente, HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y por ultimo JavaScript procesa la información y ejecuta un algoritmo determinado.

Es diseñado para ser utilizable por todos los desarrolladores de *Open Web*, puede usar varios recursos del dispositivo, que se clasifican en varios grupos según su función.

Incluso después de toda esta unión, de hecho, la estructura sigue por mucho siendo la parte más esencial de un documento. Este aporta con factores necesarios para situar contenido estático o dinámico siendo un componente básico para aplicaciones. Ahora la estructura debe proveer forma, organización y flexibilidad.

Cuando uno se requiere trabajar y desarrollar sitios webs y aplicaciones con HTML5 lo primero que debemos saber es que dispone y está constituida de las siguientes características.

- **Semántica:** Permite describir con mayor precisión cuál es su contenido.
- **Conectividad:** Permite comunicarse con el servidor de formas nuevas e innovadoras.
- **Off-line y almacenamiento:** Permite a páginas web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.

- Multimedia: Otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.
- Gráficos y efectos 2D/3D: Proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son el lienzo 2D, WebGL, SVG, etc.
- Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del *hardware*.
- Acceso al dispositivo: Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.

### 2.5.1.2 CSS

Las hojas de estilo en cascada, tienen como en función establecer reglas de representación de un documento en un medio o dispositivo. Mediante estas reglas podremos establecer medidas, colores o cualquier otra característica de representación de una página web, para que se vea reflejada en una pantalla de monitor de un dispositivo móvil, una tablet, una impresora etc.

La función principal de CSS es por lo tanto la de permitir separar el contenido y la estructura que se define en un documento HTML, de la representación, que queda a cargo de las hojas de estilos.

CSS3, Al tiempo que la versión 2 de CSS se desarrollaba, un equipo de trabajo ponía manos a la obra en lo que sería CSS3. Poco a poco se fueron conociendo sus principales características más importantes es así como hoy es que se destacan las siguientes características (Gauchat, 2012).

- Nuevas alternativas para dibujar bordes con el uso de opciones tales como color, imágenes y radio o redondeado.
- Propiedades para el trabajo con fondos, y el uso de degradados igualmente la posibilidad de incluir múltiples imágenes.
- Uso de sombras para texto (*text shadow*).
- Características relacionadas con la interfaz de usuario.
- Incorporación de transición y también funciones de animación.

### 2.5.1.3 JavaScript

Su función principal es permitir la creación de páginas dinámicas, con código que se encuentra alojado en el servidor sin embargo ante una petición desde un terminal es descargado inmediatamente a este para luego ejecutarse desde el lado cliente, disminuyendo las tareas del servidor y la cantidad de peticiones que se les solicite. Por sus características, resulta útil en acciones como validación de formularios, desplegar y aplicar efectos visuales, y exhibir avisos en pantalla, también cabe recalcar que su uso cobro mayor fuerza a partir del éxito de AJAX y el importante impulso en la web por parte de Google. La inclusión de un código JavaScript dentro de un documento HTML puede realizarse ubicándolo en el encabezado entre las etiquetas a continuación.

```
<script type="text/javascript"> y </script>
```

### 2.5.1.4 AJAX

El término AJAX proviene del acrónimo *Asynchronous JavaScript And XML*. El punto fuerte de AJAX es trabajar con datos de manera asincrónica, valiéndose de JavaScript como lenguaje del lado cliente para manejar datos que le llegan desde el servidor. De esta manera, el motor AJAX trabaja como un intermediario entre el cliente y el servidor, pero en lugar de demorar procesos, los administra de tal forma que es posible (Smith, 2013).

Esta posibilidad cambia el paradigma de la necesidad de una recarga completa de la página y permite construir aplicaciones web más potentes, emulando incluso a muchas de las soluciones que se vean posibles en software de escritorio, estos son los lenguajes que intervienen:

- HTML/XHTML y CSS: son los que permiten la representación en el navegador
- XML y JSON: son los que permiten realizar el intercambio asíncrono de los datos que sean necesarios.
- XMLHttpRequest: es el que permite realizar el intercambio asíncrono de los datos que sean necesarios

- DOM: es la interfaz que permite acceder a las partes de un documento como si fueran objetos y a partir de eso modificarlas
- JavaScript: es el lenguaje utilizado para unir y compilar todo.

#### 2.5.1.4.1 Aplicaciones Desconectadas

Hasta ahora, siempre que se pensó en una aplicación AJAX se le consideró conectada a Internet. Sin embargo se ha ido utilizando cada vez más un nuevo concepto que apareció en los últimos tiempos para aplicaciones de escritorio o widgets.

Bajo esta premisa, el objetivo es que una aplicación desarrollada con técnicas de tipo web conectada, como AJAX, pueda funcionar de manera desconectada (*offline*). Hay varias soluciones para crear este tipo de aplicaciones. Basadas en *plugins* como Adobe AIR, Google Gears o Microsoft Silverlight en futuras versiones, y basadas en estándares, como HTML5. (Smith, 2013)

Muchas plataformas móviles, como iPhone, LG, Samsung y otras tienen plataformas de desarrollo de widgets o WebApps, que no son más que aplicaciones AJAX que funcionan instaladas en los equipos como aplicaciones desconectadas con posibilidad de conectarse. Esto abre un panorama ante nosotros para incluso poder distribuir aplicaciones en tiendas descargadas.

#### 2.5.1.5 JSON

*JavaScript Object Notation* (JSON) es utilizado principalmente para el intercambio de datos con lenguajes de programación de alto nivel. En la actualidad se emplea mucho en aquellos proyectos donde con frecuencia se produce un gran nivel de flujo de datos asíncronos entre cliente/servidor, ya que permite tener la información organizada con facilidad como se observa en la figura 8 para dar acceso a información para ser manipulada (Keith, 2015) sin importar desde el lugar y lenguaje que solicite dicha información.

De este modo la nomenclatura que se ha convenido para la agrupación de información son los corchetes “{ }” para los objetos y llaves “[ ]” para los arreglos de datos (Keith, 2015). Donde aquello es igualmente es apreciado en la figura 8, ya que se observa como la información es agrupada bajo los criterios antes mencionados

paréntesis y corchetes. En el caso de la figura 8 se consta de tres objetos cuya contenido no contiene un nombre definido, por ello es que se asigna por defecto números únicamente para visualizarlos, a su vez estos objetos constan de un arreglo de tres datos como son nombre, descripción y revisión.

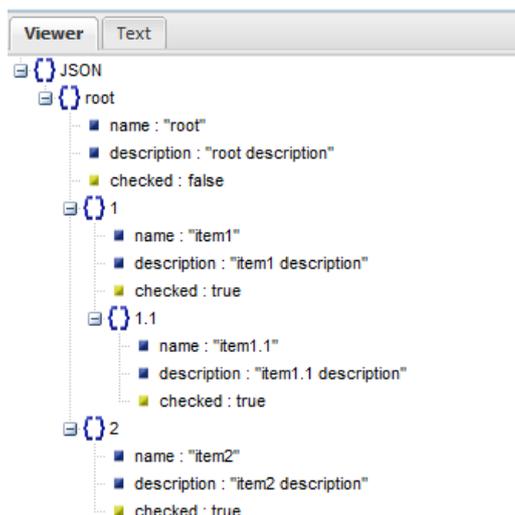


Figura 8. Ejemplo de una matriz de datos estructurado en formato JSON.

Fuente: (Autor)

### 2.5.1.6 El navegador web

Es una aplicación que permite al usuario acceder a Internet y navegar vía web. Además aquellas pueden variar según el sistema operativo donde se ejecute. Actualmente la mayoría de navegadores incluyen HTML5 y las hojas de estilo CSS3 solo se tienen en cuenta en las últimas versiones. Por otro lado, la integración de HTML5 y de CSS3 es todavía parcial y varía de un navegador a otro. Por este motivo es preciso escoger para nuestro estudio los principales actores de esta innovadora tecnología para asegurar una vista de conjunto de los aportes más importantes que afectan o afectarán a la edición y publicación de documentos web.

Uno de estos componentes es el motor que se encargara de ejecutar el código para la integración completa de HTML5, como se puede observar en la tabla 2 donde se encuentran los principales motores de JavaScript en la actualidad mayormente utilizado en el mercado.

Tabla 2.

Principales motores JavaScript de los navegadores usados en la Web.

Navegador Web	Motor de JavaScript	Versión	Licencia	Empresa
Chrome	V8	7	Apache 2.0	Google
Firefox	Spider Monkey	38	Open source	Firefox Fundación
Safari	Nitro	8.0.2	GNU/LGPL	Apple

Fuente: (Autor)

Los navegadores más utilizados tardarán todavía algunos meses o tal vez hasta años. Aunque, habiendo estado implicadas en la elaboración de HTML5 las principales firmas de software, es de suponer que la evolución de este lenguaje será de una manera rápida y eficaz.

## 2.5.2 Back-End

El Back-End en cambio trabaja del otro lado (Servidores), los mismos que forman parte de una gran variedad como lo son de archivos, correos, información, red entre otros. El *software* utilizado en este lado maneja procesos lógicos que dan servicios o cumplen con procesos solicitados por los clientes. Pero en esta parte también debemos tener en cuenta que es importante los componentes que se utilizan en *hardware* como en *software* para proveer una alta disponibilidad en su servicio.

Así deberá tener gran poder de procesamiento y de igual forma espacio de almacenamiento para poder satisfacer las necesidades de las aplicaciones y de la misma forma en la parte de software que en la actualidad se utiliza los administradores de base de datos.

### 2.5.2.1 MySQL

Es un sistema de base de datos por sus siglas en ingles SGBD, DBMS desarrollado en C/C++ muy conocido por su simplicidad y notable rendimiento.

Aunque carece de algunas características avanzadas disponibles en otros SGBD, es una opción atractiva tanto para aplicaciones comerciales como de entretenimiento y para investigación. MySQL se encuentra disponible para múltiples plataformas, la seleccionada para este proyecto es bajo Windows. Sin embargo, las diferencias con cualquier otra plataforma son prácticamente nulas, ya que la herramienta utilizada es el cliente *mysql-client*, que permite interactuar con un servidor MySQL (Local o remoto) en modo texto. De este modo es posible realizar todas las acciones, a través de Internet, sobre un servidor remoto (George Coulouris, 2013).

### **2.5.2.2 PHP**

Es un lenguaje de programación que se puede ejecutar en el servidor, en 1994 inicio fue desarrollado en lenguaje C y actualmente se encuentran en la versión 5.6 existe 3 versiones estables. Este lenguaje también tiene la opción de embeberse en un archivo HTML con un formato propio. Además PHP tiene la capacidad de conectarse con la mayoría de motores de base de datos por otra parte uno de las ventajas principales de este lenguaje también se basa en el uso de las cookies para registro temporal de variables, útil en el control de sesiones para usuarios (Smith, 2013).

## CAPÍTULO 3

# DESCRIPCIÓN DE HARDWARE Y SOFTWARE

### 3.1 Descripción de Hardware

#### 3.1.1 Gear S

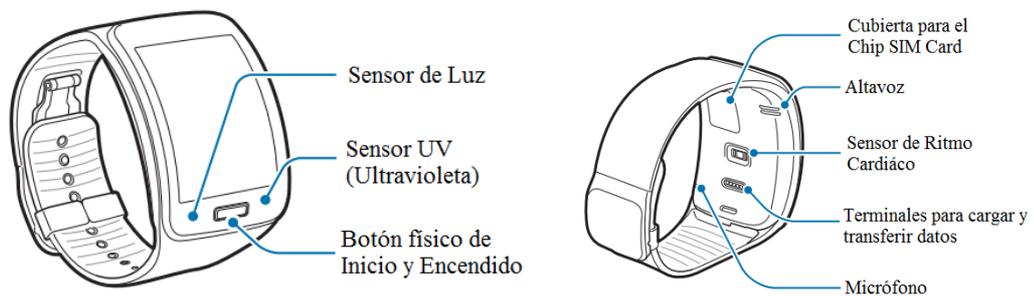
Es el primer reloj electrónico inteligente autónomo en el mundo fabricado por la compañía Samsung. Su aspecto exterior mostrado en la figura 9 permite apreciar que se encuentra provisto de una pantalla curva de tipo AMOLED acoplado a un sujetador de tipo correa a base de goma plástica. El reloj Gear-S también viene provisto de algunos servicios para el cuidado de la salud gracias a los sensores que son incluidos en el reloj que permite el sensado de cantidad de irradiación ultravioleta y ritmo cardíaco como se observan en la figura 10 (literal a y b respectivamente). Además al momento de comunicarse con otras personas el usuario no depende de ningún *Smartphone* para el intercambio de información (Calmon, 2014).



Figura 9: Dispositivo Reloj Inteligente Modelo Gear-S.

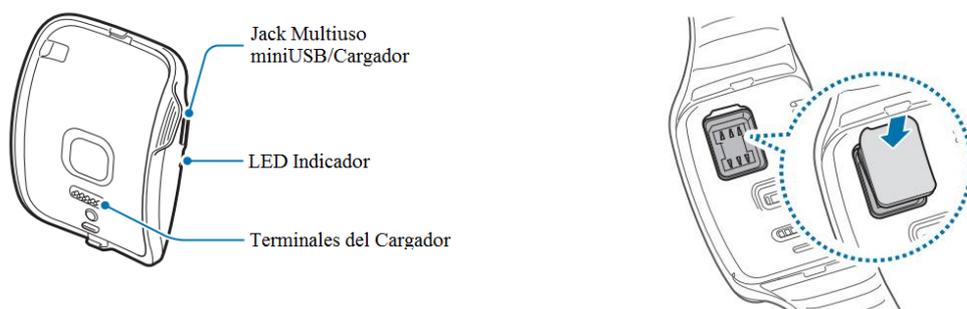
Fuente (Calmon, 2014)

Por otra parte como se observa en la figura 10 (literal d), el reloj incorpora los mismos servicios que actualmente es posible encontrar en un teléfono inteligente como una *SIM card* que brinda conexión de datos 3G y telefonía, además incorpora *Wi-Fi*, *Bluetooth* y *GPS* como sus principales características.



a) Sensores parte frontal del reloj

b) Componentes de la parte trasera del reloj



c) Cargador e interfaz de comunicación entre PC y reloj

d) Lugar donde se aloja la *SIM Card*

Figura. 10 Componentes del Reloj *Gear-S*.

Fuente: (Evans, 2011)

Este modelo de relojes inteligentes también conocidos en inglés como *Smartwatches* utilizan el sistema operativo TIZEN, como se dio a conocer en el capítulo 2 esta plataforma permite el desarrollo de aplicaciones de tipo nativas y de tipo web. De este modo en la tabla 3 se aprecia sus especificaciones técnicas disponibles hasta el momento por la compañía Samsung.

Tabla 3.

Especificaciones técnicas del Gear-S.

Categoría	Descripción	Especificaciones
Tamaño	Dimensiones	39.9 x 58.1 x 12.5 mm
	Peso	67 g / 84 g
Resistencia	Standard	- Certificación IP67: a prueba de polvo y agua
Pantalla	Tipo	Super AMOLED touchscreen capacitivo, 16M colores
	Tamaño	360 x 480 pixels, 2.0 pulgadas
Sensores	Variable	- Sensor acelerómetro
		- Giroscopio
		- Monitor de ritmo cardíaco
		- Sensor de luz ambiente
		- Sensor UV
		- Barómetro
Memoria	Agenda telefónica	Entradas y campos prácticamente ilimitados, Foto de llamada
	Registro de llamadas	Prácticamente ilimitado
	Slot de tarjeta	No
	Memoria interna	4GB, 512MB RAM
Procesamiento	CPU	- Procesador dual-core a 1GHz
Conectividad	Posicionamiento	- GPS con soporte A-GPS
	Tecnologías	Puerto infrarrojo
		- EDGE
		- 3G HSDPA / HSUPA
		GSM 900 / 1800 o 850 / 1900 - HSDPA 900 / 2100 o 850 / 1900
		- Wi-Fi 802.11 b/g/n
		- Bluetooth v4.1 LE
- USB 2.0		
BATERÍA	Activo	Standard, Li-Ion 300 mAh
	Stand-by	Hasta 2 d
Sistema Operativo	OS	Tizen OS

Continúa 

<b>Widgets</b>	GPRS	Si
	Mensajería	SMS, MMS, email, IM
	Navegador Web	Disponible a través de Samsung Store
	Reloj	Si
	Alarma	Si
	Colores	Negro, Azul, Blanco
	Organización	Cronómetro
	Salud	Podómetro
	Cámara	No
	Tipos de Ringtones	Si
	Ringtones Personalizados	Descargas
	Vibración	No
	<b>Multimedia</b>	Formatos/Funciones
- Reproductor de audio MP3/AAC/AAC+/eAAC+		
- Cancelación activa de ruido con micrófono dedicado		
- Comandos de voz		

Fuente: (Calmon, 2014)

### 3.1.1.1 Funcionamiento

En este modelo de relojes como es el Gear-S se puede encontrar tres formas de funcionamiento como se observa en la figura 11 los cuales están comprendidos en Enlace (Maestro y seguidor), Integrado e Independiente. De este modo cada funcionamiento dispone de las siguientes formas de trabajo.

- Enlace (Maestro y seguidor) para este tipo de funcionamiento es necesario que el teléfono y reloj inteligente interactúen de forma que exista una aplicación, el cual se encargue de que ambos dispositivos intercambien información en un vínculo conocido como productor-consumidor, ya que permitirá atender alguna petición proveniente de otra aplicación del mismo teléfono.
- Integrado. Para este funcionamiento igualmente será necesario de ambos dispositivos el teléfono y el reloj, la diferencia es que la aplicación encargada de formar un vínculo y además aplicaciones que se valen de este vínculo para proveer de información al reloj esta vez son uno solo, se puede apreciar mejor en la figura 11 en la parte central de la imagen.

- Independiente. Esta forma de trabajar no requiere algún hardware adicional para su funcionamiento, ni tampoco para la gestión y administración de aplicaciones. Este se debe a que con ayuda del cargador mostrado en la figura 10 literal c, es posible agregar o desinstalar aplicaciones a través de un computador con ayuda de un entorno de desarrollador llamado Tizen que se describen más adelante en el capítulo 3 ítem 3.2.1.

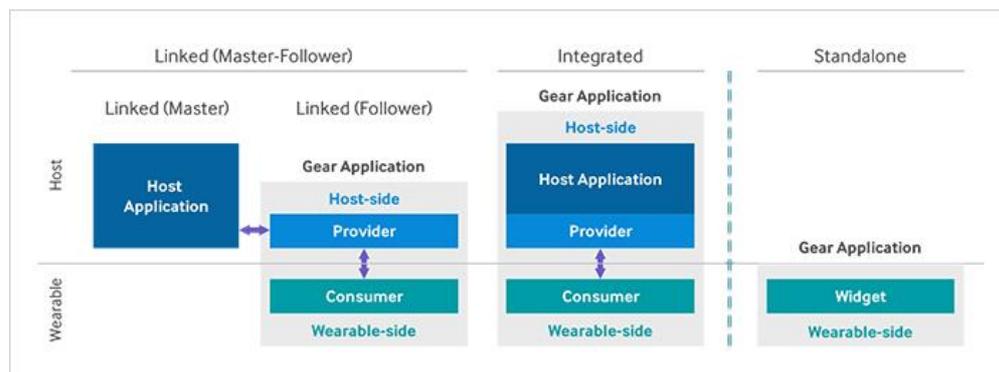


Figura 11. Formas de Funcionamiento del Gear-S.

Fuente (Samsung, 2014)

Un aspecto importante para el control del reloj es la administración de las aplicaciones el cual se realiza a través del teléfono inteligente del mismo fabricante, al momento únicamente compatible para versiones para su sincronización desde la gama de Samsung S4 en adelante. Como se puede apreciar en la figura 12, la aplicación utilizada para el control y la instalación de widgets se llama “Gear-S” y está disponible en la tienda de Google Play al igual como en Galaxy App.

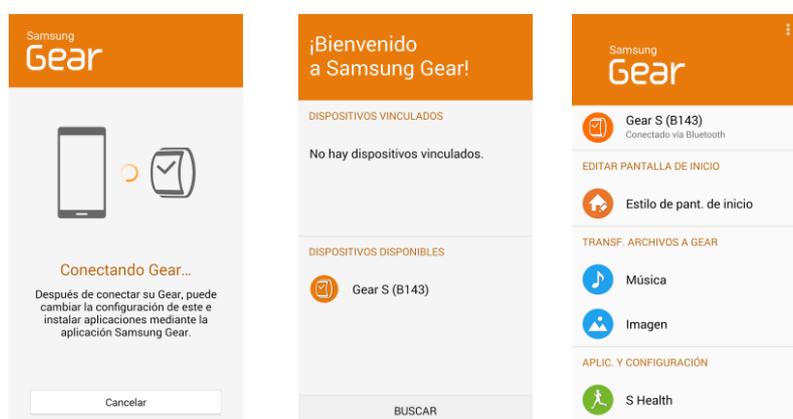


Figura 12. App Samsung Gear, Administración de widgets en el reloj desde el teléfono.

Fuente: (Autor)

### 3.1.2 Sensordrone

Es un dispositivo semiautónomo desarrollado por la empresa *Sensorcorn*<sup>®</sup> en el año 2011, su función es la adquisición de algunas variables físicas ambientales que puede ser controlada o pre programada de forma remota (Sensorcorn, 2012). A través de circuitos especializados para un uso en específico también conocidos como ASIC con Sensordrone es posible unir todas las funcionalidades en un reducido espacio como se observa en la figura 13 donde aprecia mejor el lugar donde se encuentran ubicados los sensores en el dispositivo. De esta forma la integración de *Hardware*, *Firmware* y Sensores se hace transparente para el programador o desarrollador que desea crear una aplicación en especial (Manraj Singh, 2014).

Aquel dispositivo se encuentra compuesto de 8 sensores que son capaces de sensar hasta 11 variables físicas a la vez además posee un puerto de expansión de salidas y entradas digitales (TTL UART e i2C) que permite conectar sensores como EKGs o impresoras termales (Sensorcorn, 2015).

A través de Sensordrone es posible manejar y capturar las siguientes variables a continuación:

1. Sensor de Gas de precisión Electro-química: Se encuentra calibrado para medir monóxido de carbón sin embargo es posible utilizarlo para medir alcohol e hidrógeno.
2. Sensor de Gas de tipo Oxidantes: Es utilizado para medir principalmente gases como ozono, dióxido de nitrógeno etc.
3. Sensor de Gas de Reducción: Es utilizado para medir principal mente gases como metano, propano y otros hidrocarbonos.
4. Temperatura: Encargado de medir con los últimos estándares de comunicación.
5. Humedad: Encargado de medir la cantidad de agua por cada centímetro cubico.
6. Termómetro de tipo infrarrojo: Las actualizaciones de *Firmware* pueden ser ejecutadas en la plataforma Encargado de medir la temperatura.
7. Sensor de Proximidad:
8. Intensidad de luz:
9. Presión: puede ser usado como Barómetro.
10. Altitud.

## 11. Sensor de intensidad Cromática RGB.

Para una mejor referencia del tamaño de un Sensor drone se puede apreciar en la figura 13 el tamaño que ocupa este dispositivo comparado a un llavero de auto:

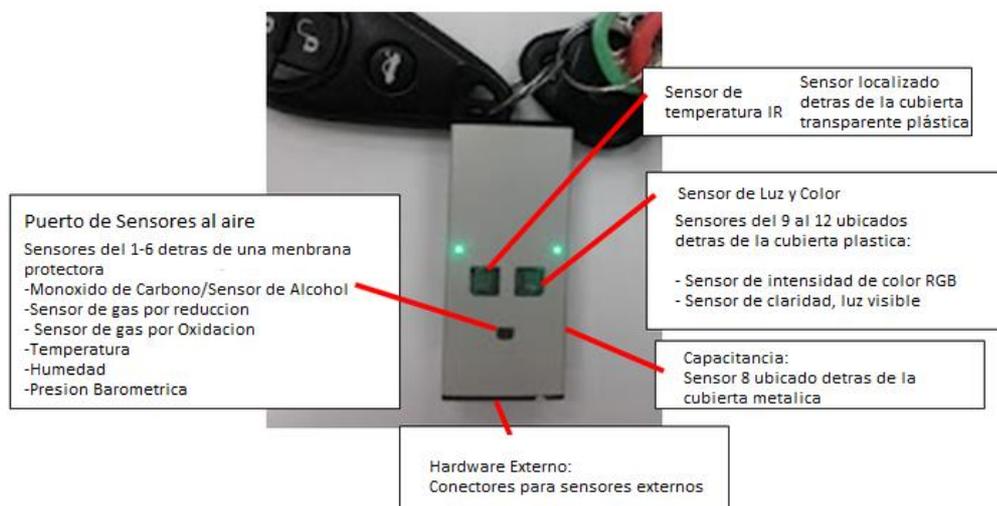


Figura 13. Descripción general de los sensores en el Sensor drone.

Fuente: (Sensorcorn Inc, 2014)

## 3.2 Descripción de Software

### 3.2.1 Entornos de Desarrollo IDE

Al momento de empezar a implementar un proyecto de software el ambiente de desarrollo integrado o IDE puede llegar a tomar un rol muy importante en la codificación de nuestro algoritmo. Es por ello que es tomado en cuenta como una herramienta importante a la hora de dimensionar el número de recursos humanos el así como el tiempo que trabajaran en el proyecto. De este modo para este proyecto se trabajado con los siguientes IDEs: Netbeans, Android Studio y Tizen; además también se hizo uso del navegador web Chrome para las pruebas de funcionamiento.

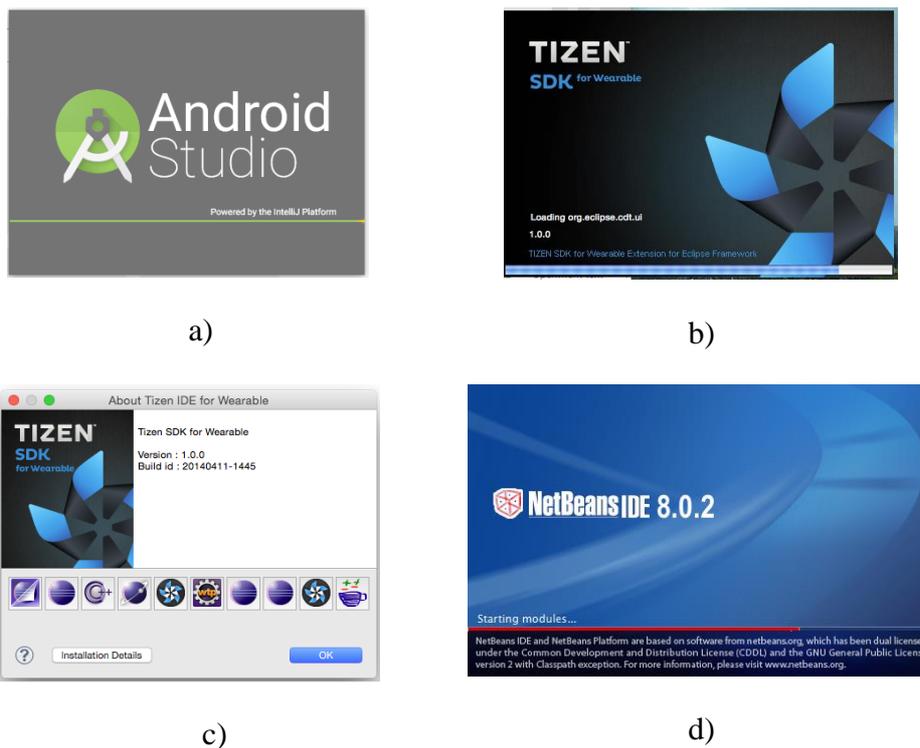


Figura 14. Herramientas para desarrollar el sistema de monitoreo con MCS.

Fuente (Autor)

- **Android Studio**

Google en un inicio dispuso de Eclipse como único IDE para el desarrollo de aplicaciones para sus dispositivos Android. Sin embargo a medida que Android surgía con nuevas aplicaciones y versiones de sistemas operativos se fueron encontrando varias inconvenientes en la manera de continuar con el desarrollo de nuevas apps esto condujo al creciente uso de metodologías para mantener la compatibilidad de una aplicación entre distintas versiones de sistema operativo. (Sakila Banu, 2013).

Debido a que Android utiliza Java es posible encontrar algunos IDEs en la web que permiten desarrollar aplicaciones para Android muchos de ellos de tipo *open-source* sin embargo estos también llegaron a tener el mismo inconveniente que tenía Eclipse. Es así como Google creador de Android lanza al mercado su IDE oficial llamado Android Studio observado en la figura 14 (literal a) que solventaba y solucionaba algunos problemas presentados en compilación y compatibilidad entre versiones de sus propios sistemas operativos.

Los IDE Android estudio al igual que Eclipse para la creación de aplicaciones móviles basan sus repositorios y bibliotecas en envoltorio de herramientas llamada ADT (*Android Developer Tool*) necesarias para probar y depurar las aplicaciones. Es así como los componentes están comprendidos:

1. Plugins.
2. Android SDK Tools.
3. Emulador Android.

- **Tizen IDE**

Esta IDE es una herramienta basada en java y personalizada de Eclipse véase la figura 14 (literales c y d) donde se aprecia las adecuaciones hechas en eclipse para este nuevo entorno. Este es IDE es relativamente nuevo ya que lleva pocos meses en el software destinado para el desarrollo. Tizen ha sido modificado teniendo en cuenta algunas para desarrollo de aplicaciones de tipo web a través de HTML5 y aquellas nativas basadas en C++.

Un punto importante mencionar es para desarrollar un aplicación y probarla en un simulador o en un dispositivo físico será necesario seguir un procedimiento para poder obtener un certificado de desarrollador para correr, descargar y distribuir mi aplicación.

- **NetBeans**

Este IDE recomendado por Oracle posee un vasto conjunto de herramientas que permiten el desarrollo de aplicaciones para varias tecnologías de desarrollo de software. Para el desarrollo del proyecto MCS se lo ha realizado con la versión más reciente 8.0.2 debido a la facilidad que presenta la creación de aplicaciones HTML5 y PHP.

## **3.2.2 Bibliotecas y Software de Terceros**

### **3.2.2.1 API Google Maps**

Esta API ha sido diseñada por la compañía estadounidense Google el cual brinda la posibilidad de interactuar con un mapa a nivel mundial que puede ser incrustado

en nuestro sistema web o una aplicación para dispositivos inteligentes. Es así como esta API utiliza JavaScript, HTML5 y AJAX para aplicaciones que requiera de un navegador web, sin embargo esta API fue también diseñada en Java para ser utilizado en aplicaciones para dispositivos móviles inteligentes.

En la actualidad Google ha designado una plataforma echa para desarrolladores en ella consta de tres versiones publicadas y distribuidas para la creación de aplicaciones que se ejecuten sobre un navegador web (Loton, 2014). Cada versión es necesaria registro y obtención de una llave que permite la ejecución e incrustación de mapas en el sitio web de quien desee utilizar este recurso para la versión 3 ya no es necesario la utilización de una llave (Svennerberg, 2014).

### 3.2.2.2 CanvasJS

Es una biblioteca realizada en JavaScript y HMTL5 que elabora una gran variedad de graficas estáticas o dinámicas como se observa en la figura 15. CanvasJs es un software propietario liberado bajo la licencia “*Creative Commons Attribution – non Commercial*”, pertenece a la empresa Fenopix que puede ser usado libremente bajo fines no comerciales (Sunil Urs, 2015). Además con la creación de objetos JSON es posible ingresar la información para la elaboración de cualquier tipo de gráfica como las que se observa en la figura 15.



Figura 15. Tipos de Graficas encontradas en CanvasJS.

Fuente: (Aubry, 2014)

### 3.2.2.3 Biblioteca Sensordrone

Esta biblioteca es construida con lenguaje Java diseñada para Android, ha sido diseñada por la empresa Sensorcorn para el control del Sensordrone, actualmente es utilizada para el desarrollo de aplicaciones que permitan controlar dicho dispositivo mediante *bluetooth*. Así mismo esta biblioteca es distribuida bajo la licencia libre y sin costo. En la tabla 4 se muestra 8 clases que componen la librería necesarios para el uso del Sensordrone a través de la librería.

Tabla 4.

Clases que componen la biblioteca Sensordrone.

Clases	Descripción y Funcionamiento
CoreDrone	Esta clase principal que permite interactuar con el Sensordrone y sus sensores, Esta es una clase abstracta que es extendida de una clase dependiente de la <i>implements</i> ; todos los métodos de conexión debería establecer allí.
Drone	La clase Drone permite controlar el Sensordrone e interactuar con los sensores de a bordo a través de Java
DroneEventHandler	Es una clase utilizada como manejadora de eventos
DroneEventListener	El DroneEventListener se utiliza para realizar acciones cuando el Sensordrone ha realizado con éxito una medición. Un bloque de numerosos eventos de estado para ser disparado. Si sólo desea trabajar con una serie de eventos particulares, no dude en revisar DroneEventHandler
DroneEventObject	Una clase personalizada utilizado para notificaciones EventListener (extiende EventObject)

Continúa



---

DroneEventObject.droneEventType	Es una extensión de la clase <code>Java.lang.enum</code> y enumera todos los tipos de eventos permitidos. Si no estás en la lista eventos, entonces usted no está adquiriendo el sensado.
DroneStatusListener	El <code>DroneStatusListener</code> se utiliza para realizar acciones cuando el <code>Sensordrone</code> ha comprobado con éxito un estado (es decir, es un sensor especial activado o desactivado). Un bloque de numerosos eventos de estado para ser disparado. Si sólo desea trabajar con una serie de eventos particulares, no dude en revisar <code>DroneEventHandler</code> .
DroneconnectionHelper	Esta clase facilita establecer la conexión con el <code>Sensordrone</code> a través de <i>Bluetooth</i> .

---

Fuente: (Sensorcorn, 2015)

### 3.1.2 Servidor XAMPP

Es una distribución de Apache que contiene varios servicios específicamente Apache, MySQL, PHP y PERL. Esta es la plataforma donde se instalará todo el sistema de monitoreo web. Actualmente es posible instarlo en cualquier sistema operativo es decir multiplataforma. De este modo si es instalado en Windows es conocido como servidor WAMP, en Linux como LAMP y en Mac MAMP (Delisle, 2012).

# CAPÍTULO 4

## IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO

### 4.1 Arquitectura del Sistema

#### 4.1.1 Escenario para el Sistema de Monitoreo con MCS

Para la elaboración de nuestro sistema de monitoreo a través de MCS primero es necesario contemplar el escenario de red donde se llevará a cabo las comunicaciones para el envío de datos provenientes de los dispositivos inteligentes. Como se puede observar en la figura 16 donde se define todos los elementos que trabajaran en conjunto para brindar el servicio de monitoreo.



Figura 16. Esquema MCS para proveedor de información a la Base de Datos.

Fuente: (Autor)

Los elementos que llegan a conformar el sistema de monitoreo son los mencionados a continuación:

- Samsung Galaxy S5
- Sensordrone
- Gear-S
- Servidor
- Access points del campus – WiFi

Así el rol de un usuario con su dispositivo inteligente así como el servicio de cobertura del campus universitario son igualmente importantes para el transporte de información del estado del ambiente y salud del usuario que detecta y recolecta datos cada vez que utiliza la aplicación desarrollada.

## **4.2 Desarrollo de la Plataforma Web**

### **4.2.1 Front-End**

Como se ha podido entender en el capítulo dos, a través de tecnologías como HTML5 (JavaScript, CSS3 y HTML) es posible enriquecer y elaborar nuestro sistema web de monitoreo con contenido agradable a la vista, sin embargo, algo que también es posible es trabajar con el navegador en segundo plano a través de AJAX.

En el esquema cliente-servidor observado en la figura 17, se aprecia el origen de los valores sensados desde el cliente proveedor u donde se crea la información. Las líneas marcadas de rojo indican que dispositivos finalmente pueden observar los datos que están siendo almacenados en el sistema en ese instante. Aquellas líneas negras representan el acuse de recibo y flujo de información realizado a través de TCP para dar confiabilidad a los datos originados en los dispositivos inteligentes.

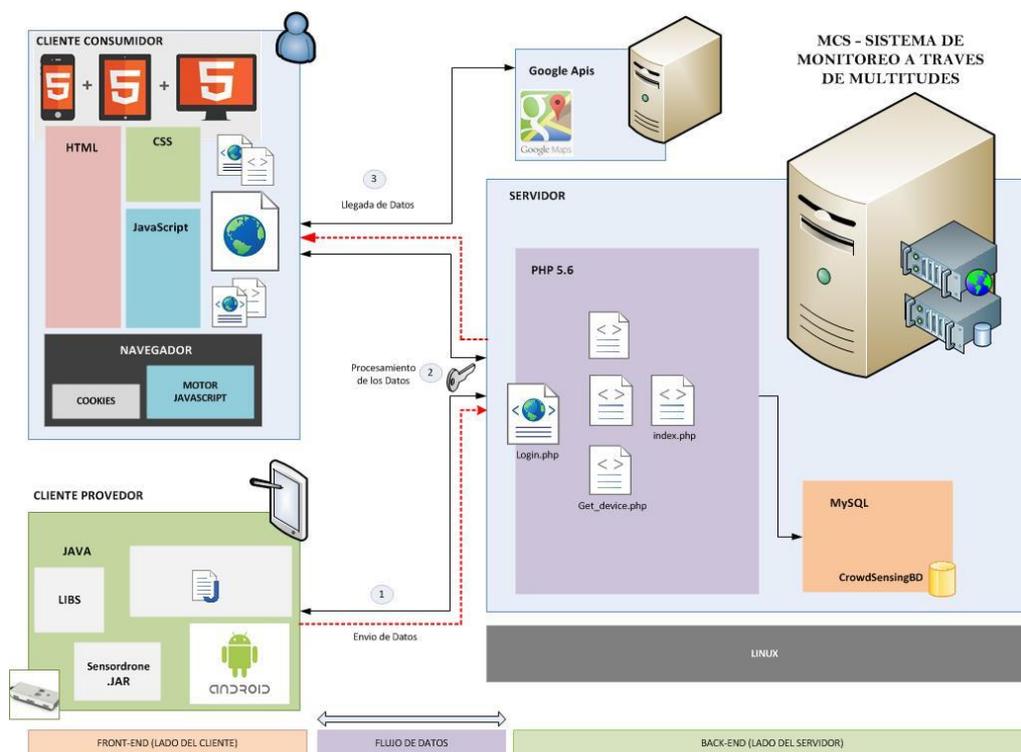


Figura 17 Esquema general de tecnologías utilizadas para la elaboración del Sistema de Monitoreo.

Fuente: (Autor)

#### 4.2.1.1 Interfaz de usuario

Esta parte del sistema es encargada de levantar un entorno gráfico en el navegador web para una fácil visualización en tiempo real del estado de las variables sensadas por los dispositivos móviles inteligentes.

Dicha interfaz de usuario consta de un algoritmo para la descarga y presentación continua de datos desde el servidor. Además considerando al alcance del proyecto, se ha desarrollado el algoritmo observado en la figura 18 de la siguiente manera:

1. Primero, se descarga desde el servidor al navegador el código y algoritmo que se ejecutará a partir que termine de descargar los documentos.
2. A continuación se validará los archivos necesarios para cargar la página web.
3. Se carga los datos a través de AJAX y PHP desde la base de datos al navegador pero todavía no son presentados al usuario.
4. Se presenta los datos antes descargados de modo gráfico la temperatura, presión y humedad con ayuda de la biblioteca de canvasJS.

5. Ahora valida si existe algún error en las gráficas, de ser así lo desplegará por consola y termina el programa.
6. Si no existió error en las gráficas, ahora cargar e inicializar el mapa para ubicar al dispositivo que en envía información y el usuario actual.
7. Si no existió error al cargar el mapa. ahora cargará el gráfico histórico
8. Si no existió error al cargar el gráfico, espera 900 milisegundos y de nuevo se dirige al punto 3, caso contrario despliega por consola el error ocurrido.

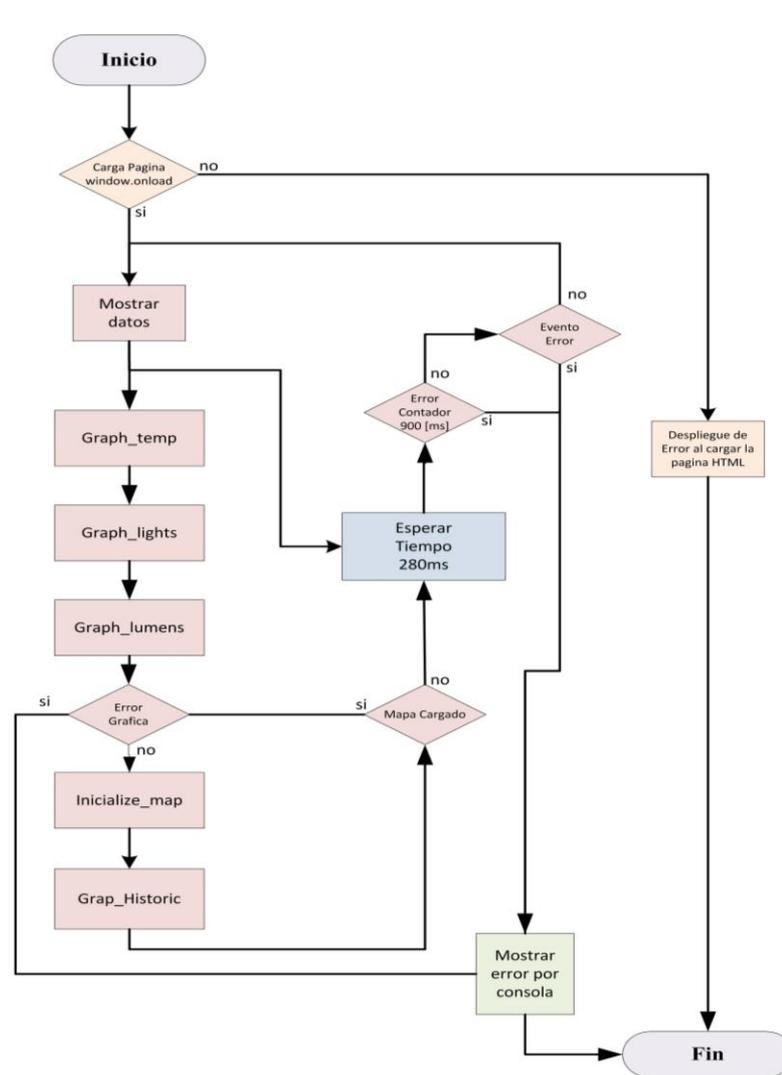


Figura 18. Algoritmo ejecutado en el lado del cliente, una vez descargado la página .php en un navegador web desde el servidor.

Fuente (Autor)

### 4.2.1.2 Visualización de variables sensadas

Gracias a técnicas de programación y comunicación respectivamente JSON y AJAX es posible el intercambio de datos en segundo desde el navegador web y el servidor PHP como se observó en la figura 18. Ahora en el algoritmo de la figura 19 los pasos necesarios para la transmisión de datos en segundo plano donde se detalla el punto 3 del interfaz de usuario antes mencionado.

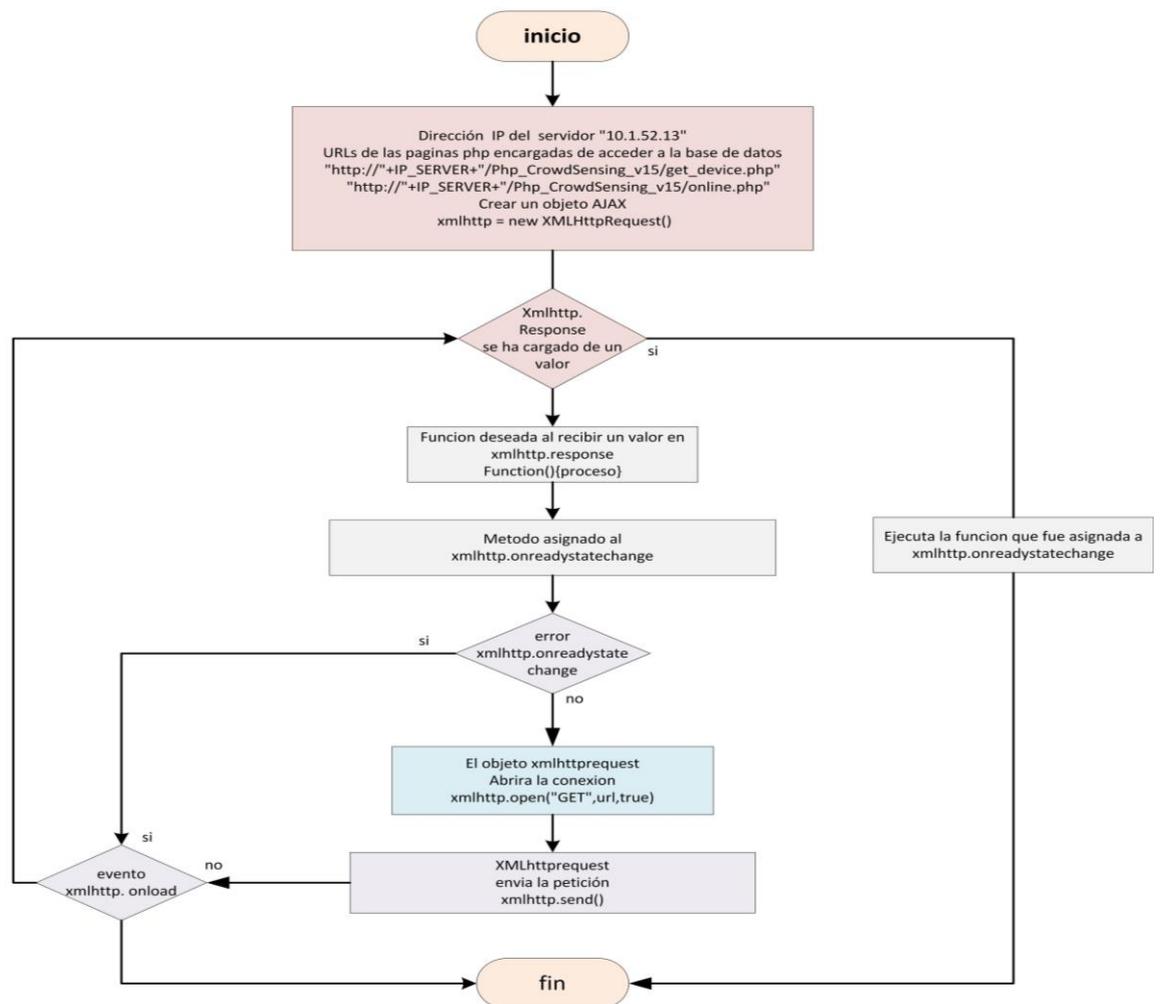


Figura 19 Algoritmo proceso de recepción y envío de información a través AJAX.

Fuente: (Autor)

Como se ha podido apreciar en la figura 19 los principales eventos que validan todas las demás acciones son tres objetos de AJAX utilizando la propiedad XMLHttpRequest, el primero es de tipo "response", "onreadystatechange" y el de tipo "onload".

### 4.2.1.3 Georreferenciación con la API de Google Maps

En la implementación para el servicio de georreferenciación se ha diseñado un algoritmo como se observa en la figura 20. Dicho algoritmo tiene como objetivo determinar los procesos llevados a cabo dentro del navegador web para determinar la ubicación de quien haya ingresado en el sistema de monitoreo.

Así antes de entender este algoritmo se debe tener presente que la mayoría de recursos necesarios para mostrar los mapas resultan de los servidores de Google. Por este motivo la API desarrollada puede ser ejecutada en la mayoría de teléfonos inteligentes ya que no ocupa muchos recursos a comparación de los utilizados en los servidores antes mencionados (Burton, 2012).

De este modo el algoritmo se encuentra conformado por los siguientes pasos para la ejecución en un navegador web:

1. Al igual que los anteriores algoritmos, una vez descargada la información desde el servidor al navegador, es donde iniciará los procedimientos para la georreferenciación.
2. A través de la biblioteca de Google-maps se enviará las credenciales respectivas para solicitar al servidor de google se acceda al servicio de mapas
3. Una vez confirmada las credenciales por Google Map, se inicializará el mapa cargando las opciones de latitud y longitud al igual como el zoom en que se representará la posición deseada por defecto.
4. Valida si el dispositivo puede realizar Geolocalización para ello verifica si dispone de GPS caso contrario lo realizara a través de WiFi.
5. Si la geolocalización es exitosa agrega la posición del usuario al mapa, caso contrario continuará desplegara en pantalla que este proceso no será posible.
6. Posteriormente cargara la posición de todos los usuarios en el mapa con ayuda de marcadores personalizados con la imagen del dispositivo.
7. Una vez cargado los usuarios actuales, no volverá a cargar usuarios hasta después de 1,29 segundos tiempo establecido para nuevamente solicitar al sistema si existe alguna actualización en la base de datos con un nuevo usuario en línea.

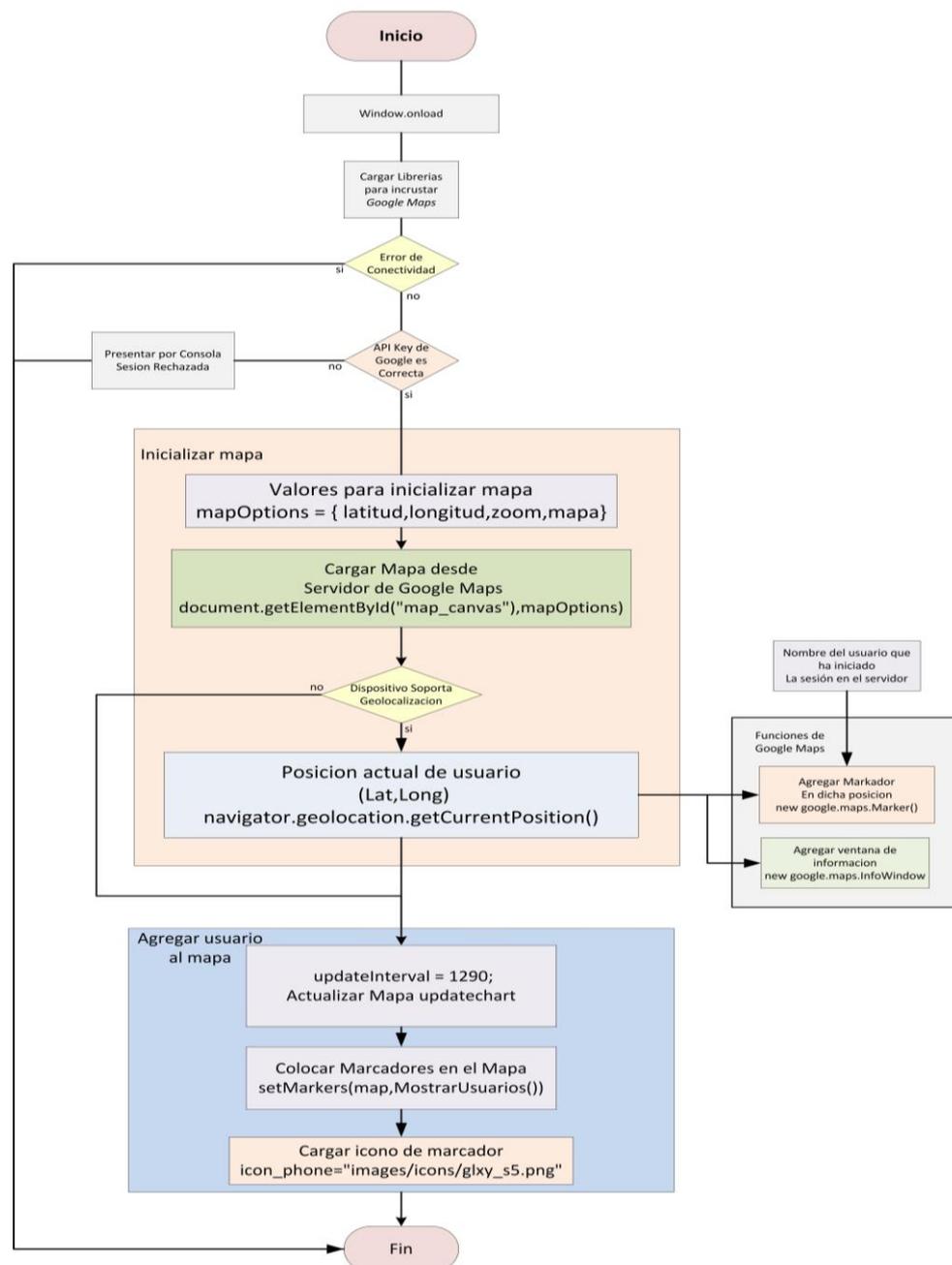


Figura 20. Algoritmo utilizado para cargar y utilizar la API de Google Maps en la plataforma Web.

Fuente: (Autor)

De esta manera Google Maps ha basado su funcionamiento en una arquitectura cliente servidor de tipo web. Como se ha apreciado en la figura 20 dos de sus componentes importantes para su funcionamiento es JavaScript y AJAX ya que permite el intercambio de información de manera asíncrona en el navegador web,

momento de descargar componentes e imágenes a medida que el usuario interactúa con el mapa.

## 4.2.2 Back-End

### 4.2.2.1 Estructuración de la Base de Datos

Dentro la elaboración de una base de datos, las buenas prácticas llevadas en el campo del diseño de software recomienda primero la elaboración de nuestro diagrama de abstracción y representación de la lógica negocio y comportamiento de nuestra aplicación.

El modelo a seguir para el diseño de nuestro diagrama entidad-relación se ha realizado bajo tres consideraciones que se podría encontrar en un sistema de monitoreo en tiempo real, mismas que son mencionadas a continuación.

- Llevar en todo momento un registro del lugar y el tiempo en el que son ingresadas los datos de tipo ambiental así como de salud de un usuario.
- Disponer de un registro de cuentas de usuarios que permita el acceso al sistema de monitoreo MCS.
- Facilitar la creación de reportes ambientales y estado de las variables ambientales y de salud.

A través del diagrama de la figura 21 con dichas consideraciones es posible contemplar el manejo del almacenamiento de información para nuestro sistema monitoreo MCS.

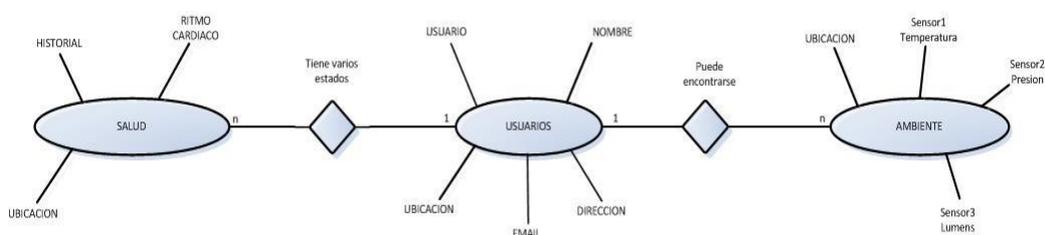


Figura 21 Esquema entidad relación para la base de datos *Crowdsensing*.

Fuente (Autor)

Una vez completado el diagrama entidad relación es posible implementar las tablas en nuestra Base de Datos “Crowdsensingbd”. Con la ayuda de nuestro sistema gestor de base de datos SGBD MySQL finalmente el posible mostrar el esquema final implementado que se apreciar a continuación (Gauchat, 2012) (Aubry, 2014) (Luca, 2011).

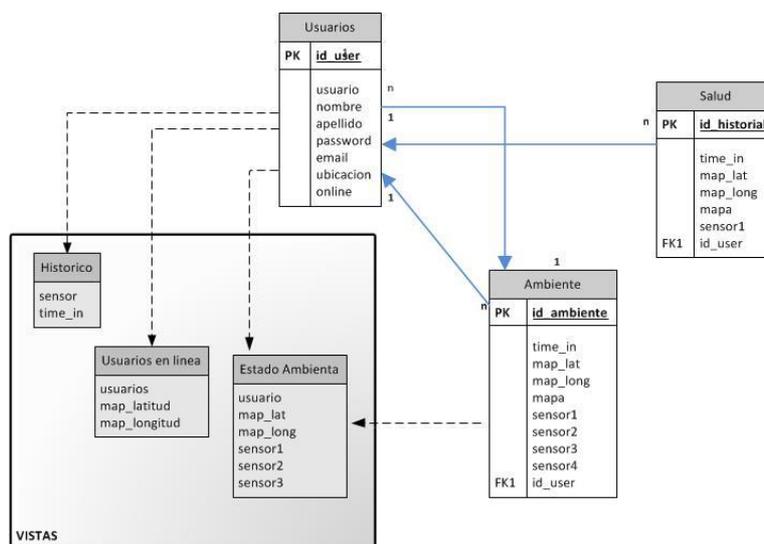


Figura 22. Esquema de tablas creadas para el almacenamiento y gestión de las variables tanto ambientales como de salud para MCS.

Fuente (Autor)

#### 4.2.2.2 PhpMyAdmin

Es una herramienta que permite facilitar el uso de un gestor de base de datos a través de una interfaz web GUI. Como se aprecia en la figura 23 es una herramienta que viene incluida en XAMPP. Por ello es posible la creación e implementación de Base de datos y así como el mantenimiento a través de cuadros y botones en este caso para la conformación y desarrollo del sistema de monitoreo a través de MCS.

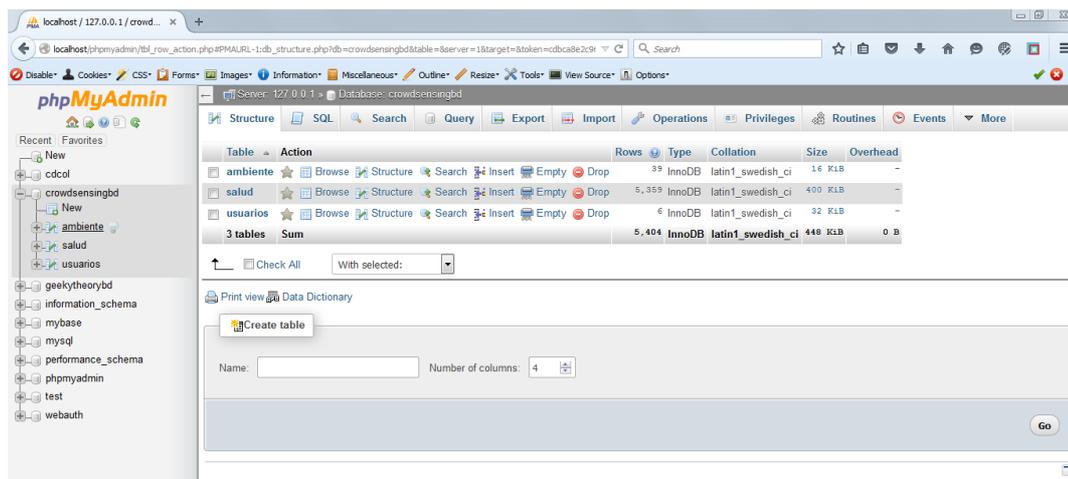


Figura 23. Interfaz GUI “phpmyadmin”, base de datos “CrowdsensingBD”.

Fuente: (Autor)

### 4.2.2.3 Implementación en PHP

Los procesos que se llevan a cabo en el servidor con ayuda de PHP es un componente importante en el intercambio de información como se aprecia en la figura 17. Allí es donde ingresan y salen los paquetes con la información del sensado del clima y del cuidado humano.

Para esta implementación de estos procesos antes mencionadas se ha seguido el algoritmo que se aprecia en la figura 24 y 25 donde muestra los pasos llevados a cabo en el servidor al momento que existe un requerimiento desde un cliente proveedor de datos sensados a un usuario de escritorio, por favor seguir en la siguiente hoja para observar también el otro algoritmo.

Así este primer algoritmo se ha desarrollado de la siguiente manera:

1. A través de mensajes HTTP de tipo POST se colecta los mensajes de provenientes del reloj y teléfono inteligente.
2. Valido si existe alguna dato entrante, si existe algún dato se conecta a la base de datos de “Crowdsensing BD” caso contrario finaliza el algoritmo.
3. Valido si se ha establecido la conexión con la base de datos, de existir algún error igualmente finaliza el algoritmo.
4. Una vez establecida una conexión exitosa con la base de datos, ingreso a esta la información proveniente de los dispositivos móviles.

5. Nuevamente regreso al punto 2. Para verificar nuevos datos tratando de ingresar al sistema.

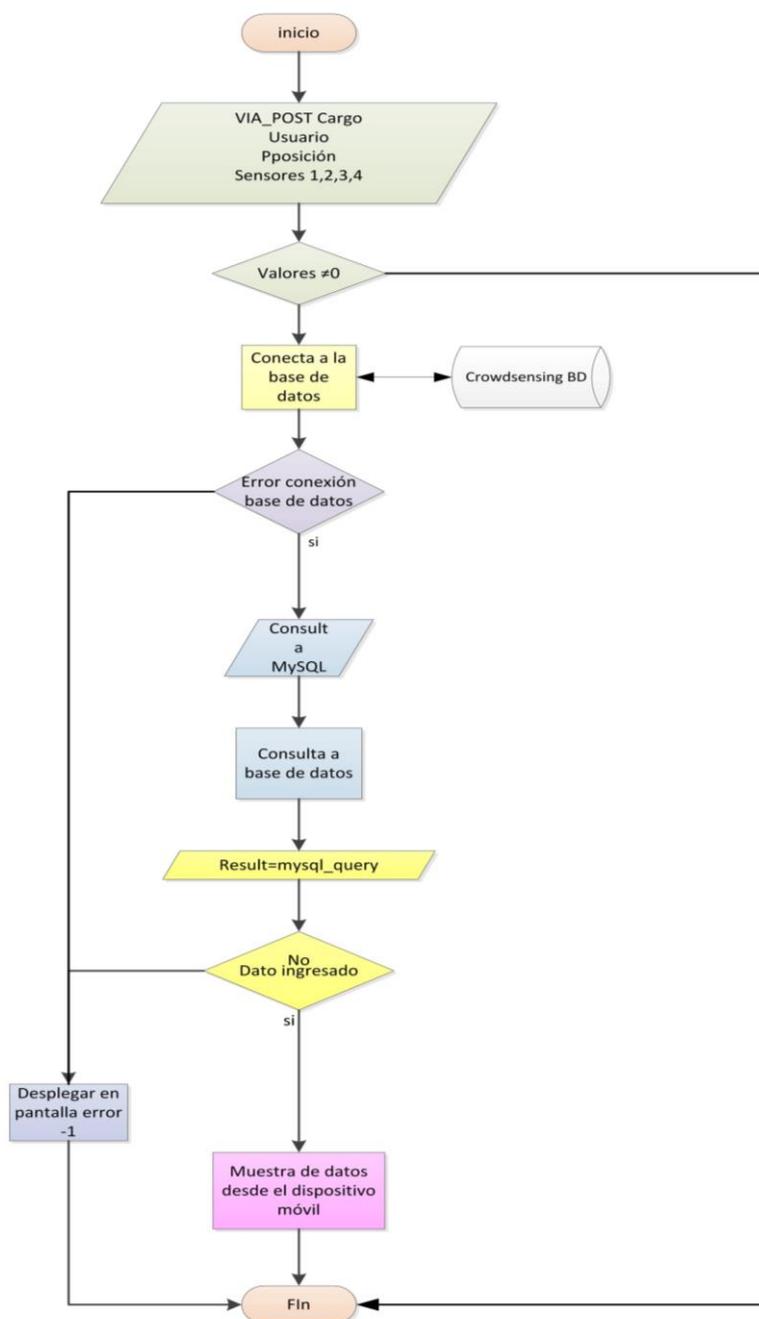


Figura 24 Diagrama de flujo descripción del funcionamiento al recibir una petición a través del método “POST”.

Fuente: (Autor)

Seguidamente, en la figura 25 muestra el algoritmo utilizado para el ingreso de valores a la base de datos desde PHP, donde a diferencia del anterior aquí se detalla el ingreso de información a la base de datos y el uso de JSON para el transporte.

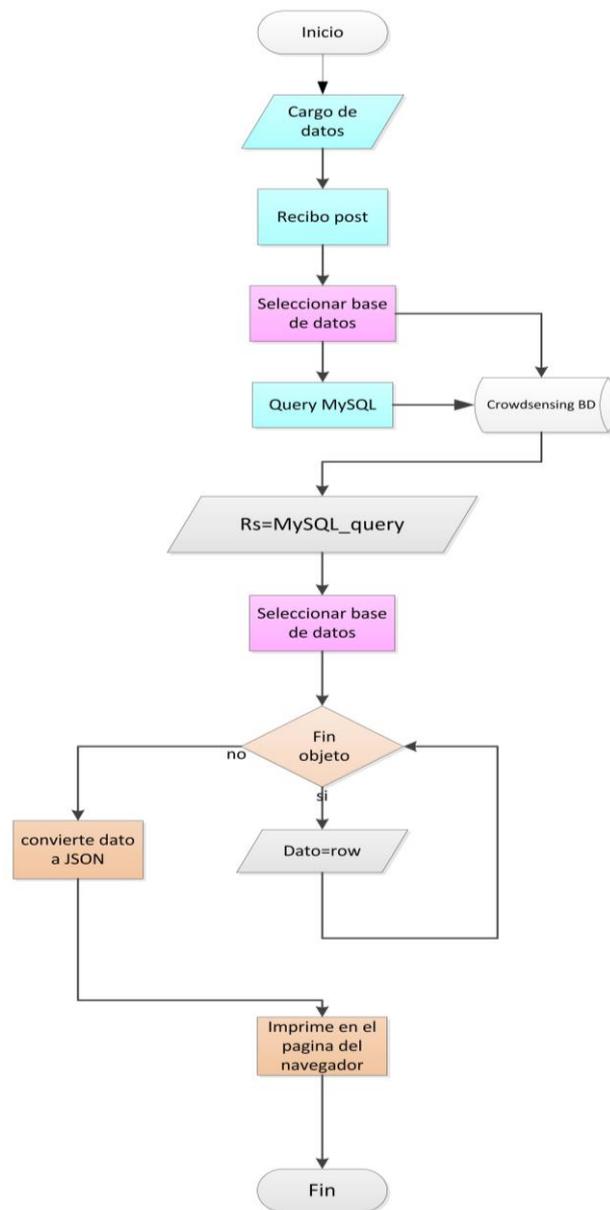


Figura 25. Diagrama de flujo descripción para ingresar los datos a la Base de datos desde PHP.

Fuente: (Autor)

Una parte también del sistema de monitoreo es la conversión de datos que son extraídos de la base de datos y que se explican en el algoritmo que se observa en la figura 25 y que a partir de la selección de la base de datos se detalla a continuación:

1. A través de paquetes HTML de tipo POST recibimos datos al sistema
2. Luego conectar la base de datos “Crowdsensing BD”.
3. Asignar el resultado de la consulta proveniente de la base de datos a una variable en el servidor PHP.
4. Desde el servidor reordenar los datos en un arreglo con ayuda de un algoritmo para un bucle de rápido ordenación conocido como *quick-sort*.
5. Dar formato tipo JSON al arreglo creado
6. Desde el servidor la variable ahora ya en formato JSON es enviada al navegador web a través de un mensaje HTTP.
7. Una vez la información se encuentre en un dispositivo con un navegador web, nuevamente es interpretada y codificada a través de JavaScript para utilizar dicha información en gráficas, como una gráfica.
8. Finaliza el despliegue de información en el navegador web.

### 4.3 Desarrollo de Aplicaciones Móviles

Esta parte del sistema conforma la creación e integración de los datos al sistema de monitoreo. Por este motivo es que se ha convenido explicar el funcionamiento de las aplicaciones en función del dispositivo que aporta con la información.

#### 4.3.1 Aplicación para Android

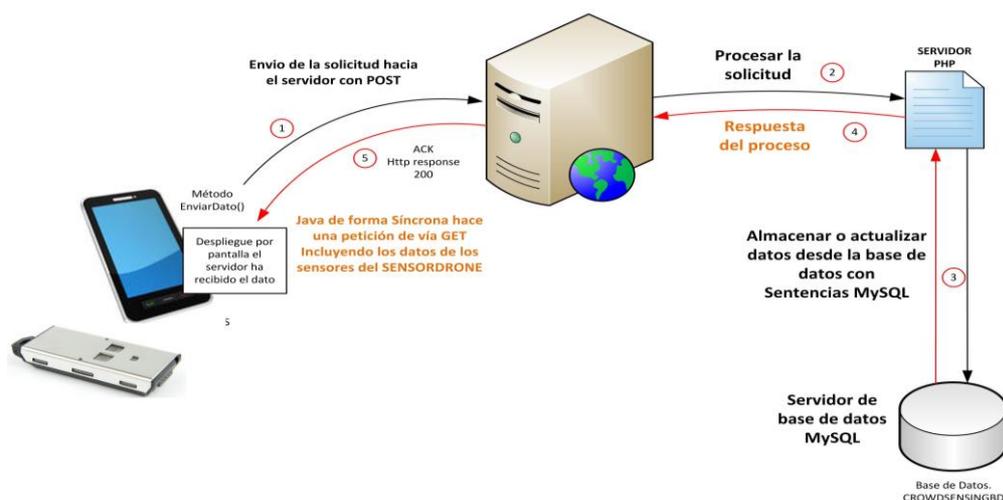


Figura 26. Pasos en la conexión lógica y física del Sensordrone y Smartphone.

Fuente: (Autor)

Para la elaboración de la interfaz gráfica para el teléfono inteligente, como se explicó en el capítulo 3 en el ítem 3.2.1 a través de Android Studio, se ha previsto la elaboración de dos botones que permitan conectar vía *bluetooth* al Sensordrone con el Smartphone. Seguido de esto también se contempló la creación de un botón que permita un refresco en la vista de los datos actuales del Sensordrone capturado en el dispositivo. En la figura 27 es posible apreciar la interfaz GUI final que se ha desarrollado para la aplicación para el MCS.

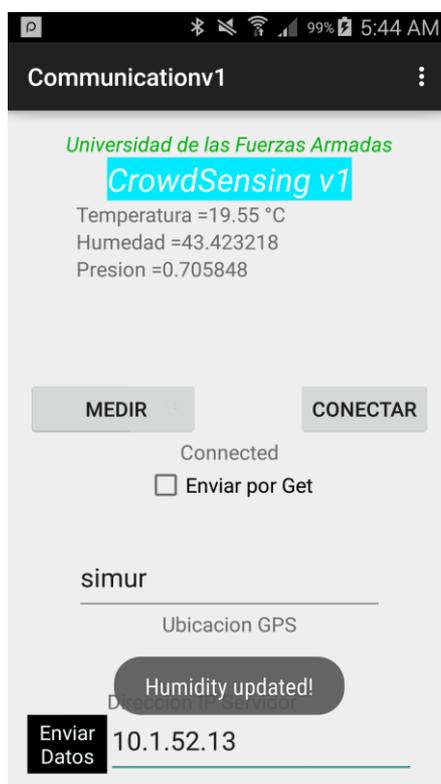


Figura 27. Interfaz GUI desarrollada para la comunicación entre el Sensordrone, teléfono y el servidor.

Fuente: (Autor)

A través de esta aplicación los datos adquiridos desde el Sensordrone una vez que se presione el botón negro “Enviar datos”, obsérvese la figura 27 extremo inferior izquierdo, ese instante empezará a enviar la información de los sensores.

De este modo empezará el envío de información hasta llegar al servidor encargado y posteriormente a la base de datos como se observa en la figura 26. Esto implica que para que la base de datos note alguna petición de ingreso, tendrá que

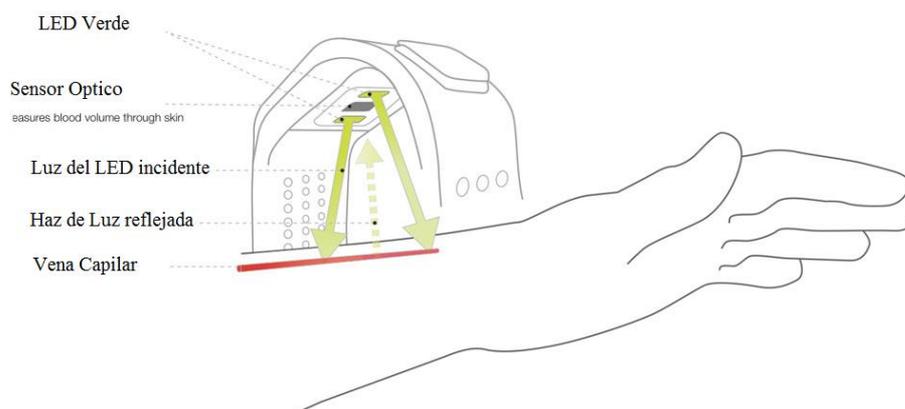
pasar un tiempo viajando por un medio de comunicación hasta llegar al servidor, de manera que el tiempo en que se registre el dato al llegar a almacenarse en la base de datos en realidad existirá un retardo dependiendo de la velocidad del medio.

### 4.3.2 Widget para Tizen

Al momento de diseñar una aplicación para la recolección de información busca que un sensor como es el “oxímetro” a través de una aplicación sea capaz de detectar los latidos del corazón como muestra la figura 28 (literal b). De este modo se pueda medir desde alguna extremidad del cuerpo humano el capilar de una vena y así enviar una base datos en tiempo real (Valencia, 2015).



a) Gráfica del ritmo cardiaco



b) Oxímetro incluido en el Gear-S

Figura 28. Widget desarrollado para la medición de ritmo cardiaco.

Fuente: (Autor)

La aplicación o widget (obsérvese la figura 28 literal a) se ha diseñado de modo que una gráfica y una tabla histórica represente los latidos del corazón por minuto que están siendo capturados en ese momento. Así Tizen proporciona la medición de este signo vital a través del método “motioncallback” que está incluido dentro de las bibliotecas nativas en el dispositivo (Guangnan Ni, 2014).

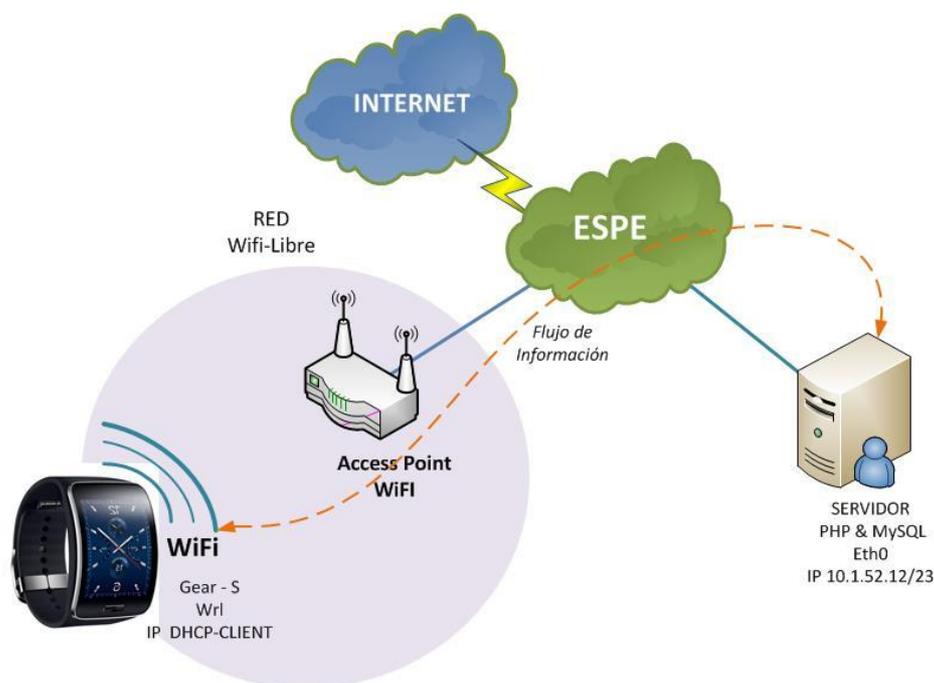


Figura 29. Esquema de red lógica para el envío de información desde el reloj inteligente hacia la base de datos.

Fuente (Autor)

Para la transmisión de la información sensada desde el reloj, de acuerdo a las características del dispositivo vestible revisado en el ítem 3.1.1 del capítulo 3, este dispositivo maneja JavaScript y sus aplicaciones son semejantes a una aplicación web desconectada. Por este motivo es posible utilizar HTML y AJAX para transmitir datos hacia el servidor como se muestra en la figura 30, debido a esto también es posible reutilizar el código implementado en la visualización de los sensores en el navegador web.

Es necesario mencionar además que este widget desarrollado utiliza HTTP, de manera que se ayuda de TCP para asegurar el envío de información.

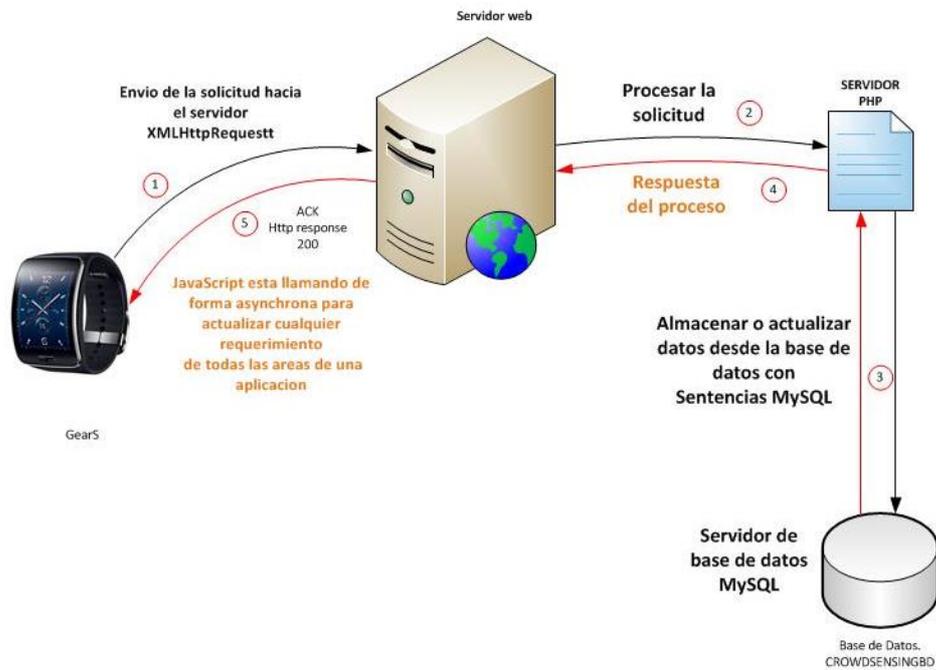


Figura 30. Procedimiento para el intercambio de información con ayuda de AJAX (*Asynchronous JavaScript XML*).

Fuente (Autor)

Como se ha podido apreciar en este capítulo se explicó los algoritmos que fueron necesarios para la implementación para el sensando de multitudes móviles. Finalmente en el siguiente capítulo se comprobará el funcionamiento de algoritmos así como las bibliotecas que se mencionaron en el presente capítulo.

# **CAPÍTULO 5**

## **PRUEBAS DE FUNCIONAMIENTO**

### **5.1 Sistema de Monitoreo**

El sistema de monitoreo se ha implementado acorde los objetivos del proyecto. Así se ha logrado a través de dispositivos móviles medir dentro del campus ESPE la temperatura, presión y humedad, observe la figura 31. De este modo como parte final de nuestro sistema en el presente capítulo se evaluará dos etapas de funcionamiento.

La primera etapa es aquella donde se muestra los valores sensados por el Sensordrone y Gear-S para estimar el error que existirá con respecto a una medición más confiable como el de un sensor especializado. Es por ello que se ha basado la obtención del error en base a los datos de la agencia climatológica del distrito metropolitano de Quito a través de su portal de la secretaria de ambiente y movilidad.

En la segunda etapa se ha dispuesto del análisis de la transmisión de datos desde un cliente hasta el servidor para almacenar la información para ello se ha contemplado por facilidad de entendimiento el uso de herramientas que permitan observar el flujo de información.

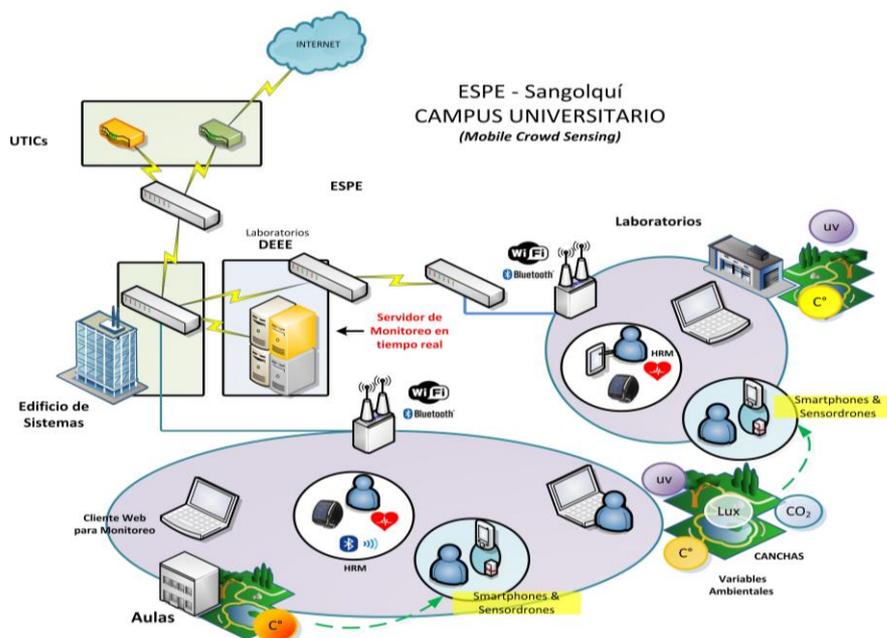


Figura 31. Esquema general del sistema de monitoreo en base a MCS diseñado para la ESPE (campus Sangolquí).

Fuente (Autor)

### 5.1.1 Envío de información con ayuda de Sensordrone y Gear S.

Un aspecto que limita a un dispositivo encargado de sensado es también la distancia máxima que puede soportar el dispositivo antes que este pierda la posibilidad de transmitir sus datos. De esta manera, al referirse a los dispositivos inteligentes Sensordrone y Gear-S de acuerdo a las experimentaciones físicas realizadas en el laboratorio de SAT del departamento de Electrónica.

Para el Gear S no existe ningún inconveniente de envío de información debido a que este se conecta directamente a través de *Wi-Fi*, de manera que no depende de ningún dispositivo intermedio para enviar la información sensada.

### 5.1.2 Recolección de Información Sensada

Para iniciar la recolección es necesario que el nuestro Smartphone previamente encienda su *Bluetooth* como muestra la figura 32 se deberá verificar en el sistema se encuentre como un dispositivo vinculado, en caso de aparecer el sensor con su nombre requiriendo una contraseña para su vinculación se deberá escribir “0000”.



Figura 32. Comunicación establecida con los dispositivos encargados de la recolección de información

Fuente: (Autor)



Figura 33. Interfaz de usuario de la aplicación actualizando los datos extraídos desde el Sensordrone.

Fuente: (Autor)

### 5.1.3 Análisis de la información de variables Climáticas

En términos de medición la evaluación del sistema se basa en la aproximación que nuestro dato llega a acercarse al compararse con un valor patrón, es decir, realizar una comparación con valor fiable ajeno a la captura de información por este sistema. De esta manera se ha tomado la siguiente muestra correspondiente de la base de datos de un lapso de tiempo aleatorio en el que se puede comparar con una medida patrón proporcionada por la plataforma de la municipalidad de Quito llamada Red de mejoramiento atmosférico para la CORPAIRE (Corporación para el mejoramiento del Aire de Quito) el cual tiene por objetivo determinar calidad de los gases existente en Quito por ello se ha comparado con nuestro dispositivo de medición Sensordrone.



Figura 34. Estación de monitoreo Los Chillos, altitud 2453 m sobre el nivel del Mar, coordenadas (78°27'36" W, 0°18'00" S), equipo PM10, MET.

Fuente: (REMMAQ, 2015)

Además gracias a la municipalidad de la ciudad a través de la secretaria de ambiente en el año 2013 también se extendió la plataforma que sería encargada de medir el estado continuo de la concentración de variables ambientales a los valles de Quito (Municipio de Quito, 2015). Es así que en las figuras 35 y 36 se aprecia dicha plataforma disponible en el valle de los Chillos para el sensado de datos y observación de variables físicas principalmente de gases y temperatura. Es por esto que se ha utilizado esta plataforma para conocer los datos en particularmente para el sector del valle de los chillos como se puede observar en la figura 35, de este modo los siguientes datos presentado en las tablas 5, 6,7 pertenecerá a este sector.

SECRETARÍA DE  
**AMBIENTE**

Reporte Datos:  
 Datos diez minutos   Datos diarios   Datos horarios   Datos semihorarios   Datos octohorarios

Reporte Gráficos:  
 Gráfico diez minutos   Gráfico diarios   Gráfico horarios   Gráfico semihorarios   Gráfico octohorarios

Ingreso de parámetros - Gráfico Diez Minutos

Fecha Inicio:    Fecha Fin:

Estaciones:  
 LOS CHILLOS (selected)  
 COTOCOLLAO  
 CARAPUNGO  
 BELISARIO  
 JIPIJAPA  
 EL\_CAMAL  
 CENTRO  
 GUAMANI  
 TUMBACO  
 LOS CHILLOS

Magnitudes:  
 Seleccione...

Buscar

Figura 35. Sistema de monitoreo de la calidad de aire, secretaria del ambiente municipio de Quito.

Fuente: (Municipio de Quito, 2015)

Dentro de esta plataforma que servirá para referencia las variables encontradas en el sistema como la temperatura, humedad y presión como se puede corroborar en la figura 36.

SECRETARÍA DE  
**AMBIENTE**

Reporte Datos:  
 Datos diez minutos   Datos diarios   Datos horarios

Reporte Gráficos:  
 Gráfico diez minutos   Gráfico diarios   Gráfico horarios

Ingreso de parámetros - Gráfico Diez Minutos

Fecha Inicio:    Fecha Fin:

Estaciones:  
 LOS CHILLOS

Magnitudes:  
 6 - MONOXIDO DE CARBONO - CO  
 7 - MONOXIDO DE NITROGENO - NO  
 8 - DIOXIDO DE NITROGENO - NO2  
 10 - PARTICULAS MENORES A 2.5 MICROMETROS PM2.5  
 12 - OXIDOS DE NITROGENO TOTALES - NOx  
 14 - OZONO - O3  
 81 - VELOCIDAD DEL VIENTO - VEL  
 82 - DIRECCION DEL VIENTO - DIR  
 83 - TEMPERATURA MEDIA - TMP  
 86 - HUMEDAD RELATIVA - HUM  
 87 - PRESION BAROMETRICA - PRE  
 88 - RADIACION SOLAR - RS  
 89 - PRECIPITACION - LLU  
 1005 - PARTICULAS MENORES A 10 MICROMETROS PM10  
 83 - TEMPERATURA MEDIA - TMP, 86 - HUMEDAD RELATIV.

Buscar

Figura 36. Red de monitoreo de la Secretaria de Ambiente, sección reporte, variables físicas ambientales, recuadro opciones de sensores medidos.

Fuente: (Municipio de Quito, 2015)

Una vez realizado la consulta a través de esta plataforma se puede extraer los siguientes datos para las tres variables estudiadas, en la fecha para la muestra analizada se obtuvo una media de temperatura de 27,7 [C°], para la humedad se obtuvo de 46,8% y a su vez para presión se obtuvo una presión de 0,7081 [atm] (Municipio de Quito, 2015), además como se observa en la tabla 5 se muestra como en promedio, nuestras mediciones presentan un error absoluto 1,003% como se observa en la figura 37 debido a sus datos iniciales ya no se ajusta a una correcta medición.

Tabla 5.

Cálculo del error y datos de la temperatura extraídos del sistema de monitoreo estimado en un intervalo de tiempo de 3 horas, realizadas desde el Sensor dron.

Temperatura [C°]					
Muestra	Tiempo [h]	Temperatura	Referencia	error(A)	error (R)
1	12:48	27,46	28	0,7	2,6
2	13:11	27	28,08	1,9	7
3	13:20	27,18	28,5	1,9	7,1
4	13:30	27,19	27	0,19	0,7
5	14:00	27,61	27,6	0,61	2,25
6	14:10	27,01	27	1,01	3,74
7	14:20	26,4	26,9	0,93	3,51
8	14:35	26,99	27	0,01	0,03
9	15:00	27,42	26,9	1,48	5,48
10	15:11	26,62	27,01	0,38	1,4
11	15:25	26,77	27,01	1,93	7,14

Fuente: (Autor)

De este modo como se puede apreciar en la figura 37 la gráfica de temperatura en función del número de muestra el cual se puede visualizar la aproximación que tienen nuestras mediciones (nótese la función en color rojo) denotan que las se asemejan a los datos que fueron sensados a través de los instrumentos del distrito metropolitano de Quito (nótese la función en color amarillo).

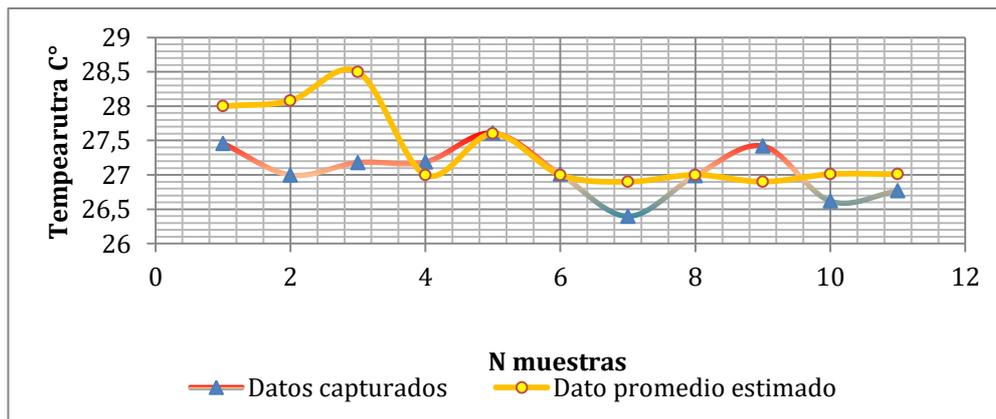


Figura 37. Aproximación de los valores de temperatura medidos con referencia a un valor externo al sistema.

Fuente: (Autor)

Seguido de la temperatura, la humedad como se observa la figura 38 una curva mucho más cercana a los datos obtenidos por los obtenidos por la plataforma del municipio, así en la tabla el error absoluto promedio presentado de la tabla 6 es del 0,7 y un error relativo de 49% debido a un error en la octava medición que se aleja en gran medida de la obtenida entre las obtenidas por el Sensordrone y el de la plataforma del municipio.

Tabla 6.

Cálculo del error y datos de la humedad extraídos del sistema en un intervalo de tiempo de tres horas realizadas desde el Sensordrone.

Humedad [%]					
Muestra	Tiempo[h]	Humedad	Referencia	Error(A)	Error(R)
1	12:48	46,355999	47	0,644011	1,37
2	13:11	48,540405	47	1,54	3,27
3	13:20	45,009094	47	1.990.906	4,23
4	13:30	46,098712	47	0,901288	1,91
5	14:00	47,202026	47	0,202026	0,42
6	14:10	45,6026	47	13.974	2,97
7	14:20	46,330627	47	0,669373	1.424.197
8	14:35	45,025439	47	1.974.561	4.201.193
9	15:00	46,825439	47	0,174561	0,37
10	15:11	46,636292	47	0,363708	0,773846
11	15:25	46,649414	47	0,350586	0,745927

Fuente: (Autor)

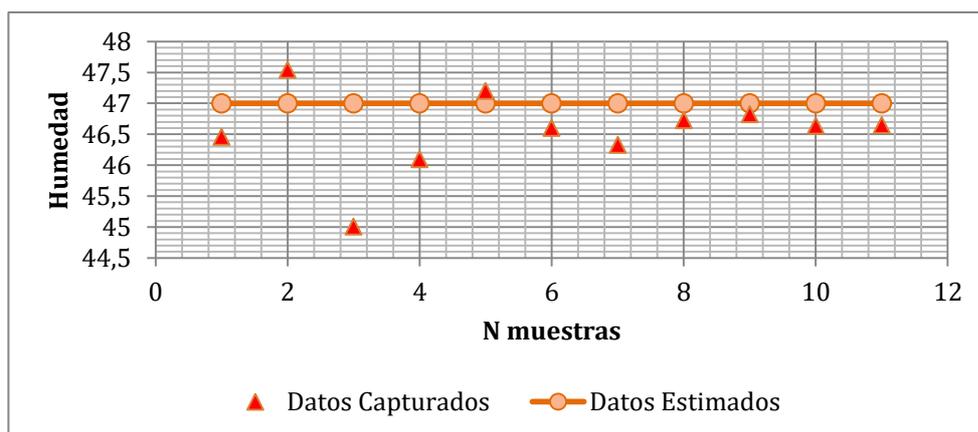


Figura 38. Aproximación de los valores de humedad medidos con referencia a un valor externo al sistema.

Fuente: (Autor)

Finalmente en la medición de la presión como se aprecia en la tabla 4 el número de muestras capturas desde el Sensordrone con respecto a la plataforma del municipio ha presentado un error absoluto del 0,33 y un error relativo 47,69%.

Tabla 7.

Cálculo del error y datos de la presión extraídos del sistema en un intervalo de tiempo de tres horas de 10 min/muestra, realizadas desde el Sensordrone.

Presión [atm]					
Muestra	Tiempo[h]	Presión	Referencia	Error(A)	Error[R]
1	12:48	0,705413	0,70821	0,002901	0,409564
2	13:11	0,705492	0,708139	0,003	0,42354
3	13:20	0,704113	0,708319	0,002901	0,409564
4	13:30	0,705532	0,708397	0,002782	0,392763
5	14:00	0,705137	0,708294	0,003177	0,448598
6	14:10	0,70565	0,708314	0,002664	0,376104
7	14:20	0,705019	0,708314	0,003295	0,465307
8	14:35	0,705782	0,708299	0,006532	0,933189
9	15:00	0,704979	0,708314	0,003335	0,470836
10	15:11	0,704703	0,708119	0,003611	0,509802
11	15:25	0,705432	0,708314	0,002882	0,406881

Fuente: (Autor)

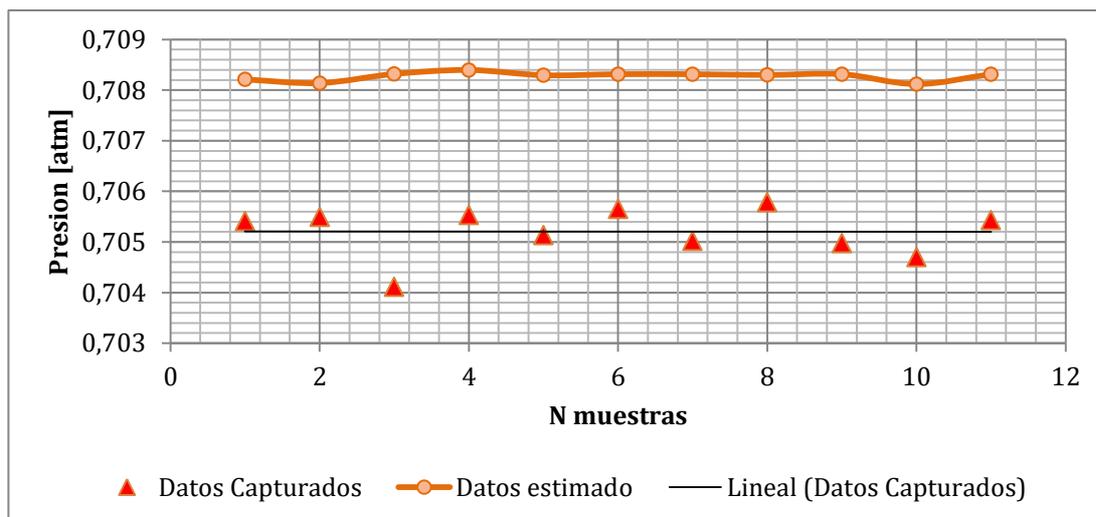


Figura 39. Aproximación de los valores de presión atmosférica comparados con referencia a un valor externo al sistema de monitoreo.

Fuente: (Autor)

En general al analizar las figuras producidas por las muestras es posible observar que entre las cantidades 3,4 y 5 muestra la existencia de un cambio en los datos reflejadas en las tablas, así en un lapso aproximado de una 2 horas se puede concluir que existió una depresión en la atmosfera.

## 5.2 Transporte de la Información

Un aspecto para analizar una aplicación en tiempo real, es el retardo y el *throughput* o número de envío de datos efectivos realizados (Parziale, 2006). Por ello una manera de observar detalladamente la transferencia de datos que son originados desde los teléfonos inteligentes que son recibidos al servidor de monitoreo y desde el mismo hacia un equipo de escritorio será a través de un analizador de paquetes o *Sniffer*. Es así que para esta prueba de funcionamiento se ha pensado realizarlo con WireShark.

Lo primero que se debe tener en cuenta al utilizar un analizador es entender la topología donde está pensado efectuar el análisis por ello para los siguientes análisis que se muestran a continuación se basarán en el diseño de la figura 30 y 33 seguido de esto es saber qué tipo de paquetes deseamos observar para ello nos ayudaremos de la información mostrada en la tabla 8.

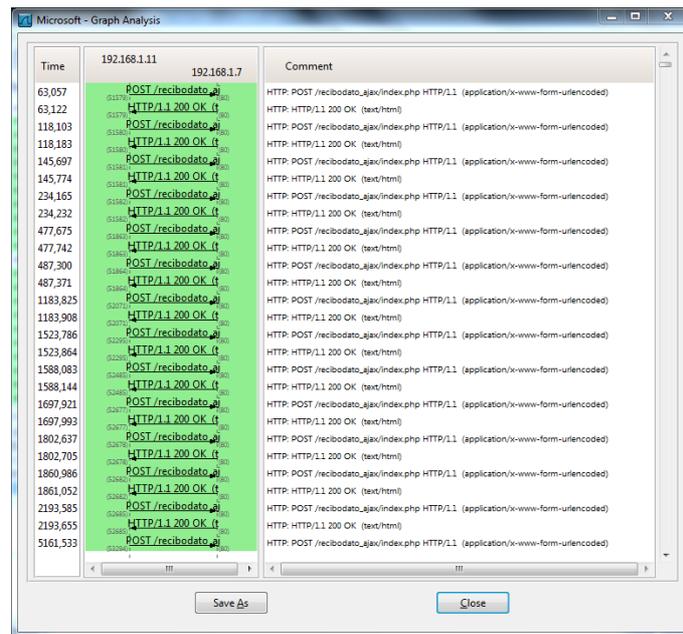


Figura 40. Comunicación entre un *Smartphone* y el Servidor de monitoreo en el envío de variables climáticas, visto a través de un *sniffer*.

Fuente (Autor)

De manera general la figura 41 nos ayudará a observar ambos dispositivos interactuando con el servidor.

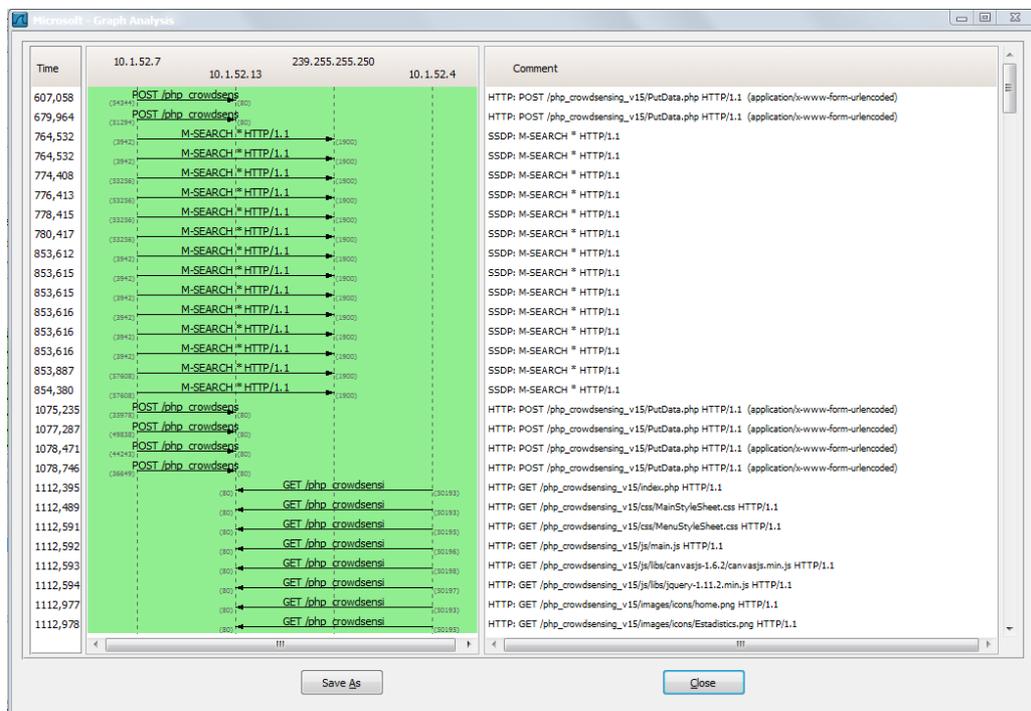


Figura 41. Gráfica del entrada y salida de información referida en el servidor encargado del sistema de monitoreo a través de WireShark.

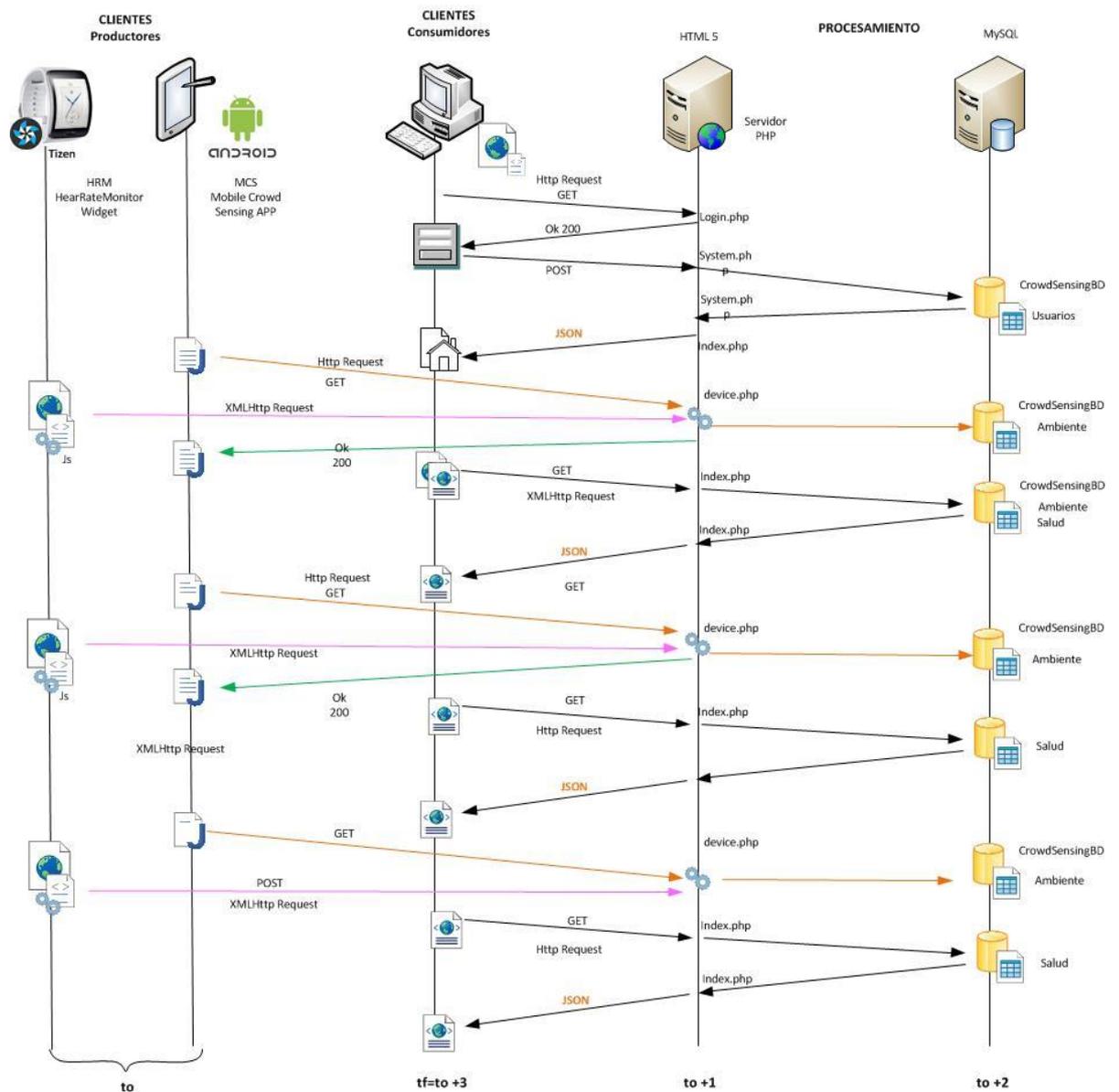


Figura 42 Diagrama de secuencia y transmisión de la información.

Fuente: (Autor)

En la captura de paquetes en el servidor para el sistema de monitoreo se aplica el siguiente filtro para observar las transmisiones de datos que existe en el servidor que aloja la computadora y el teléfono inteligente.

Tabla 8.

Descripción de comandos para la creación del filtro en Wireshark.

FILTRO	DESCRIPCIÓN
<b>ip.address</b>	Actúa a nivel de capa tres del modelo OSI, permitiendo paquetes de tipo IP en particular el campo de dirección de origen o destino.
<b>http.request.method</b>	Actúa a nivel de la capa presentación del modelo OSI, únicamente permite el tráfico de solicitudes a través de métodos GET y POST.

Fuente: (Autor)

Filtro (comandos utilizados):

(ip.addr == 10.1.52.7 and http.request.method) or (ip.addr == 10.1.52.4 and http.request.method)

Igualmente al utilizar la herramienta incluida en wireshark medidor de paquetes se puede observar la siguiente gráfica que indica la cantidad de paquetes que ingresan y salen al servidor.

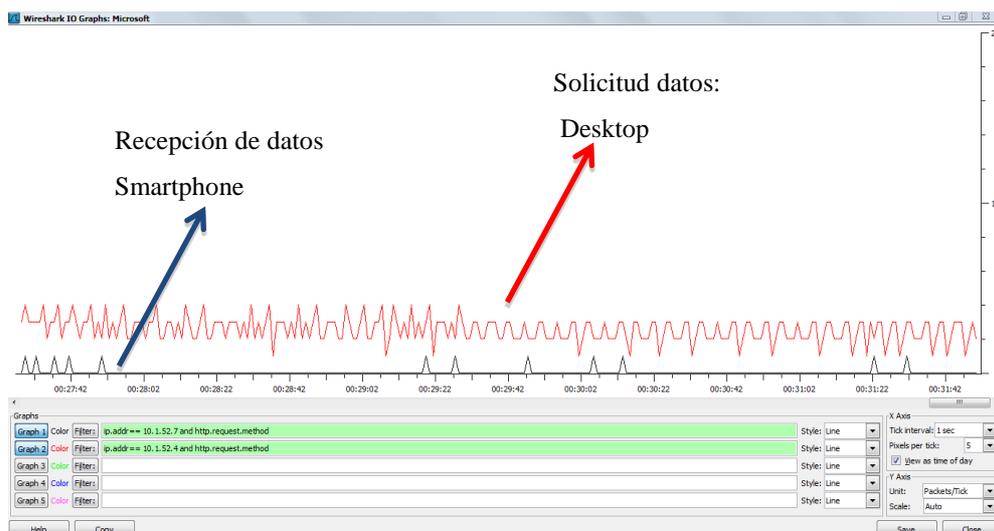
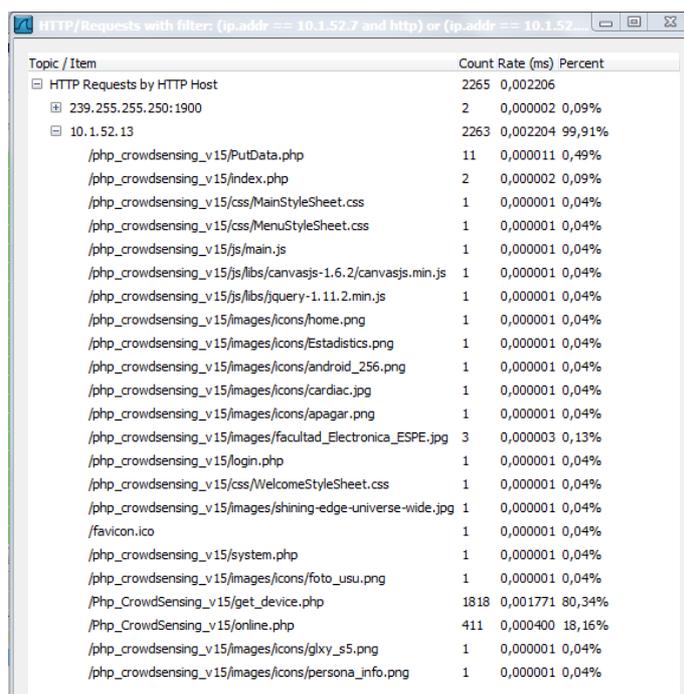


Figura 43. Cantidad de paquetes en función del tiempo a través de WireShark (Línea Azul) teléfono inteligente, (Línea Roja) paquetes de salida al servidor.

Fuente: (Autor)

## 5.2.1 Análisis de la cantidad de información

De acuerdo al sensado colectivo por móviles visto en el ítem 2.1 la transmisión de datos ha sido ampliamente estudiada debido a la necesidad del ancho de banda que puede llegar a utilizar un sistema como este. Así los algoritmos desarrollados en la sección 4.2 proveen además del sistema de monitoreo una herramienta que mide de manera indirecta el porcentaje de consumo mínimo que una aplicación puede llegar a tener para el MCS. Es por ello que a través de este proyecto implementado como se observó en la figura 43 el tráfico que un usuario produce para observar esta información es mucho mayor a la que puede llegar a ser generado para proveer de dicha información.



Topic / Item	Count	Rate (ms)	Percent
HTTP Requests by HTTP Host	2265	0,002206	
239.255.255.250:1900	2	0,000002	0,09%
10.1.52.13	2263	0,002204	99,91%
/php_crowdsensing_v15/putData.php	11	0,000011	0,49%
/php_crowdsensing_v15/index.php	2	0,000002	0,09%
/php_crowdsensing_v15/css/MainStyleSheet.css	1	0,000001	0,04%
/php_crowdsensing_v15/css/MenuStyleSheet.css	1	0,000001	0,04%
/php_crowdsensing_v15/js/main.js	1	0,000001	0,04%
/php_crowdsensing_v15/js/libs/canvasjs-1.6.2/canvasjs.min.js	1	0,000001	0,04%
/php_crowdsensing_v15/js/libs/jquery-1.11.2.min.js	1	0,000001	0,04%
/php_crowdsensing_v15/images/icons/home.png	1	0,000001	0,04%
/php_crowdsensing_v15/images/icons/Estadistics.png	1	0,000001	0,04%
/php_crowdsensing_v15/images/icons/android_256.png	1	0,000001	0,04%
/php_crowdsensing_v15/images/icons/cardiac.jpg	1	0,000001	0,04%
/php_crowdsensing_v15/images/icons/apagar.png	1	0,000001	0,04%
/php_crowdsensing_v15/images/facultad_Electronica_ESPE.jpg	3	0,000003	0,13%
/php_crowdsensing_v15/login.php	1	0,000001	0,04%
/php_crowdsensing_v15/css/WelcomestyleSheet.css	1	0,000001	0,04%
/php_crowdsensing_v15/images/shining-edge-universe-wide.jpg	1	0,000001	0,04%
/favicon.ico	1	0,000001	0,04%
/php_crowdsensing_v15/system.php	1	0,000001	0,04%
/php_crowdsensing_v15/images/icons/foto_usu.png	1	0,000001	0,04%
/Php_CrowdSensing_v15/get_device.php	1818	0,001771	80,34%
/Php_CrowdSensing_v15/online.php	411	0,000400	18,16%
/php_crowdsensing_v15/images/icons/glxy_s5.png	1	0,000001	0,04%
/php_crowdsensing_v15/images/icons/persona_info.png	1	0,000001	0,04%

Figura 44. Porcentaje de paquetes transmitidos hacia el sistema de sensado, observados desde WireShark.

Fuente: (Autor)

Además como se menciona en el internet de la cosas en el ítem 2.2 MCS transporta sin modificar ningún campo en el armado del paquete de forma que sea mejor utilizado por un servidor encargado del almacenamiento, sin embargo también esto acarrea que el ancho de banda aumente ya que no existe todavía algún estándar

o técnica desarrollada conocida para minorar el uso de este recurso. Todo esto puede fácilmente llegar a complicar el desarrollo de software encargado de levantar la comunicación del sistema de monitoreo. A través de este paradigma se pudo observar que las aplicaciones están siendo realizadas están acordes a la transacción de cualquier página web.

Así nuevamente en la figura 44 se puede corroborar nuevamente de la mismas captura realizadas por WireShark como las páginas web que contienen los algoritmos y contenido multimedia representa un consumo de 0,04% de ancho de banda que a su vez juntas representan alrededor de un 30% del total de ancho de banda posible de la conexión que el servidor tiene para conectarse a la red.

Sin embargo el consumo de ancho de banda que representa únicamente los datos al momento que se transportan y que contienen la información de los sensores (temperatura, humedad y presión) de acuerdo a wireshark como se observa en la figura 44 en la parte central archivo “get\_device.php” , da un resultado de un 80,34% tres veces más que lo genera el consumo de ancho de banda para el transporte de contenido multimedia necesario para la carga de la página web, esto se debe a que este tipo de sistemas interactúan en tiempo real, y necesariamente deben intercambiar información en cortos tiempos.

Otro valor que resulta del análisis de tráfico que existe en el sistema, es aquel que se observa en la figura 44 producido por el archivo “online.php” que es encargado de mostrar los usuarios conectados al sistema de monitoreo que se encuentran dotando de información al sistema en ese momento. Estos datos generados representan alrededor de un 18% del total del ancho de banda consumido. Esto se debe a que los algoritmos desarrollados como se describen el ítem 4.3.1 y se observan en la figura 24 y 25, el envío de información al navegador web se lo realiza cada 1290 milisegundos es decir cada determinado tiempo ya que de no ser así esto sobrecargaría el consumo de ancho de banda haciendo que nuestra aplicación se torne lenta y comprometa el funcionamiento del sistema.

# **CAPÍTULO 6**

## **CONCLUSIONES Y RECOMENDACIONES**

### **6.1 Conclusiones**

Un sistema de sensado colectivo a través de dispositivos móviles permitió demostrar la tendencia que hoy en día existe con el uso de teléfonos inteligentes como dispositivo fuente de información. Así MCS preestablece posibles estrategias para el cambio del comportamiento ambiental en base de los datos en una región en especial y habitat que rodea al humano.

MCS visto desde otra perspectiva además promueve el estudio climatológico en la cual además de los beneficios observados basados en la recolección de información. Promueve a usuarios de dispositivos móviles el uso de teléfonos inteligentes y con ello la posibilidad que sistemas de este tipo luego evolucione a un sistema de alerta temprana debido a la cantidad de usuarios que lo usan.

Las técnicas de comunicación como AJAX y JSON son actualmente las tecnologías más utilizadas dentro de un navegador web, ya que es el uso de estos permite el proceso y envío de información en segundo plano. De acuerdo al nuevo estándar de HTML5 WebSockets es la tecnología que está pensada reemplazar AJAX ya que esa diseñada para soportar aplicaciones en tiempo real.

El uso de batería es el principal limitante de este paradigma ya que utiliza varios recursos de los dispositivos inteligentes, el usuario al percibir que rápidamente se descarga la batería lo que percibe como un proceso innecesario al momento con respecto a las demás aplicaciones alojadas.

## **6.2 Recomendaciones**

A medida que un mayor número de usuarios accedan al mismo tiempo al sistema, se deberá aprovechar de mejor forma el ancho de banda de la transmisión al servidor. Para ello se recomienda mejorar el algoritmo basado en MCS, reduciendo la frecuencia o el número de consultas que son realizadas al servidor a través de XMLHttpRequest de AJAX. Así estableciendo solicitudes y actualización de los datos mayor al lapso de tiempo actual.

En la actualidad hay navegadores que aún no logran soportar todas las características de HTML5 y la mayoría de sus funciones se encuentran actualmente en estado de desarrollo. Por ello para el uso del sistema de monitoreo se recomienda usar Chrome, Firefox o Safari.

Para el desarrollo de aplicaciones independientes o widgets en Tizen es posible separar el diseño o presentación del desarrollo de su funcionamiento, para lo cual es recomendable que el diseño se lo realice con Chrome ya que este permite simular la dimensión de la resolución de pantallas de dispositivos como relojes tablets y teléfonos inteligentes.

Las cookies son de gran ayuda al realizar un portal web a través de PHP debido a que ayuda a la gestión y control de sesión de usuarios para el ingreso al sistema de monitoreo, por ello es recomendable si se desea que este sistema sea utilizado en Internet se ha implementado con las respectivas normas que regulan su uso.

El tiempo de carga de una página web en ocasiones llega a comprometer el grado de satisfacción que tiene el usuario con respecto a los servicios que brinda nuestro sistema de monitoreo. Por ello es recomendable mejorar dicho tiempo a través de ofuscadores y compresores que permiten que el código de JavaScript se reduzca en espacio de almacenamiento y con ello minore el tiempo de descarga desde el servidor.

### 6.3 Trabajos Futuros

Una vez implementado el sistema de monitoreo para la adquisición de datos colectivo a través dispositivos móviles inteligentes, para continuar con el desarrollo de este sistema algunas de las propuestas para trabajos futuros que podrían resultar de notorio aporte son nombrados a continuación.

En inicio se podría evaluar el mismo sistema pero llevado a una mayor escala semejante a toda la ciudad de Quito, es decir que esta vez el mismo servicio no solamente pueda ser usado por los usuarios dentro del campus sino más bien que este sistema sea utilizado por una población entera. Con ello será posible definir un mejor modelo del comportamiento climático de la ciudad de Quito.

Seguido de esto también se podría evaluar el sistema implementado utilizando varios Sensordrones y Gear-S, con esto sería posible dotar de más datos para los análisis de las variables climáticas estudiadas en la implementación. Además aquello permitirá observar y evaluar la confiabilidad y estabilidad de nuestro sistema de monitoreo al contemplar ahora una mayor demanda de recursos del lado del servidor.

Además se puede llegar a proponer como una solución alternativa para la adquisición de datos en caso de no constar con varios Sensordrones. Podría ser desarrollar un dispositivo portable o usable semejante al utilizado en el sistema como es el Sensordrone, capaz de medir algunas variables físicas a la vez y además pueda ser controlado vía *Bluetooth* desde un teléfono inteligente.

Como sucede actualmente debido a la disponibilidad del Sensordrone en el mercado para MCS se utilizó únicamente un único dispositivo, mismo que limitó un mejor análisis de los datos en el sistema implementado. Por ello la creación de un dispositivo de las características del Sensordrone promoverá que dichos sensores

sean de fácil adquisición en el mercado con la ventaja de que estos puedan llegar a ser deseables por el usuario y lleguen a convertirse fácilmente en una tendencia o moda, así también contribuyendo al desarrollo del MCS.



(a) Banda adicional conectada al Reloj a través de Bluetooth.



(b) Sensores adicionales de temperatura y frecuencia cardíaca.



(c) Aspecto y funcionamiento de un Simband.

Figura 45. Reloj inteligente de Samsung “Simband” diseñado específicamente para el cuidado de la salud humana.

Fuente: (Kuo, 2015)

También se podría considerar para este sistema el uso del dispositivo “Simbad” de Samsung, ya que este dispositivo portable o usable además de incluir todas las características de un Gear-S. Simbad incorpora también sensores que permiten medir la temperatura corporal, el nivel de glucosa en la sangre y a su vez puede cargarse de manera inalámbrica (Kuo, 2015). Con ello sería posible ampliar el nivel de monitoreo hasta llegar a disponer de un sistema preventivo de salud en línea.

# BIBLIOGRAFÍA

- Alan Mainwaring, J. P. (2012). Wireless Sensor Networks for Habitat Monitoring. *ETCS*, 97-98.
- Albert Alonso, A. T. (29 de Mayo de 2015). mHealth Assessment Workshop. *mHealth Competence Center of Mobile World Capital (MWC)*, (págs. 1-6). Barcelona.
- Ameneiro, S. B. (2002). *Fronteras de la Computacion*. Madrid, España: Dias de Santos.
- Angel., C. (13 de Agosto de 2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones de Diaz de Santos.
- Aransay, A. L. (2009). *Computacion Ubicua*. Pontevedra: Universidad de Vigo.
- ARCOTEL-Supertel. (2012). Evolución de la Telefonía Móvil en Ecuador. *Revista Institucional*, 16, 34.
- ARCOTEL-Supertel. (Agosto de 2014). Dispositivos Móviles Registrados con LTE. *Tecnología 4G en Ecuador. I*. Quito: Revista Institucional.
- Aubry, C. (2014). *HTML5 y CSS3 Revoluciones el diseño de sus sitios web* (Vol. 2 edicion). (V. S. Andrea Garcia Vega, Ed.) Barcelona, Catalun, Espana: ENI.
- Bin Guo, F. C. (Agosto de 2014). Mobile Crowdsensing. *IEEE Communications*, 52(8), 166.
- Burton, M. (2012). *Android Application Development for Dummies* (Segunda ed.). United States: Wiley.
- C Tanas, C. P. (29 de Septiembre de 2014). *Senda*. Obtenido de Mobile Crowd Sensing: <http://senda.uab.es/node/15>
- Cabello, A. L. (2014). *Implantacion de aplicaciones web en entornos internet, intranet y*

- extranet* (Primera ed.). Malaga, España: IC.
- Calmon, A. (Septiembre de 2014). Guia Esencial para Samsung Galaxy. (J. M. Andre Pires, Ed.) págs. 27-30. Recuperado el 19 de Julio de 2015.
- Chen Jing, P. h.-l. (2015). Analysis of Android Kernel. *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, 5, 129-133.
- Cisco System. (2006). *Networking Basics*. (T. A. Wendell Odom, Ed.) San Francisco, EEUU: by Software, Bonson.
- Cortes R, B. X. (12 de Junio de 2015). Seize New IoT Opportunities with the Cisco IoT System. *Procedia Computer Science* , 52. Obtenido de Internet of Things: <http://www.cisco.com/web/solutions/trends/iot/portfolio.html>
- David Cuartielles, A. G. (2015). *Professional Android Wearables*. Indianapolis: Wiley.
- Delgado Guerrero, P. A. (2007). *Estudio y análisis de los factores técnicos y requerimientos para promover la implementación de la red de internet-2*. Tesis Pregrado, EPN, Electronica y Telecomunicaciones, Quito.
- Delisle, M. (2012). Overview. En *Mastering phpMyAdmin 3.4 for Effective MySQL Management* (pág. 394). PACKT.
- Evans, D. (2011). *Internet de las cosas la proxima evolucion de Internet lo cambia todo*. San Francisco: CISCO IBSG.
- Fabricio Carvalho, T. M. (2014). *Quality of Service in Telecommunication Services* (Vol. II). Berlin, Franklinstr, Alemania.
- Fang-Jing, Y. F. (2011). *From wireless sensor networks towards cyber physical systems* (Vol. 7). WU.
- G Anastasi, M. C. (Noviembre de 2012). Performance Measurements of Mote Sensor Networks. 19.
- Gauchat, J. D. (2012). *EL gran libro de HTML5, CSS3 y JavaScript*. Barcelona, Catalun, Spain: marcombo.s.a.
- George Coulouris, J. D. (2013). *Sistemas distribuidos: conceptos y diseño* (Tercera ed.). (P. E. 2001, Ed., & J. B. Junquera, Trad.)
- Grossman, L. (Noviembre de 2007). the best invention of 2007. *TIMES Magazine New York*.
- Guangnan Ni, C. S. (2014). Tizen OS Development. *Tizen Developers Conference*. San Francisco: Samsung.
- Harish Doddala, H. G. (Abril de 2014). Internet of things: Role of Oracle fusion Middleware. *Oracle White paper*, 24-26.
- Huadong Ma, D. Z. (2014). Opportunities in Mobile Crowd Sensing. *IEEE* , 33-36.

- Ianace, P. (30 de Abril de 2015). *No magic*. Obtenido de Internet of Things, Remote Healthcare: <https://www.linkedin.com/pulse/delivering-iot-promise-one-example-pete-ianace>
- INEC. (Diciembre de 2011). Obtenido de Sitio Web INEC: Telecomunicaciones : <http://www.ecuadorencifras.gob.ec/documentos/web-inec/Infografias/Telecomunicaciones.pdf>
- INEC. (15 de Marzo de 2012). *INEC*. Recuperado el 1 de Diciembre de 2014, de [http://www.inec.gob.ec/sitio\\_tics2012/presentacion.pdf](http://www.inec.gob.ec/sitio_tics2012/presentacion.pdf)
- INEC. (5 de Septiembre de 2013). *INEC*. Recuperado el 1 de Diciembre de 2014, de Ecuador en Cifras: <http://www.ecuadorencifras.gob.ec/tecnologias-de-la-informacion-y-comunicacion-tic/>
- Ioannis Krontiris, M. L. (2014). Trust and Privacy in Mobile Experience Sharing: Future Challenge Avenues for research. *IEEE, Mobile Crowd Sensing*, 50-54.
- Jong Deok Choi, P. K. (2012). What is the Web Application & Web Runtime. *Tizen Developer Conference* (págs. 8-13). Ming Jin: Samsung.
- Kaikai Liu, X. L. (2014). FindingNemo: Finding Your Lost Child in Crowds via Mobile Crowd Sensing. *IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, 6-7.
- Keith, A. (12 de Jun de 2015). A Specification for serialising RDF in JSON. *CEUR Workshop Proceeding*, 2-4. Obtenido de JSON Org: <http://json.org/json-es.html>
- Kuo, R. (11 de Enero de 2015). *Samsung SSIC*. Obtenido de Strategy and Innovation Center.
- Linux Foundation. (22 de Enero de 2014). *Tizen*. Recuperado el 21 de Mayo de 2015, de About Linux Foundation: <https://www.tizen.org/es/about/about-linux-foundation>
- Loton, B. (2014). *Android Wear Smart-watch book*. (L. Tech., Ed.) EEUU.
- Luca, D. (2011). *HTML5* (Primera ed.). (USERS, Ed.) Buenos Aires, Argentina: Fox Andina & Gradi SA.
- Manraj Singh, A. K. (12 de Febrero de 2014). Temperature Detection (AETED) by using IoT. *International Journal of Computer Applications*, 121-122. Recuperado el 28 de Noviembre de 2014, de Sensordrone: <http://www.developer.sensordrone.com/getting-started/>
- Municipio de Quito. (29 de Agosto de 2015). *Secretaria de Ambiente*. Obtenido de <http://190.152.144.75/reportes/Reporte10MinutoGraph.aspx>
- Ortiz, G. L. (2013). Tecnologías involucradas en la internet del futuro. 115-127.
- Parziale, L. L. (2006). *IBM TCP/IP Tutorial and Technical OverView*. (L. P. Dr. Wei Liu, Ed.) California, EEUU: Red Book.
- Paz Felicia, D. K. (12 de Mayo de 2015). *Catregories for sport wearable technology*.

Obtenido de <http://www.worldtvpc.com/blog/category/sports-streams/>

Peter Middleton, P. K. (19 de Octubre de 2014). *Internet of Things: Manage the Complexity, Seize the Opportunity*. Obtenido de ORACLE:

<http://www.oracle.com/us/solutions/internetofthings/iot-manage-complexity-wp-2193756.pdf?ssSourceSiteId=ocomlad>

Prior, M. (Septiembre de 2014). *Sitio Web: MRP.net* . Obtenido de IPv6 Survey: Ecuatorian Universities: [http://www.mrp.net/ipv6\\_survey/diagnostics/espe.edu.ec.html](http://www.mrp.net/ipv6_survey/diagnostics/espe.edu.ec.html)

REMMAQ. (31 de Agosto de 2015). *Red Metropolitana de Monitoreo Atmosferico*.

Obtenido de CORPORACION PARA EL MEJORAMIENTO DE QUITO:  
<http://www.quito.gob.ec/index.php/quito-por-temas/ambiente>

Ricardo Barranco Fragoso, I. S. (Junio de 2012). *Que es el Big Data* . Mexico: IBM Developer Works. Obtenido de IBM Developers Work:

<http://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>

Ronan Schwarz, P. D. (2013). *The Android Developer's Cookbook* (Second ed.). EEUU: Addison-Wesley.

Sakila Banu, K. V. (Julio de 2013). An IDE for Android Mobile Phones with Extend Functionalities using best develop methodologies. *International Journal of Computer Networks & Communications (IJCNC)*, 5(4).

Samsung. (3 de Noviembre de 2014). *Tizen Architecture Samsung Developers*. EEUU: Samsung. Recuperado el 27 de Marzo de 2015, de Introduction to Gear Architecture: [http://developer.samsung.com/technical-doc/view.do?v=T000000168#gLink\\_certificate](http://developer.samsung.com/technical-doc/view.do?v=T000000168#gLink_certificate)

Sensorcorn. (14 de Junio de 2015). *Guide of Class to Control the Sensordrone*. Obtenido de Sensordrone: <http://developer.sensordrone.com/java/api/>

Sensorcorn Inc. (2014). *Sensordrone Guide Specification*. New York: Sensorcorn.

Shrestha, B. (2014). Drone to the Rescue: Relay-Resilient Authentication. *Eighteenth International Conference on Financial Cryptography and Data Security*.

Smith, P. (2013). *Professional Website Performance Optimizing the Front End and Back End*. Wiley.

Sunil Urs, A. J. (11 de Agosto de 2015). *Canvas JS*. Obtenido de <https://angel.co/canvasjs/jobs>

Svennerberg, G. (29 de Agosto de 2014). *Google Maps API 3 developers*. (S. A. Clay Andres, Ed.) EEUU: Springer Science. Obtenido de <https://developers.google.com/maps/documentation/business/support>

Tizen Org. (24 de Enero de 2014). *Tizen*. Recuperado el 20 de Agosto de 2015, de Tizen: <https://www.tizen.org/es/about>

- Trends, D. (Enero de 2015). *Digital Trends Best wearables 2014 AWARD*. Obtenido de <http://www.digitaltrends.com/ces/best-of-ces/>
- Valencia, R. (29 de Mayo de 2015). *SMART HEALTH*. Obtenido de Mobile World Congress: [https://www.b2match.eu/system/echalliance-hw2015-mwc/files/Smart\\_City\\_Health\\_\\_Wellness.pdf?1428405342](https://www.b2match.eu/system/echalliance-hw2015-mwc/files/Smart_City_Health__Wellness.pdf?1428405342)
- VDC Research Inc. (19 de Mayo de 2014). *SmartThings Samsung*. Obtenido de Smarthings: <http://www.smarthings.com/>
- Victor Pankratius, F. L. (agosto de 2013). *MIT National Science Foundation under INSPIRE*. Obtenido de Mahali Space Weather Monitoring Project: <http://mahali.mit.edu/>
- Wen Sun, Q. L.-K. (2014). Wireless Deployed and Participatory Sensing System for Enviromental Monitoring. *IEEE SECON 2014 International Conference on Sensing, Communcations and Networking*, 1-3.
- Zikopoulos, P. (Octubre de 2011). *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. McGraw-Hill.