



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



**Máster Universitario**  
en Tecnologías, Sistemas y  
Redes de Comunicaciones

# Arquitectura e Implementación de un Sistema Distribuido de Detección de Sismos para Alerta Temprana.

*Autor:* Ana María Zambrano Vizuete.

*Director:* PhD. Manuel Esteve Domingo.

*Fecha de comienzo:* Junio 2012.

*Lugar de trabajo:* Laboratorio de Sistemas de Tiempo Real Distribuidos.

### *Objetivos*

1. Estudiar, analizar, diseñar y desarrollar una arquitectura capaz de cumplir los requisitos en un sistema en tiempo real de alertas tempranas sísmicas eficiente y confiable.
2. Aprovechar la enorme difusión informática (Smartphones) como dispositivos finales en una red de sensores sísmicos que proporcionen información acerca de un evento sísmico.
3. Realizar una aplicación eficiente, de fácil uso y bajo consumo de batería que motive el uso en la ciudadanía capaz de brindar al usuario tiempo para una respuesta adecuada antes y durante un desastre sísmico.

### *Metodología*

Se define las necesidades de un país con constante riesgo sísmico que obliga la vinculación de ecuaciones de atenuación, estructuras de suelos y otros términos sismológicos necesarios; Se propone una arquitectura de Gestión de Alertas Tempranas en 2 capas apta para solventar dichas necesidades usando un modelo JDL (Join Directors of Laboratories) donde integra teléfonos inteligentes como clientes, para lo cual se hace un estudio exhaustivo de precios, mercados, sensores, utilidad, batería y sistemas operativos con el fin de realizar una aplicación capaz de enviar y recibir información en tiempo real; Y una aplicación de escritorio capaz de enviar avisos (notificaciones) al cliente en tiempo real cuando determine un evento sísmico. Por último se monta una pequeña muestra del sistema planteado y se realizan las pruebas necesarias solventando y justificando nuestros objetivos.

### *Desarrollos teóricos realizados*

La realización del proyecto necesitó una investigación previa en términos de sismología para entender, programar, y decidir valores óptimos logrando así, los mejores resultados tanto en cliente y servidor. Se investiga los términos necesarios para el estudio de acelerogramas y los procesos necesarios a seguir para que el móvil pueda comportarse como un acelerógrafo. Se compara las maneras de comunicación en ambos sentidos encontrando como solución el protocolo MQTT para las notificaciones al usuario y los beneficios del protocolo UDP para el envío de muestras al servidor. El reloj (sincronización de tiempos) cumple un papel indispensable en una aplicación de tiempo real por lo que es necesario el uso del protocolo NTP (Network Time Protocol).

### *Desarrollo de prototipos y trabajo de laboratorio*

Se plantea una arquitectura Cliente-Servidor; El cliente que corresponde a los diferentes dispositivos móviles que envían información acerca de sus cambios de aceleración, y un servidor encargado de detectar y notificar a la aplicación cliente la ocurrencia de un evento sísmico. Se desarrolla un estudio acerca del funcionamiento y características del acelerómetro además del GPS. Se escoge un IDE (Infraestructura de Datos Especiales) adecuado (ECLIPSE) para poder interactuar en bajo nivel con estos sensores en el Sistema Operativo escogido (ANDROID). Además bajo el mismo IDE se realizan validaciones y métodos necesarios para que la aplicación realizada no sea un peso para el usuario sino más bien una herramienta de ayuda.

## 2 Arquitectura e Implementación de un Sistema Distribuido de Detección de Sismos para Alerta Temprana.

Por otro lado, el servidor consiste en tres partes, la primera encargada de comunicarse con el Usuario y realizada en JAVA, la segunda un gestor de base de datos MySQL para el almacenamiento de la información; y por último la tercera, para la detección y visualización de datos realizada en MATLAB.

### *Resultados*

Se obtiene una aplicación móvil realizada para el sistema operativo ANDROID capaz de detectar movimientos sísmicos con una gestión de batería y reconexiones de red adecuada, y una aplicación de escritorio, que representa al servidor de la arquitectura, realizado en dos lenguajes de programación MATLAB y JAVA encargado de determinar si ha existido un evento o no. Los dos logran una comunicación en TIEMPO REAL garantizando tiempo útil para la toma de una mejor decisión ante un evento de la que puede depender una o más vidas.

### *Líneas futuras*

Ampliar el sistema a una arquitectura en 3 capas logrando un Sistema de Gestión de Emergencias Global. El tercer nivel será el centro de mando, el lugar donde se toman las decisiones. Por otro lado lograr ya no solo un proceso PRE-EVENTO, sino también uno POST-EVENTO el cual presente el compartimiento de recursos multimedia, ayudas al usuario con rutas óptimas a centros de refugio y también, dar aviso de rescate de usuarios que notifiquen la necesidad de ayuda a través de la aplicación móvil instalada. Por último, validar la calidad del sistema con datos reales obtenidos del Instituto Geofísico Nacional del Ecuador (IGEPN).

### *Publicaciones*

QUAKE DETECTION SYSTEM USING SMARTPHONE-BASED WIRELESS SENSOR NETWORK FOR EARLY WARNING

REALWSN 2013 -Fifth Workshop on Real-World Wireless Sensor Networks – Italy.

**Estado:** ENVIADO – Notificación de aceptación: 08/08/2013

### *Abstract*

An early warning management for emergencies has gained notoriety among the population because of the crucial role it plays in reducing the losses of facilities, goods, properties, and more important, human lives. However, this requires an adequate preventive emergency system, which is not usually deployed in developing countries. In this paper a reliable, scalable and adaptable 2-layered architecture of sensor nodes is proposed in order to detect seismic events using smartphones as accelerographs, forming a low cost wireless sensor. The first contribution is a low-energy consumption application developed in Android OS that allows distinguishing normal user movements from a seismic event. The key contribution of this research is the definition of a detection process to send real-time notifications to users within fixed error and reliability ranges. This detection process is enhanced by introduction of two features that have not been used in previous works: a spatial-zoning-analysis using attenuation equations.

**Autor:** Zambrano Vizueté, Ana María email: [anzamvi@posgrado.upv.es](mailto:anzamvi@posgrado.upv.es)

**Director:** Esteve Domingo, Manuel email: [mesteve@dcom.upv.es](mailto:mesteve@dcom.upv.es)

**Fecha de entrega:** 15-07-13

## **ÍNDICE**

<b>I. INTRODUCCIÓN .....</b>	<b>5</b>
<b>II. ESTADO DEL ARTE.....</b>	<b>6</b>
<b>II.1. SISTEMAS DE GESTIÓN DE ALARMAS TEMPRANAS .....</b>	<b>6</b>
<b>II.2. TRABAJOS PREVIOS.....</b>	<b>6</b>
<b>III. DEFINICIÓN DE LA ARQUITECTURA .....</b>	<b>7</b>
<b>III.1. MODELO JDL (Join Directos of Laboratories).....</b>	<b>8</b>
<b>III.2. PRIMERA CAPA - CLIENTE .....</b>	<b>9</b>
<b>III.2.1. SENSORES EN UN SMARTPHONE .....</b>	<b>10</b>
<b>III.2.2. PROCESO DE UN ARRAY DE DATOS DE ACELERACIÓN .....</b>	<b>11</b>
<b>III.2.3. OBTENCIÓN DE LA INFORMACIÓN .....</b>	<b>12</b>
<b>III.2.4. DETERMINACIÓN DE LA INTENSIDAD .....</b>	<b>12</b>
<b>III.2.5. TRANSFORMADA DE FOURIER.....</b>	<b>14</b>
<b>III.2.6. DETECCIÓN DE PICOS STA/LTA.....</b>	<b>15</b>
<b>III.2.7. NTP (NETWORK TIME PROTOCOL).....</b>	<b>15</b>
<b>III.3. SEGUNDA CAPA - SERVIDOR.....</b>	<b>16</b>
<b>III.3.1. ESQUEMA DE BASE DE DATOS.....</b>	<b>16</b>
<b>III.3.2. DIAGRAMA DE FLUJO.....</b>	<b>17</b>
<b>III.3.3. RANGO DE LEJANÍA AL SERVIDOR CON HAVERSINE.....</b>	<b>17</b>
<b>III.3.4. TAMAÑO DE LA MUESTRA MÍNIMA .....</b>	<b>19</b>
<b>III.3.5. HISTOGRAMA DE LA INTENSIDAD EN EL TIEMPO.....</b>	<b>20</b>
<b>III.3.6. VENTANAS DESLIZANTES .....</b>	<b>21</b>
<b>III.3.7. ANÁLISIS DE LA VARIANZA KRUSKAL-WALLIS.....</b>	<b>22</b>
<b>IV. COMUNICACIONES.....</b>	<b>23</b>
<b>IV.1. CLIENTE - SERVIDOR .....</b>	<b>23</b>
<b>IV.2. SERVIDOR - CLIENTE .....</b>	<b>23</b>
<b>IV.2.1. MQTT (MESSAGE QUEUE TELEMETRY TRANSPORT).....</b>	<b>23</b>
<b>IV.2.1.1. AGENTES NECESARIOS.....</b>	<b>24</b>
<b>IV.2.1.2. PARÁMETROS ADICIONALES.....</b>	<b>25</b>
<b>IV.2.2. ESCENARIO A REALIZAR.....</b>	<b>25</b>
<b>IV.2.3. NOTIFICACIONES AL USUARIO .....</b>	<b>26</b>

<b>V. RESULTADOS PRELIMINARES</b> .....	<b>26</b>
<b>V.1. SERVIDOR MATLAB</b> .....	26
<b>V.2. SERVIDOR JAVA</b> .....	27
<b>V.3. APLICACIÓN CLIENTE (e-Quake)</b> .....	27
<b>V.4. PRUEBAS REALIZADAS</b> .....	28
<b>V.4.1. NOTIFICACIONES MQTT</b> .....	28
<b>V.4.2. CONSUMO DE BATERÍA</b> .....	29
<b>V.4.3. DETECCIÓN DE PICOS EN MOVIL</b> .....	29
<b>VI. CONCLUSIONES</b> .....	<b>30</b>
<b>VII. TRABAJO FUTURO</b> .....	<b>30</b>
<b>VII.1. ARQUITECTURA EN 3 CAPAS</b> .....	31
<b>VII.2. GESTIÓN POST-EVENTO</b> .....	31
<b>VII.3. COMUNICACIONES WSN (WIRELESS SENSOR NETWORK)</b> .....	32
<b>VII.4. VALIDACIÓN DE CALIDAD DEL SISTEMA A TRAVÉS DE DATOS REALES</b> ..	33
<b>AGRADECIMIENTOS Y DEDICATORIA</b>	
<b>Referencias</b> .....	<b>34</b>
<b>ANEXO 1 - Determinación de frecuencia de muestreo óptima:</b> .....	<b>36</b>
<b>ANEXO 2 - Selección del Mecanismo de Notificación Óptimo</b> .....	<b>37</b>
<b>ANEXO 3 – Publicación enviada</b> .....	<b>38</b>

## I. INTRODUCCIÓN

La actividad sísmica es sin duda uno de los desastres naturales más frecuentes y destructivos que causa pérdidas de millones de euros en bienes, estructuras, propiedades y peor aún, vidas humanas. Se dice que cerca del 80% de terremotos terrestres son originados en la parte Circui-Pacífica, un área de montañas y volcanes delimitado por el Océano Pacífico, del cual Ecuador forma parte. Es impresionante la cantidad de muertos que dejaron tres sismos, el que afectó al eje cafetero de Colombia en 1999; el sismo de Pisco de 2007 en Perú y el sismo de 2010 que causó tanto daño en la región central de Chile. En el último siglo, el Instituto Geofísico de la Escuela Politécnica Nacional del Ecuador (IGEPN) [1] registró un total de 20 sismos con intensidad de 8 grados en la escala de Mercalli, entre ellos el del 31 de enero de 1906, considerado el más fuerte del país y el quinto más fuerte del mundo [2] reflejando también la gran vulnerabilidad sísmica a que Ecuador se ve afectado.

¿Qué pasaría si ocurriera un terremoto de categoría 7, como el que azotó a Haití en 2010? Las decenas de casas destruidas, edificios colapsados, víctimas atrapadas entre los escombros y la desesperación de sobrevivientes no están lejos de la realidad del país por la condición sísmica anteriormente citada, y sobre todo por la vulnerabilidad en las provincias que no están preparadas para enfrentarlo. Esto es un simple justificativo de lo importante que sería mantener un sistema de gestión de riesgos y alertas tempranas donde cientos (miles) de personas puedan ser prevenidos, **brindándoles segundos para una mejor decisión la cual puede ser la diferencia entre permanecer vivo o no** [3]. Vulcanólogos y técnicos del IGEPN tienen el temor de que mientras en el país ni autoridades ni la población asumen sus responsabilidades, bajo la corteza terrestre la tierra no deja de temblar, la energía se acumula a un ritmo continuo y el riesgo de un gran terremoto está cada vez más cerca, por el comportamiento histórico de los hechos sísmicos en el país.

Una gestión de emergencias que prevenga eventos de catástrofe es crucial, una o muchas vidas pueden estar a cargo de una sola decisión la cual está ligada al tiempo que alguien tiene para tomarla. Por ejemplo en México (1995) en un sismo de magnitud 8.0, miles de vidas se salvaron porque las personas fueron alertadas con 50 segundos de antelación al evento usando un sistema de alertas tempranas [4] pero sin embargo no todos los países en constante riesgo realizan una adecuada gestión pre-evento.

A finales del 2013 el número de dispositivos conectados a internet excederá el número de personas en la Tierra, y para el 2017 habrá cerca de 1.4 móviles por cada una, lo que significaría más de 10 billones de dispositivos móviles conectados [5]. Aprovechando este boom electrónico, el smartphone, una pequeña computadora multi-propósito, multi-sensor y multi-red, se define como parte de la solución planteada comportándose como un acelerógrafo; Así se logra una red de sensores de **bajo costo, factible, de fácil mantenimiento que obtiene muestras correspondientes a movimientos sísmicos y los envían a un servidor para procesarlas en tiempo real.**

## II. ESTADO DEL ARTE

### II.1. SISTEMAS DE ALERTAS TEMPRANAS

“Hoy en día, se reconoce ampliamente que la alerta temprana es mucho más que una cuestión científica o tecnológica relacionada con el monitoreo de las amenazas, los pronósticos, las telecomunicaciones, la climatología, la vulcanología y la sismología.” [6]

Sistemas de alertas temprana, que son la clave para la reducción efectiva de riesgos, tiene como propósito fundamental el fortalecer la capacidad de anticipar una posible amenaza natural, en este caso un evento sísmico, para mejorar la preparación, mitigación y la respuesta humana, la cual puede ser vital para una o más personas. Son más que sistemas de monitoreo científicamente avanzados ya que toda la tecnología sofisticada no valdría la pena si no se llega a la comunidad.

Los procesos de alerta temprana eficaces estudian las posibilidades actuales de la sociedad y de la zona, adaptando el sistema a necesidades y requerimientos económicos, políticos y sociales del lugar para brindar a la ciudadanía el mayor “tiempo útil” para la toma de una decisión adecuada; El término “tiempo útil” es relativo y no significa que debe ser recibido de manera inmediata, sino que, debe ser recibida antes o HASTA el tiempo que esa información sea útil; Si la información llega luego de este periodo de tiempo, dejará de ser útil para el control ya que la toma de decisiones se retrasaría dejando de ser óptimo y funcional.

Tan importante es la toma de decisiones antes de un evento de catástrofe que la decisión durante y después del mismo, por tanto sería necesario completar el sistema con procesos de gestión Post-evento que preste una ayuda global agilizando y optimizando tiempos de respuesta, rescate, y lo más importante decisiones.

### II.2. TRABAJOS PREVIOS

La gestión de emergencias es una preocupación constante convirtiéndose en un reto para investigadores, por tal razón existen trabajos dedicados a mejorar la detección temprana de sismos que entre los más destacados citamos a continuación:

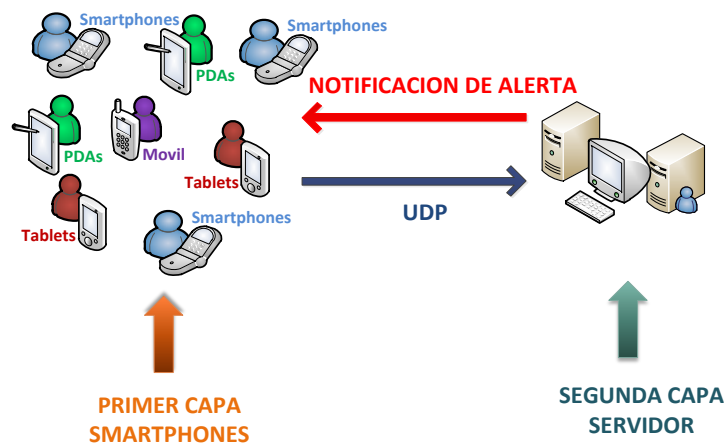
- **QuakeCast [7]:** Un proyecto que logra determinar el epicentro de un sismo y la intensidad del mismo con el uso de acelerómetros, para luego enviar alarmas cuando se considere que ha existido un evento. QuakeCast conforma su equipo de detección mediante un acelerómetro en 3 ejes y un ordenador personal, brindando una gran precisión pero a costa de una movilidad nula, por lo que es orientado a la cooperación de centros de primeros auxilios más no a alertar a la ciudadanía. El uso de sensores fijos permite la detección mediante ondas primarias P y secundarias S lo cual hace a QuakeCast muy exacto, pero aun así limita mucho el escenario de ayuda. Por contrario a este proyecto, la arquitectura propuesta aprovecha

recursos de la misma ciudadanía para ser alertados antes de que un evento se produzca teniendo un amplio escenario, y pudiendo alertar a un creciente número de clientes a través de una aplicación móvil que sea totalmente transparente para el usuario.

- **IShake [8]:** Sistema desarrollado por la Universidad Berkeley el cual usa smartphones para medir la aceleración y determinar la llegada de un sismo usando solamente dispositivos APPLE iOS aunque la tendencia de mercado se encuentra en el uso de ANDROID. La arquitectura realizada complementa a IShake y toma ventaja del mismo al incorporar un análisis espacial en el servidor el cual este no considera y será detallado más adelante.
- **Community Sensor Network CSN [9]:** Creado por el Instituto de Tecnología en California es uno de los más completos proyectos. Toma muestras de smartphones para la detección de un sismo y distribuye la información con la nube de Google al usar servidores virtuales provenientes de la arquitectura “Google App Engine” [10]. El uso de terceras partes, obliga a cumplir condiciones para su implementación, limita el proyecto a ciertas ventajas y depende de su estado funcional, que en el caso de un evento sísmico donde las comunicaciones usualmente caen, es muy probable que este deje de brindar su servicio dejando totalmente inhabilitado las funcionalidades de CSN.

### III. DEFINICIÓN DE LA ARQUITECTURA

Se plantea una arquitectura cliente-servidor mostrada en la Fig. 1 para solventar las necesidades de un sistema de alertas tempranas que aproveche los recursos de la sociedad formando una red de sensores móviles de fácil manejo, que se acople a cambios como la llegada de más usuarios (sensores) al sistema convirtiéndose en escalable, y siendo indispensable las notificaciones en tiempo real para entregar tiempo útil de decisión a un usuario en un evento sísmico. Tomando en cuenta estas características escogemos el modelo JDL [11] para fusión de data en sensores la cual ayudará a cumplir los objetivos planteados.



**Figura 1:** Arquitectura Cliente – Servidor.



La arquitectura propuesta es una plataforma independiente, modificable a los requerimientos y cambios de la zona y de la población, multi-red que en el caso de un evento sísmico pueda ofrecer algún tipo de comunicación alternativo por ejemplo WIFI, WiMAX, 3G o 4G que garantice la notificación al usuario. Además, siendo desarrollada bajo JAVA – Android no dependemos de terceras partes sometiéndonos a cambios constantes.

La mayor diferencia con trabajos previos es la segunda capa de la arquitectura (servidor), el cual manipula, integra y procesa las muestras usando un proceso de Análisis de la Varianza ANOVA llamado Kruskal-Wallis [12]; un algoritmo de ventanas deslizantes; usa MQTT [13] como protocolo de mensajería en tiempo real introduciéndonos en el campo de internet de las cosas, y por último la mayor contribución es el análisis espacial por zonas usando ecuaciones de atenuación y el algoritmo Haversine [14] para lograr encontrar el rango óptimo (radio de distancia) que cada servidor debería cubrir y aceptar muestras mejorando así, el proceso de correlación y a su vez un servidor más eficiente y preciso.

### III.1. MODELO JDL (Joint Directors of Laboratories)

Es un modelo conocido dentro de la comunidad de la fusión de datos originado por U.S Joint Directos of Laboratory y el departamento de defensa americano DoD9 en 1997. Presenta cinco niveles de procesado mostrados en la Fig. 2 acoplados a nuestro escenario, conjunto con una base de datos y un bus de conexión entre los componentes:

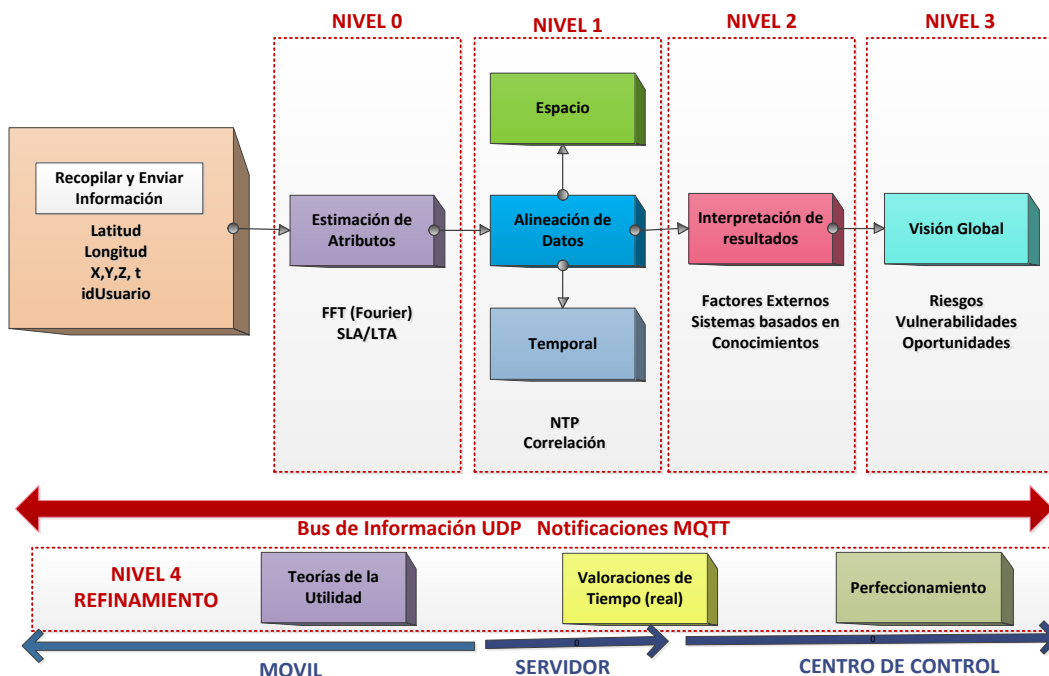


Figura 2: Modelo JDL.

- **FUENTES:** Es cada sensor/smartphone el cual obtiene muestras (Latitud; longitud; [x, y, z] aceleraciones; tiempo; identificativo de Usuario).

- **INTERACCIÓN PERSONA-SENSOR:** Aplicación en Android que denominaremos “e-Quake” capaz de transmitir los datos obtenidos al servidor y recibir la notificación de alerta.
- **BBDD:** El sistema de gestor de base de datos MySQL capaz de recopilar la información de cada fuente.
- **FUSION DE SEÑAL (NIVEL 0):** Sirve para reducir la cantidad de datos y mantener información útil diferenciando entre movimientos usuales del cliente como caminar, trotar, correr o sentarse, y un evento sísmico.
- **VALORACIÓN DE OBJETOS (NIVEL 1):** Alineación espacio temporal, correlación, eliminación de falsos positivos.
- **VALORACIÓN DE LA SITUACIÓN (NIVEL 2):** Valoración de la situación, identificación de la situación más probable.
- **VALORACIÓN DEL IMPACTO (NIVEL 3):** Para identificar riesgos, vulnerabilidades y oportunidades operacionales. Valorar el nivel de riesgo y amenaza de las actividades detectadas por el nivel 1.
- **REFINAMIENTO DEL PROCESO DE FUSIÓN (NIVEL 4):** Refinamiento del proceso de fusión, monitorear el rendimiento del sistema.

**NOTA:** Los niveles 2, 3 y 4. Son parte del trabajo futuro con una tercera capa jerárquica llamado Centro de Control.

### III.2. PRIMERA CAPA - CLIENTE

Esta primera capa, mostrada en la Fig. 3, corresponde a una aplicación móvil que toma el nombre “e-Quake” realizada en ANDROID [15] escogido por sus claras ventajas citadas en la Tabla. 1, donde a través de sus sensores recopila, procesa y valida información para determinar picos de aceleración que correspondan a un evento sísmico diferenciándolos de actividades usuales del cliente.

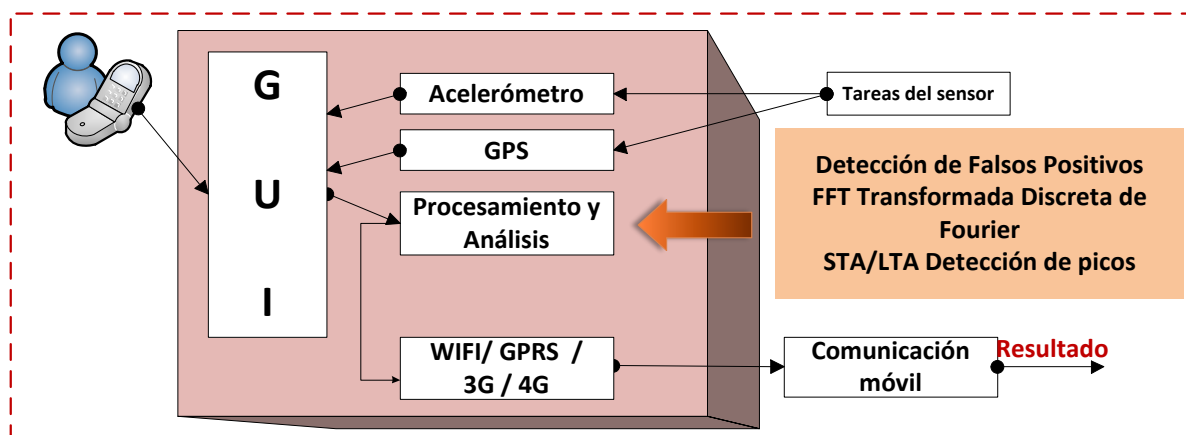


Figura 3: Diagrama de bloques Cliente.

Mercado	Tiene una amplia gama de móviles ya que puede instalarse en prácticamente todo tipo de dispositivos, sean móviles, portátiles e incluso microondas encontrando así más posibles clientes para la arquitectura. Por otro lado otros SO se ven obligados a estar rezagados en un terminal o estar limitado a una determinada marca.
Código Abierto	Android se encuentra liberado con una licencia APACHE y código abierto lo cual permite a un desarrollador modificar código a su antojo y necesidades.
Costo	Es un sistema gratuito, tanto para el programador como para el Cliente. (en la mayoría de aplicaciones). “e-Quake” será gratuito a la comunidad.
Multi-Tarea	Presenta un sistema multitarea inteligente capaz de gestionar varias aplicaciones a la vez e incluso poder dejar algunos en background. “e-Quake” debe ser parte de una red de sensores que envía información a un servidor de manera que para el cliente sea imperceptible.
Libertad	Este SO permite al usuario la libertad de instalar cualquier aplicación, desde un ejecutable (.apk), música, documentos, así como cualquier aplicación colgada por Android Market que es el lugar donde se espera publicar la “e-Quake” siendo totalmente libre.
Personalizable	Ya que presenta códigos abiertos, el programador puede modificar y hasta mejorar código ayudando así a acoplarse de la mejor manera a las necesidades sociales y requerimientos del sistema.
Comunidad	No solamente cuenta con la comunidad más grande de desarrolladores sino que ofrece múltiples vías de comunicación y ayudas que facilitan el trabajo del programador.
Integración con Google	Se adapta sin problemas a los servicios ofrecidos por Google (Gmail, Youtube, Google Talk, Google Chrome, Reader, Google Plus, Google Maps, Google Earth, Picassa, etc.).

**Tabla 1:** Ventajas de S.O Android.

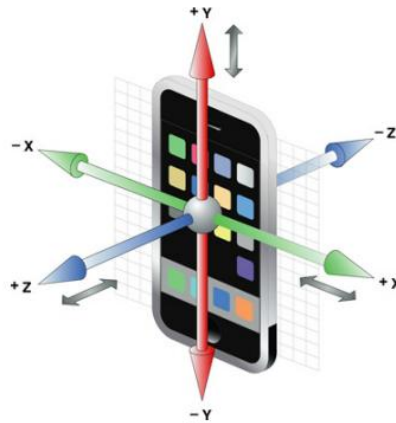
Un smartphone no ha sido diseñado para comportarse como un acelerógrafo, y está expuesto a las actividades que el usuario se encuentre realizando como caminar, trotar o saltar, no es posible usar los valores puros de los sensores porque se verían distorsionados por estas actividades, razón por la cual es necesario someter a esta información a un proceso de limpieza detallado a continuación.

### *III.2.1. SENSORES EN UN SMARTPHONE*

Hoy en día un smartphone presenta un sin número de características que en un inicio eran solamente imaginables: ha pasado de ser un dispositivo netamente orientado a la comunicación a un dispositivo de entretenimiento multimedia, monitoreo, ayuda, comunicación, entre otros y esto gracias a los

sensores que éste con el pasar del tiempo ha ido incorporando; Entre los sensores más comunes en un smartphone están: Acelerómetro, brújula o magnetómetro, giroscopio, proximidad, micrófono, GPS (Sistema de Posicionamiento Global).

El sensor primordial en esta investigación es el acelerómetro, un dispositivo que mide los cambios de aceleración en 3 ejes [x, y, z], como muestra la Fig. 4, en  $m/s^2$  o en términos de la fuerza de gravedad “g” en el rango [- 1g y + 2g]. En otras palabras mide que tan rápidos son los cambios de velocidad de un objeto.



**Figura 4:** Ejes de un acelerómetro en un Smartphone.

Por otro lado GPS que en un inicio fue utilizado solamente para aplicaciones militares y en 1980 fue habilitado para aplicaciones civiles, permite determinar la localización de un objeto mediante una su latitud y longitud con pocos metros de precisión; Sin embargo algunos smartphones incorporan AGPS o “Assisted GPS” que incorpora un servidor intermedio, que cuando no puede conectar con los satélites (muy frecuente indoor), este provee una satélite cercano con información adicional para determinar la ubicación exacta.

“e-Quake” hace uso de estos dos sensores para simular un acelerógrafo y obtener información de picos de aceleración que puedan corresponder a un evento sísmico conjunto con localización, la cual es primordial para la validación en el servidor quien notifica al cliente.

### *III.2.2. PROCESO DE UN ARRAY DE DATOS DE ACELERACIÓN*

El poder destructivo de un sismo se define por 3 puntos: valores picos, duración del movimiento y contenido espectral; La manera más precisa de determinar un sismo es mediante un análisis de ondas Primarias (P) y ondas Secundarias (S) [16], sin embargo este proceso necesita mucha exactitud y estabilidad, lo cual los sensores usados no presentan por su escenario de uso diario, por esta razón enfocamos nuestro estudio a analizar los dos primeros puntos (valores picos y duración del movimiento) para la detección de sismos en un smartphone; para ello usamos el acelerómetro [17]

con el que obtenemos una historia en el tiempo de las aceleraciones registradas. Se plantea un algoritmo para limpiar lo mejor posible la señal mostrada en la Fig. 5:

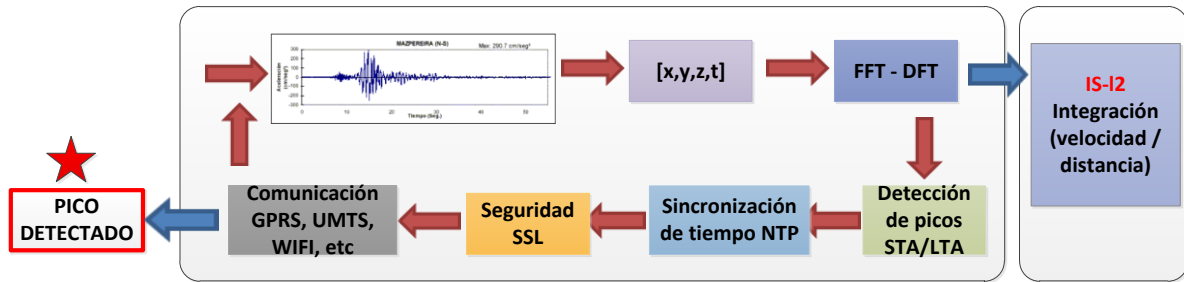


Figura 5: Diagrama de bloques del proceso de datos de aceleración.

El diagrama de bloques se convierte en un bucle infinito donde siempre se encuentra tomando nuevas muestras del acelerómetro. Todo el proceso debe ser ejecutado en background, con el fin de que el usuario sea parte de la arquitectura ayudando con el envío de muestras constantes sin que interfiera con el uso habitual del usuario.

### III.2.3. OBTENCIÓN DE INFORMACIÓN

Un sismógrafo de los que utiliza actualmente el Geofísico Nacional tiene una precisión de alrededor de 10 a 12 bits y obviamente los acelerómetros MEMS utilizados en móviles inteligentes no alcanzan esta precisión, pero se **pretende conseguir mayor precisión cuanto mayor número de muestras (obtenidas por la red de sensores) sean fusionadas [18].**

Sin duda alguna el éxito sería que el usuario siempre permita el envío de información con cada movimiento, pero se convierte inviable ya que este sensor puede recoger datos alrededor de 35 veces por segundo, ver *ANEXO 1*, con lo cual devastaríamos a la batería del sensor. Con todo esto nos centramos en el problema de definir la frecuencia de muestreo, o en otras palabras, cada que tiempo se debe enviar muestras para llegar a un buen equilibrio en Ec. 1. La tasa o frecuencia de muestreo es el número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta durante el proceso necesario para convertirla de analógica en digital. Se obtiene una frecuencia de muestreo siguiendo la siguiente fórmula detallada en *ANEXO 1*:

$$f = 2W = 2AB = 2 * 10Hz \quad \therefore f_{Nyquist} = 20Hz \quad \therefore t_{muestreo} = 0.05seg \quad (1)$$

### III.2.4. DETERMINACIÓN DE LA INTENSIDAD MINIMA

Parte muy importante es saber que el valor que realmente interesa es la INTENSIDAD, más no la magnitud ni la aceleración total, aunque se encuentren muy ligadas; La magnitud de un sismo es un parámetro para determinar la cantidad de energía liberada, y por el contrario, la intensidad, es una

medida que evalúa el daño ocasionado por el mismo [19] medida en una escala [0-12]. Mediante ecuaciones de atenuación Ec. 2, que relaciona la intensidad con la aceleración, intentamos obtener el rango de aceleraciones para las cuales la Intensidad es demasiado baja, por lo que pierde sentido el anunciarlo como sismo.

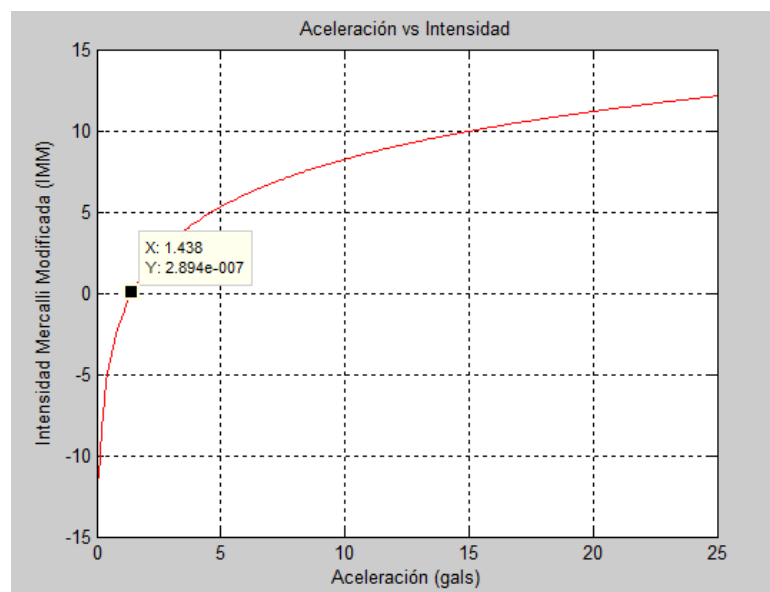
Según Aguiar y Castro (2009) – La Ecuación para Ecuador según datos macro sísmicos:

$$\text{Log}(A_{\max}) = 0.3629 + 0.2355 * I_{MM} \tag{2}$$

Donde  $I_{MM}$  es la Intensidad de Mercalli Modificada y  $A_{\max}$  es la aceleración máxima registrada, que despejando se obtienen los valores mostrados tanto en la Tabla. 2 y representados en la Fig. 6:

$I_{MM}$	$\text{Log}(A_{\max})$ [gals]	$A_{\max}$ [gals]
0	0.3629	1.4375
1	0.5984	1.8192
2	0.8339	2.3023
3	1.0694	2.9136
4	1.3049	3.6873
5	1.5404	4.6665
6	1.7759	5.9056
7	2.0114	7.4738
8	2.1479	8.5668
9	2.4824	11.970
10	2.7179	15.148
11	2.9534	19.171
12	3.1889	<b>24.262</b>

**Tabla 2:** Valores de Aceleración vs. Intensidad.



**Figura 6:** Gráfica de Aceleración vs. Intensidad

Existen valores de aceleración para los cuales la intensidad se convierte en negativa perdiendo sentido, por tanto eliminamos este rango de aceleraciones cuando  $I_{MM}=0$  obteniendo la Ec. 3:

$$I_{MM} = \frac{\log(A_{\max}) - 0.3629}{0.2355} = 0 \quad (3)$$

$$e^{\log(A_{\max})} = e^{0.3629} \rightarrow A_{\max} = 1.4375$$

Las ventajas del análisis son:

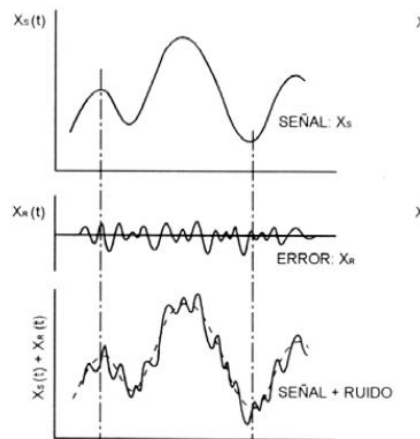
- Obtenemos menos datos en el servidor para ser procesados.
- El móvil envía menos información al servidor, por lo tanto ahorramos y cuidamos la batería.
- Mayor confiabilidad al reducir los falsos positivos enviados.

### III.2.5. TRANSFORMADA DE FOURIER

La transformada de Fourier [20] es una herramienta matemática para convertir la señal en el dominio del tiempo al dominio de la frecuencia y de esta manera poder limpiar la señal  $a(t)$  en la Ec. 4, de frecuencias que solamente representan ruido.

$$a(t) = X_s(t) + X_R(t) \quad (4)$$

Siendo  $a(t)$  el acelerograma sin corregir,  $X_s(t)$  la señal sísmica y  $X_R(t)$  el ruido mostrada en la Fig.7:



**Figura 7:** Presencia de ruido en una señal.

Existen dos tipos de ruidos a los que una señal puede verse afectada, que se citan posteriormente:

- **El ruido de alta frecuencia:** Cuando aparece por encima de la banda de frecuencias en la que trabajamos. Este ruido afecta fundamentalmente a los picos de aceleración pero NO afecta al área de derivación (velocidad, distancia) ya que el área bajo la curva es prácticamente la misma que para la señal corregida, dado que las desviaciones por encima y por debajo se compensan.

- **El ruido de baja frecuencia:** Cuando aparece por debajo de la banda de frecuencias en la que trabajamos. Afecta al proceso de integración básicamente, ya que al integrar la aceleración para obtener velocidad y desplazamiento, estamos obteniendo el área bajo la curva.

Las frecuencias menores a 0.5 Hz se asocian a condiciones meteorológicas. Las de aproximadamente 1 Hz relacionados a vientos locales y condiciones meteorológicas. Y las mayores a 1Hz asociadas a las actividades humanas. De esto podemos deducir que las frecuencias que queremos eliminar es el de alta frecuencia usando filtros pasa baja para eliminarlas y conseguir una señal sin perturbaciones.

Siendo imposible realizar la transformada de una señal infinita, se debe realizar por intervalos de tiempo llamadas ventanas, pero la división de ventanas es complicada ya que al ser muy corta se obtiene una buena resolución temporal pero puede ocultar componentes de mayor periodo; Y al ser muy larga, se obtiene una resolución en frecuencia precisa pero aumenta la incertidumbre temporal. En este caso se usa la transformada rápida de Fourier FFT, donde los registros digitales se encuentran formados por N muestras tomadas en intervalos regulares de tiempo separadas un periodo  $T = 1/f$  donde f es la frecuencia de muestreo antes definida.

### *III.2.6. DETECCIÓN DE PICOS STA/LTA*

El aumento de aceleración no siempre representará el que un evento sísmico haya ocurrido; así un acelerómetro puede medir un valor alto mientras el cliente está realizando una actividad física, por tanto es imposible definir un pico solamente al sobrepasar un valor fijo. El algoritmo STA/LTA [21], por medio de reglas de decisión que se basan en el establecimiento de umbrales variables, determina cuando realmente se ha detectado un pico de aceleración que corresponderá a un evento sísmico. Si la señal sobrepasa este umbral obtenido se dirá que ha existido un evento y un mensaje será enviado al servidor. La idea principal es comparar la medida de los datos en una ventana de tiempo de corta duración (STA), frente a la correspondiente en una ventana de larga duración (LTA). Esta detección de picos es muy importante al evitar picos cuando el móvil está sometido a una actividad periódica como correr, trotar, sentarse, hablar, etc.

### *III.2.7. NTP (NETWORK TIME PROTOCOL)*

Al trabajar en una estructura de gestión de alarmas tempranas un parámetro crítico es el tiempo. NTP es un protocolo para conocer la hora exacta de un proceso sincronizando los relojes de una red y se encuentra organizado en una jerarquía dinámica de servidores comenzando con el Stratus-0. Trabaja sobre el puerto 123 UDP y mantiene ejecutando un demonio en el servidor para que los clientes puedan hacer la petición de la hora correcta.



Recordamos también que nuestro propósito es la sincronización de los dispositivos de la red, mas no que todos los equipos tengan la hora exacta del día.

### III.3. SEGUNDA CAPA – SERVIDOR

El servidor correspondiente a la segunda capa de la arquitectura, no es más que un ordenador con capacidades físicas necesarias para abastecer las peticiones de los clientes que estén en la zona y donde los móviles de la primera etapa enviarán su información para ser procesada y fusionada de manera rápida garantizando que la respuesta sea en tiempo real. El proceso que sigue este servidor se encuentra expuesto en la Fig. 8:

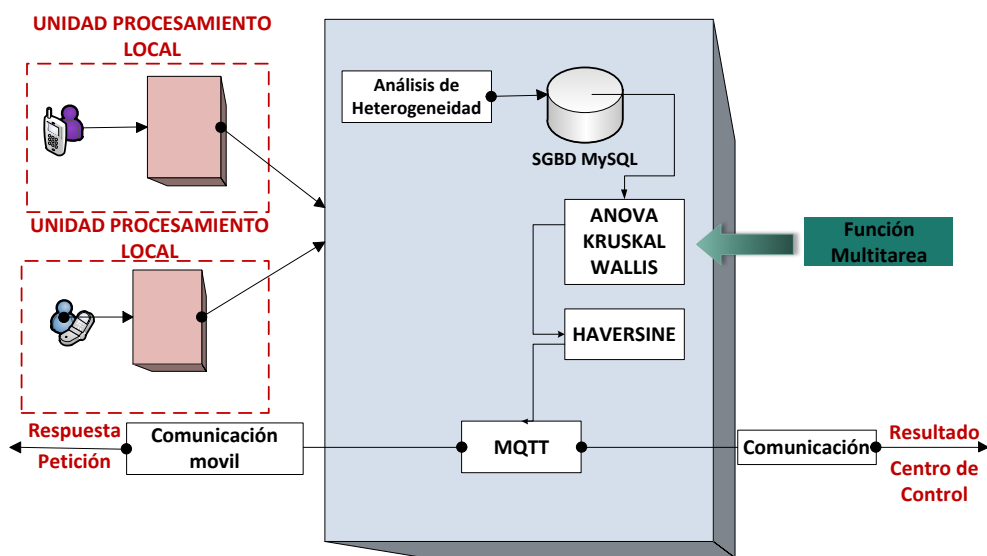


Figura 8: Diagrama de bloques Servidor.

#### III.3.1. ESQUEMA DE BASE DE DATOS

La base de datos presenta la estructura que se muestra en la Fig. 9 y su detalle en la Tabla. 3:

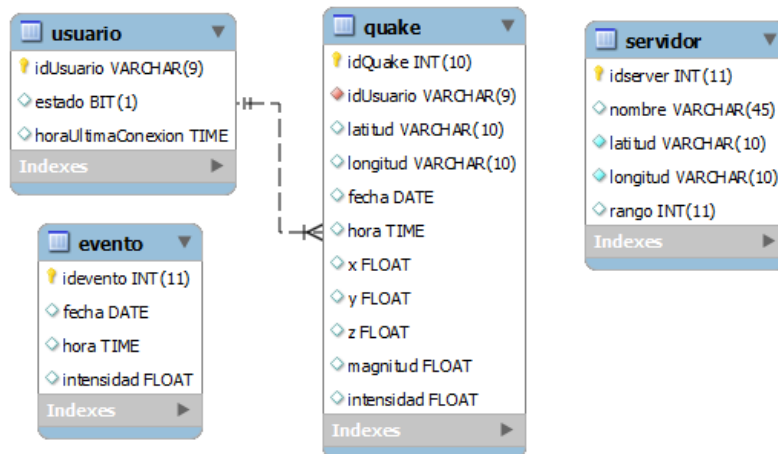


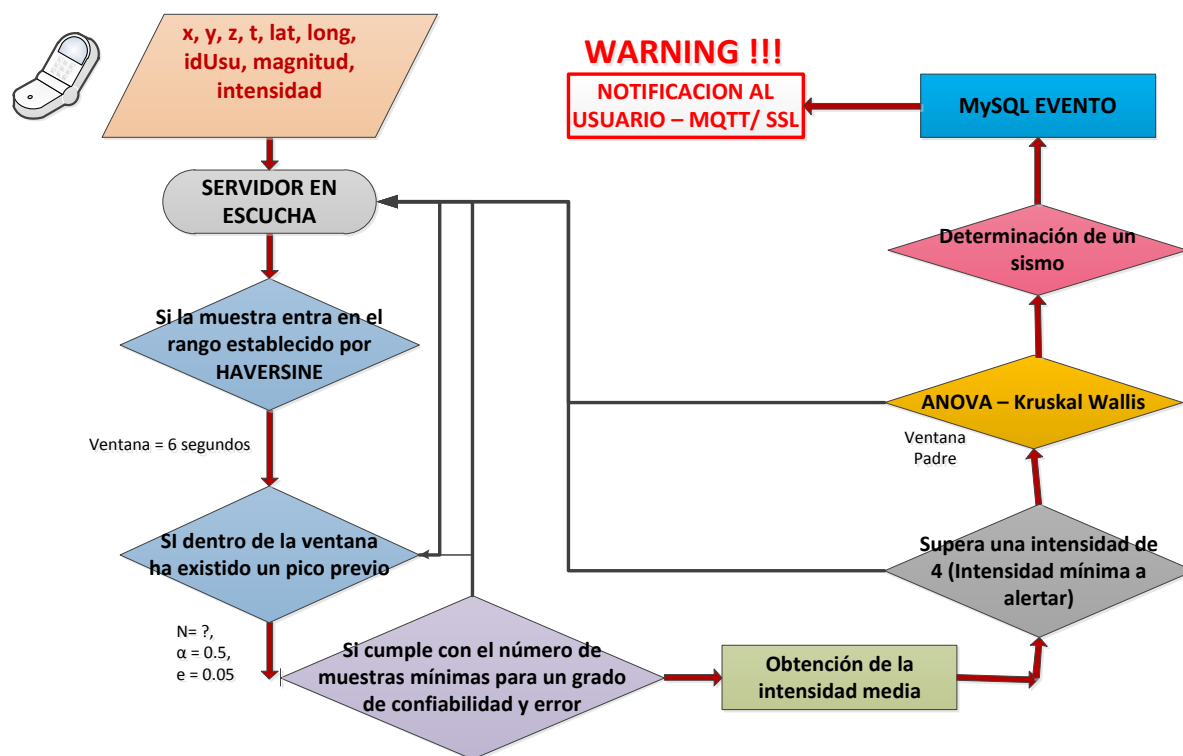
Figura 9: Modelo de base de datos en MySql.

<b>Usuario</b>	Define los usuarios que hayan instalado la aplicación y su estado (activo =1, no activo = 0) que es indispensable para la validación de muestra mínima detallado más adelante.
<b>Quake</b>	Recopila la información enviada por los sensores con la siguiente información: [id Usuario; latitud; longitud; fecha; hora; x, y, z aceleraciones; magnitud; intensidad]
<b>Servidor</b>	Representa cada servidor detallando su ubicación, y el rango que este cubre para ser o no ser tomadas en cuenta las medidas. Ej. Si una medida se encuentra a más de 35 km. no será tomada en cuenta. La determinación de este valor se detalla más adelante usando la fórmula de HAVERSINE
<b>Evento</b>	Picos detectados por el programa, por tanto también representa el número de notificaciones enviadas a los usuarios que sirve para comprobar la eficiencia del servidor y otras validaciones en el algoritmo de ventanas deslizantes.

**Tabla 3:** Descripción de tablas de la base de datos en MySQL.

### III.3.2. DIAGRAMA DE FLUJO

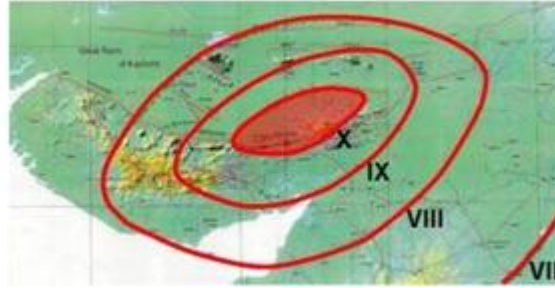
En la Fig. 10 se presenta un diagrama del proceso que sigue el servidor para determinar si se ha producido un evento sísmico o no:



**Figura 10:** Diagrama de flujo de Servidor

### III.3.3. RANGO DE LEJANÍA AL SERVIDOR CON HAVERSINE

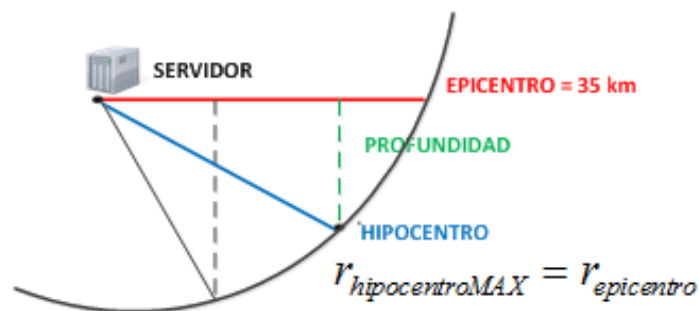
La Intensidad de un sismo es inversamente proporcional a la distancia donde es medido, por tanto, en el epicentro será mayor que en un lugar más lejano e irá disminuyendo siguiendo las leyes de atenuación tal y como se muestra en la Fig. 11.



**Figura 11:** Variación de Intensidad vs. Distancia.

En consecuencia todas las muestras no pueden ser procesadas de igual manera porque sería mezclar todo el rango de valores de intensidad [0-12], por tanto es necesario subdividir las muestras segmentando por radios de distancia (km) el cual es estudiado y determinado para luego guardarlo a la base de datos en la tabla SERVIDOR.

La distancia HIPOCENTRAL y la EPICENTRAL son diferentes términos, como se muestra en la Fig. 12, el Hipocentro es el punto donde se origina el sismo a una determinada profundidad, mientras que la EPICENTRAL es la distancia desde el punto que baja perpendicularmente de la superficie de la tierra al Hipocentro hasta el lugar medido; Por otro lado, para saber cómo varía la aceleración con respecto a la distancia Hipocentral usamos las leyes de atenuación de la zona [22] citada en la Ec. 5, y tan solo observando deducimos que la distancia a calcular (distancia de epicentro) corresponde a la distancia máxima de Hipocentro que es cuando la aceleración es máxima para una determinada magnitud.



**Figura 12:** Distancia Epicentral y Distancia Hipocentral.

$$\log(A_{gals}) = 1.2474 + 0.3735 * M_w - 0.4383 * \ln(D + 10) \quad (5)$$

Donde  $A_{gals}$  = Aceleración en [gals] y,  $M_w$  = Magnitud de Kanamori o momento Sísmico. (La magnitud a diferencia de la Intensidad se mide en una escala [0-10]) y D la distancia Hipocentral.

Obtenemos la Distancia Hipocentral para un sismo con magnitud igual a 5.5, que es el valor de la magnitud mínima desde el cual el Servidor va a alertar puesto que para valores anteriores a cinco no

existen daños materiales y a veces hasta imperceptibles para un humano [19]. Así con un sismo de 5.5 y 6 grados en magnitud, tenemos un rango de cobertura definido en la Tabla. 4:

Magnitud	Rango (km)
5.5	34.27 = <b>35 km</b>
6	56.16 = <b>56 km</b>

**Tabla 4:** Rango obtenido de las leyes de atenuación para sismos de magnitud 5.5 y 6.

Todas las muestras que no se encuentren dentro de este rango de cobertura serán descartadas por el servidor, pero sin embargo serán recogidas por otro servidor del mismo nivel que cubra esa área de cobertura.

**NOTA:** Físicamente el servidor puede situarse en cualquier lugar. En la base de datos se designa un valor (longitud, valor) con el cual se determina el valor lógico del servidor, pudiendo existir varios servidores lógicos en un mismo servidor físico.

### III.3.4. TAMAÑO DE LA MUESTRA MÍNIMA

Para que la aplicación sea fiable necesita la menor tasa de falsos positivos (que mande una alerta al cliente sin que realmente haya existido un evento), así que un solo móvil detecte un pico no determina que ha existido un evento si existen en la base de datos N usuarios más registrados. Este análisis es necesario para determinar si el número de sensores que han enviado un pico es representativo con respecto al total de sensores registrados.

Hay que tomar en cuenta que el smartphone puede no enviar picos por diferentes razones, por ejemplo que no encuentre localización alguna (no tenga su GPS activado), que no posea salida a internet, o en realidad que no haya registrado un pico.

**NOTA:** Cada sensor es registrado en base de datos (tabla Usuario) con su identificador único “número telefónico” y estado “activo”, cuando se instala por primera vez la aplicación.

El número total de clientes, correspondiente a N en la Ec. 6, no representa a los clientes registrados en base de datos, sino más bien, los usuarios REGISTRADOS-ACTIVOS por tanto, el número de sensores que están ingresando valores constantemente es el total de usuarios a trabajar; Sin tomar en cuenta esta diferencia el análisis fallaría al aumentar indiscriminadamente número total de clientes. Para solventar este problema se hacen validaciones tanto en cliente como en servidor: En cliente, cada vez que detecta un cambio en la red intenta automáticamente reconectar y además, envía “beacons” periódicos con el fin de notificarle al servidor que móviles permanecen activos realmente; Por parte del servidor, hace una validación cada cierto tiempo t (30 minutos) que verifica si el móvil en ese tiempo por lo menos ha enviado un pico, caso contrario se toma como inactivo; El servidor hace la validación con la tabla Usuario en Mysql que presenta los campos ESTADO y

HORAULTIMACONEXION; si el servidor analiza que en este periodo no ha existido un pico modifica el estado del usuario y de esta manera éste sensor no cuenta para el análisis de muestra mínima restándole al total de sensores REGISTRADOS-ACTIVOS.

El algoritmo a seguir es mostrado en Ec. 6, donde cuanto mayor número usuarios se tenga se tendrá mucha mayor precisión, y por el contrario cuanto menos usuarios la confiabilidad irá decreciendo.

$$\eta = \frac{N * Z_{\sigma/2}^2 * \sigma^2}{(N - 1) * e^2 * N * Z_{\sigma/2}^2 * \sigma^2} \tag{6}$$

Donde **N** = Población total = número de usuarios en estado 1 = Usuarios activos.; **n** = Porción de población = Número de usuarios que han enviado un pico.; **e** = máximo error permitido (1%) (0.05%); **Z** = nivel de confianza (0.95 = 1.96 sigma/errores típicos).

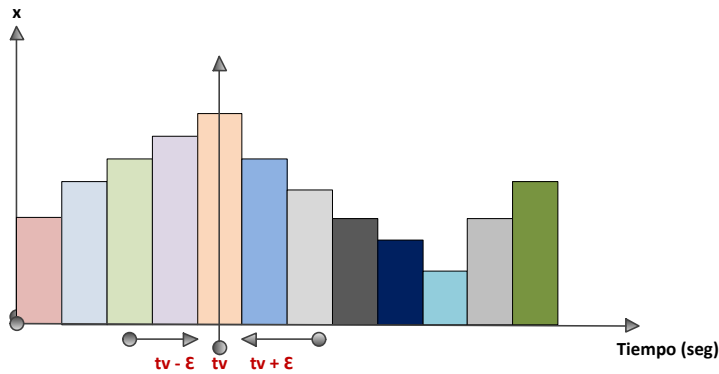
Para este caso en particular se ha trabajado con los valores mostrados en la Tabla. 5, haciendo cumplir que el número de móviles que enviaron un pico sea igual o mayor que la mínima población calculada necesaria para ser tomada en cuenta como representativa.

<b>Z</b>	<b>1.96</b>
<b>S = σ = sigma</b>	<b>0.5</b>
<b>Error = e</b>	<b>0.05</b>

**Tabla 5:** Valores tomados para análisis de muestra mínima.

### III.3.5. HISTOGRAMA DE LA INTENSIDAD EN EL TIEMPO

Este proceso es importante para ver cuánto se parecen unas muestras con respecto a otras en el transcurso del tiempo. La Fig. 13 muestra a x, una variable aleatoria que representa la intensidad del sismo y la cual queremos relacionar con otras para observar que tan correlacionados están los datos en espacios de tiempos  $\epsilon$  (segundos).



**Figura 13:** Histograma Intensidad vs tiempo.

La variación del parámetro  $\epsilon$ , en segundos, debe ser escogido correctamente ya que: (i) Entre más grande sea el valor de  $\epsilon$  se pierde resolución, sin lograr el objetivo de una correlación. (ii) Entre menor sea el valor de  $\epsilon$  se hacen histogramas de mejor resolución. (iii) Entre menor sea el valor de  $\epsilon$  puede llegar a dejarse rectángulos vacíos, lo cual no necesariamente represente que no ha existido un evento.

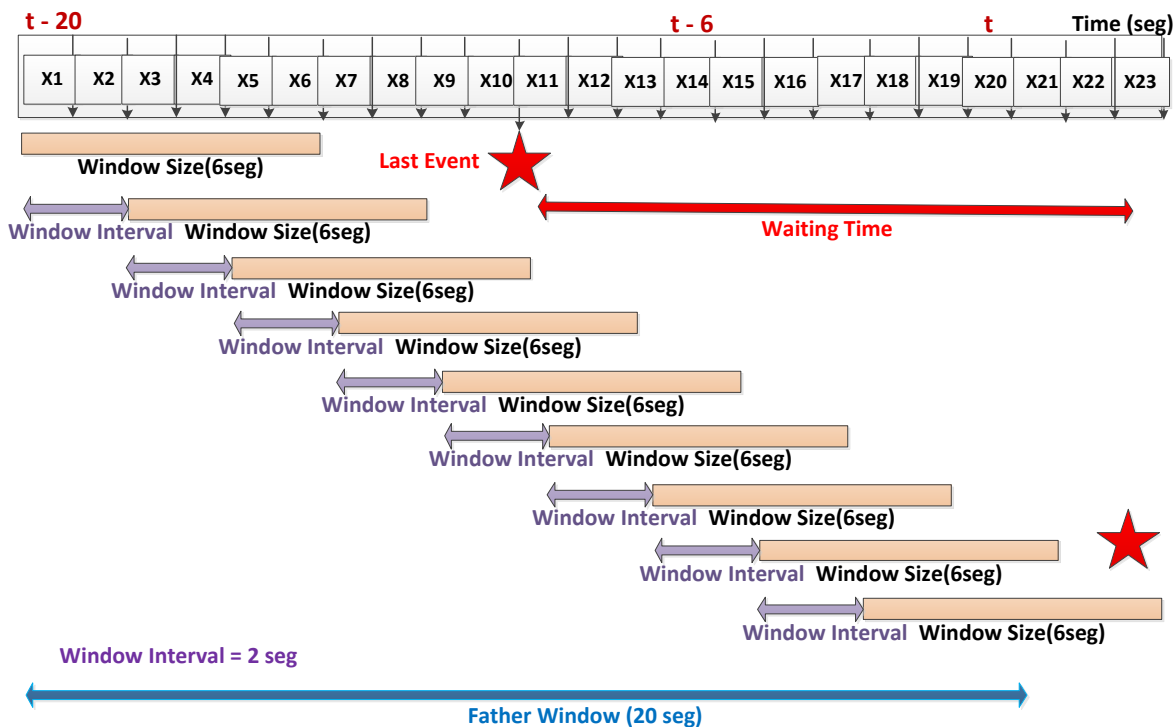
Para poder determinar el valor de  $\epsilon$  nos ayudamos de la escala de Richter que como se detalla en la Tabla. 6 para una magnitud de 5.5 la sensación en la población es moderado y para una magnitud de 6.0 la sensación sube considerablemente. Tomamos una magnitud de 5.5 como el valor mínimo para alertar a las personas y con una media de duración 6 segundos realizamos ventanas de valor  $\epsilon = 6$  segundos garantizando así una correlación de intensidades.

MAGNITUD	DURACIÓN	DAÑOS	SENSACIÓN
5.5	6	Muy leve	Moderado
6	12	Leve	Fuerte

**Tabla 6:** Sensación y daños con respecto a la magnitud de un sismo.

### III.3.6. VENTANAS DESLIZANTES

Una vez encontrado el valor de una ventana, se realiza un proceso de ventanas deslizantes, mostrada en la Fig. 14, para luego en cada una hacer un análisis de la varianza ANOVA:



**Figura 14:** Ventanas deslizantes.

El algoritmo se repite periódicamente cada 2 segundos (Windows Interval) recolectando la intensidad de todas las muestras guardadas en la base de datos en una ventana de 6 segundos (Windows Size). Cada 20 segundos, que corresponde una Ventana Padre, toma de nuevo las intensidades para realizar un análisis de varianza. Y por último con el fin de eliminar el riesgo de notificar un mismo evento o réplicas, existe un tiempo de espera (Waiting Time) de 20 minutos entre el evento actual y el último detectado.

### III.3.7. ANALISIS DE LA VARIANZA KRUSKAL-WALLIS

El Análisis de la Varianza conocido como ANOVA, sirve para probar dos hipótesis a través de un análisis entre grupos; y aunque existen un sinnúmero de posibilidades se ha decidido usar Kruskal-Wallis o también llamada prueba H [12] que es una prueba no paramétrica que utiliza rangos de datos muestrales de tres o más poblaciones independientes para probar la hipótesis nula de que las muestras independientes provienen de poblaciones con medianas iguales; y una hipótesis alternativa, que es la aseveración de que las poblaciones tienen medianas diferentes. Se considera que dos grupos son distintos cuando la variabilidad ENTRE los grupos, es mayor que la variabilidad DENTRO de los grupos probando si una misma media procede a la misma población.

Para aplicar la prueba calculamos el estadístico de prueba H con la fórmula mostrada en Ec. 7, y luego una distribución aproximada a la distribución CHI CUADRADA donde el número de grados de libertad es  $k-1$ , y donde  $k$  es el número de muestras mostrada en la ecuación.

$$H = \frac{12}{N(N+1)} \sum_{i=1}^k \frac{R_i^2}{n_i} - 3(N-1) \quad (7)$$

$$H = \frac{12}{N(N+1)} \left( \frac{R_1^2}{n_1} + \frac{R_2^2}{n_2} + \dots + \frac{R_k^2}{n_k} \right) - 3(N+1)$$

Donde  $N$  = Número total de observaciones en todas las muestras combinadas,  $k$  = número de muestras,  $R_1$  = suma de rangos de la muestra 1, que se calcula utilizando el procedimiento que se describe a continuación y  $n_1$  = número de observaciones de la muestra 1; para la muestra 2, la suma de los rangos es  $R_2$  y el número de observaciones es  $n_2$  y se utiliza una notación similar para las otras muestras.

Las validaciones se realizan de acuerdo a la Tabla. 7 donde los valores críticos ( $p$ ) usualmente usados son  $p < 0.05$ ,  $p < 0.01$ .

<b>P = Kruskal-Wallis(X, grupo)</b>		
<b>p ≈ 0</b>	Valor significativo	Pone en duda la hipótesis nula y sugiere que al menos una mediana difiere significativamente unas de otras. <b>PROVIENE DE DIFERENTES POBLACIONES</b>
<b>p &gt; 0</b>	Valor NO significativo	Hipótesis nula correcta. Proviene de la misma población. <b>PROVIENE DE UNA MISMA POBLACION</b>

**Tabla 7:** Valores críticos para el algoritmo Kruskal-Wallis.

#### IV. COMUNICACIONES

A continuación se detalla los protocolos utilizados para una comunicación en tiempo real en ambos sentidos:

##### IV.1. CLIENTE – SERVIDOR



El protocolo que permite que la aplicación móvil alcance al servidor es UDP [23] mediante datagramas permitiendo una comunicación viable, efectiva, que no represente consumo de batería significativo, CPU ni memoria, que casi no tiene retardo y se puede decir que funciona en tiempo real. Para el acceso mediante datos móviles (fuera de la red local) es necesaria la habilitación de un puerto y al tener el servidor una dirección IP pública se puede acceder sin problemas con cualquier tecnología GPRS, 3G, 4G con salida a Internet.

##### IV.2. SERVIDOR – CLIENTE



Esta comunicación hace referencia a la notificación que recibirá el cliente cuando el servidor detecte un evento. El tipo de notificación a implementar es del tipo PUSH al usar el protocolo MQTT que solventa y cumple plenamente los requerimientos del sistema *ANEXO 2*.

##### IV.2.1 (MQTT MESSAGE QUEUE TELEMETRY TRANSPORT)

MQTT [13] es un protocolo de suscripción/publicación inventado por And Stanford para IBM en 1999 usando TCP/IP por el puerto 1883 reservado por la IANA, y el 8883 registrado para MQTT con seguridad SSL. Es un protocolo abierto de mensajería que habilita la transferencia de datos a



través de una red donde su principal característica, además de brindar escalabilidad, es su ligereza por lo cual es muy utilizado desde pequeñas redes de sensores hasta grandes servidores ubicados en internet; Es ideal para dispositivos móviles porque es pequeño, usando muy poca batería, mínimo envío de paquetes y distribución eficiente de 1 a miles de receptores.

#### IV.2.1.1. AGENTES NECESARIOS

A continuación se detalla los agentes involucrados para la utilización de notificaciones MQTT:

- **CLIENTE:** Smartphones que pedirán tener notificaciones de determinados “Grupos” a través de la aplicación cliente “e-Quake”.
- **TOPIC:** “Grupos” a los que el cliente se suscribe para recibir notificaciones.
- **PAYLOAD:** La información que será mostrada en la notificación.
- **BROKER:** Es el encargado de conseguir los paquetes hacia los clientes. Existe una gran variedad de Brokers, entre los más representativos cuentan WebSphere o RSBM de IBM (Proprietarios) mientras que en software libre destacan MOSQUITTO, MOSQUETTE o ActiveMQ de Apache. El Broker utilizado es Mosquitto [24] configurado con los parámetros de la Tabla. 8:

<b>Cliente:</b>	Representa que clientes están autorizados a recibir una notificación. En nuestro caso es definido SIN AUTENTICACION para que cualquier persona nueva pueda pedir notificaciones.
<b>Seguridad</b>	Utiliza SSL como seguridad en sus mensajes. También se configura que solo clientes con un determinado PREFIJO oculto puedan acceder (En nuestro caso la palabra “quake/”). Así evitamos que otras aplicaciones o “personas mal intencionados” puedan suscribirse a la aplicación sin permiso alguno.
<b>Calidad de Servicio</b>	Puede definir 3 tipos de calidad de servicio haciendo aún más eficiente: <ol style="list-style-type: none"> <li>1. <b>[0]:</b> Es el más rápido, sin calidad de servicio.</li> <li>2. <b>[1]:</b> Puede que le llegue más de una vez la notificación hasta que el servidor reciba una notificación de llegada.</li> <li>3. <b>[2]:</b> Asegura que llegue una y solo una vez la notificación.</li> </ol>
<b>Tópicos</b>	Son los grupos en los que un cliente puede estar suscrito. En nuestro caso se realizan los siguientes tópicos pensando en que cada uno está representado por un servidor de segundo nivel en cada ciudad del Ecuador. #quake/localización_1 #quake/localización_2
<b>Accesos de red</b>	Se configura para que desde cualquier red pueda alcanzar el servidor. Así desde cualquier dirección IP el móvil pueda mandar una notificación sin importar que tipo de conexión mantenga.
<b>Accesos de red</b>	Se configura para que desde cualquier red pueda alcanzar el servidor. Así desde cualquier dirección IP el móvil pueda mandar una notificación.

**Tabla 8:** Configuración del Broker MQTT.

#### IV.2.1.2. PARMÁMETROS ADICIONALES

MQTT realiza reconexiones automáticas configuradas con las especificaciones en la Tabla. 9:

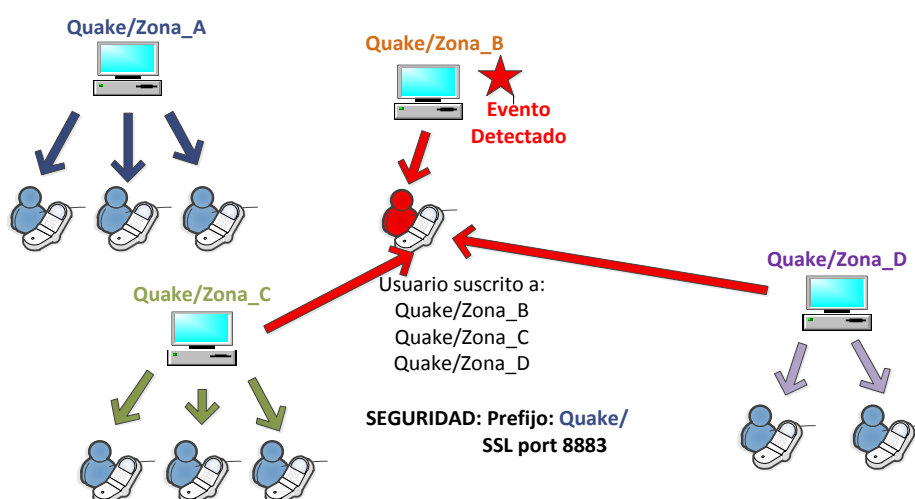
<b>MQTT-HOST</b>	Ip del Servidor: 158.42.188.46
<b>MQTT-PUERTO</b>	8883 –seguridad SSL.
<b>INTERVAL_KEEP_ALIVE</b>	15 MINUTOS: Cada este tiempo detecta si está arriba la comunicación.
<b>QoS</b>	Calidad de Servicio igual a 0. Lo más rápido posible y consumo de energético casi nulo.
<b>INITIAL_RETRY_INTERVAL</b>	10 SEGUNDOS: Intervalo cada cuanto quiere reconectar el servicio MQTT.
<b>MAXIMUM_RETRY_INTERVAL</b>	30 MINUTOS: Hasta este tiempo intenta reconectarse caso contrario deja de intentarlo ya que es demasiado consumo de batería para el dispositivo.

**Tabla 9:** Parámetros de reconexión MQTT.

#### IV.2.2. ESCENARIO A REALIZAR

El cliente mediante “e-Quake” puede decidir de qué lugares (“grupos”) recibir notificaciones inmediatas por medio de una pantalla que configura un archivo oculto para que esta permanezca guardada incluso si el móvil se reinicia o se apaga.

Por ejemplo, en la Fig. 15 el cliente en verde, se ha suscrito a 3 ciudades diferentes, por tanto recibirá esas notificaciones de alerta incluso sin actuar como sensor en todas ellas.



**Figura 15:** Escenario de Notificaciones MQTT.

### IV.2.3. NOTIFICACIONES AL USUARIO

En un escenario de alertas tempranas el aviso de un evento debe ser notificado de una manera distinguida y notoria que obligue al usuario a interrumpir la acción que se encuentre realizando. Nos ayudamos de las funcionalidades de Android para construir una notificación con la mayor prioridad, que ataquen a la mayor percepción del usuario usando luces parpadeantes, una melodía de alerta repetitiva hasta que el usuario lo cancele, y por ultimo una vibración incesante. **Los segundos obtenidos pueden significar la vida entre la vida o la muerte de una o más personas.**

## V. RESULTADOS PRELIMINARES

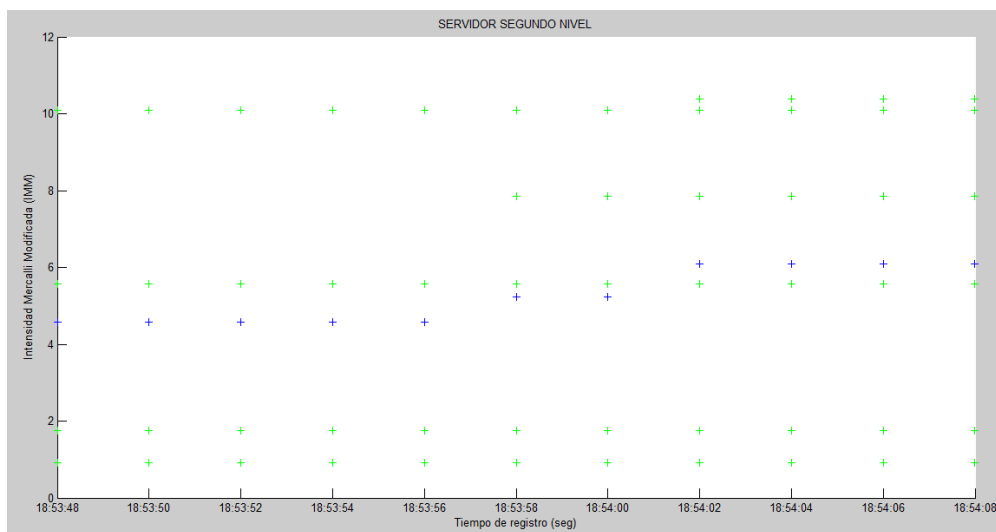
Aunque no es posible realizar las pruebas en un ambiente con condiciones reales, se ha realizado pruebas de laboratorio que consisten en 10 dispositivos actuando como clientes y un PC como servidor detallando sus características en la Tabla. 10:

Capa	Tipo	Numero	Dispositivo	Sistema Operativo
Capa 1	Sensor	2	Samsung Galaxy Ace II	Android (2.3.6)
Capa 1	Sensor	2	Samsung Galaxy S3	Android (4.1.2)
Capa 1	Sensor	3	HTC Desire	Android (2.3.5)
Capa 1	Sensor	2	LG Optimus P700 - L7	Android (4.0.3)
Capa 1	Sensor	1	LG Optimus P470 – L5	Andorid (2.2.0)
Capa 2	Servidor	1	Dell Inspiron Core i5	Microsoft Windows 7

**Tabla 10:** Descripción de dispositivos de pruebas.

A pesar de que el número de participantes es limitado esperamos que en un escenario real este número crezca significativamente proporcionándole mayor confiabilidad.

### V.1. SERVIDOR MATLAB



**Figura 16:** Gráfica de visualización generada por Servidor Matlab.

Este servidor es una utilidad extra utilizada para visualizar la variación constante de las intensidades en cada ventana de tiempo de las muestras que llegan a un servidor. En la Fig. 16 se muestra en verde cada valor de intensidad recopilado, en azul la media, y el rojo el pico detectado.

**NOTA:** Este servidor no mantiene una comunicación con los clientes.

## ***V.2 SERVIDOR JAVA***

El principal objetivo del servidor es trabajar lo suficientemente rápido para proporcionar al usuario tiempo útil para una mejor decisión en caso de catástrofe, por tanto se intenta obtener resultados con respecto a tiempos:

- Con ayuda de un proceso que crea alrededor de 2000 muestras aleatorias probamos cuanto demora el servidor en procesarlas todas tomando un valor medio de 1.2 segundos.
- El retardo que existe desde que el servidor ha detectado un pico y el usuario es notificado que ha existido un evento sísmico, es menor a un segundo en los dos escenarios probados: WIFI y 3G.

A través de estas pruebas podemos concluir que la arquitectura propuesta satisface los requerimientos de tiempo de un sistema en tiempo real ya que a pesar que éste sea en orden de los segundos (depende de la duración del sismo) o minutos puede llegar a ser vital salvando cientos o tal vez miles de vidas cumpliendo así el objetivo de esta investigación.

## ***V.3. APLICACIÓN CLIENTE (e-Quake)***

Se obtiene una aplicación totalmente funcional que manipula los sensores del móvil sin un consumo alto de batería y presenta las pantallas que presenta mostradas en la Fig. 17 y detalladas a continuación:

- 1. Pantalla de Inicio:** Creada con la ayuda de Google Maps que muestra automáticamente la ubicación actual del cliente usando su sensor GPS.
- 2. Mapa:** La pantalla de inicio permite realizar zoom y desplazarse por el mapa.
- 3. MQTT Notificaciones:** A través de esta pantalla se permite al usuario escoger de una lista (actualmente de Ecuador) las ciudades de las cuales desea recibir notificaciones tempranas.
- 4. Archivo de Configuración:** Archivo transparente para el usuario, que guarda las configuraciones de notificaciones. Cambia cada vez que el cliente modifica las ciudades en la pantalla anterior (MQTT).
- 5. Información “e-Quake”:** Pantalla de información acerca de la aplicación.
- 6. Prevención:** Pantalla que detalla las mejores acciones a realizar en un evento.
- 7. Notificaciones:** Las últimas 2 pantallas representa la manera visual en que llega la alarma.

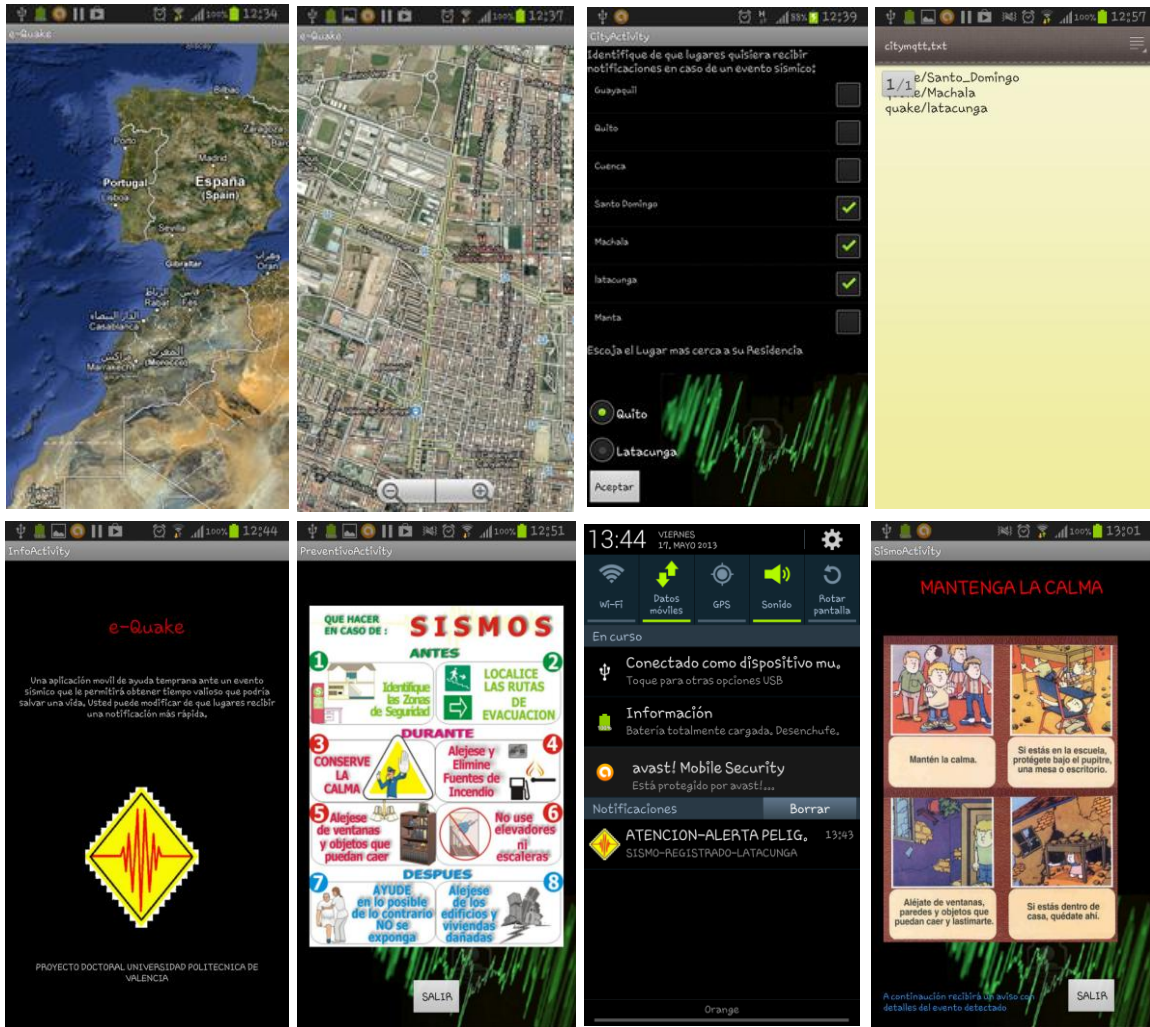


Figura 17: Pantallas “e-Quake”.

#### V.4. PRUEBAS REALIZADAS

Cada una de las pruebas fueron realizadas en dos escenarios WIFI y 3G pudiendo compararlas.

##### V.4.1. NOTIFICACIONES MQTT

Se realizan pruebas de cuanto consumo de batería representa mandar el envío continuo de “beacons” desde el Cliente al Servidor obteniendo los resultados en la Tabla. 11. Podemos observar que cuanto más frecuente es el envío de “beacons” utiliza mayor batería.

	5 minutos		15 minutos	
	3G	WIFI	3G	WIFI
Batería/hora	0.1553%	0.0245%	0.0021%	0.00037%

Tabla 11: Porcentaje de batería/hora por tipo de conexión.

#### V.4.2. CONSUMO DE BATERÍA

El consumo de batería es uno de los puntos que más se ha cuidado en este trabajo ya que puede ser la razón de que el cliente desinstale la aplicación perdiendo así sensores en la red. Hasta el momento la aplicación consume un 15 % de batería, cerca de 42 horas sin carga, el cual si comparamos con otras aplicaciones no representa demasiado, como por ejemplo WhatsApp [25] el cual tiene un consumo del 13% de batería según la Fig. 18 tomada de un Samsung S3, sin considerar el hecho de que esta aplicación puede salvar vidas. Sin embargo este consumo se puede reducir considerablemente al reducir la tasa muestreo, aumentando el tiempo entre reconexiones y número de “beacons”.

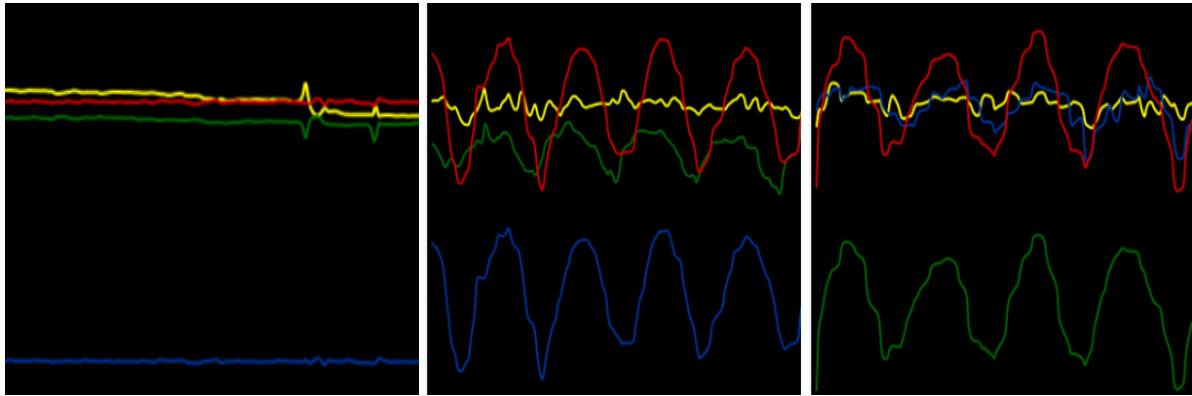


**Figura 18:** Consumo de batería de “e-Quake” en un smartphone Samsung Galaxy S3.

#### V.4.3. DETECCIÓN DE PICOS EN EL MOVIL

El proceso de detección de picos es parte primordial de esta investigación, por tal razón realizamos diferentes pruebas para corroborar el proceso teórico planteado:

- La aplicación es muy precisa cuando los móviles no tienen ruido alguno, por ejemplo mientras se encuentra cargándose en las noches donde usualmente se encuentra en reposo, se puede detectar variaciones muy pequeñas, incluso la de la vibración de un mensaje de texto.
- La aplicación no siempre detecta un pico al tener aceleraciones altas, todo depende del algoritmo usado STA/LTA. De esta manera si el usuario se mantiene caminando o corriendo (realizando un proceso periódico) y su aceleración es alta, no enviará ningún pico ya que la relación entre la ventana corta y ventana larga del algoritmo se mantiene constante como muestra la Fig. 19. Se determina que el proceso funciona con un 85.1% de eficiencia.



**Figura 19:** Pantallas de picos en acelerómetro.

Así la cantidad de picos enviados se reduce considerablemente haciendo el uso de batería mucho mejor y reduce el trabajo del servidor al tener mucho menos datos de procesar.

## VI. CONCLUSIONES

- La aplicación “e-Quake” realizada en Android cumple con un buen equilibrio entre eficiencia y consumo de batería, incluso pudiendo ser mejorado variando valores de frecuencia de muestreo, tiempos en ventanas deslizantes y tiempos de petición de reconexiones.
- Con el trabajo realizado tanto teórico como práctico se cumple a cabalidad los objetivos planteados a un inicio logrando realizar una arquitectura funcional completa. Un cliente realizado en Android que interactúe rápidamente con el Servidor con retardos mínimos al usar una comunicación por datos con respecto a WIFI gracias al protocolo MQTT.
- Con los datos recogidos en las pruebas podemos asegurar que la arquitectura funciona en tiempo real cumpliendo así el objetivo principal: En caso de que el servidor detecte un evento sísmico, el usuario obtendría tiempo extra para la toma de una decisión adecuada. Un segundo de anticipación puede representar la vida de una persona.
- Se culmina con las primeras dos capas de una gran arquitectura de gestión de emergencias sísmicas capaz de ayudar al usuario antes, durante y después de un evento. Sin duda alguna, esta arquitectura podrá ser implantada en lugares de riesgo constante y por falta de recursos (u otra razón) no mantengan una adecuada gestión.

## VII. TRABAJO FUTURO

El proyecto realizado es la primera parte de una gran arquitectura pensada para solventar los problemas de una nación en constante riesgo sísmico.

### VII.1. ARQUITECTURA EN 3 CAPAS

A nuestro esquema actual aumentamos un nivel jerárquico más representando su procesamiento en un diagrama de bloques mostrado en la Fig.20. Se agrega un tercer nivel de mayor jerarquía con el cual conseguimos una mejor distribución ya que si por algún motivo (muy factible en un sismo) llega a caer un punto intermedio, el TERCER NIVEL seguirá teniendo información de los otros sitios del mismo nivel de otras zonas. Este también es llamado el Centro de Mando en el cual se toman las decisiones más importantes.

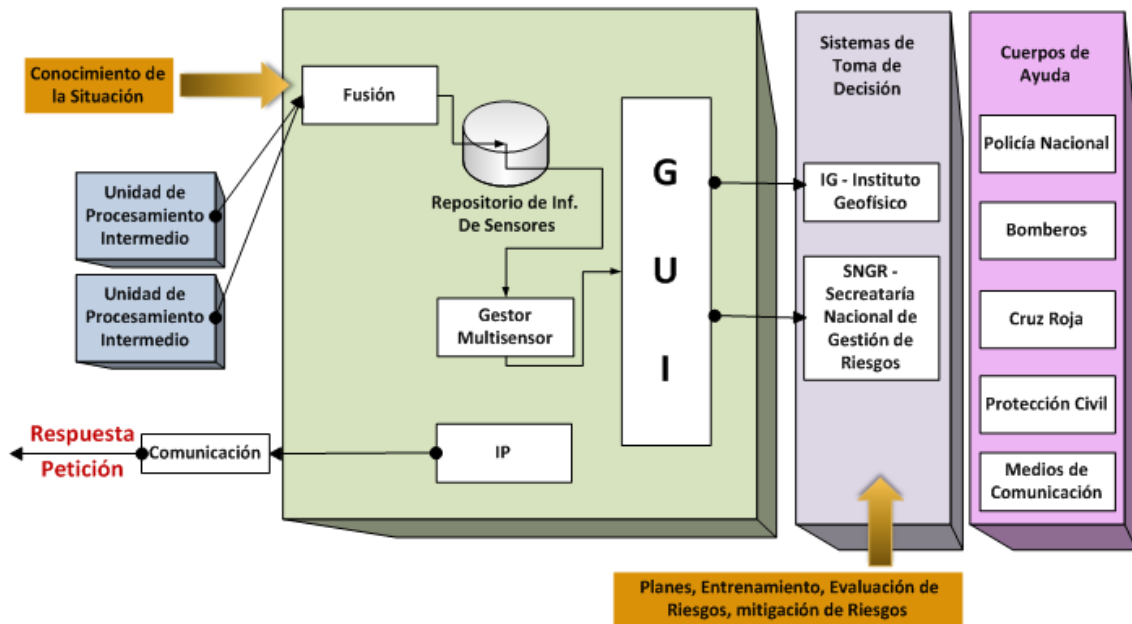


Figura 20: Tercera capa – Centro de Mando y Control.

### VII.2. GESTIÓN POST- EVENTO

Tan importante como tomar una decisión antes y durante una catástrofe es tomar una decisión adecuada luego del mismo. La gestión post-evento debería habilitar el envío de información multimedia, fotos y comentarios que serán útiles para poder definir los daños producidos. **NOTA:** Este requerimiento fue pedido por el IGEPN ya que actualmente este envío de información es manual y vía telefónica.

También se prevé un mecanismo de búsqueda de personas atrapadas a través de la misma aplicación móvil pudiendo ser utilizada no solamente en emergencias sísmicas, sino también en otras como inundaciones, deslizamientos, lluvias, incendios, deslaves, alertando así a los diferentes cuerpos de ayuda que una persona se encuentra en problemas y necesita socorro inmediato, agilizando y optimizando el tiempo de búsqueda.

En la Fig. 21 y Fig. 22, se muestra la manera de interactuar los diferentes niveles en esta nueva arquitectura:



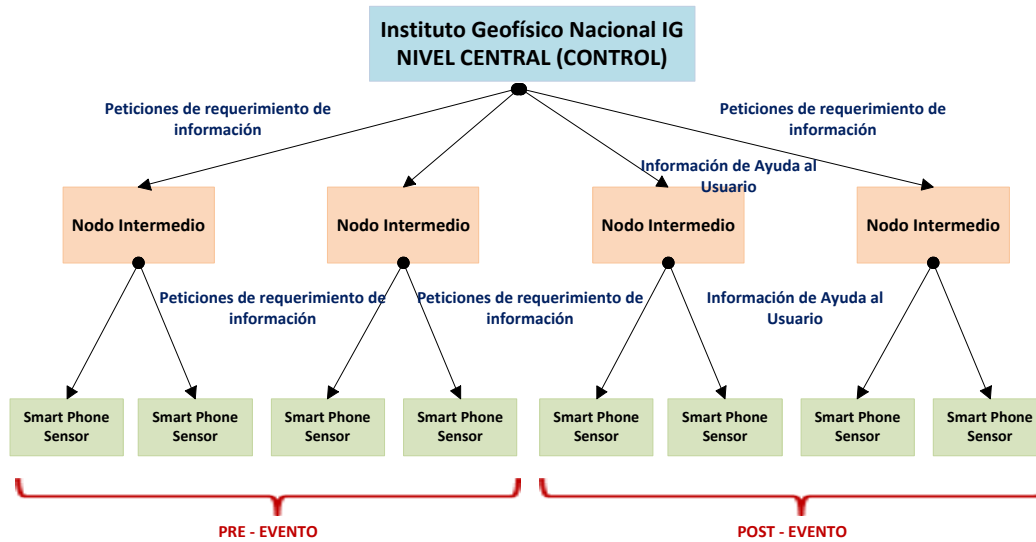


Figura 21: Comunicación “down” de una arquitectura 3 capas.

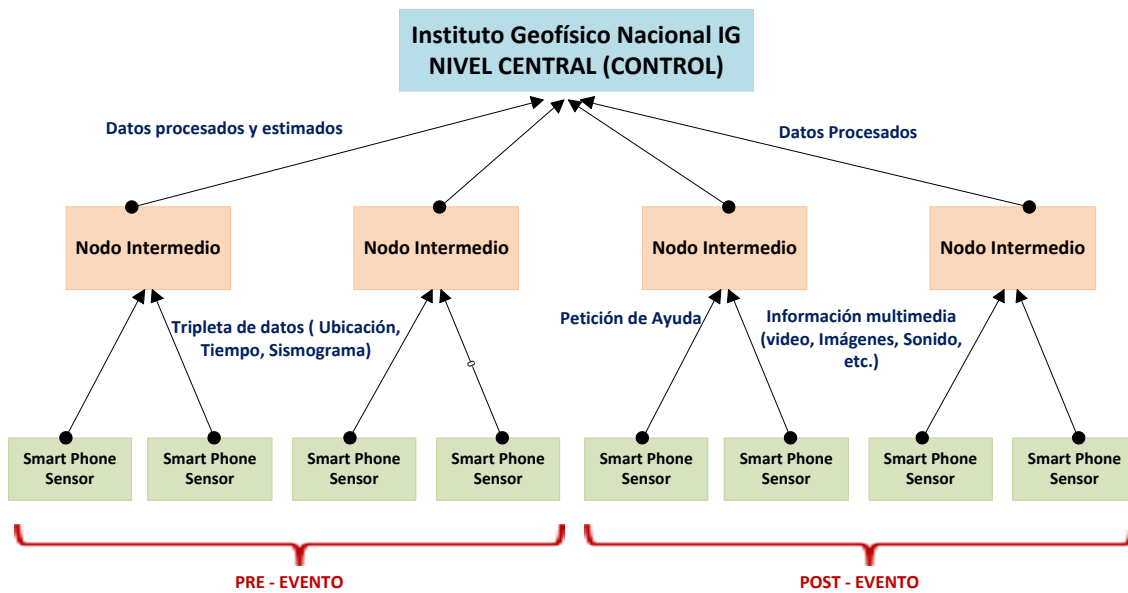


Figura 22: Comunicación “up” de una arquitectura 3 capas.

### VII.3. COMUNICACIÓN WSN (WIRELESS SENSOR NETWORK)

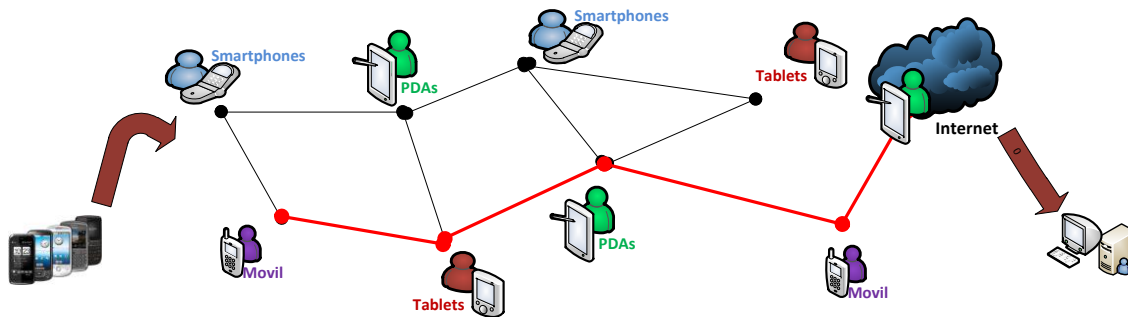


Figura 23: Red MESH de una red de sensores móviles.

Durante y una vez ocurrido el evento es muy posible que las redes telefónicas y conexiones colapsen dejando al usuario sin ningún tipo de comunicación. Por lo cual queremos brindar un nuevo plan de comunicación realizando una red tipo MESH, como muestra la Fig. 23, formada de smartphones que se encuentran repartidos en un territorio. Sería necesario formar una red de múltiples saltos que encuentre la ruta con el mínimo coste y rápida. Uno de los puntos problemáticos que no debemos olvidar es que esta red de sensores móviles será incluso más difícil de darse y acoplarse que siendo portátiles, ya que estos presentan potencias de señal aún más bajas y siendo un caso de emergencia, uno de nuestros puntos críticos será el ahorro al máximo de las baterías.

#### ***VII.4. VALIDACIÓN DE LA CALIDAD DEL SISTEMA A TRAVÉS DE DATOS REALES.***

En un sistema como el planteado se torna difícil encontrar el escenario de pruebas, sin embargo es necesaria la validación del sistema con datos reales para lo cual contamos con la ayuda del IGEPN, quien proporciona información obtenida de sismos anteriormente medidos por sus equipos para poder validar de una mejor manera la calidad del sistema. El proyecto cuenta con la ayuda del IGEPN ya que puede llegar a ser de gran ayuda a nivel nacional.

#### **AGRADECIMIENTOS Y DEDICATORIA.**

Sin duda alguna, mi primer agradecimiento es a Dios, quien es el pilar de mi vida.

Agradezco a mis compañeros del laboratorio de SISTEMAS EN TIEMPO REAL Y DISTRIBUIDOS de la UPV, quienes día tras día me han acompañado en el trajinar de mi investigación, y han creado un ambiente de trabajo idóneo para poder cumplir con este reto.

No puedo olvidar agradecer al Estado Ecuatoriano y a su Secretaría Nacional de Educación Superior, Ciencia y Tecnología (Senascyt), por haberme dado la oportunidad de escalar un peldaño más en la fabulosa e interminable carrera de la adquisición de conocimiento y formación del ser humano. Además al IGEPN por la ayuda y apertura en la información prestada para una eficaz arquitectura de alertas tempranas.

De manera especial, agradezco a mi director el Phd. Manuel Esteve y Phd Israel Pérez, quienes con su conocimiento y experiencia, me han guiado en el transcurso de este proyecto con sus consejos y sugerencias.

Por último, no podría pensar en otras personas a quien pueda dedicar este pequeño logro que a mis padres, Rocío e Iván, quienes han sido mi motivación e inspiración, y los cuales, con sus bendiciones han ayudado a que llegue a finalizar este proyecto.

## REFERENCIAS

- [1] Servicio Nacional de Sismología y Vulcanología. Sitio Oficial: <http://www.igepon.edu.ec/>
- [2] Grandes Terremotos – Servicio Nacional de Sismología y Vulcanología Ecuador. Sitio Oficial: <http://www.igepon.edu.ec/index.php/sismos/sismicidad/grandes-terremotos.html>
- [3] Kontogiannis, T. and Kossiavelou, Z; (1999) - Stress and team performance: principles and challenges for intelligent decision aids, *Safety Science*, Vol.33, 103 -106.
- [4] Allen, R. – (2011) Seconds Before the Big One: Progress in Earthquake Alarms. *Scientific American* 2011. Sitio Oficial: <http://www.scientificamerican.com/article.cfm?id=tsunami-seconds-before-the-big-one>
- [5] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017- (2013) Cisco Systems, Inc: [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-520862.pdf](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf)
- [6] Secretaría Interinstitucional de la Estrategia Internacional para la reducción de desastres. (2004) - Vivir con el Riesgo. Informe mundial sobre las iniciativas para la reducción de desastres.
- [7] Chandy, R., Rita A. and Qistein Skjellu; (2009) - QuakeCast: Distributed Seismic Early Warning. *Caltech Undergraduate Research Journal Winter 2011*.
- [8] Ervasti, M., Dashti, S., Reilly, J., Bray, J.D., Bayen, A., Glaser, S; (2011)- IShake: Mobile Phones as Seismic Sensors – User Study Findings; *Proceedings of the 10th international Conference on Mobile and Ubiquitous Multimedia*, NUM 11, 43-52
- [9] Faulkner, M., Olson, M., Chandy, R., Krause, J., Chandy, M., Krause, A; (2011) - The next big one: Detecting earthquakes and other rare events from community-based sensors. *ACM/IEEE International Conference on Information Processing in Sensor Networks IPSN'11*, 10.
- [10] Google Cloud Platform – página oficial. Sitio Oficial: <http://cloud.google.com/>
- [11] Steinberg, A. Bowman, C. and White, F; (1999) – Revisions to the JDL Data Fusion Model; *Sensor Fusion: Architectures, Algorithms, and Applications*, Vol. III, 430
- [12] Kruskal, W. and Wallis, W.A; (2004) – Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*. Vol. 3.
- [13] Hunkeler, U. Truonh H.L and Standford-Clark A; (2008) – MQTT-S A publish/subscribe protocol for Wireless Sensor Networks. *Communication Systems Software and Middleware and Workshops*. 978-1-4244-1796-4. 791-798.
- [14] Robusto, C. (1957) – The cosine-Haversine formula. *The American mathematical Monthly*, 38-40.
- [15] Bimal, G., Khushbu, S; - Analysis of The Emerging Android Market; Master Thesis – San José State University - Department Of General Engineering; (2010).
- [16] Bormann, P., Engdahl, B., & Kind, R. (2012). Seismic Wave Propagation and Earth models. In P. Bormann (Ed.), *New Manual of Seismological Observatory Practice 2 (NMSOP2)* (pp. 1-105).
- [17] Chen, G; (2011) - How to Convert 3-Axis Directions and Swap X-Y Axis of Accelerometer Data within Android Driver; *Freescale Semiconductor*
- [18] P. K. Varshney; (1997); *Multisensor Data Fusion; ELECTRONICS & COMMUNICATION ENGINEERING JOURNAL*.

- [19] Escala de Mercalli, Servicio Nacional de Sismología y Vulcanología:  
<http://www.igepn.edu.ec/index.php/escala-de-mercalli.html>
- [20] Sheng, X., Zhang, Y., Pham, D., Lambaré G. (2005) –Antileakage Fourier transform for seismic data regulation, Geophysics, Vol. 70, no. 4, 87-95
- [21] Sharma, B., Klumar, A., Murthy, VM. (2010)- Evaluation of Seismic Events Detection Algorithms, Journal Geological Society of India, Vol.75, 533-538.
- [22] Beauval, C., Yepes, H., Bakun, W., Egred, J., Alvarado, A., Singaicho, JC. (2010) – Historical Earthquakes in the Sierra of Ecuador (1587 – 1996). Geophysical Journal International, 181 (3), 1613-1633.
- [23] Panagiotis, P. and Vassilis, T. (2006) – On Transport Layer Mechanisms for Real-Time QoS; Journal of Mobile Multimedia, Vol. 1, no. 4; 343 – 363. UDPPP
- [24] Mosquitto MQTT Broker – Página Oficial: <http://mosquitto.org/>
- [25] WhatsApp – Página Oficial: <http://www.whatsapp.com/>

## ANEXO 1: DETERMINACIÓN DE FRECUENCIA DE MUESTREO ÓPTIMA

Para este procedimiento se trabajó con 4 smartPhones: 3 SAMSUNG GALAXY ACE 2 Y 1 LG OPTIMUS BLACK P470:

Mediante un pequeño código realizado en ANDROID obtenemos cuantas muestras puede el acelerómetro medir en un periodo de tiempo determinado, tal como muestran las siguientes capturas de pantallas:



Definiendo un periodo de 45 segundos, que es el tiempo que el sensor GPS actualiza la posición del usuario, obtenemos que en media se recopila 1402 muestras de aceleración o en otras palabras 35 muestras por segundo.

1357	1365	1350	1379	1337	1406
1380	1379	1558	1401	1504	1408
<b>PROMEDIO</b>					<b>1402</b>

La tasa o frecuencia de muestreo es el número de muestras por unidad de tiempo que se toman de una señal continua para producir una señal discreta, durante el proceso necesario para convertirla de analógica en digital. Según el teorema de muestreo de Nyquist-Shannon, para poder replicar con exactitud (es decir, siendo matemáticamente reversible en su totalidad) la forma de una onda es necesario que la frecuencia de muestreo sea superior al doble de la máxima frecuencia a muestrear.

Si las señales sísmicas trabajan normalmente de 1 a 10hz, basándonos en el teorema de Nyquist-Shannon, podremos comenzar trabajando con el mínimo muestreo crítico.

$$f = 2 W = 2 AB = 2 * 10 \text{ Hz} \quad \therefore f_{Nyquist} = 20 \text{ Hz} \quad \therefore t_{muestreo} = 0.05 \text{ seg}$$

**NOTA:** Si se quisiera equiparar las medidas de un sismógrafo de grandes características deberíamos muestrear a una tasa de 50 Hz a 100 Hz. Se utiliza un número de muestras múltiplo de 2 para optimizar tiempo y cálculos.

**ANEXO 2: SELECCIÓN DEL MECANISMO DE NOTIFICACIÓN ÓPTIMO:**

Los diferentes tipos de notificación son citados en la siguiente tabla:

<b>SMS</b>	Lleva un coste adicional para las compañías notificadores. Pero la recepción de la notificación es prácticamente en tiempo real e independiente. Pero ocupa muchos menos recursos.
<b>MECANISMOS DE PULL</b>	Si el tiempo real no es un requisito indispensable. Se realizan consultas periódicas al servidor en busca de novedades por tanto un mayor número de consultas al servidor y mayor será el consumo de batería del terminal.
<b>CONEXIONES PERSISTENTES</b>	Es la penalización del consumo energético. Se refiere a dejar un canal de comunicación siempre abierto, pero presenta la ventaja que es muy rápido. Este mecanismo se vuelve inviable ya que el SO Android termina cortando la comunicación luego de un tiempo determinado. También consume muchos mayores recursos en Servidor, tanto Hardware como Software al mantener tantas conexiones abiertas para su clientes.
<b>MECANISMOS DE PUSH</b>	El servidor inicia la comunicación para notificar el mismo momento que ocurre el evento y el cliente está siempre esperando ante un mensaje. Este tipo de comunicaciones es usado por Google en C2DM y muchos servicios de mensajería también como: WhatsApp, Google Talk, Facebook Messenger, Line, etc.

Luego del análisis se concluye que el mejor mecanismo de notificación es PUSH, el cual solventa los requerimientos del sistema mostrados en la tabla siguiente:

<b>Precio del servicio.</b>	Si las notificaciones serían mediante SMS se conlleva a un gasto considerable a pagar por la compañía de teléfono. Esto se debe evitar ya que se quiere una aplicación de uso gratuito (Es una de las razones por las que escogimos ANDROID).
<b>Periodo de recepción del mensaje</b>	Debe ser en tiempo real proporcionando tiempo útil al cliente en caso de un evento sísmico.
<b>Consumo de batería en el Cliente</b>	Se debe minimizar el impacto sobre la batería del mismo, caso contrario se puede llegar a que los clientes desinstalen la aplicación.
<b>Dependencia de servicio a terceros</b>	Esto limitaría el fin del proyecto o tendríamos que estar sujetos a cambios, exigencias y limitaciones de estos terceros.
<b>Seguridad de la información</b>	Garantizar algún tipo de autenticación y seguridad de la información ya que se envía información personal que al ser interceptada puede ser mal usada.
<b>Independencia del tipo de notificación</b>	El servidor debe ser capaz de alcanzar cualquier terminal con ANDROID sin importar las capacidades y diseño de cada terminal.

**ANEXO 3: PUBLICACIÓN ENVIADA.**

# Quake Detection System Using Smartphone-Based Wireless Sensor Network For Early Warning

**Abstract.** An early warning management for emergencies has gained notoriety among the population because of the crucial role it plays in reducing the losses of facilities, goods, properties, and more important, human lives. However, this requires an adequate preventive emergency system, which is not usually deployed in developing countries. In this paper a reliable, scalable and adaptable 3-layered architecture of sensor nodes is proposed in order to detect seismic events using smartphones as accelerographs, forming a low cost wireless sensor. The first contribution is a low-energy consumption application developed in Android OS that allows distinguishing normal user movements from a seismic event. The key contribution of this research is the definition of a detection process to send real-time notifications to users within fixed error and reliability ranges. This detection process is enhanced by introduction of two features that have not been used in previous works: a spatial-zoning-analysis using attenuation equations. Also, the architecture includes a post-event management that can be used by the control center and aid-centers to enhance the decisions.

## 1 Introduction

Seismic activity is one of the most frequent and destructive natural disasters, costing thousands millions euros in facilities, goods, properties and worst, lives. About 80% of terrestrial earthquakes are originated in the Pacific-circuit, an area of mountain and volcanic zones delimited by the Pacific Ocean: Earthquakes as Ecuador 1949, Colombia 1999, Peru 2007 and Chile 2010, all of them left 7674 deaths in seconds. And even so some countries, like Ecuador [1], do not have yet a proper early warning management policy in case of an earthquake. Deploying a real-time early warning architecture that would be efficient, fast, scalable and functional, that leverages the social resources and does not represent an extra cost to the country is the main idea of this work.

By the end of 2013 the number of mobile-connected devices will exceed the number of people in the earth, and by 2017 there will be nearly 1.4 mobile devices per capita, which means over 10 billion mobile-connected devices including machine-to-machine (M2M) modules that will exceed the worlds population at that time (7.6 billion) [2]. Taking advantage of this obvious technology boom, the smartphone (SP), which is an advanced multi-purpose computer and multi-sensor device, will be a key part of the proposed architecture to behave as an accelerograph, and, although SP capabilities are small compared to specific seismic hardware, it is also true that it would be economically impossible for a nation to get thousands of sensors covering a vast territory. Having a huge low-cost network of self-maintained devices allows the gathering of relevant seismic information to help Control center (CC) for making decisions in order to optimize time and human resources, which is one key contribution of this work.

It is necessary to merge the information from the sensor network using a multiple data integration process corresponding to the same actual-world object into a consistent, accurate, and useful representation. This paper aims to build an accelerograph network in a hierarchical architecture for early warning, following the JDL model [3] showed in Fig. 1, which perfectly covers the 3 layers and coupled equally to meet the objectives of secure data fusion, agile and fast. In the first layer which corresponds to Level-0, SPs are used as sensors and processing units that will send samples to the second layer or Intermediate Server (IS); IS corresponding to Level-1, must identify whether there was an event or not and immediately notify to the user who owns each sensor and at the same time, must communicate to CC this incident. Finally, in the third layer corresponding to level-2 and level-3, the CC will show this information and generated alarms to other ISs (to notify their SPs), commanders in charge of the operation (who makes the decisions) and emergency management agencies.

The care of a person's life is of invaluable cost and is in our hands to know how to prevent catastrophic events, so having a prevention mechanism is as important as making good decisions during and after a catastrophic event. This research envisions a post-event management, where each SP helps the CC by sending information such as comments, pictures and videos; helping in the process of achieving a global view of a disaster evolves.

This work is organized as follows: First previous and related projects in the area with their respective contribution are shown. In the next section, the proposed architecture is detailed and justified; the signal processing chain corresponding to the first layer, the server process and Control Center follows. Finally the conclusions of our research are presented.

## 2 Motivation and Previous Work

When a seismic event occurs, the first decision is crucial, it is the difference between being alive or not, and it depends on how much time someone has for thinking the best option [4], for example in Mexico 1995 (earthquake 8.0 magnitude), thousands of lives were saved because people could be warned 50 seconds ahead just using an early warning management system [5]. The relevant approach is detecting the occurrence of seismic activity in a short time window, typically in the order of minutes or seconds and although it can be considered a very tight frame, most studies reveal that is enough to save thousands of lives (considering that is enough time to get away from buildings or big structures) [5].

Earthquake detection researchers are interested in improving early monitoring systems using different procedures; some of the more relevant seismic-detection researches in this area are:

1. Quake Cast [6] is a project that detects events using a 3-axis-accelerometers, locates its epicenter, determines the intensity and sends alarms when considers that an earthquake has been detected. However, it uses a static device composed of a fixed accelerometer and a personal computer, providing good accuracy but no mobility or spatial distribution and only oriented to warning first-responders but no for social warnings. Quake Cast gains precision in the time axis by having immobile sensors and this allows to use P and S waves [7] as peak detection mechanism. Although it is very efficient, the scenario is fairly limited; Contrary Quake Cast our application is made completely



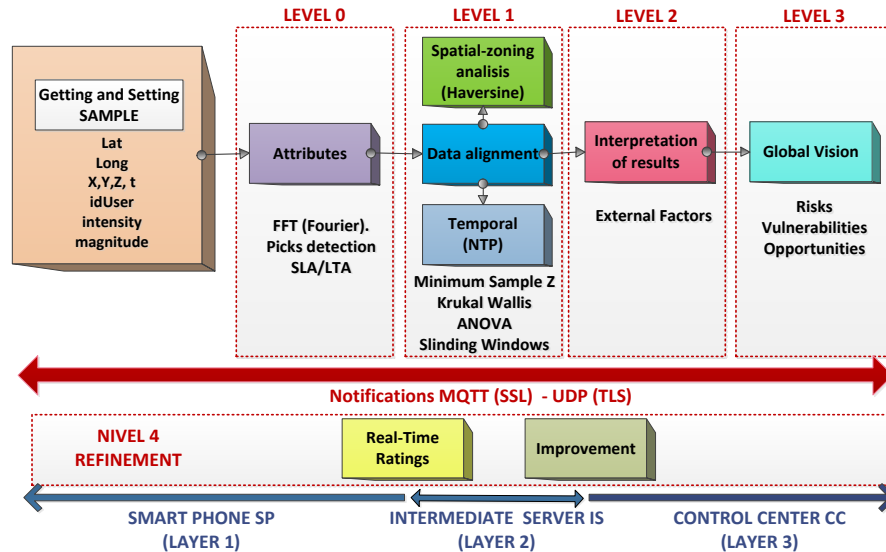


Fig. 1. Model JDL.

transparent to the user. The system will achieve greater reliability, is the number of user increase.

2. IShake [8] is a system developed by Berkeley University that uses smart-phones to measure acceleration and then determine the arrival of an earthquake but his application is limited to Apple iOS whereas current market trend is the use of Android [9]. The work done complements this project as well, and it takes advantage of the spatial-zoning-analysis in server which IShake does not consider.
3. Community Sensor Network CSN [10], created by the California Institute of Technology is one of the most complete on-going projects. It gathers data from SPs and implements a distributed decision-making process in the Google cloud, using virtual servers provided by the Google App Engine architecture [11]. The usage of third parties, forces us to comply certain conditions for their use, limit ourselves to its advantages, depend in their functional state and, even worst, the fact that in the case of a seismic event, the most possible is to stop giving the service. CSN depends completely of an architecture that, if a failure occurs, the system becomes disabled entirely.

The proposed architecture has been developing using an independent platform, modifiable to the requirements and changes of the zone and population, becoming a multi-network architecture that in the case of a seismic event, when links usually fall down, may ensure some kind of communication i.e. WIFI, WiMax, 3G or 4G. Furthermore, It has been developed using free software, Java-Android, that does not depend on a third party platform like Google which binds us to constant changes. The Android application must collect data, transparently to the user, always caring about the batterys power and trying to differentiate between peaks and habitual movements.

The main difference with previous work is the second layer process, which manipulates, integrates and processes the samples using Kruskal Wallis test ANOVA [12]; a sliding windows algorithm; MQTT [13] as messaging protocol of choice for real-time notification introducing us into the new term Internet of things, and finally other contribution in a spatial-zoning-analysis of samples by means of Attenuation Equations and Haversine distance [14], in order to find the optimum range (radius distance) that each IS should cover to accept samples improving the correlation process and hence, making a more accurate IS.

Another advantage is the incorporation of the third layer CC which allows a post-event management to help aid-centers and users to make a better post event decision as: (i) know the closest aid-centers to the user. (ii) Choose the best route. (iii) Give knowledge to aid-centers of areas that require immediate assistance, (iv) quantify the damage, and others.

### 3 Architecture

The hierarchical architecture Fig. 2 is composed of three layers: the first is basically a network of sensor nodes SPs which gather data (x, y, z accelerations; latitude; longitude; date; hour; magnitude and intensity); the second, composed of computers (servers) with enough resources; receiving and sending data to all SPs which are in the coverage range and also to the third and last layer, the CC which obtains a certain, current and global vision of the situation, being responsible to organize and order the whole architecture and handle the aid-centers management.

Each SP needs an application which computes the samples in order to determine whether a seismic event has occurred, and in such case sends data (time; latitude; longitude; intensity and event evaluation) to IS; this application has been developed for Android, designed to be non-intrusive with user normal activities. Low power consumption has been taken into account in order to the user not to experience high impact and dismiss application usage.

ISs are always listening a port waiting for SPs data which will be stored at IS by means of MySQL DBMS usage; Next, every so often, ISs take all samples and compute as fast as possible in order to determine whether the data from all SPs represents a real hazard and if it should notify to users or not; Moreover, each IS is connected to a CC to inform that the other two layers have detected an earthquake. CC, which has the best detection equipment nationwide checks immediately whether the occurrence of an earthquake is likely, if it is, sends a notification with accurate and updated data to all ISs which in turn notify all his users, the place, intensity, damages and help information like close aid-centers.

Between SPs and ISs, several communication technologies (Wi-Fi, GPRS, WIMAX, 3G or 4G) can be used due to current smartphone devices multi interface capabilities. From the application point of view, it can seamlessly handover from one communication network to another if connectivity loss is detected. Communications need to be fast that's the reason to include MQTT protocol (Message Queue Telemetry Transport) which is an extremely lightweight publish/subscribe messaging transport protocol. It is also ideal for mobile applications because of its small size; low power usage; minimized data packets and efficient distribution of information to one or many receivers. Between IS and CC nodes, the communications scenario becomes less tactical but even in such

case data links can be destroyed, so satellite may be used. Communications are always real-time oriented so, for instance, UDP [15] is used as transport in order to avoid retransmissions and connection oriented processes.

The architecture has the following features: (i) Simple Maintenance: as it is the responsibility of the user, he/she only has to spend a few seconds installing or upgrading it; (ii) Adaptive and Scalable: There will be more processing in the ISs but SPs will not perceive the change between 100, 1000 or even 10000 smartphones in the system. This architecture, first, allows entering new servers in order to help requests, notifications and process to another server which needs it; and second, the communication uses a lightweight protocol which allows a minimum CPU usage, freeing IS resources. (iii) Agile: achieved by using real-time protocols and technologies as data validity expiration is very short; (iv) Secure: first layer smartphones send personal information but this could be intercepted during the communication and could be misused. TLS is proposed as security mechanism to support data exchange over UDP and MQTT is used for secure notification over SSL. (v) Standard: by means of using standard protocols and technologies at every layer and COTS devices as consumer electronics networking hardware and smartphones. (vi) Time synchronization: NTP (Network Time Protocol) is used for clock synchronization between layers where CC is defined as Stratum 0 server NTP.

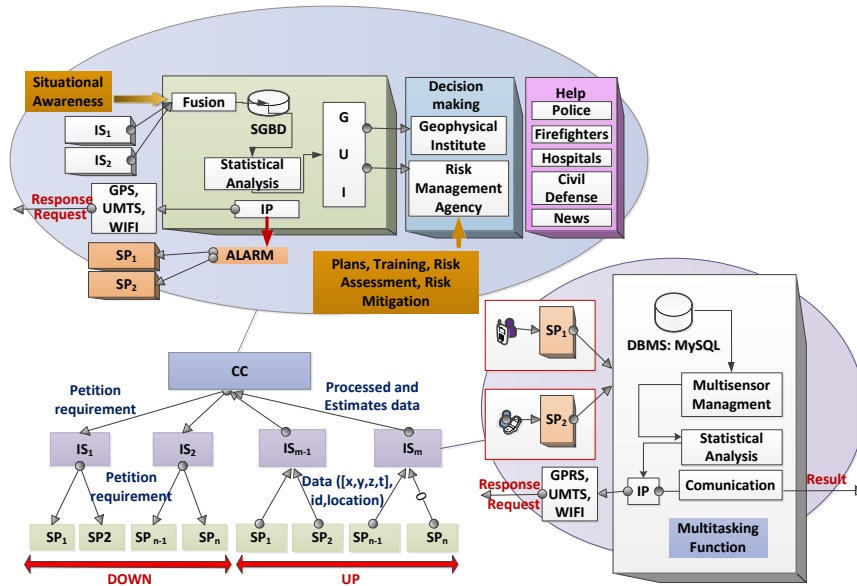


Fig. 2. Architecture

#### 4 Client Application and Acceleration Processing

SPs application must be simple; non-interfering; non-battery consuming; as well as good helper during and after a seismic catastrophe in order to assist crisis managers to make better decisions. Regarding the implementation issues, the Android operating system was chosen due to the following advantages: (i) it has a strong position in the market guaranteeing a huge amount of potential

users, (ii) it is a free open source software, (iii) the application development is relatively easy and, (iv) It have a properly manageable life cycle. The application has been developed adapting to the capabilities and limitations of a Google API 2.2 platform ensuring that more than 95% of users using Android OS [16] would be able to install it smoothly. Furthermore, one of our goals is to have as much sensor nodes as possible, versions of the mobile application for Windows Mobile and iOS are currently under development. System signal processing loop is as follows in Fig. 3:

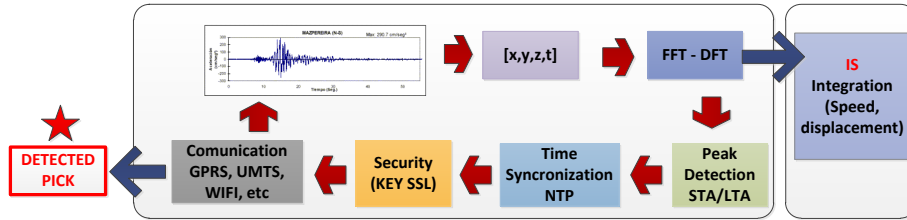


Fig. 3. Process acceleration

In order to work with seismic signals ranging from 1 to 10 Hz, sampling interval must be 0.05 seconds; therefore it has to process data 20 times per second. Though it seems large sampling rate, it is not, considering that the accelerometer can catch around 35 to 40 samples per second.

Peaks indicate that there was an event. They must be handled with care because a false positive can cause chaos in the area and also lose credibility in the system. Time domain is not enough to properly treat the signal, so a background process collects acceleration samples and, by means of Discrete Fourier Transform (DFT), each coefficient is located in time and frequency [17]. Then STA/LTA [18] is used to represent the signal to noise Ratio (SNR) in order to define if the peak at a short window (STA) versus one long (LTA) exceeds a threshold. Finally a NTP Client calls server situated in CC to get an accurate clock time. And, in order to handle user privacy (id, location), a security mechanism with TLS server key is used.

On the other hand the application, by means of a subscription/publication model with MQTT M2M protocol, brings the opportunity to select which ISs require an early warning showed in Fig. 4. MQTT also allows us to provide security with SSL and authentication with prefixes, but not everyone can get into the system. Despite MQTT being a protocol over TCP, it is very fast because it works with 3 different qualities of service (QoS). In this work we have used QoS 0, the lowest, without ack as we want an application that delivers higher velocity. When the SP receive a MQTT message from IS, SP understands that it has happened a seismic event and it notifies immediately with an alarm to SP users.

## 5 Server Intermediate Application

For the second layer IS, the core software has been developed in JAVA and currently can run in 64-bit Windows and Unix platforms as its main feature is a number crunching process to achieve accuracy and prompt response. IS is always listening the SPs sent samples using different threads. It should keep its clock

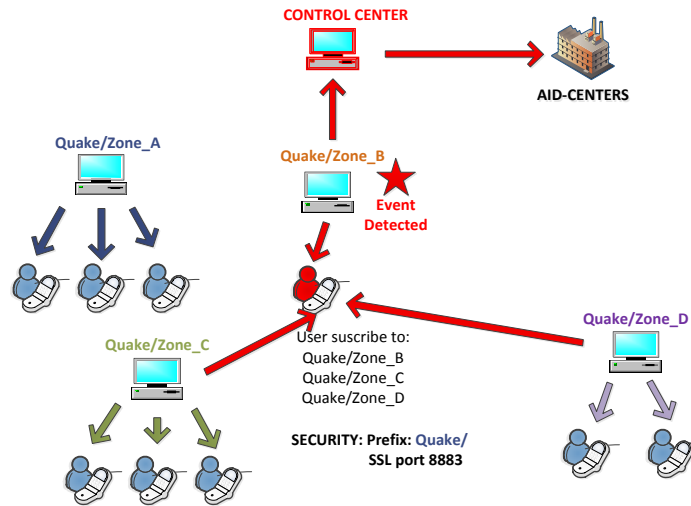


Fig. 4. MQTT Notification

synchronized with the clock CC by means of a NTP server before entering them into a MySQL database because of the real-time scenario. Then, to determine the existence of an earthquake, the server performs some tests and validations in order to reduce false alarms which decrease system reliability. Fig. 5 presents an overview about IS process:

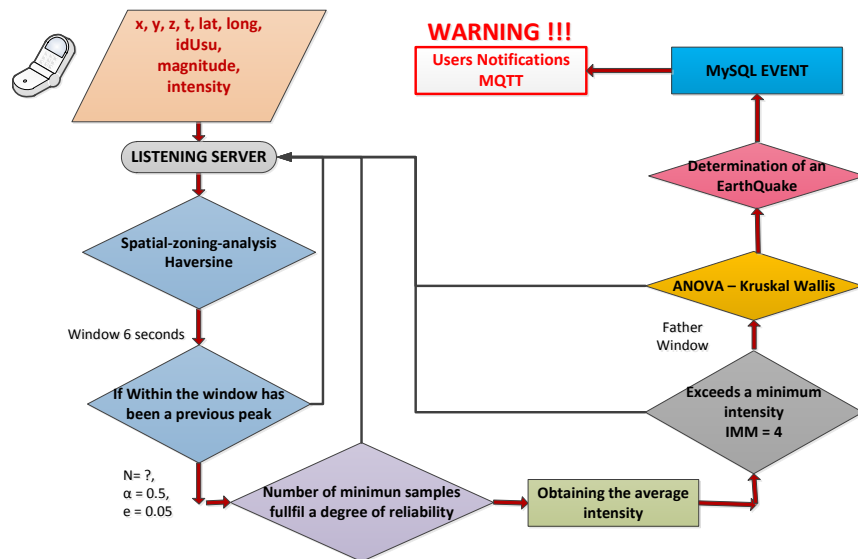


Fig. 5. Flow Diagram IS

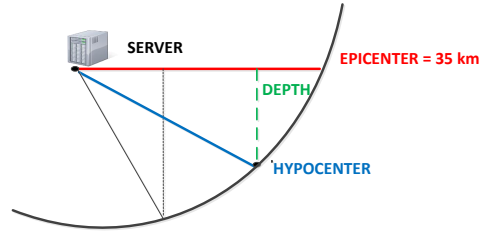
1. A spatial-zoning-analysis is a must, Eq. 1 define the attenuation equations. They show intensity ratio decreases as the distance increases, so if there was an event, the SPs in the area "A" would measure a greater acceleration than the SPs in farther zone "B". Therefore, an IS must analyze only samples that

correspond to a single location or, in other words, spatially-correlated samples. Using attenuation equations we force an IS to analyze only samples that are within its range. Earthquake intensity varies in relation to the Eq. 1, and in turn they depend on soil types, speeds and other; so each geographic zone has its own equations. We work with one of Ecuador attenuation equations [19] as follow:

$$\log(A_{gals}) = 1.2474 + 0.3735M_w - 0.4383 \ln(D + 10) \quad (1)$$

Where  $A_{gals}$  is the acceleration [ $\text{cm}/\text{s}^2$ ],  $M$  the magnitude and  $D$  the hypocenter distance. We have set a magnitude of 5.5 and the maximum SP acceleration using Eq. 2 to have a balance between the number of samples and the effectiveness. If  $D$  is too small it might be the case that IS samples run out and if it is too large, the samples are uncorrelated obtaining the maximum distance to hypocenter equal to 35 kilometers which corresponds to the epicenter distance [21] as show Fig: 6.

$$A_{max} = \sqrt{x^2 + y^2 + z^2} - g = \sqrt{19.6^2 + 19.6^2 + 19.6^2} - 9.8 = 24.14 \quad (2)$$



**Fig. 6.** Maximum Hypocenter Distance = Epicenter Distance

Samples that have a latitude and longitude that do not satisfy Haversine function [14] between its location and the servers location saved in MySQL are discarded and will be considered by another IS closer.

2. Minimum Sample test show how many SPs (active SPs) of all registered are enough to generalize the population with a percent of reliability and a margin of error as defined in Eq. 3:

$$\eta = \frac{N \cdot Z_{\sigma/2}^2 \cdot \sigma^2}{(N - 1) \cdot e^2 + N \cdot Z_{\sigma/2}^2 \cdot \sigma^2} \quad (3)$$

Where  $N$  = Total Population,  $\eta$ = portion of population - Number of users who has sent a pick,  $e$  = permitted error (1%, 0.05%),  $Z$  = confidence level ( $0.95 = 1.96$  *sigma/error*).

Both SP and IS do validations to determine which SPs are alive (active) and which not. First, SPs send beacons and are constantly monitoring the network for reconnections, and second the IS validates the last connection time and after a fixed time period (30 minutes) changes the phone state on database to inactive. These validations are important because we have considered a finite population which can change over time having SPs which do not have data location or network connectivity or have a dead battery, etc.

3. Kruskal Wallis or test H, is a non-parametric statistical test to probe whether a set of data comes from the same population (correlated) or not (uncorrelated), given that two groups are different when the variability between groups is greater than the variation within group.  
To use Kruskal Wallis was developed a periodical sliding windows algorithm showed in Fig. 7. The process repeats every 2 seconds collecting the intensity of all samples stored in database in a window of 6 seconds to process them. Each 20 seconds that correspond to a Father Window in Fig. 7, takes again the intensities to test the correlation using Kruskal Wallis and finally, trying to eliminate the risk of notifying the user the same event or replicas by validating that the time between the last and the present event has been at least 20 minutes.

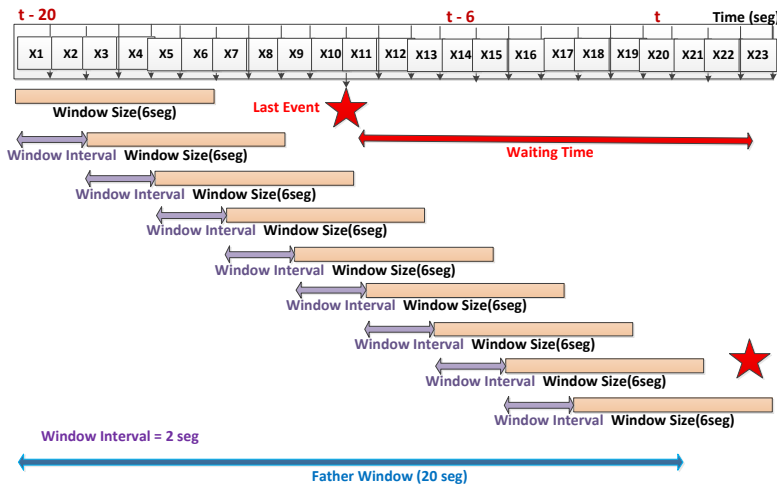


Fig. 7. Sliding Window Algorithm

Once an earthquake is detected, the user is informed immediately giving time to make a better decision. IS uses a MQTT server called Mosquitto [22], to publish to SPs who have subscribed to the topic whenever it detects an event.

## 6 Control Center

The Control Center is considered a central node, located at an emergency management agency headquarters acting as a command and control post, delivering to the emergency management commanders accurate information about what is going on to help them make proper decision due to the expensive and certain hardware (seismographs, accelerometers, inclinometer, etc.) that it has, e.g. in Ecuador, the Geographic Institute at the National Polytechnic School [23].

It allows us to extend from a pre-event to a post-event management, where first, each SP helps to CC by sending information such as comments, pictures and videos; helping in the process of achieving a global view of what is going on in a disaster; And second helps to users to make a better post-event decisions providing information about closest aid-centers, the safer and faster route, achieving an entire Emergency Management System to the whole area using low-cost devices.

## 7 Preliminary Results:

Our test equipment consists of 10 sensor-nodes and two PCs showed in table 1:

**Table 1.** Test-Equipment Used

Layer	Type	Num	Device	Operative System (Version)
Layer 1	Sensor	2	Samsung Galaxy Ace II	Android (2.3.6)
Layer 1	Sensor	2	Samsung Galaxy S3	Android (4.1.2)
Layer 1	Sensor	3	HTC Desire	Android (2.3.5)
Layer 1	Sensor	2	LG Optimus P700 - L7	Android (4.0.3)
Layer 1	Sensor	1	LG Optimus P470 L5	Andorid (2.2.0)
Layer 2	Sensor	1	Dell Inspiron Core i5	Microsoft Windows 7
Layer 3	Sensor	1	HP Pro Book Core i5	Microsoft Windows 7

First, SP application was tested simulating a seismic event by hitting a demonstration table such as [24], where the smartphones rests following conclusions where obtained (numbers obtained are an average of all SPs measures):

- Although SPs do not have same quality accelerometers as specific devices, the application can gather movements as small as a message vibration, verifying so, the accelerometers has enough quality to this research. (Obviously, these movements are not sent to IS).
- A relevant point to measure is how much percentage of battery it consumes because the user could stop using the application for this reason. At present, the application consumes a maximum of 15% (42 hours without charging) which if we compare with other applications is not too much, e.g. WhatsApp [25] which consumes a 13% in average without from the fact of our application could be the difference between be alive or not. This percentage could be decreased significantly modifying the sampling rate, reconnection time or numbers of beacons.

Also, the application monitors this consumption often and if it detects a low level of battery, it stops its work and just keeps running MQTT protocol for alerting the user. So, even if we lose a sensor, we can save a life just by leaving the functional smartphone.

- We needed 10 volunteers to carry each smartphone while performing some usual activities as walking, sitting, or even running, in order to test if our application can distinguish between a seismic-peak to these usual movements by means STA/LTA algorithm. Here, the application works fine with an 85.1% of accuracy, or in other words, 15 to 100 times sent a message to server when really it is not a seismic event.

The main goal of this work is providing extra time to end-user for making a better decision in case of a seismic event. Being a research that should work in real-time and unable to test in real conditions, the tests are performed in laboratory using the same android devices (numbers obtained are an average of all SPs measures):

- Time that IS takes to process around 2000 samples sent by a test-program developed to check each window size of value 6 seconds is: 1.2 seconds
- A delay of less than one second between IS detects a seismic event and user is notified.



We can conclude that the architecture satisfies the real-time system requirements because delivers time that, although may be only in the order of seconds (depend of the duration seismic event) or minutes, it is vital and can save hundreds or maybe thousands of peoples lives.

## 8 Conclusions and Future Work

A 3-layered architecture has been presented and developed satisfying the objectives of an early warning system in terms of time, scalability, security and adaptability in all layers, which increase its efficiency the more users are in the system. A smartphone application was made for Android OS taking care of battery consumption, network reconnections, together with a notification mechanism MQTT that fulfills with a real-time system and with a seismic detection mechanism achieves smartphones behave as accelerometers that differentiate between usual user movements and seismic events in order to notify to an intermediate server which manipulates sent samples covering a calculated distance. This Server developed in JAVA, the main contribution of this research, collects samples of a wireless sensor network formed by SPs in each geographic zone and processes them to determinate a seismic event and notify the user bringing time to make a better decision. In third layer, Control Center can make decisions to better global manage as society, servers and aid-centers. Finally, the preliminary results justify the used methods and raised architecture which could be implemented in a short time, with low monetary resources, leaving the efficiency to the number of clients in system, that we hope it will be greater and greater. So we have obtained a low cost, fast, scalable architecture that can be implemented anywhere specially in constant risk seismic places like Ecuador.

For future work we propose to extend the functionality in post-event management. Once that the event had been verified by CC, the SP application will present a unique and really simple screen that will allow the users ask for help to CC; thus CC which will know the specify location of each user, can manage in a better way the aid-centers and optimize the time of help or rescue. This justifies why the proposed and developed architecture implements security protocols anticipating this plus and ensuring the full user privacy.

## References

1. Secretaría Técnica de Gestión de Riesgos Ecuador; (2012) - Guía para la incorporación de la variable de riesgo en la Gestión integral de nuevos proyectos de infraestructura; 99-108. Available: <http://www.gestionderiesgos.gob.ec/> (last visited January 2013)
2. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017,- (2013) Cisco Systems, Inc, Available at: URL: [http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-520862.pdf](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf)
3. Steinberg, A. Bowman, C. and White, F; (1999) - Revisions to the JDL Data Fusion Model; Sensor Fusion: Architectures, Algorithms, and Applications, Vol. III, 430
4. Kontogiannis, T. and Kossiavelou, Z; (1999) - Stress and team performance: principles and challenges for intelligent decision aids, Safety Science, Vol.33, 103 -106.
5. Allen, R.- (2011) Seconds Before the Big One: Progress in Earthquake Alarms. Scientific American 2011. Permanent Address: <http://www.scientificamerican.com/article.cfm?id=tsunami-seconds-before-the-big-one>

6. Chandy, R., Rita A. and Qistein Skjellu; (2009) - QuakeCast: Distributed Seismic Early Warning. Caltech Undergraduate Research Journal Winter 2011.
7. Bormann, P., Engdahl, B., and Kind, R. (2012). Seismic Wave Propagation and Earth models. In P. Bormann (Ed.), *New Manual of Seismological Observatory Practice 2 (NMSOP2)* (pp. 1-105).
8. Ervasti, M., Dashti. S., Reilly, J., Bray, JD., Bayen, A., Glaser, S; (2011)- IShake: Mobile Phones as Seismic Sensors User Study Findings; Proceedings of the 10th international Conference on Mobile and Ubiquitous Multimedia, NUM 11, 43-52
9. Bimal, G., Khushbu, S; - Analysis of The Emerging Android Market; Master Thesis San Jos State University - Department Of General Engineering; (2010).
10. Faulkner, M., Olson, M., Chandy, R., Krause, J., Chandy, M., Krause, A; (2011) - The next big one: Detecting earthquakes and other rare events from community-based sensors. ACM/IEEE International Conference on Information Processing in Sensor Networks IPSN11, 10.
11. Google Cloud Plataform- official Site. Available at <http://cloud.google.com/>
12. Kruskal, W. and Wallis, W.A; (2004) Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*. Vol. 3.
13. Hunkeler, U. Truonh H.L and Standford-Clark A; (2008) MQTT-S A publish/subscribe protocol for Wireless Sensor Networks. *Communication Systems Software and Middleware and Workshops*. 978-1-4244-1796-4. 791-798.
14. Robusto, C. (1957) The cosine-Haversine formula. *The American mathematical Monthly*, 38-40.
15. Panagiotis, P. and Vassilis, T. (2006) On Transport Layer Mechanisms for Real-Time QoS; *Journal of Mobile Multimedia*, Vol. 1, no. 4; 343-363.
16. Developer Android. Available at: <http://developer.android.com/about/dashboards/index.html>
17. Sheng, X., Zhang, Y., Pham, D., Lambar G. (2005) Antileakage Fourier transform for seismic data regulation, *Geophysics*, Vol. 70, no. 4, 87-95
18. Sharma, B., Klumar, A., Murthy, VM. (2010)- Evaluation of Seismic Events Detection Algorithms, *Journal Geological Society of India*, Vol.75, 533-538.
19. Beauval, C., Yepes, H., Bakun, W., Egred, J., Alvarado, A., Singaicho, JC. (2010) Historical Earthquakes in the Sierra of Ecuador (1587-1996). *Geophysical Journal International*, 181 (3), 1613-1633.
20. Cochran, W.G. (1963) *Sampling Techniques*, 2nd Edition, new York: John Wiley and Sons, Inc.
21. Instituto Geofísico de la Escuela Politécnica Nacional del Ecuador (2011) Informe sísmico para el Ecuador. Available: URL: [http://www.igepn.edu.ec/images/collector/collection/informes-de-actividad-sismica/informe\\_ssmico\\_para\\_el\\_ao\\_2011\\_final.pdf](http://www.igepn.edu.ec/images/collector/collection/informes-de-actividad-sismica/informe_ssmico_para_el_ao_2011_final.pdf).
22. Mosquitto MQTT Broker official site. Available at: URL: <http://mosquitto.org>
23. Servicio Nacional de Sismología y Vulcanología official site: <http://www.igepn.edu.ec/>
24. Suzuki, M., Saruwatari, S., Kurata, N., and Morikawa, H. (2007) Demo Abstract: A High-Density Earthquake Monitoring System Using Wireless Sensor Networks, *Proceedings of the 5th international conference on Embedded networked sensor systems*, 373-374
25. WhatsApp official site. Available at: <http://www.whatsapp.com/>