



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**FACULTAD DE INGENIERÍA**

ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

**“Trabajo de grado previo a la obtención del Título de Ingeniero en Sistemas  
y Computación”**

**TRABAJO DE GRADUACIÓN**

**Título del proyecto**

ANÁLISIS DE LA RED AD-HOC NETWORKS EN DISPOSITIVOS  
MÓVILES Y SU APLICACIÓN EN LA MITIGACIÓN Y ALERTAS DE  
DESASTRES EN LA UNIVERSIDAD NACIONAL DE CHIMBORAZO

**Autor:**

Jonny Eduardo Vasco Barreno

**Directora:**

Ing. Lady Espinoza

**Riobamba – Ecuador**

**2015**

## PÁGINA DE REVISIÓN

Los miembros del Tribunal de Graduación del proyecto de investigación de título: **ANÁLISIS DE LA RED AD-HOC NETWORKS EN DISPOSITIVOS MÓVILES Y SU APLICACIÓN EN LA MITIGACIÓN Y ALERTAS DE DESASTRES EN LA UNIVERSIDAD NACIONAL DE CHIMBORAZO**, presentado por: Sr. **VASCO BARRENO JONNY EDUARDO**, y dirigida por: **Ing. LADY MARIELIZA ESPINOZA TINOCO**.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

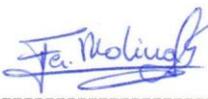
Ing. Danny Velasco  
**Presidenta del Tribunal**

  
-----  
**Firma**

Ing. Lady Espinoza  
**Miembro del Tribunal**

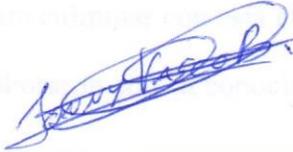
  
-----  
**Firma**

Ing. Fernando Molina  
**Miembro del Tribunal**

  
-----  
**Firma**

## AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: VASCO BARRENO JONNY EDUARDO e ING. LADY ESPINOZA; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



---

Vasco Barreno Jonny Eduardo

*Jonny Eduardo Vasco Barreno*



---

Ing. Lady Espinoza

## **AGRADECIMIENTO**

Mi agradecimiento en primer lugar a Dios, por haberme dado valor y fuerza para culminar esta etapa profesional.

A su vez a la Ing. Lady Espinosa, por su apoyo al realizar el trabajo investigativo.

A mis familiares que me han apoyado en todo el transcurso de mis estudios dándome apoyo y ánimos para culminar con esta etapa.

Gracias a todos quienes colaboraron con su conocimiento en este proyecto.

***Jonny Eduardo Vasco Barreno***

## **DEDICATORIA**

Yo, Jonny Eduardo Vasco Barreno dedico esta investigación a Dios, a mi Esposa, a mi Hijo, a mis Padres, Hermanos y Amigos, mismos que con su apoyo moral han influido para terminar esta etapa más de mi vida profesional.

*Jonny Eduardo Vasco Barreno*

## ÍNDICE GENERAL

ÍNDICE GENERAL.....	v
ÍNDICE DE TABLAS .....	x
ÍNDICE DE GRÁFICOS .....	xi
RESUMEN.....	xii
SUMMARY .....	xiii
INTRODUCCIÓN .....	1
CAPÍTULO I .....	2
1.1. TÍTULO DEL PROYECTO.....	2
1.2. PROBLEMATIZACIÓN.....	2
1.2.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA .....	2
1.2.2. ANÁLISIS CRÍTICO .....	3
1.2.3. PROGNOSIS .....	4
1.2.4. DELIMITACIÓN.....	4
1.2.5. FORMULACIÓN DEL PROBLEMA.....	4
1.2.6. OBJETIVOS.....	5
1.2.7. JUSTIFICACIÓN .....	6
CAPÍTULO II .....	7
2.1. RED.....	7
2.1.1. Topologías Básicas de Red .....	7
2.2 COMPONENTES DE REDES INALÁMBRICAS.....	8
2.2.1 Punto de Acceso.....	8
2.2.2 Clientes Inalámbricos.....	9
2.3 Ad-Hoc Network.....	10
2.3.1 Multihop Ad-Hoc Networks Ad-Hoc .....	11
2.3.2 Lista de Protocolos de Enrutamiento Ad-Hoc.....	11
2.3.3 Protocolos de Enrutamiento Ad-Hoc .....	12

2.4 Redes Ad-Hoc .....	13
2.4.1 Aplicaciones De Las Redes Ad-Hoc .....	14
2.4.2 Funciones de la red ad-hoc .....	16
2.4.3 Encaminamiento de redes ad-hoc .....	16
2.4.4 Funciones de movilidad .....	16
2.5 Instalación de una red inalámbrica ad-hoc.....	17
2.5.1 Requisitos del sistema: .....	17
2.5.2 Elección de la tarjeta:.....	17
2.5.3 Adaptadores PCMCIA:.....	17
2.5.3.1 Adaptadores miniPCI: .....	18
2.5.3.2 Adaptadores PCI:.....	18
2.5.3.3 Adaptadores USB: .....	19
2.6 Instalación del adaptador: .....	19
2.6.1 Configuración:.....	20
CAPÍTULO III .....	25
3. SISTEMA OPERATIVO ANDROID .....	25
3.1. Características .....	25
3.2. Arquitectura .....	29
3.3. Plataforma Android.....	30
3.3.1. Kernel de Linux.....	31
3.3.2. Bibliotecas .....	31
3.3.3. Entorno de ejecución .....	31
3.3.4. Marco de aplicación .....	32
3.3.5. Aplicaciones .....	32
3.3.6. Usos y dispositivos .....	33
3.3.7. Aplicación para Android.....	34

3.3.8.	Estructura Básica de Android .....	34
3.3.9.	Diseño de HLMP para Android .....	35
3.3.9.1.	HLMP-Java .....	36
3.3.10.	Android-Ad-Hoc.....	36
3.3.10.1.	Recepción de Mensajes SMS.....	37
3.3.10.2.	Envío de Mensajes SMS .....	40
3.4.	DESASTRES DE LA NATURALEZA.....	49
3.5.	DEFINICIÓN DE TÉRMINOS BÁSICOS .....	51
3.6.	HIPÓTESIS.....	52
3.7.	IDENTIFICACIÓN DE VARIABLES .....	52
3.7.1	VARIABLE INDEPENDIENTE .....	52
3.7.2	VARIABLE DEPENDIENTE .....	52
	CAPÍTULO IV .....	53
4.1	DETERMINACIÓN DE REQUERIMIENTOS .....	53
4.1.1	Definición de actores.....	53
4.2	Identificación de actores y tareas .....	54
4.2.1	Escenarios .....	54
4.3	Diagramas de casos de uso.....	55
4.4	Propósito .....	57
4.4.1	Ámbito de la aplicación .....	58
4.4.2	Beneficios del sistema.....	58
4.4.3	Definiciones, acrónimos y abreviaturas .....	58
4.4.4	Definiciones .....	58
4.4.5	Acrónimos .....	58
4.4.6	Visión general del proyecto .....	59
4.5	DESCRIPCIÓN GENERAL DEL PROYECTO .....	59
4.5.1	Perspectivas del producto.....	59

4.5.2 Funciones del sistema de Alerta de Desastres.....	59
4.5.3 Características de los usuarios .....	60
4.5.4 Funcionamiento.....	60
4.5.5 Suposiciones y dependencias .....	60
4.6 REQUISITOS DEL SERVIDOR SAD.....	60
4.6.1 Requisitos de interface .....	60
4.6.2 Interfaces externas.....	61
4.6.2.1 Interfaces de usuario .....	61
4.6.2.2 Hardware .....	61
4.6.3 Interfaz del servidor .....	61
4.6.4 Interfaz del cliente.....	62
4.7 PROCESAMIENTO DE DATOS .....	62
4.8 SALIDA DE DATOS .....	62
4.9 DESCRIPCIÓN DEL DISEÑO DE LA APLICACIÓN FINAL .....	63
4.9.1 Visualización:.....	63
4.10 DEFINICIÓN DE LAS HERRAMIENTAS A UTILIZAR .....	64
4.11 EVALUACIÓN DEL PROTOTIPO.....	65
4.12 IMPLEMENTACIÓN.....	65
4.13 METODOLOGÍA .....	74
4.13.1 TIPO DE ESTUDIO .....	74
4.13.2 POBLACIÓN Y MUESTRA .....	74
4.13.2.1 POBLACIÓN .....	74
4.13.2.2 MUESTRA .....	74
4.13.3 OPERACIONALIZACIÓN DE VARIABLES.....	75
4.13.4 PROCEDIMIENTOS.....	76
CAPÍTULO V.....	79
5.1 ANÁLISIS .....	82
5.2. COMPROBACIÓN DE LA HIPÓTESIS.....	82

5.2.1. Hipótesis.....	82
CAPÍTULO VI.....	86
6. CONCLUSIONES Y RECOMENDACIONES.....	86
6.1. CONCLUSIONES.....	86
6.2 RECOMENDACIONES.....	87
BIBLIOGRAFÍA .....	88
ANEXOS .....	90
ANEXO I. HOJA DE RECEPCION DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO.....	91
ANEXO II. ENCUESTA .....	92
ANEXO III. ANÁLISIS DE LAS PREGUNTAS 1, 2, 4, 5 .....	93
ANEXO IV. SIMULACRO .....	97
ANEXO V. MANUAL DE USUARIO .....	101
ANEXO VI. MANUAL TÉCNICO.....	110
ANEXO VII. TABLA DE CHI-CUADRADO.....	117

## ÍNDICE DE TABLAS

TABLA N° 1 CARACTERÍSTICAS SISTEMA OPERATIVO ANDROID .....	29
TABLA N° 2 ESCENARIO ADMINISTRAR ALERTA .....	54
TABLA N° 3 ESCENARIO CLIENTES.....	55
TABLA N° 4 OPERACIONALIZACIÓN DE LAS VARIABLES .....	75
TABLA N° 5 PROCEDIMIENTOS .....	76
TABLA N° 6 RESULTADOS DE LA PREGUNTA 3 .....	79
TABLA N° 7 RESULTADOS DE LA PREGUNTA 6.....	81
TABLA N° 8 FRECUENCIA OBSERVADA .....	83
TABLA N° 9 TABLA DE CONTINGENCIA DE LA HIPÓTESIS .....	84

## ÍNDICE DE GRÁFICOS

GRÁFICO N. 1 TOPOLOGÍA DE RED .....	7
GRÁFICO N. 2 ENLACE PUNTO A PUNTO EN MODO AD-HOC .....	10
GRÁFICO N. 3 VARIACIONES DE LA TECNOLOGÍA AD-HOC .....	11
GRÁFICO N. 4 REDES AD-HOC .....	14
GRÁFICO N. 5 ADAPTADORES PCMCIA.....	18
GRÁFICO N. 6 ADAPTADORES PCI.....	18
GRÁFICO N. 7 ADAPTADORES PCI.....	19
GRÁFICO N. 8 ADAPTADORES USB .....	19
GRÁFICO N. 9 ANDROID.....	29
GRÁFICO N. 10 PLATAFORMA ANDROID.....	30
GRÁFICO N. 11 INTERFAZ DE LA CENTRAL DE COMUNICACIONES .....	44
GRÁFICO N. 12 CUADRO JERÁRQUICO DE USUARIOS .....	53
GRÁFICO N. 13 CASO DE USO VISITANTE .....	56
GRÁFICO N. 14 CASO DE USO CLIENTE.....	56
GRÁFICO N. 15 CASO DE USO ADMINISTRADOR, CLIENTE .....	57
GRÁFICO N. 16 INTERFAZ REALIZADA CON MICROSOFT WORD DEL SERVIDOR.....	61
GRÁFICO N. 17 INTERFAZ REALIZADA CON MICROSOFT WORD DEL CLIENTE.....	62
GRÁFICO N. 18 BOTONES NECESARIOS .....	63
GRÁFICO N. 19 DISEÑO FINAL SERVIDOR.....	63
GRÁFICO N. 20 DESCARGA DE JAVA .....	66
GRÁFICO N. 21 PANTALLA DE BIENVENIDA DE INSTALACIÓN DE JAVA.....	66
GRÁFICO N. 22 PREGUNTA 3 RESULTADOS .....	80
GRÁFICO N. 23 PREGUNTA 6 RESULTADOS .....	81

## RESUMEN

Esta investigación se basa en el análisis de la red ad-hoc networks para dispositivos móviles para el uso en caso de desastres en la Universidad Nacional de Chimborazo, la red ad-hoc networks que son utilizadas en caso de emergencias por su fácil implementación.

En el desarrollo de la aplicación que permite la mitigación y alerta de desastres a través del envío de alertas e información a dispositivos móviles con sistema operativo Android, se utilizó la herramienta open source eclipse versión 3.5 usando el plugin de herramientas de desarrollo de Android, la aplicación funciona en redes LAN es decir una red interna por lo que permite su funcionamiento en una red ad-hoc networks que en si es una red LAN.

Para la comprobación se ha realizado la implementación de una red ad-hoc networks e instalado la aplicación a 15 estudiantes, a su vez la realización de un simulacro, y la obtención de datos a través de encuestas. Los resultados obtenidos determinan que se debería disponer de un sistema para alertar a la Universidad Nacional de Chimborazo puesto que chi-cuadrado  $\chi_c^2 = 9.48$  es mayor que cuadrado teórico  $\chi_t^2 = 5.99$  por lo cual, es necesario instalar la aplicación, para que las alertas que se reciban por esta aplicación contribuirán con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo.



UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA  
CENTRO DE IDIOMAS



19 de Mayo del 2015

Lic. Luis Oswaldo Guadalupe Bravo

### SUMMARY

This research is based on mobile network analysis of ad-hoc networks to mobile devices for use in a disaster at the Universidad Nacional de Chimborazo, the mobile ad-hoc networks that are used in emergencies for easy implementation.

In developing the application that allows mitigation and disaster warning by sending alerts and information to mobile devices with an Android operating system, the open source tool eclipse version 3.5 was used using the plugin development tools Android, the application works on LANs which is an internal network by allowing operation in a mobile ad-hoc networks that is a LAN.

For the verification a mobile ad-hoc network was implemented and installed to 15 students, and at the same time the simulation was performed, and data was collected through surveys. The results determined that the Universidad Nacional de Chimborazo should have an alert system, given that chi-square  $\chi_c^2 = 9.48$  is greater than theoretical square  $\chi_i^2 = 5.99$  so you need to install the application, so that the alerts received by this application will contribute to mitigating disaster at the Universidad Nacional de Chimborazo.



## **INTRODUCCIÓN**

Ad-Hoc o Topología de redes distribuidas en redes configurables logran operar sin la necesidad de una estructura arreglada donde el punto de conexión es wireless. (ad-hoc\_network)

La red ad-hoc (expresión latina que significa literalmente "para esto", "para fin específico"), cada nodo está deseando enviar datos hacia los otros nodos, por eso que la determinación a que nodo enviar los datos se establece dinámicamente basado en la conectividad de la red. (Buettrich, 2004)

La topología Ad-Hoc es conveniente para un rápido despliegue de una red wireless en un ambiente móvil o establecido. Su mínima configuración y fácil implementación hace a las redes Ad-Hoc útiles para situaciones de emergencia como desastres naturales, conflictos militares, y situaciones de emergencia médicas.

## **CAPÍTULO I**

### **MARCO REFERENCIAL**

#### **1.1. TÍTULO DEL PROYECTO**

Análisis de la red Ad-Hoc Networks en dispositivos móviles y su aplicación en la mitigación y alertas de desastres en la Universidad Nacional de Chimborazo.

#### **1.2. PROBLEMATIZACIÓN**

##### **1.2.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA**

Las diferentes tecnologías de comunicación en la actualidad son de mucha importancia, resaltando entre estas las redes móviles como claro, movistar y CNT dentro de nuestro país, por lo cual se han incrementado el número de dispositivos móviles en nuestra ciudad.

Las aplicaciones que se encuentran en los dispositivos son de varios aspectos como educativos, redes sociales, oficina móvil, entre otras en la Universidad Nacional de Chimborazo no existe una aplicación que permita mitigar los riesgos en caso de desastres, y que sea desarrollado con software libre.

Trabajar con aplicaciones para dispositivos móviles con software libre para mitigación de riesgos que permitan conectarse a través de la red Ad-Hoc puede reducir el número de pérdidas humanas en caso de desastres.

En la actualidad la comunicación entre dispositivos móviles a través de las diferentes redes de comunicación tienen diferentes motivos, a nivel nacional existen varias formas de comunicación entre estas muy utilizadas las redes móviles de las operadoras movistar, claro y alegro.

Estas operadoras permiten comunicación con un costo, lo cual en caso de un desastres las personas que cuentan con saldo se comunican tratando de saber que se debe hacer

o qué tipo de alerta está pasando en caso de que las operadoras no colapsen su servicio.

El principal problema que se enfrenta es la falta de alertas tempranas en el cantón Riobamba, las cuales permitan informar y prevenir a los habitantes de desastres naturales por diferentes medios de telecomunicación, se trata de integrar una aplicación a través de la red Ad-Hoc Network, la que detallará el tipo de alerta, esto lograría que no sean afectados severamente por fenómenos naturales intensos, causando un descenso brusco en el nivel de vida de sus habitantes.

La unión de varias tecnologías de comunicación que permitan la mitigación de desastres en el cantón Riobamba permite colocarse en uno de los primeros cantones en el país con sistemas de comunicación que ayuden a prevenir las pérdidas humanas en casos de desastres.

La red Ad-Hoc Networks permite comunicaciones entre nodos, cada uno es un dispositivo móvil y gracias a la unión de la misma con una aplicación para los mismos se pueden realizar aplicaciones que permitan mitigar los desastres.

### **1.2.2. ANÁLISIS CRÍTICO**

En el proceso de analizar las redes Ad-Hoc, se presentan ciertos aspectos que se detallan a continuación:

- Para realizar el análisis de las redes Ad-Hoc se debe conocer las topologías de red básicas.
- Se debe investigar las propiedades multisalto de los nodos de una red Ad-Hoc.
- Es necesario realizar un simulacro para obtener datos para utilizar la prueba de chi-cuadrado.
- Se debe realizar una aplicación que permita trabajar sobre las redes Ad-Hoc que permita el envío de alertas e información en caso de desastres en la Universidad Nacional de Chimborazo.

### **1.2.3. PROGNOSIS**

El propósito de esta investigación es determinar si las redes Ad-Hoc permiten realizar envío de alertas en caso de algún desastre este debe mejorar la calidad en la emisión de alertas por la Universidad Nacional de Chimborazo.

### **1.2.4. DELIMITACIÓN**

En el transcurso de este proyecto de investigación se realizará el análisis de las redes Ad-Hoc, se elegirá la mejor alternativa para desarrollar la aplicación para el envío de alertas las cuales puedan recibir los usuarios que se encuentren conectados a la misma red del Servidor y cuenten con sistema operativo Android en sus dispositivos móviles.

La aplicación instalada a los usuarios del sistema operativo Android de la Universidad Nacional de Chimborazo deben conocer la dirección IP que este asignada al Servidor y el puerto.

### **1.2.5. FORMULACIÓN DEL PROBLEMA**

¿En qué forma La red Ad-Hoc Networks en los dispositivos móviles con el sistema Android, permite contribuir con la mitigación de desastres en la Universidad Nacional de Chimborazo?

## **1.2.6. OBJETIVOS**

### **1.2.6.1. GENERAL**

Analizar la red Ad-Hoc Networks en dispositivos móviles y su aplicación en la mitigación y envío de alertas de desastres de la Universidad Nacional de Chimborazo.

### **1.2.6.2. ESPECÍFICOS**

- Estudiar las necesidades de una red Ad-Hoc Networks.
- Examinar la capacidad multisalto de una red ad-hoc para asegurar el reenvío de paquetes a través de los nodos de la red.
- Desarrollar una aplicación que capture y envíe información, envíe alertas en caso de desastres.

### **1.2.7. JUSTIFICACIÓN**

En la actualidad la tecnología se encuentra en cada hogar tanto desde un pequeño radio, televisión, celulares, computadoras, y muchos otros los cuales se los dan varios usos. En el proyecto se intenta lograr que esa tecnología nos permita mitigar los desastres.

Gracias a las nuevas tecnologías que nos permiten crear una red Ad-Hoc Networks se pretende lograr una comunicación de las autoridades de la Universidad Nacional de Chimborazo y los estudiantes.

La red Ad-Hoc Networks puesto que se pueden realizar con dispositivos móviles que tienen o cuentan con baterías lograrían una comunicación en caso de que el fluido eléctrico deje de funcionar o cualquier tipo de telefonía fallase la red Ad-Hoc Networks empezara a funcionar comunicando a las autoridades.

Con la comunicación se permite reducir los riesgos en desastres a través de alarmas o información correcta sobre que se debe hacer en esos casos.

## CAPÍTULO II

### FUNDAMENTACIÓN TEÓRICA

#### 2.1. RED

##### 2.1.1. Topologías Básicas de Red

La topología de una red significa la disposición de los enlaces que conectan los nodos de una red. (Escudero, 2007)

Las redes pueden tomar varias formas dependiendo de cómo están interconectados los nodos. Las dos formas de describir la topología de una red son física o lógica. La topología física se refiere a la configuración de antenas, cables, computadores y otros dispositivos de red, mientras la lógica hace referencia a un nivel más abstracto, considerando por ejemplo el flujo y método de la información transmitida entre nodos.

A continuación se da una descripción de 6 topologías de red básicas:

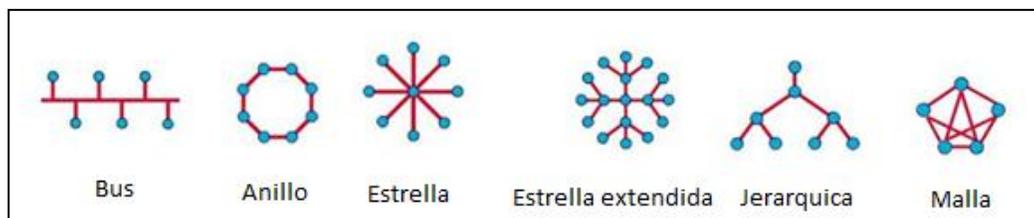


GRÁFICO N. 1 TOPOLOGÍA DE RED

Fuente: Investigador

#### **Topología**

#### **Descripción**

##### **1 Estrella**

En esta topología cada nodo se conecta directamente a un concentrador central, todos los datos pasan a través del concentrador antes de alcanzar su destino.

Esta es una topología común tanto en redes Ethernet como

inalámbricas.

- 2 Línea (o multiconcentrador)** Un conjunto de nodos conectados en una línea. En esta topología cada nodo se conecta a sus dos nodos vecinos con la excepción del nodo final que tiene sólo un nodo vecino.
- 3 Árbol** Esta es una combinación de las topologías de bus y estrella. Un conjunto de nodos configurados como estrella se conectan a una dorsal (backbone).
- 4 Anillo** Todos los nodos se conectan entre sí formando un lazo cerrado, de manera que cada nodo de esta topología se conecta directamente a otros dos dispositivos. Típicamente la infraestructura es una dorsal con fibra óptica
- 5 Malla completa** Existe enlace directo entre todos los pares de nodos de la red. Una malla completa con  $n$  nodos requiere de  $n(n-1)/2$  enlaces directos. Debido a esta característica, es una tecnología costosa pero muy confiable. Se usa principalmente para aplicaciones militares.
- 6 Malla parcial** Algunos nodos están organizados en una malla completa, mientras otros nodos se conectan solamente a uno o dos nodos de la red. Esta topología es menos costosa que la malla completa pero por supuesto, no es tan confiable ya que el número de enlaces redundantes se reduce.

## **2.2 COMPONENTES DE REDES INALÁMBRICAS**

### **2.2.1 Punto de Acceso**

Un punto de acceso es un “concentrador” inalámbrico. El transmisor/receptor conecta entre sí los nodos de la red inalámbrica y normalmente también sirve de puente entre ellos y la red cableada. Un conjunto de puntos de acceso (coordinados) se pueden conectar unos con otros para crear una gran red inalámbrica.

Desde el punto de vista de los clientes inalámbricos (como las computadoras portátiles o las estaciones móviles), un punto de acceso provee un cable virtual entre los clientes asociados. Este “cable inalámbrico” conecta tanto a los clientes entre sí, como los clientes con la red cableada.

Un punto de acceso deben distinguirse de un enrutador inalámbrico, que es muy común en el mercado actual. Un enrutador inalámbrico es una combinación entre un punto de acceso y un enrutador, y puede ejecutar tareas más complejas que las de un punto de acceso. Considere un enrutador inalámbrico como un puente inalámbrico (entre la red inalámbrica y la red Ethernet) y un enrutador (con características de enrutamiento IP).

Los clientes se conectan a un punto de acceso mediante su nombre. Este mecanismo de identificación se conoce como SSID-Service Set Identifier- (Identificador del Conjunto de Servicio), es el mismo para todos los miembros de una red inalámbrica específica. Todos los punto de acceso y clientes que pertenecen a un mismo ESS - Extended Service Set- (Conjunto de Servicio extendido) se deben configurar con el mismo ID (ESSID).

Conectarse a una red inalámbrica con SSID “x” es equivalente a conectar su computador a un punto de red con la etiqueta “x”. (León, 2011)

### **2.2.2 Clientes Inalámbricos**

Un cliente inalámbrico es cualquier estación inalámbrica que se conecta a una red de área local (LAN–Local Área Network) inalámbrica para compartir sus recursos. Una estación inalámbrica es cualquier computador con una tarjeta adaptadora de red inalámbrica instalada que transmite y recibe señales de Radio Frecuencia (RF).

Algunos de los clientes inalámbricos más comunes son las computadoras portátiles, PDAs, teléfonos inalámbricos de VoIP y equipos de vigilancia. (Escudero, 2007)

### 2.2.3 Modos de Operación de Redes Inalámbricas

El conjunto de estándares 802.11 definen dos modos fundamentales para redes inalámbricas:

1. Ad-hoc
2. Infraestructura



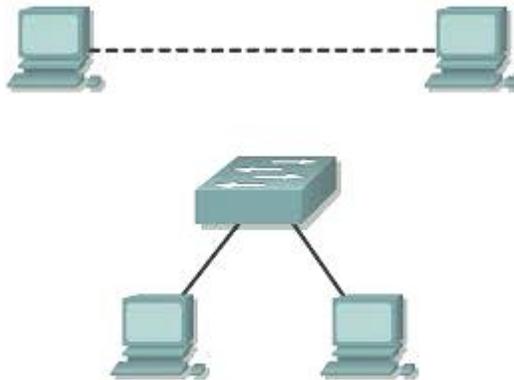
GRÁFICO N. 2 ENLACE PUNTO A PUNTO EN MODO AD-HOC  
Fuente: Redes avanzadas 2012/2013, 2013

### 2.3 Ad-Hoc Network

Ad-Hoc o Topología de redes distribuidas aplicadas a redes configurables que pueden operar sin la necesidad de una estructura arreglada donde el punto de conexión es wireless. (VARSHAVSKY, 2005)

La red es ad-hoc (expresión latina que significa literalmente "para esto", "para fin específico"), cada nodo está deseando enviar datos hacia los otros nodos, por lo que la determinación a que nodo enviar los datos se establece dinámicamente basado en la conectividad de la red. (Buettrich, 2004)

La topología Ad-Hoc es conveniente para un rápido despliegue de una red wireless en un ambiente móvil o establecido. Su mínima configuración y fácil implementación hace a las redes Ad-Hoc útiles para situaciones de emergencia como desastres naturales, conflictos militares, y situaciones de emergencia médicas. Topología Ad-rápido móvil mínima configuración fácil implementación Ad-útiles médicas.



**GRÁFICO N. 3 VARIACIONES DE LA TECNOLOGÍA AD-HOC**  
Fuente: Investigador

La figura de la en la parte superior es single hop ad hoc networks, como el nombre lo indica, cada usuario tiene su capacidad funcional de comunicación directa con cualquiera de los otros usuarios. “single-networks”.

### **2.3.1 Multihop Ad-Hoc Networks Ad-Hoc**

En algunas aplicaciones Ad-Hoc, donde los usuarios a lo mejor estén distribuidos sobre una amplia área, puede ser que el usuario solo pueda alcanzar una porción de la señal de cobertura debido a limitaciones de poder de la transmisión de la señal. (Galeas, 2004)

En estas situaciones los puntos de acceso de los nodos o usuarios deben colaborar en llevar mensajes a través de la red que se encuentra dispersa entre ellos.

Las redes diseñadas a funcionar de esta manera son llamadas “Multihop Ad-hoc networks”.

La configuración de la red multipad (Multihop ad-hoc networks) fue originalmente diseñada para tácticas militares, sobre dispersas áreas geográficas.

### **2.3.2 Lista de Protocolos de Enrutamiento Ad-Hoc**

El protocolo de enrutamiento Ad-hoc es un convenio o estándar que controla como los nodos llegan a entenderse por cual camino enrutar paquetes entre los nodos en una red móvil ad vil ad-hoc (MaNet). (Galeas, 2004)

En las redes ad-hoc, los nodos no tiene un conocimiento de la topología de la red que los rodea, ellos tienen que descubrirlo.

La idea básica es que un nuevo nodo (opcional) anuncie su presencia y escuche los anuncios emitidos por sus vecinos. El nodo aprende sobre sus nuevos nodos cercanos y los caminos para alcanzarlos, y tal vez anuncie que él también puede alcanzarlos (nodos cercanos).

En forma que el tiempo avanza, cada nodo conoce sobre todos los otros nodos y una o varios caminos para alcanzarlos.

### **2.3.3 Protocolos de Enrutamiento Ad-Hoc**

Los algoritmos de enrutamiento deben:

Mantener la tabla de enrutamiento razonablemente pequeña

Escoger la mejor ruta para determinado destino (esta puede ser la más rápida, más fiable, tasa de transferencia)

Mantener actualizada la tabla cuando el nodo muere, se mueve o alguien ingresa

Requiere de pequeños mensajes y el tiempo para converger.

1. Pro-active (Table-driven)
2. Reactive (On-demand)
3. Hybrid (Pro-Active/Reactive)
4. Hierarchical
5. Geographical
6. Power aware
7. Multicast
8. Geographical Multicast (Geocasting)
9. Other

Pro-active (Table-driven), en este algoritmo, mantiene actualizada la lista de los destinatarios y sus rutas a través de tablas de enrutamiento periódicamente de la red.

La mayor desventaja de este algoritmo es:

Desperdicio de ancho de banda en transmitir las rutas de enrutamiento.

Desperdicio en mantener rutas que jamás serán utilizadas en el futuro.

Simulaciones han demostrado que ciertos algoritmos Proactive como DSDV nunca podrán converger en redes extensas.

Reactive (On-demand), este protocolo encuentra la ruta en demanda a través de inundar a la red con paquetes de Route Request

La mayor desventaja de este algoritmo es:

1. Demora en encontrar la ruta
2. Excesiva inundación que puede hacer que la red se atasque.

Hybrid (Pro Active/Reactive), HSLS (Hazy Sighted Link State routing protocol), usa la optimización matemática para mezclar el estado de conexión y enrutamiento reactivo para optimizar los datos de actualización en tiempo y espacio.

Power aware, la energía requerida para transmitir es aproximadamente proporcional a, donde  $d$  es la distancia y  $\Rightarrow 2$ , es el factor de atenuación o exponente de pérdida, el cual depende del medio de transmisión. Cuando  $\alpha = 2$  (el caso óptimo), transmitir una señal mitad de la distancia se requiere un cuarto de energía y si existe un nodo en el medio deseando gastar otro cuarto de energía para la segunda parte, los datos serán transmitidos por la mitad de la energía que a través de una transmisión directa un factor que constituye el inverso cuadrado de las leyes físicas. Este método introduce retardo. Aware, energía  $d^\alpha$ ,  $\alpha$  atenuación transmisión.  $\alpha$  señal energía sería energía través transmisión – físicas. (Galeas, 2004)

#### **2.4 Redes Ad-Hoc**

Una red "Ad-hoc", consiste en un grupo de ordenadores que se comunican cada uno directamente con los otros a través de las señales de radio si usar un punto de acceso.

Las configuraciones "Ad-hoc", son comunicaciones de tipo punto a punto.

Solamente los ordenadores dentro de un rango de transmisión definido pueden comunicarse entre ellos. La tecnología es utilizada en varios campos como en el ejército, celulares y juegos de videos. En fin, en la tecnología "Ad-hoc", cada terminal de comunicación se comunica con sus compañeros para hacer una red "peer to peer".

También conocidas como MANET "ad-hoc networks". AD-HOC viene del latín y se refiere a algo improvisado, mientras que en las comunicaciones el propósito de ad-

hoc es proporcionar flexibilidad y autonomía aprovechando los principios de auto-organización.

La conexión es establecida por la duración de una sección. Los artefactos descubren otros artefactos cercanos o en rango para formar el “networks”. Los artefactos pueden buscar nodos que están fuera del área de alcance conectándose con otros artefactos que estén conectados a la red y estén a su alcance. Las conexiones son posibles por múltiples nodos.

Las redes ad-hoc presentan cambios de topología. Estos cambios son frecuentes debido a su movilidad. Estas características impiden la utilización de protocolos de encaminamiento. Y esto crea nuevos retos de investigación que permitan ofrecer soluciones para problemas tales como topología dinámica y recursos de ancho de banda entre otros.

Se pretende utilizar una topología jerárquica la cual se puede lograr utilizando un proceso conocido como “clustering”; este es un proceso en el cual los nodos de una red se organizan en grupos llamados clusters. Que es la forma en la que los nodos en MANET se organizan. Sin embargo la mayoría de los estudios de técnicos sobre “clustering” en ad hoc o MANET asume el conocimiento sobre la posición de los nodos. (López, 2004)

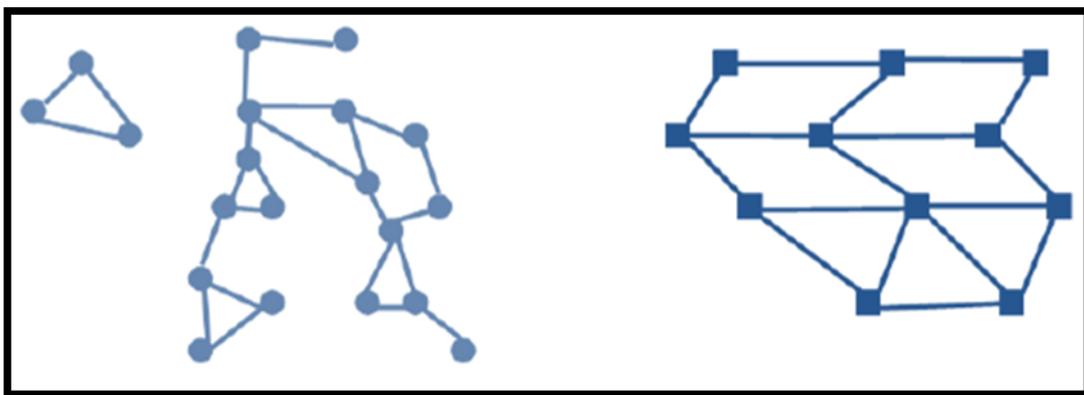


GRÁFICO N. 4 REDES AD-HOC  
Fuente: Investigador

#### 2.4.1 Aplicaciones De Las Redes Ad-Hoc

Las Aplicaciones Militares, donde una configuración de la red descentralizada es una ventaja operativa o incluso una necesidad.

En la parte comercial, los equipos para informática inalámbrica móvil no han sido disponibles a un precio atractivo para los grandes mercados.

A medida que va aumentando la capacidad de los ordenadores móviles de manera uniforme, también hay expectativa que crezca la necesidad de formación de redes ilimitadas.

Las redes comerciales ad hoc se podrían usar en situaciones allí donde no se dispone de ninguna infraestructura (fija o celular).

Ejemplos:

- Operaciones de rescates en zonas remotas
- Cuando la cobertura local debe ser desplegada rápidamente en un sitio de construcción.
- Acceso inalámbrico público en zonas urbanas.

A nivel local, las redes ad-hoc que enlazan ordenadores portátiles o de mano (palmtop) podrían ser usadas para difundir y compartir información entre los participantes de una conferencia.

Apropiada para aplicaciones de redes domésticas, donde los dispositivos pueden comunicarse directamente para intercambiar información, tal como audio/video, alarmas, y actualizaciones de configuración.

Aplicaciones de mayor alcance como las redes más o menos autónomas, de robots domésticos interconectados que limpian, lavan los platos, cortan el césped, realizan vigilancia de seguridad, y otras labores parecidas.

Hay personas que han propuesto incluso redes de saltos múltiples ad-hoc (denominadas redes censoras)

Ejemplo:

- Monitorización medioambiental- donde las redes se podrían utilizar para predecir la polución del agua o para alertar con tiempo de la aproximación de un tsunami.

Las redes ad-hoc de corto alcance pueden simplificar la intercomunicación entre diversos dispositivos móviles.

Ejemplo:

- Entre celular y una PDA

(Mercado, Redes inalámbricas Ad-hoc, 2006)

### **2.4.2 Funciones de la red ad-hoc**

#### **Seguridad**

La seguridad es un motivo de preocupación en una red ad-hoc.

- ¿Cómo puede estar seguro un usuario de que nadie está curioseando en el tráfico mediante un nodo de reenvío?
- ¿Es el usuario del otro extremo realmente la persona que dice ser?

Los servicios ad-hoc no implican muchos problemas nuevos.

Los requisitos relativos a la autenticación, la confidencialidad, y la integridad o no repudio son los mismos que para otras muchas redes de comunicaciones públicas.

En una red inalámbrica ad-hoc, la confianza es un problema fundamental.

No se puede confiar en el medio, la única elección que queda es usar la criptografía, lo que nos fuerza a confiar en las claves criptográficas usadas.

El método que se describe más abajo se puede usar para distribuir relaciones de confianza a toda una red ad-hoc.

El método se basa en un procedimiento de clave pública y como ejemplo se usa una pequeña red ad-hoc. (Mercado, Redes inalámbricas Ad-hoc, 2006)

### **2.4.3 Encaminamiento de redes ad-hoc**

Las redes móviles ad-hoc el encaminamiento de paquetes entre cualquier par de nodos llega a convertirse en una tarea comprometida, porque los nodos se pueden mover de manera aleatoria dentro de la red.

Un camino que se consideraba óptimo en un punto dado del tiempo podría no funcionar en absoluto unos pocos momentos después.

Para escenarios bajo techo- el cierre de una puerta podría ser causa de que se interrumpiese un camino.

### **2.4.4 Funciones de movilidad**

En una red ad-hoc, pueden moverse los nodos del centro de la red o más bien, la totalidad de la red está basada en la idea del dispositivo que sirven al mismo tiempo tanto de encaminadores como de anfitriones.

En una red ad-hoc, la movilidad es gestionada directamente por el algoritmo de encaminamiento.

## **2.5 Instalación de una red inalámbrica ad-hoc**

Este tipo de redes **AD-HOC**, podrían usarse para crear un grupo de trabajo con el objetivo de realizar intercambio de archivos o juegos en red, sin tener que realizar ningún tipo de instalación adicional, ya sea hardware o software, y de una forma sencilla y rápida.

### **2.5.1 Requisitos del sistema:**

Un ordenador, puede servir cualquier ordenador con sistema operativo actualmente disponibles, sea Windows o Linux.

Es posible usar en la red diferentes tipos de Sistema Operativos, teniendo en cuenta las limitaciones propias de la conexión entre equipos con diferentes Sistemas. Me refiero a la necesidad de usar algún software adicional si hay que compartir recursos entre Windows y Linux.

Además del Sistema Operativo un adaptador inalámbrico.

### **2.5.2 Elección de la tarjeta:**

Cuando el adaptador inalámbrico venga incorporado en el ordenador, como ocurre con los portátiles con tecnología Centrino, este punto se puede omitir y se continúa directamente al siguiente punto.

Los adaptadores inalámbricos que se pueden instalar pueden ser de varios tipos y la elección dependerá de las necesidades y de las características del equipo, pudiendo elegir entre adaptadores **PCMCIA, miniPCI, PCI o USB**. (Peña, 2012)

### **2.5.3 Adaptadores PCMCIA:**

Los adaptadores de red inalámbrica **PCMCIA**. Son casi de uso exclusivo de ordenadores portátiles, que normalmente son los que vienen equipados con este tipo de conector. En la figura se puede apreciar la forma de este dispositivo.



**GRÁFICO N. 5 ADAPTADORES PCMCIA**  
**Fuente:** Clanar Internacional, Internet y redes inalámbricas 2011.

A la izquierda de la tarjeta se aprecia los conectores de la misma. Al insertarla en el correspondiente slot PCMCIA sólo quedará a la vista la pieza negra que aparece a la derecha, que es la antena

### **2.5.3.1 Adaptadores miniPCI:**

Este tipo de adaptador son los usados habitualmente por los portátiles y los routers inalámbricos. Es un pequeño circuito parecido a una memoria de los computadores portátiles, como se puede ver en la fotografía.



**GRÁFICO N. 6 ADAPTADORES PCI**  
**Fuente:** Clanar Internacional, Internet y redes inalámbricas 2011.

Incluye la antena, aunque en la mayor parte de los dispositivos se puede incorporar una antena externa adicional.

### **2.5.3.2 Adaptadores PCI:**

Los adaptadores PCI, similares a las tarjetas de red llevan una pequeña antena para recepción-emisión de la señal. Su uso está indicado en ordenadores de escritorio. Se puede apreciar en la imagen su similitud con las tarjetas Ethernet que solemos instalar en los computadores.



**GRÁFICO N. 7 ADAPTADORES PCI**  
**Fuente:** Clanar Internacional, Internet y redes inalámbricas 2011.

### **2.5.3.3 Adaptadores USB:**

Los adaptadores USB son los más habituales por su precio y facilidad de instalación, pudiendo ser usado en cualquier computadora que disponga de puertos USB, sea una computadora de escritorio o una portátil. Incluso es posible adaptarlos a cualquier aparato electrónico que disponga de ese tipo de conexión. Se puede ver en la fotografía un ejemplo de este adaptador.



**GRÁFICO N. 8 ADAPTADORES USB**  
**Fuente:** Clanar Internacional, Internet y redes inalámbricas 2011.

## **2.6 Instalación del adaptador:**

El proceso de instalación del adaptador para redes inalámbricas es sencillo, sobre todo si se trata de un adaptador **PCMCIA** o **USB**, ya que no hay más que introducirlo en su correspondiente ubicación y seguir las instrucciones del manual de instalación. En el caso de los adaptadores **PCI**, el proceso es el mismo que el habitual en las tarjetas de este tipo, apagar el computador, desconectar los cables de alimentación, destapar la caja, localizar un slot PCI libre e instalar la tarjeta en él. Una vez encendido el computador de nuevo, detectará la nueva tarjeta e instalará el software correspondiente.

## **2.6.1 Configuración:**

### **Redes inalámbricas multisalto en GNU/Linux**

Las redes inalámbricas multisalto no necesitan de infraestructura fija para compartir datos entre distintos PC's, cada uno de los nodos que forman la red hacen las labores de un router de forma que se enrutan datos de unos a otros para formar una infraestructura de red dinámica. Si uno de los nodos tiene una conexión a Internet, los demás nodos pueden utilizar a éste como puente.

### **Requisitos previos**

Como paso previo hay que hacer lo siguiente:

En el kernel tienes que tener compilado el "Netfilters"

Habilitar el Packet Forwarding. En /etc/sysconfig/network añadir una línea:

```
FORWARD_IPV4=yes
```

Tener configuradas las tarjetas wireless.

Montar la red ad-hoc

En lo relativo a la configuración de las tarjetas, es muy útil el comando iwconfig que permite la configuración de las tarjetas de forma similar al clásico ifconfig.

En Debian, ese comando está en el paquete wireless-tools.

Si se tiene una tarjeta Wireless (en mi caso PC Card WL100) configurada en un portátil con Debian.

El iwconfig nos dice que se tiene que realizar:

```
eth1 IEEE 802.11-DS ESSID:"COMPAQ" Nickname:"Prism I"
```

```
Mode:Managed Frequency:2.457GHz Access Point: 00:02:A5:2D:06:35
```

```
Bit Rate:11Mb/s Tx-Power=15 dBm Sensitivity:1/242700000
```

Retry min limit:8 RTS thr:off Fragment thr:off

Encryption key:off

Power Management:off

Link Quality:49/1 Signal level:-53 dBm Noise level:-144 dBm

Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0

Tx excessive retries:0 Invalid misc:0 Missed beacon:0

Es el momento de observar dos cuestiones importantes, la primera de ellas es el Mode que está en "Managed" (utilizado para asociarse a puntos de acceso) y el segundo es el ESSID.

Hay que señalar que para que esta tarjeta pueda formar parte de una red ad-hoc todas las tarjetas tienen que tener el mismo ESSID y estar puestas en modo ad-hoc.

Para ello, y utilizando el iwconfig ejecutar:

```
homer:~# iwconfig eth1 mode ad-hoc
```

```
homer:~# iwconfig eth1 essid ARCO
```

```
homer:~# iwconfig
```

```
eth1 IEEE 802.11-DS ESSID:"ARCO" Nickname:"Prism I"
```

```
Mode:Ad-Hoc Frequency:2.457GHz Cell: 02:23:7A:CF:F1:E5
```

```
Bit Rate:5.5Mb/s Tx-Power=15 dBm Sensitivity:1/242700000
```

```
Retry min limit:8 RTS thr:off Fragment thr:off
```

```
Encryption key:off
```

```
Power Management:off
```

```
Link Quality:0 Signal level:0 Noise level:0
```

```
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
```

```
Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

El segundo ordenador a utilizar en nuestra red es un ordenador de escritorio con una tarjeta PCI-PCMCIA y con una tarjeta Wireless PC Card WL110. La salida del iwconfig es similar a la anterior, mientras que la salida del lsmod muestra el módulo orinoco\_plx que es el driver que se ha usado.

Ahora es el momento de configurar las direcciones IP que van a formar parte de la red ad-hoc. Idealmente, un nodo ad-hoc debería ser capaz de auto configurarse para formar parte de la red. En principio se va a asignar estas direcciones de forma estática.

Edita el archivo /etc/networks/interfaces y añade la configuración de la interfaz eth1, por ejemplo:

```
auto eth1

iface eth1 inet static

address 10.10.0.4

netmask 255.255.255.0

gateway 10.10.0.1
```

De forma que si se ejecuta:

```
radia:~# ifdown eth1

radia:~# ifup eth1

radia:~# ifconfig

eth1   Link encap:Ethernet HWaddr 00:02:A5:6F:28:FC

inet addr:10.10.0.4 Bcast:10.255.255.255 Mask:255.255.255.0
```

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
RX packets:1091 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:89 errors:20 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:100
```

```
RX bytes:81419 (79.5 KiB) TX bytes:20266 (19.7 KiB)
```

```
Interrupt:11 Base address:0xd800
```

Se ve la interfaz configurada con la dirección IP asociada.

Hacer lo mismo tanto en el portátil como el otro PC.(obviamente con direcciones IP distintas).

Bien, en este momento se debe realizar un simple ping de un ordenador a otro a las direcciones que se hayan puesto, siempre y cuando estemos en el ámbito de cobertura.

Enrutando paquetes

Para tener una verdadera red inalámbrica multisalto se debe hacer que cada nodo enrute paquetes, para ello debemos instalar un algoritmo de enrutado, en nuestro caso vamos a usar el algoritmo AODV cuya implementación (una de ellas) podéis bajar de `aodv_kernel`

Siguiendo los pasos del archivo README no debe haber ningún problema. Básicamente:

Te bajas el archivo y lo descomprimes.

```
homer:~# tar xvzf kernel-aodv_v2.2.2.tgz
```

Ejecutas

```
homer:~# make
```

Cambias los ficheros start.sh y start\_gateway.sh según los parámetros de la subred

ejecutar start.sh si eres cliente sin acceso a internet

ejecutar start\_gateway.sh si vas a funcionar como gateway dentro de la red ad-hoc

En este momento detectará todos los nodos dentro de su alcance y los añadirá a la tabla de enrutado. Para ver la tabla de enrutado:

```
homer:~# less /proc/aodv/route_table
```

## CAPÍTULO III

### ANDROID

#### 3. SISTEMA OPERATIVO ANDROID

Android es un sistema operativo basado en Linux, diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android, Inc., Google respaldó económicamente y más tarde compró en el año 2005, Android fue desvelado en el año 2007 junto con la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. (Valverde, 2014)

##### 3.1. Características

<b>Diseño de dispositivo</b>	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
<b>Almacenamiento</b>	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
<b>Conectividad</b>	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+ y WiMAX.
<b>Mensajería</b>	SMS y MMS son formas de mensajería, incluyendo mensajería

	<p>de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.</p>
<b>Navegador web</b>	<p>El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3.</p>
<b>Soporte de Java</b>	<p>Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma.</p> <p>El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados.</p> <p>El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.</p>
<b>Soporte multimedia</b>	<p>Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.</p>

<b>Soporte streaming</b>	<b>para</b> Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
<b>Soporte hardware adicional</b>	<b>para</b> Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
<b>Entorno desarrollo</b>	<b>de</b> Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando el plugin de Herramientas de Desarrollo de Android.
<b>Google Play</b>	Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
<b>Multi-táctil</b>	Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero.  La funcionalidad fue originalmente desactivada a nivel de

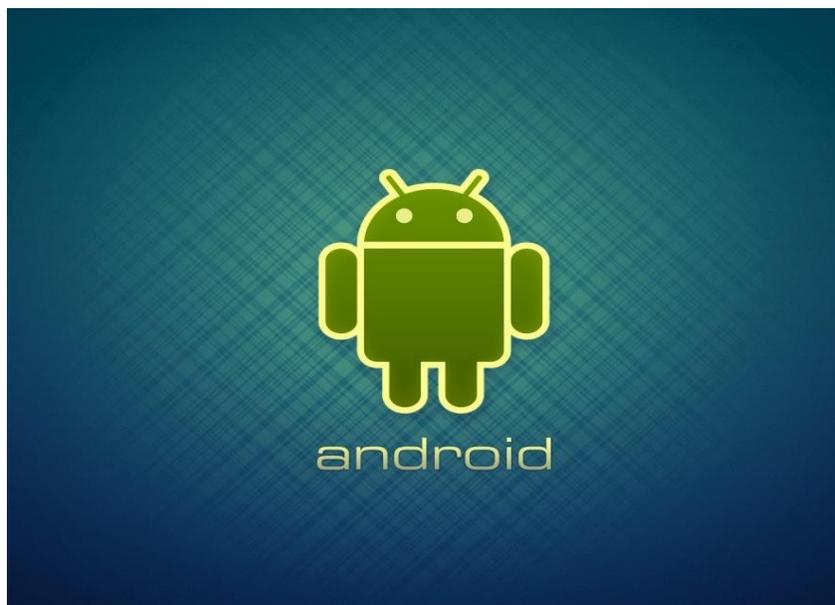
	<p>kernel (posiblemente para evitar infringir patentes de otras compañías).</p> <p>Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa.</p>
<b>Bluetooth</b>	<p>El soporte para A2DF y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.&lt;/ref&gt; Los cambios incluyeron:</p>
<b>Video llamada</b>	<p>Android soporta video llamada a través de Google Talk desde su versión HoneyComb.</p>
<b>Multitarea</b>	<p>Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj, a diferencia de otros sistemas de la competencia en la que la multitarea es congelada (Como por ejemplo iOS, en el que la multitarea se limita a servicios internos del sistema y no a aplicaciones externas) 5)</p>
<b>Características basadas en voz</b>	<p>La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.</p>
<b>Tethering</b>	<p>Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los</p>

	<p>teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo PdaNet).</p> <p>Para permitir a un PC usar la conexión de datos del móvil android se podría requerir la instalación de software adicional.</p>
--	--

**TABLA N° 1 CARACTERÍSTICAS SISTEMA OPERATIVO ANDROID**  
**Fuente: EL GRAN LIBRO DE ANDROID (2° EDICIÓN)**

### **3.2. Arquitectura**

Android es la plataforma móvil más popular del mundo que tiene el poder de cientos de millones de dispositivos móviles en más de 190 países de todo el mundo. Es la mayor base instalada de cualquier plataforma móvil que se encuentra creciendo rápidamente. Todos los días más de un millón de usuarios encienden un dispositivo Android, y empiezan a buscar aplicaciones, juegos y otros contenidos digitales. (Valverde, 2014)



**GRÁFICO N. 9 ANDROID**  
**Fuente: EL GRAN LIBRO DE ANDROID (2° EDICIÓN)**

### 3.3. Plataforma Android

Android es un Sistema Operativo además de una plataforma de Software basada en el núcleo de Linux. Diseñada en un principio para dispositivos móviles, permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptados por Google mediante el lenguaje de programación Java. Es una plataforma de código abierto. Esto quiere decir, que cualquier desarrollador puede crear y desarrollar aplicaciones escritas con lenguaje C u otros lenguajes y compilarlas a código nativo de ARM (API de Android). Inicialmente, fue desarrollada por Google Inc. aunque poco después se unió Open Handset Alliance, un consorcio de 48 compañías de Hardware, Software y telecomunicaciones, las cuales llegaron a un acuerdo para promocionar los estándares de códigos abiertos para dispositivos móviles. Google sin embargo, ha sido quien ha publicado la mayoría del código fuente de Android bajo la licencia de Software Apache, una licencia de software libre y de código abierto a cualquier desarrollador. (Valverde, 2014)

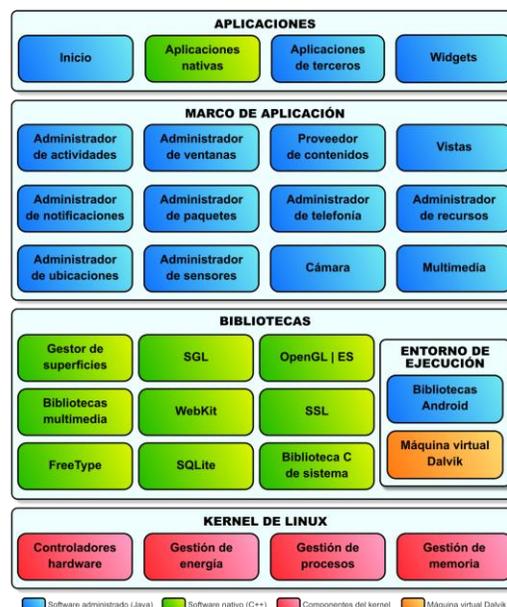


GRÁFICO N. 10 PLATAFORMA ANDROID  
Fuente: EL GRAN LIBRO DE ANDROID (2ª EDICIÓN)

### **3.3.1. Kernel de Linux**

El núcleo del sistema operativo Android es un kernel Linux versión 2.6, similar al que puede incluir cualquier distribución de Linux, como Ubuntu, solo que adaptado a las características del hardware en el que se ejecutará Android (normalmente Smartphone).

Proporciona una capa de abstracción para los elementos hardware a los que tienen que acceder las aplicaciones. Esto permite que se pueda acceder a esos componentes sin necesidad de conocer el modelo o características precisas de los que están instalados en cada teléfono. De esta forma, si una aplicación necesita, por ejemplo, la brújula, podrá utilizar la que incluya el teléfono, sea cual sea. Para cada elemento hardware del teléfono existe un controlador (o *driver*) dentro del kernel que permite utilizarlo desde el software.

Además de proporcionar controladores hardware, el kernel se encarga de gestionar los diferentes recursos del teléfono (energía, memoria) y del sistema operativo en sí: procesos, elementos de comunicación (*networking*), etc. (Valverde, 2014)

### **3.3.2. Bibliotecas**

La capa que se sitúa justo sobre el kernel la componen las bibliotecas nativas de Android (*librerías*). Estas bibliotecas están escritas en C o C++ y compiladas para la arquitectura de hardware específica del teléfono, tarea que normalmente realiza el fabricante, que también se encarga de instalarlas en el terminal antes de ponerlo a la venta. Su cometido es proporcionar funcionalidad a las aplicaciones, para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de la forma más eficiente.

### **3.3.3. Entorno de ejecución**

El componente principal del entorno de ejecución de Android es la máquina virtual Dalvik, componente que ejecuta todas y cada una de las aplicaciones no nativas de Android. Las aplicaciones se codifican normalmente en Java y son compiladas, pero no para generar un ejecutable binario compatible con la arquitectura hardware específica del dispositivo Android. En lugar de eso, se compilan en un formato específico para la máquina virtual Dalvik, que es la que las ejecuta. Esto permite

compilar una única vez las aplicaciones y distribuirlas ya compiladas teniendo la total garantía de que podrán ejecutarse en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera cada aplicación.

Aunque las aplicaciones se escriben en Java, Dalvik no es realmente una máquina virtual Java. Java se usa únicamente como lenguaje de programación, pero los ejecutables que se generan con el SDK de Android no son programas Java convencionales, por lo tanto, no se pueden ejecutar en máquinas virtuales Java convencionales. Durante el proceso de compilación de los programas Java (normalmente archivos .java) se genera, de forma intermedia, el bytecode habitual (archivos .class). Pero esos archivos son convertidos al formato específico de Dalvik en el proceso final (.dex, de *Dalvik executable*).

Los archivos .dex son mucho más compactos que los class equivalentes (hasta un 50% menos de tamaño), lo que permite ahorrar espacio en el teléfono (algo siempre escaso) y acelerar el proceso de carga. Además, a diferencia de las máquinas virtuales tradicionales, Dalvik se basa en registros en lugar de una pila para almacenar los datos, lo que requiere menos instrucciones. Esto permite ejecuciones más rápidas en un entorno con menos recursos. (Valverde, 2014)

#### **3.3.4. Marco de aplicación**

La siguiente capa la forman todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones, se apoyan en las bibliotecas y en el entorno de ejecución. La mayoría de los componentes de esta capa son bibliotecas Java que acceden a los recursos a través de la máquina virtual Dalvik.

#### **3.3.5. Aplicaciones**

La capa superior de esta pila software la forman las aplicaciones. En este se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, tanto las nativas (programadas en C o C++) como las administradas (programadas en Java), tanto las que vienen de serie con el dispositivo como las instaladas por el usuario.

Aquí está también la aplicación principal del sistema: **Inicio** (*Home*), también llamada a veces lanzador (*launcher*), porque es la que permite ejecutar otras

aplicaciones proporcionando la lista de aplicaciones instaladas y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso pequeñas aplicaciones incrustadas o *widjets*, que son también aplicaciones de esta capa. Lo principal a tener en cuenta de esta arquitectura es que todas las utilizan el mismo marco de aplicación para acceder a los servicios que proporciona el sistema operativo.

Ahora es el turno de comunicar los dispositivos móviles a la Central de Comunicaciones y que ésta identifique el transmisor, el mensaje, cuándo y de donde lo está enviando. Una vez que la central les ha enviado a los móviles el mensaje “L” éstos ejecutarán un proceso “escucha” en un programa en java en la plataforma Android esperando el *token* indicado. El Móvil emplea la capacidad de mensajería SMS para enviar y recibir mensajes hacia y desde una central u otro móvil. En el entorno de desarrollo de Eclipse crear un proyecto en el cual se modificará el archivo *AndroidManifest.xml* los permisos necesarios para la funcionalidad mencionada. De este modo, cuando la aplicación está instalada, será claro para el usuario los permisos de acceso específicos requeridos por la aplicación. (Valverde, 2014)

### **3.3.6. Usos y dispositivos**

El sistema operativo Android se usa en teléfonos inteligentes, ordenadores portátiles, netbooks, tabletas, Google TV, relojes de pulsera, auriculares y otros dispositivos., siendo este sistema operativo accesible desde terminales de menos de 100 dólares hasta terminales que superan los 1600 dólares, obviando, evidentemente, sus diferencias técnicas.

La plataforma de hardware principal de Android es la arquitectura ARM. Hay soporte para x86 en el proyecto *Android-x86*, y Google TV utiliza una versión especial de *Android x86*.

El primer teléfono disponible en el mercado para ejecutar Android fue el HTC Dream, dado a conocer al público el 22 de octubre del 2008. A principios del 2010 Google ha colaborado con HTC para lanzar su producto estrella en dispositivos

Android, el Nexus One. Google ha continuado la comercialización de la gama Nexus en el 2010 con el Samsung Nexus S, en 2011 con el Galaxy Nexus y en 2012 con el Nexus 4 (y los tablets Nexus 7 y Nexus 10). Los dispositivos Nexus son utilizados para el desarrollo e implementación de **Android**, siendo los dispositivos que estrenan las nuevas versiones disponibles. En la actualidad existen más de 650.000 aplicaciones para Android y se estima que 1.000.000 teléfonos móviles se activan diariamente.

IOS y Android 2.3.3 "Gingerbread" pueden ser configurados para un arranque dual en un iPhone o iPod Touch liberados con la ayuda de OpeniBoot e iDroid. (Valverde, 2014)

### **3.3.7. Aplicación para Android**

Una preocupación constante entre toda persona que vive en un país con alta incidencia de desastres naturales, es, además de saber cómo reaccionar ante un evento de tal magnitud, el poder monitorear y si es posible, prevenir con antelación un sismo que pueda provocar daños en la instalación en la cual se encuentre y a las personas que estén con él/ella. (Valverde, 2014)

### **3.3.8. Estructura Básica de Android**

Las aplicaciones en Android se deben programar en el lenguaje de programación Java junto a otros recursos de configuración – y se compilan con el Android SDK con el fin de generar un único archivo (de extensión .apk) para ser instalado en los dispositivos. Una vez instalado, la aplicación vivirá en su propia sandbox por motivos de seguridad y ejecución de procesos en Android. Adicionalmente, y de manera opcional, se pueden utilizar componentes nativas de Linux que no son accesibles desde la API de Android. Estas rutinas deben ser escritas en los lenguajes C o C++ y son compiladas con el Android NDK para ser utilizadas desde la aplicación escrita en Java mediante JNI.

La aplicación se ejecuta con permisos de usuario básico, desde esta perspectiva, si se requiere hacer cambios en la configuración del sistema operativo, solo podrá ser utilizando lo que ofrece la API de Android. Sin embargo, si se modifica el sistema

operativo Android para obtener acceso de súper usuario, se podrán hacer cambios desde cualquier rutina sobre todo el sistema operativo.

Toda aplicación a desarrollar debe contener un único archivo de configuración de nombre *AndroidManifest.xml* – ubicado en el directorio raíz del desarrollo. En este archivo se declaran los elementos a utilizar por la aplicación, la forma en que se relacionan y los permisos necesarios, además de otras configuraciones. Existen dos elementos importantes en esta configuración, primero la clase a instanciar para mantener un estado global de la aplicación, segundo las clases a instanciar de las diferentes componentes de aplicación que constituyen la aplicación.

Existen cuatro tipos de componentes de aplicación para desarrollar una aplicación en Android, cada una con diferentes propósitos y ciclos de vida que definen cómo se crean y destruyen. Se tomarán en cuenta las dos de mayor interés desde el punto de vista de un usuario final; los *Activities* y *Services*. La primera representa un proceso con interfaz de usuario desplegada en el dispositivo y la segunda un proceso sin interfaz de usuario que se ejecuta paralelamente a otras del tipo anterior. De esta manera, una aplicación puede tener más de un componente de aplicación y estas componentes pueden iniciar otras mediante diferentes mecanismos, sin embargo al momento de ejecutar la aplicación comenzará con la ejecución de solo un componente configurado en el archivo de configuración de la aplicación.

A veces los componentes de aplicación requieren compartir contenido entre ellas, para esto existen al menos dos formas, una es por medio de solicitudes de acciones mediante *Intents* (además estas solicitudes poseen la capacidad de iniciar componentes de aplicación), la otra forma es manteniendo actualizado el estado global de la aplicación por medio de la instancia de la clase configurada en el archivo de configuración de la aplicación. (Valverde, 2014)

### **3.3.9. Diseño de HLMP para Android**

A diferencia de las otras implementaciones de HLMP para Windows, esta implementación no solo es una biblioteca para utilizar en un desarrollo, sino que es la estructura inicial para comenzar una aplicación que utilizará HLMP en Android.

HLMP para Android se compone de tres elementos, el primero denominado *Android-Ad-Hoc* encargado de la administración del WiFi en el dispositivo, el segundo denominado *HLMP-Java* relacionado al comportamiento del protocolo HLMP, y el tercero denominado *HLMPConnect* es el esqueleto de la aplicación que administra los recursos anteriores y permite la comunicación con el usuario.

#### **3.3.9.1. HLMP-Java**

Es la implementación del protocolo HLMP y está basada en el desarrollo de HLMP para Windows XP, Vista y Mobile. Esta biblioteca se componen de tres módulos y tres sub-protocolos, los módulos son Red, Comunicación e Interoperabilidad, están encargados respectivamente de la apertura y control de sockets, administración de eventos y mensajes y por último el control de la conexión WiFi del dispositivo. De igual manera, los tres sub-protocolos son Ping, Chat y FileTransfer, encargados respectivamente de comprobar la conexión de otros dispositivos, proveer un chat entre los usuarios y permitir la transferencia de archivos.

#### **3.3.10. Android-Ad-Hoc**

Es una aplicación para Android que permite iniciar una conexión WiFi del tipo Ad-Hoc, destacando entre ellos el Nexus One de HTC.

La principal ventaja de esta aplicación es que posee una estructura simple y fácil de portar a otros proyectos, como es el caso del protocolo HLMP para Android. Consta principalmente de 4 elementos, entre ellos dos componentes de aplicación – un Activity para que el usuario comience o termine la conexión Ad-Hoc y un Service que administra dicha conexión – la extensión, mediante una subclase, del estado global Application para acceder y configurar los componentes de aplicación, y por último los procedimientos nativos en el lenguaje C y C++ que permiten la conexión Ad-Hoc.

La mayor parte de los cambios de Bernacle a Android-Ad-Hoc constaron en remover elementos adicionales a la conexión Ad-Hoc, por ejemplo el control de usuarios, las tasas de transmisión, entre otras cosas. Otros cambios importantes fueron agregar condiciones que permitieron crear sockets en Android y la reestructuración del código para que este servicio pudiese ser portado y configurado en otro proyecto.

Android-Ad-Hoc un desarrollo minimalista basado en la aplicación Bernacle, dejando solo la capacidad de configurar algunos parámetros de la red Ad-Hoc y la interfaz para conexión y desconexión.

Esta aplicación requiere tener permiso de súper usuario en tiempo de ejecución, debido a que Android no soporta desde su API la conexión WiFi a redes Ad-Hoc ni la creación de estas, por lo tanto, debe ejecutar rutinas nativas del Sistema Operativo Linux (Android es un tipo de distribución de Linux) para configurar adecuadamente la conexión WiFi tipo Ad-Hoc en Android.

Las rutinas antes mencionadas están escritas en los lenguajes C, C++ y Bash. Las dos primeras son utilizadas desde Java por medio de JNI, mientras que la última es utilizada por las dos primeras. La compilación de estas rutinas se automatizó con el Android NDK mediante reglas de compilación (ver Anexo A), las cuales tienen como objetivo generar los binarios y bibliotecas de C o C++.

Estas rutinas, en parte, emulan el comportamiento de las herramientas de configuración de red en Linux como *ipconfig*, *iwconfig* y *wpa\_supplicant*, además del manejo de algunos controladores de tarjetas inalámbricas en los dispositivos móviles (activación y desactivación de los módulos del Kernel de Linux). (Vukovic, 2004)

### **3.3.10.1. Recepción de Mensajes SMS**

Se pueden interceptar los mensajes SMS entrantes mediante un objeto BroadcastReceiver. Para ver la forma de recibir mensajes SMS desde la aplicación para Android, en el archivo de AndroidManifest.xml se agrega el elemento <receiver> para que los mensajes SMS entrantes puedan ser interceptados por la clase SmsReceiver:

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="net.learn2develop.SMSMessaging"

android:versionCode="1"

android:versionName="1.0.0">
```

```

<application android:icon="@drawable/icon" android:label="@string/app_name">
<activity android:name=".SMS"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<receiver android:name=".SmsReceiver">
<intent-filter>
<action android:name=
"android.provider.Telephony.SMS_RECEIVED" />
</intent-filter>
</receiver>
</application>
<uses-permission android:name="android.permission.SEND_SMS">
</uses-permission>
<uses-permission android:name="android.permission.RECEIVE_SMS">
</uses-permission>
</manifest>

```

Se Añade un nuevo archivo de clase al proyecto y el nombre como SmsReceiver.java.

Agregando el archivo SmsReceiver.java al proyecto de Eclipse

En la clase SmsReceiver, se extiende la clase BroadcastReceiver y se sobrescribe el método la OnReceive ():

```

package net.learn2develop.SMSMessaging;
import android.content.BroadcastReceiver;

```

```

import android.content.Context;
import android.content.Intent;

public class SmsReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
    }
}

```

Cuando se reciben mensajes SMS, el método `onReceive ()` será invocada. El mensaje SMS es contenido y adjunto al objeto *Intent* que es el segundo parámetro en el método `onReceive ()` a través de un objeto de *Bundle*. Los mensajes se almacenan en un array de objetos en el formato PDU. Para extraer cada mensaje, se utiliza el método estático `createFromPdu ()` de la clase *SmsMessage*. El mensaje SMS se muestra a continuación utilizando la clase *Toast*:

```

package net.learn2develop.SMSMessaging;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;

import android.telephony.gsm.SmsMessage;
import android.widget.Toast;

public class SmsReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        //---obtiene el mensaje SMS pasado

```

```

Bundle bundle = intent.getExtras();

SmsMessage[] msgs = null;

String str = "";

if (bundle != null)

{

//---recupera el mensaje recibido---

Object[] pdu = (Object[]) bundle.get("pdu");

msgs = new SmsMessage[pdu.length];

for (int i=0; i<msgs.length; i++){

msgs[i] = SmsMessage.createFromPdu((byte[])pdu[i]);

str += "SMS from " + msgs[i].getOriginatingAddress();

str += " :";

str += msgs[i].getMessageBody().toString();

str += "\n";

}

//---despliega el nuevo mensaje---

Toast.makeText(context, str, Toast.LENGTH_SHORT).show();

}

}

}

```

### 3.3.10.2. Envío de Mensajes SMS

El envío de mensajes SMS potencialmente incurrirán en un costo adicional en el extremo del usuario, lo que indica que los permisos de SMS en el archivo de

AndroidManifest.xml permitirá al usuario decidir si permite la aplicación para instalar o no y agregar los dos permisos *SEND\_SMS* y *RECEIVE\_SMS* tal como se muestra en el siguiente segmento de código:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="net.learn2develop.SMSMessaging"
android:versionCode="1"
android:versionName="1.0.0">
<application android:icon="@drawable/icon" android:label="@string/app_name">
<activity android:name=".SMS"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
<uses-permission android:name="android.permission.SEND_SMS">
</uses-permission>
<uses-permission android:name="android.permission.RECEIVE_SMS">
</uses-permission>
</manifest>
```

En el archivo de main.xml localizado en la carpeta res / layout, se agrega el siguiente código para que el usuario puede introducir un número de teléfono, así como un mensaje a enviar:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Enter the phone number of recipient"
    />
    <EditText
        android:id="@+id/txtPhoneNo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Message"
    />
    <EditText
        android:id="@+id/txtMessage"
        android:layout_width="fill_parent"
        android:layout_height="150px"
        android:gravity="top"
    />
    <Button
        android:id="@+id/btnSendSMS"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```

```
android:text="Send SMS"
```

```
/>
```

```
</LinearLayout>
```

El Código de arriba crea la interfaz de usuario. En la actividad de SMS se vincula la vista del botón “Enviar Mensaje” de tal manera que cuando el usuario le dé click se va a comprobar que el número de teléfono del destinatario y el mensaje se ingresa antes de enviar el mensaje utilizando la función `sendSMS ()`, la cual se definirá en breve:

```
package net.learn2develop.SMSMessaging;
import android.app.Activity;

import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.gsm.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SMS extends Activity
{
    Button btnSendSMS;
    EditText txtPhoneNo;
    EditText txtMessage;
    /** Se le llama cuando la actividad se creó por primera vez. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
```

```

txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
txtMessage = (EditText) findViewById(R.id.txtMessage);
btnSendSMS.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View v)
    {
        String phoneNo = txtPhoneNo.getText().toString();
        String message = txtMessage.getText().toString();
        if (phoneNo.length()>0 && message.length()>0)
            sendSMS(phoneNo, message);
        else
            Toast.makeText(getApplicationContext(),
                "Ingrese el número telefónico and Mensaje.",
                Toast.LENGTH_SHORT).show();
    }
});
}
}
}

```

(Valverde, 2014)



**GRÁFICO N. 11** INTERFAZ DE LA CENTRAL DE COMUNICACIONES  
**Fuente:** Investigador

La función sendSMS() se define como sigue:

```
public class SMS extends Activity

{

    /** Se le llama cuando la actividad se creó por primera vez. */

    @Override

    public void onCreate(Bundle savedInstanceState)

    {

        //...

    }

    //---envía un mensaje SMS a otro dispositivo---

    private void sendSMS(String phoneNumber, String message)

    {

        PendingIntent pi = PendingIntent.getActivity(this, 0,

        new Intent(this, SMS.class), 0);

        SmsManager sms = SmsManager.getDefault();

        sms.sendTextMessage(phoneNumber, null, message, pi, null);

    }

}
```

Para enviar un mensaje SMS, se utiliza la clase SmsManager. A diferencia de otras clases, no es directamente una instancia de esta clase, sino que se le llama al método estático getDefault() para obtener un objeto SmsManager. El método sendTextMessage () envía el mensaje SMS con un PendingIntent. El objeto

PendingIntent se utiliza para identificar un objetivo para invocar en un momento posterior. Por ejemplo, después de enviar el mensaje, puede utilizar un objeto para mostrar PendingIntent otra actividad. En este caso, el objeto PendingIntent (pi) está apuntando a la misma actividad (SMS.java), así que cuando el SMS se envía, no pasará nada.

Para controlar el estado del proceso de envío de SMS mensaje, se puede utilizar realmente dos objetos PendingIntent junto con dos objetos BroadcastReceiver, así:

*//---envía un mensaje SMS message hacia otro dispositivo---*

```
private void sendSMS(String phoneNumber, String message)
{
String SENT = "SMS_SENT";
String DELIVERED = "SMS_DELIVERED";
PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
new Intent(SENT), 0);
PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,
new Intent(DELIVERED), 0);

//--- cuando el SMS ha sido enviado ---

registerReceiver(new BroadcastReceiver(){
    @Override
    public void onReceive(Context arg0, Intent arg1) {
        switch (getResultCode())
        {
```

```
case Activity.RESULT_OK:

    Toast.makeText(getBaseContext(), "SMS sent",
        Toast.LENGTH_SHORT).show();

    break;

case SmsManager.RESULT_ERROR_GENERIC_FAILURE:

    Toast.makeText(getBaseContext(), "Generic failure",
        Toast.LENGTH_SHORT).show();

    break;

case SmsManager.RESULT_ERROR_NO_SERVICE:

    Toast.makeText(getBaseContext(), "No service",
        Toast.LENGTH_SHORT).show();

    break;

case SmsManager.RESULT_ERROR_NULL_PDU:

    Toast.makeText(getBaseContext(), "Null PDU",
        Toast.LENGTH_SHORT).show();

    break;

case SmsManager.RESULT_ERROR_RADIO_OFF:

    Toast.makeText(getBaseContext(), "Radio off",
        Toast.LENGTH_SHORT).show();

    break;

}
```

```

    }

    }, new IntentFilter(SENT));

    //--- cuando el SMS se ha entregado ---

    registerReceiver(new BroadcastReceiver(){

        @Override

        public void onReceive(Context arg0, Intent arg1) {

            switch (getResultCode())

            {

                case Activity.RESULT_OK:

                    Toast.makeText(getBaseContext(), "SMS delivered",

                    Toast.LENGTH_SHORT).show();

                    break;

                case Activity.RESULT_CANCELED:

                    Toast.makeText(getBaseContext(), "SMS not delivered",

                    Toast.LENGTH_SHORT).show();

                    break;

            }

        }

    }, new IntentFilter(DELIVERED));

    SmsManager sms = SmsManager.getDefault();

    sms.sendTextMessage(phoneNumber, null, message, sentPI, deliveredPI);

}

```

El código anterior se utiliza un objeto PendingIntent (sentPI) para supervisar el proceso de envío. Cuando un mensaje SMS es enviado, el primer caso OnReceive BroadcastReceiver se disparará. Aquí es donde se puede comprobar el estado del proceso de envío. El objeto PendingIntent segundo (deliveredPI) supervisa el proceso

de entrega. En el segundo caso OnReceive BroadcastReceiver se activará cuando un SMS es entregado con éxito.

Así, cuando el programa recibe un token “L” en un proceso “escucha”, se levanta un proceso “WAKE\_UP”:

```
if(message.startsWith("L") && numero.compareTo(numeroorigen)==0
/*Seccion para crear un nuevo intento de actividad*/

Intent broadcast = new Intent("com.TaxistaGPPI.WAKE_UP");
```

### **3.4. DESASTRES DE LA NATURALEZA**

#### **3.4.1. DESASTRES NATURALES.**

Los sucesos de gran magnitud se denominan como desastres, que afectan a la sociedad en su diario vivir, ocasionando víctimas y daños o pérdidas de bienes materiales, o su forma de sustento sobrepasando la capacidad de las organizaciones o personas para enfrentarlas por lo que se requiere de acciones extraordinarias.

Un desastre no es un fenómeno natural, sino las consecuencias del impacto de un determinado fenómeno de origen natural o vinculado a la tecnología sobre una sociedad, una comunidad y los elementos vulnerables existentes en un territorio dado.

En un intento de síntesis podríamos acordar que un desastre constituye un evento súbito, inesperado, provocado por la naturaleza o por el hombre, que por su magnitud afecta las estructuras bio-psicosocio-ecológicas, alterando la cotidianidad, causando daños, pérdidas materiales y humanas que colocan a la población en una situación de crisis. (Espinoza, 2008)

Los desastres por la fuerza de la naturaleza se tienen:

#### **3.4.2. DESASTRES GENERADOS POR PROCESOS DINÁMICOS EN EL INTERIOR DE LA TIERRA.**

- a. **Sismos.**- Son movimientos de la corteza terrestre que generan deformaciones intensas en las rocas del interior de la tierra, acumulando energía que súbitamente es liberada en forma de ondas que sacuden la superficie terrestre.

- b. **Tsunamis.**- Movimiento de la corteza terrestre en el fondo del océano, formando y propagando olas de gran altura.
- c. **Erupciones Volcánicas.**- Es el paso del material (magma), cenizas y gases del interior de la tierra a la superficie.

### **3.4.3. DESASTRES GENERADOS POR PROCESOS DINÁMICOS EN LA SUPERFICIE DE LA TIERRA.**

- a. **Deslizamiento de Tierras.**- Que ocurren como resultado de cambios súbitos o graduales de la composición, estructura, hidrología o vegetación de un terreno en declive o pendiente:
- b. **Derrumbes.**- Es la caída de una franja de terreno que pierde su estabilidad o la destrucción de una estructura construida por el hombre.
- c. **Aludes.**- Masa de nieve que se desplaza pendiente abajo.
- d. **Aluviones.**- Flujos de grandes volúmenes de lodo, agua, hielo, rocas, originados por la ruptura de una laguna o deslizamiento de un nevado.
- e. **Huaycos.**- Desprendimientos de lodo y rocas debido a precipitaciones pluviales, se presenta como un golpe de agua lodosa que se desliza a gran velocidad por quebradas secas y de poco caudal arrastrando piedras y troncos.

### **3.4.4. DESASTRES GENERADOS POR FENÓMENOS METEOROLÓGICOS O HIDROLÓGICOS.**

- a. **Inundaciones.**- Invasión lenta o violenta de aguas de río, lagunas o lagos, debido a fuertes precipitaciones fluviales o rupturas de embalses, causando daños considerables. Se pueden presentar en forma lenta o gradual en llanuras y de forma violenta o súbita en regiones montañosas de alta pendiente.
- b. **Sequías.**- Deficiencia de humedad en la atmósfera por precipitaciones pluviales irregulares o insuficientes, inadecuado uso de las aguas subterráneas, depósitos de agua o sistemas de irrigación.
- c. **Heladas.**- Producidas por las bajas temperaturas, causando daño a las plantas y animales.
- d. **Tormentas.**- Fenómenos atmosféricos producidos por descargas eléctricas en la atmósfera.

- e. **Granizadas.**- Precipitación de agua en forma de gotas sólidas de hielo.
- f. **Tornados.**- Vientos huracanados que se producen en forma giratoria a grandes velocidades.
- g. **Huracanes.**- Son vientos que sobrepasan más 24 Km. /h como consecuencia de la interacción del aire caliente y húmedo que viene del océano Pacífico con el aire frío.

#### **3.4.5. DESASTRES DE ORIGEN BIOLÓGICO.**

- a. **Plagas.**- Son calamidades producidas en las cosechas por ciertos animales.
- b. **Epidemias.**- Son la generalización de enfermedades infecciosas a un gran número de personas y en un determinado lugar.

#### **3.4.6. DESASTRES TECNOLÓGICOS.**

- a. Incendios.
- b. Explosiones.
- c. Derrames de Sustancias Químicas.
- d. Contaminación Ambiental.
- e. Guerras.
- f. Subversión.
- g. Terrorismo.

(Roque, 2013)

### **3.5. DEFINICIÓN DE TÉRMINOS BÁSICOS**

Se describen los términos utilizados en este proyecto:

CITIC: Centro Internacional de Investigación Científica.

MANET's: Es la estandarización de enrutamiento IP.

Mobile: Movilidad.

Networks: Redes.

Protocolos: conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.

Enrutamiento: buscar un camino entre todos los posibles en una red.

Mitigación: reducción de la vulnerabilidad, es decir la atenuación de los daños potenciales sobre la vida y los bienes.

### **3.6. HIPÓTESIS**

Las alertas recibidas por redes Ad-Hoc contribuyen con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo.

### **3.7. IDENTIFICACIÓN DE VARIABLES**

#### **3.7.1 VARIABLE INDEPENDIENTE**

RED AD-HOC NETWORKS

#### **3.7.2 VARIABLE DEPENDIENTE**

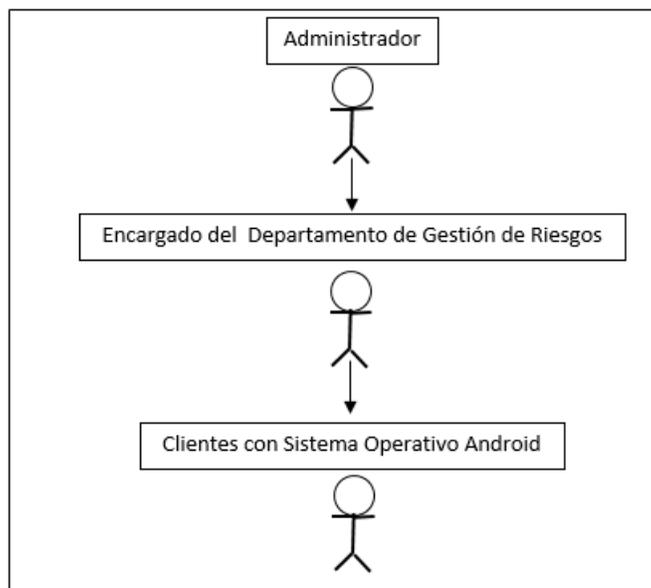
MITIGAR Y ALERTAR DESASTRES

## CAPÍTULO IV DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

### 4.1 DETERMINACIÓN DE REQUERIMIENTOS

En este punto se va a identificar los actores y las tareas que realizan en el sistema SAD Sistema de Alerta de Desastres. Luego se determinan los escenarios para cada tarea y tipo de actor.

#### 4.1.1 Definición de actores



**GRÁFICO N. 12 CUADRO JERÁRQUICO DE USUARIOS**

**Fuente:** Investigador

**Administrador:** Tiene acceso a el servidor para él envió de alertas e información a los clientes conectados.

**Encargado del Departamento de Gestión de Riesgos:** Es el usuario que se encuentre a cargo en el departamento de Gestión de Riesgos de la Universidad Nacional de Chimborazo.

**Cliente:** Es el usuario que se conecta a través de la misma red en la que se encuentra el servidor y tiene la aplicación instalada en su dispositivo.

#### 4.2 Identificación de actores y tareas

##### Administrador

- Administrar selección de alertas como:
  - ✓ Amarilla
  - ✓ Naranja
  - ✓ Roja
  - ✓ Normal

##### Cientes (Personas que tienen instalado la aplicación SAD en su dispositivo Android)

- Recibir alertas
- Recibir información

#### 4.2.1 Escenarios

**TABLA N° 2 ESCENARIO ADMINISTRAR ALERTA**

<b>Escenario</b>	<b>Administrar Alertas</b>	<b>Usuario Administrador</b>
Contexto	Consiste en seleccionar el tipo de alerta a ser enviado a los clientes conectados.	
Objetivo	Conocer sobre los riesgos y tener autoridad para poder alertar a las personas que asisten a la Universidad Nacional de Chimborazo.	

Acciones	<ul style="list-style-type: none"> <li>• Conocer el tipo de alerta por la que está cursando la Universidad Nacional de Chimborazo.</li> <li>• Tener autorización para el envío de alertas.</li> <li>• Tener información para ser enviada a los clientes conectados.</li> <li>• Enviar alerta e información a los clientes conectados.</li> </ul>
----------	--

**TABLA N° 3 ESCENARIO CLIENTES**

<b>Escenario</b>	Acceso a la Red y al Servidor	<b>Usuario Cliente</b>
Contexto	El usuario cliente accede a la red a la que se encuentra el servidor para conectarse a través de la aplicación en su dispositivo Android.	
Objetivo	Conectar la aplicación a la red para poder recibir las alertas e información, en caso de desastres.	
Acciones	<ul style="list-style-type: none"> <li>• Acceder a la red.</li> <li>• Acceder a la aplicación a través de su dispositivo.</li> <li>• Dejar en segundo plano la aplicación.</li> </ul>	

### 4.3 Diagramas de casos de uso

**Uno:** Diagrama de caso de uso para el administrador

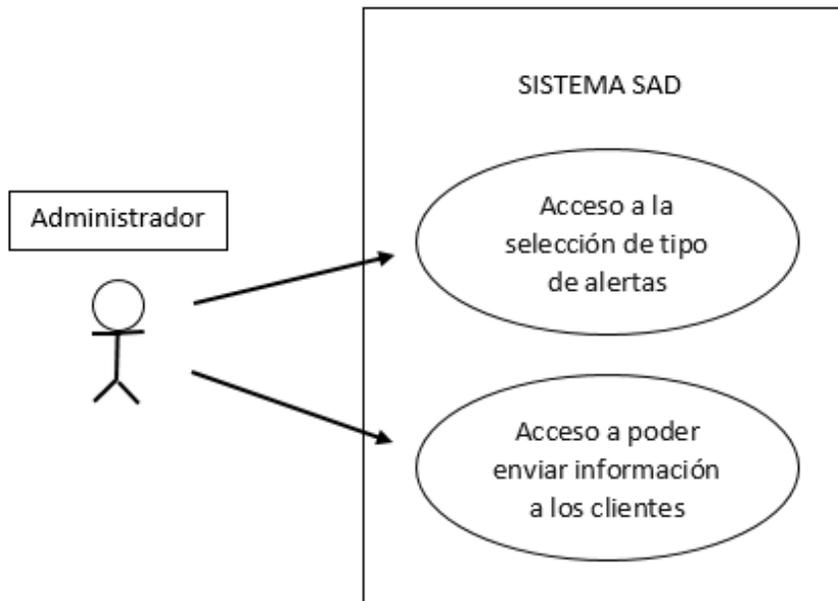


GRÁFICO N. 13 CASO DE USO VISITANTE

Fuente: Investigador

**Dos:** Diagrama de caso de uso para el cliente.

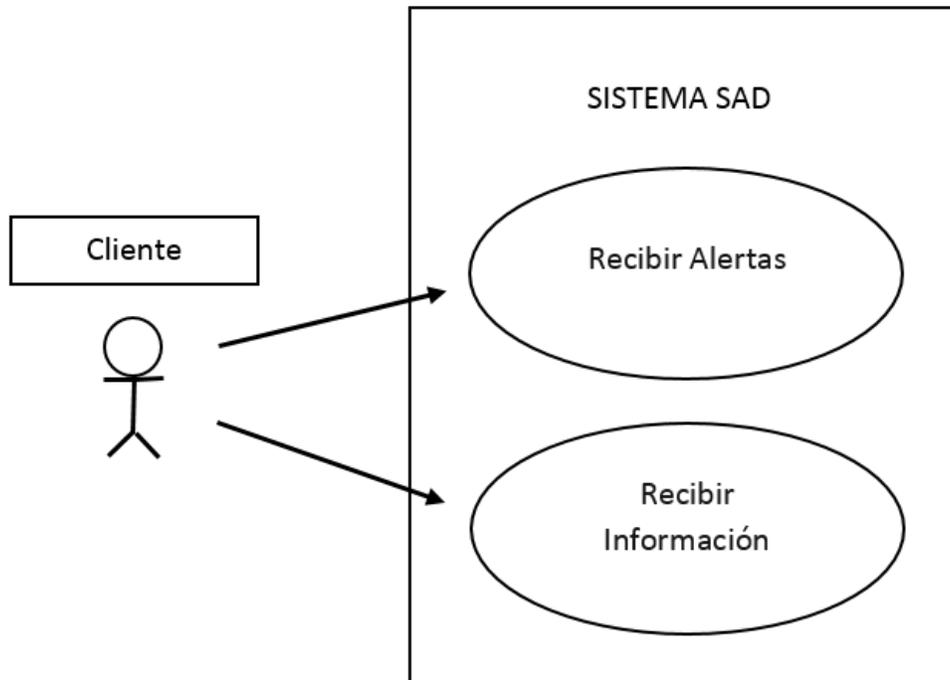


GRÁFICO N. 14 CASO DE USO CLIENTE

Fuente: Investigador

**Tres:** Diagrama de casos de uso para todos los usuarios.

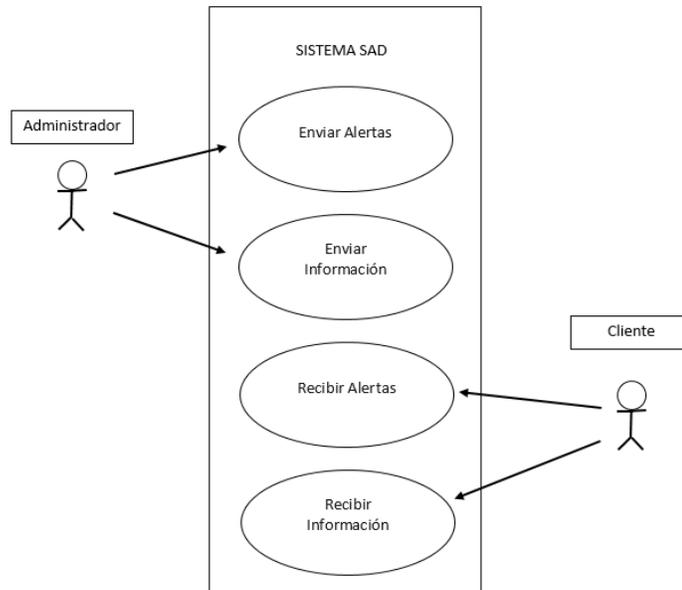


GRÁFICO N. 15 **CASO DE USO ADMINISTRADOR, CLIENTE**

Fuente: **Investigador**

#### **4.4 Propósito**

El propósito de este documento es llevar en forma clara y detallada todo el análisis de las necesidades y requerimientos del Sistema de Alerta de Desastres.

Su principal tarea consiste en la emisión de alertas de desastres e información sobre la alerta, a la vez los clientes a través de sus dispositivos móviles con sistema operativo Android la reciban. De esta manera, se pretende mitigar desastres en la Universidad Nacional de Chimborazo.

El documento está dirigido al personal del departamento de gestión de riesgos de la Universidad Nacional de Chimborazo, que posteriormente les sirva de guía para incrementar su funcionalidad; así como al usuario para tener en claro que todos los puntos detallados en este documento han sido claramente atendidos, cumpliendo con las expectativas del sistema SAD.

#### **4.4.1 Ámbito de la aplicación**

El presente Sistema de Alertas de Desastres, es un programa orientado a optimizar el proceso que realiza la Universidad Nacional de Chimborazo para alertar en caso de desastres.

#### **4.4.2 Beneficios del sistema**

- Las personas encargadas del departamento de gestión de riesgos de la Universidad Nacional de Chimborazo.
- Las personas que cuentan con sistema operativo Android en sus dispositivos móviles.
- A través de las alertas emitidas por el sistema SAD su expansión hacia las otras personas es mucho más rápido por lo que los beneficiarios son todas las personas que asisten cotidianamente a la Universidad Nacional de Chimborazo.

#### **4.4.3 Definiciones, acrónimos y abreviaturas**

##### **4.4.4 Definiciones**

- **LAN** – interconexión de quipos en un entorno local limitado.
- **SISTEMA OPERATIVO** – es un programa o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación.
- **ANDROID** - Android es un sistema operativo basado en el núcleo Linux.

##### **4.4.5 Acrónimos**

- **SAD** – Sistema de Alerta de Desastres.

#### **4.4.6 Visión general del proyecto**

- En los capítulos siguientes se realizará una descripción general de la perspectiva de la aplicación.
- Realizar una descripción en cuanto a las limitaciones generales, tanto de hardware como de software.
- Especificar los requisitos funcionales del software.
- Describir requisitos mínimos de hardware.
- Se considera también las limitaciones de diseño que se deben tomar.
- Se realizará una breve descripción de requisitos de interfaces externas, requisitos pedagógicos en cuanto a las pantallas.
- Realizar una descripción de la respectiva documentación que será requerida por el usuario en cuanto al sistema se refiere.

### **4.5 DESCRIPCIÓN GENERAL DEL PROYECTO**

#### **4.5.1 Perspectivas del producto**

El Sistema de Alerta de Desastres desarrollado por el Señor: Vasco Barreno Jonny Eduardo Egresado de la Universidad Nacional de Chimborazo, Escuela de Ing. Sistemas y Computación, que pretende apoyar al departamento de gestión de riesgos de la Universidad Nacional de Chimborazo.

#### **4.5.2 Funciones del sistema de Alerta de Desastres**

Las funciones principales que realizará son:

- Interfaz de fácil intuición.
- Escoger tipo de alerta a ser enviada.
- Poder enviar información del motivo de la alerta.

### **4.5.3 Características de los usuarios**

El Sistema de alerta de desastres será utilizado por usuarios que contengan el sistema operativo Android.

### **4.5.4 Funcionamiento**

- El Servidor es multi-plataforma
- Para su ejecución se requiere de un servidor con JAVA.

### **4.5.5 Suposiciones y dependencias**

- Para el óptimo desempeño del sistema, se recomienda que se disponga de computadores con procesador de 3.00 MHz o superior con Windows XP; 2GB de RAM; 500 GB de espacio en disco disponibles en caso de correr el sistema localmente o una conexión LAN fast ethernet 100/1000 Mbps; se recomienda una tarjeta gráfica de 256 MB o superior.
- Las personas que manejarán el Sistema deben poseer conocimientos básicos previos de computación.
- Tomamos en cuenta también que el Administrador debe conocer sobre el tipo de información a ser enviada a los clientes.

## **4.6 REQUISITOS DEL SERVIDOR SAD**

### **4.6.1 Requisitos de interface**

- Pantalla de seguridad.
- Menú de botones para la alerta.

## 4.6.2 Interfaces externas

### 4.6.2.1 Interfaces de usuario

- El usuario podrá ejecutar la aplicación en cualquier dispositivo con sistema operativo Android.
- La interfaz primaria gráfica del usuario se mantendrá estática.

### 4.6.2.2 Hardware

Los dispositivos Android requieren conexión Wireless.

### 4.6.3 Interfaz del servidor

La interfaz debe ser muy sencilla y de fácil entendimiento, de tal forma que no se necesite de un manual puesto que es una aplicación para el envío de alertas lo mejor es que se pueda hacer el envío lo más rápido posible. La interfaz debería quedar parecida al siguiente diseño.

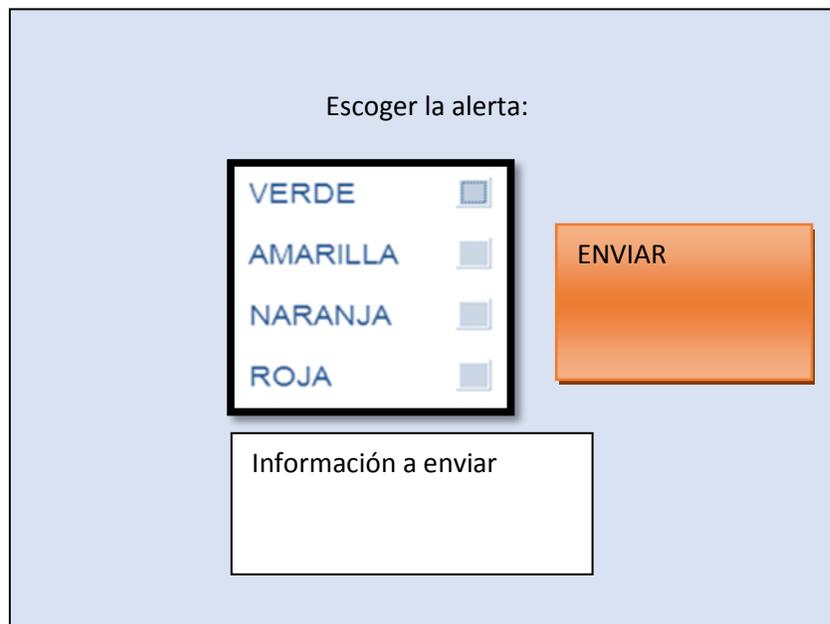


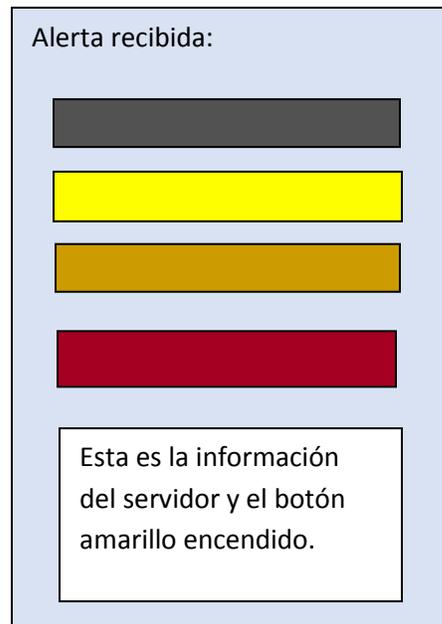
GRÁFICO N. 16 INTERFAZ REALIZADA CON MICROSOFT WORD DEL SERVIDOR

Fuente: Investigador

#### 4.6.4 Interfaz del cliente

En la interfaz del cliente debe ser fácil de entender para que al momento de recibir la información del servidor se pueda captar muy rápido lo que se debe realizar.

Por ejemplo la interfaz que debería mostrarnos en caso de una alerta amarilla por gripe debería ser como la siguiente.



**GRÁFICO N. 17 INTERFAZ REALIZADA CON MICROSOFT WORD DEL CLIENTE**  
Fuente: Investigador

#### 4.7 PROCESAMIENTO DE DATOS

##### Edición de datos.

Pueden modificarse el tipo de alerta y el mensaje a ser enviado.

#### 4.8 SALIDA DE DATOS

Botones de alerta, Mensaje de la descripción de la alerta.

## 4.9 DESCRIPCIÓN DEL DISEÑO DE LA APLICACIÓN FINAL

### 4.9.1 Visualización:

La interfaz permite ver e interactuar con la información de la aplicación se remplazó los botones de selección siguientes



GRÁFICO N. 18 **BOTONES NECESARIOS**

Fuente: **Investigador**

Por botones puesto que se facilita al usuario una forma más comprensible de escoger el tipo de alerta.



GRÁFICO N. 19 **DISEÑO FINAL SERVIDOR**

Fuente: **Investigador**

**4.9.2 Administración:** se encarga de manejar la información que va a ser emitida por el sistema.

## **4.10 DEFINICIÓN DE LAS HERRAMIENTAS A UTILIZAR**

### **a. SOFTWARE**

#### **LENGUAJES DE PROGRAMACIÓN**

**Eclipse.-** Es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

#### **SERVIDOR**

El servidor puede ser cualquier computadora que esté conectada a la red, que tenga instalada JAVA para el funcionamiento de la aplicación SERVIDOR.

### **b. HARDWARE**

#### **SERVIDOR**

Los servidores para el funcionamiento de la aplicación requieren para su adecuado funcionamiento las siguientes especificaciones:

- Procesador de 1.8MHz mínimo.
- Memoria RAM 500MB mínimo.
- Espacio libre en el disco duro de 30GB mínimo.
- Conexión a la red.

- Tener instalado JAVA.

## **CLIENTES**

En general, los clientes de la aplicación requerirán para su acceso al sistema, un teléfono con el sistema operativo android.

## **RECURSOS HUMANOS**

Con el propósito de administrar la aplicación se requiere:

**Administrador del sistema:** Una persona encargada de la administración general de la aplicación, que deberá ser un profesional capacitado.

### **4.11 EVALUACIÓN DEL PROTOTIPO**

La evaluación tiene el fin de consolidar los requisitos identificados y presentados al tutor a través del prototipo. Los resultados de esta evaluación o validación deben ser integrados a los productos del análisis siguiendo las fases pertinentes.

Además esta evaluación permite definir los requisitos de rendimiento y las limitaciones de recursos que caracterizan el diseño.

### **4.12 IMPLEMENTACIÓN**

#### **SERVIDOR**

- **COMPROBACION DEL SOFTWARE NECESARIO**
  - Para saber si el computador tiene instalado JAVA, se utiliza la utilidad ¿Tengo Java? de la propia página de Java: <http://www.java.com/es>, si aparece el mensaje de java instalado en el ordenador se puede seguir al siguiente paso, caso contrario aparece el mensaje Descarga gratuita de java, presionar en descargar e inicia el proceso de instalación.



GRÁFICO N. 20 DESCARGA DE JAVA  
Fuente: Página oficial de JAVA

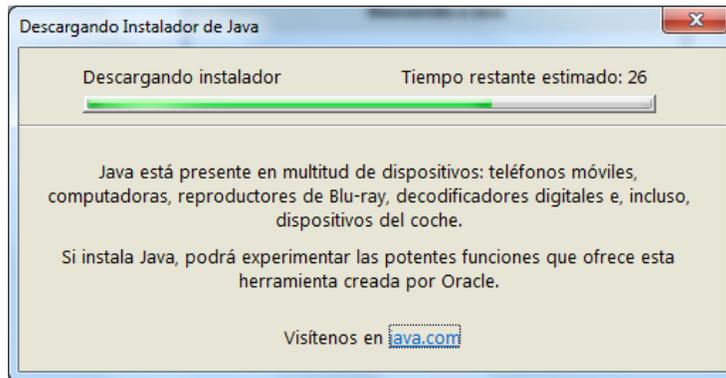
- Aparece la pantalla de bienvenida de JAVA.
- Presionar en instalar.



GRÁFICO N. 21 PANTALLA DE BIENVENIDA DE INSTALACIÓN DE JAVA  
Fuente: Investigador



- Aparece la ventana que indica que se está descargando lo



necesario para la instalación.

Aparece ventana de ofertas al instalar JAVA

Desmarcar las opciones

Presionar en el botón siguiente

Aparece la ventana de instalación de JAVA



Aparece ventana de información de instalación



Aparece que Java se ha instalado correctamente y presionar cerrar.

- Se ha terminado de instalar el software necesario para el funcionamiento del servidor.
- Copie la aplicación SAD en el computador que será el servidor, recordando que este debe estar conectado a la RED con la que van a conectarse también los dispositivos móviles.
- Ejecute la aplicación

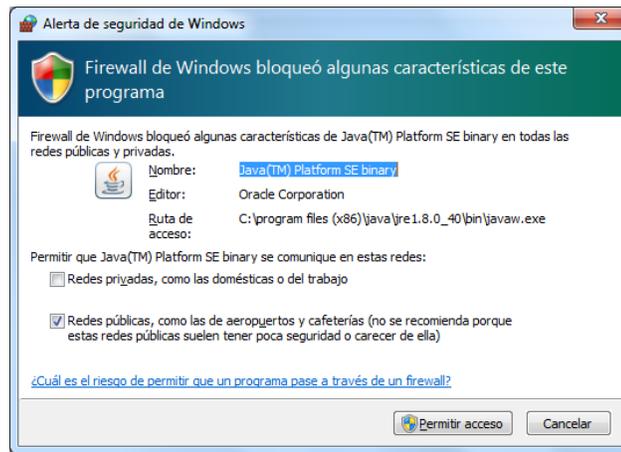


Ingrese el Usuario y Contraseña.

Hay dos usuarios un Administrador y su contraseña es AdminSAD, y el usuario emergencia y la contraseña SAD, los cuales solo deben ser entregados a las personas que están autorizadas a enviar alertas.

Una vez colocado el usuario y la contraseña correcta, aparecerá una alerta de seguridad de Windows.

Presionar Permitir acceso.



- Utilizar nuestra aplicación servidor:



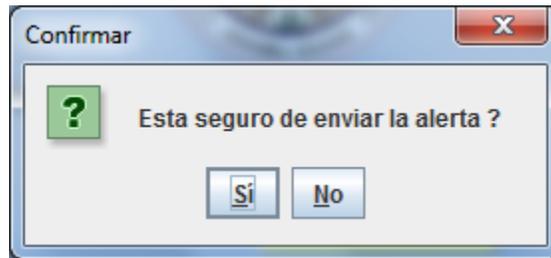
- Observar que nos muestra una ip y un puerto el cual servirá para que los clientes puedan conectarse.
- Para el uso de la aplicación seleccionar el tipo de alerta presionando en el botón que se requiera para notar que ha sido seleccionado se encenderá el botón.



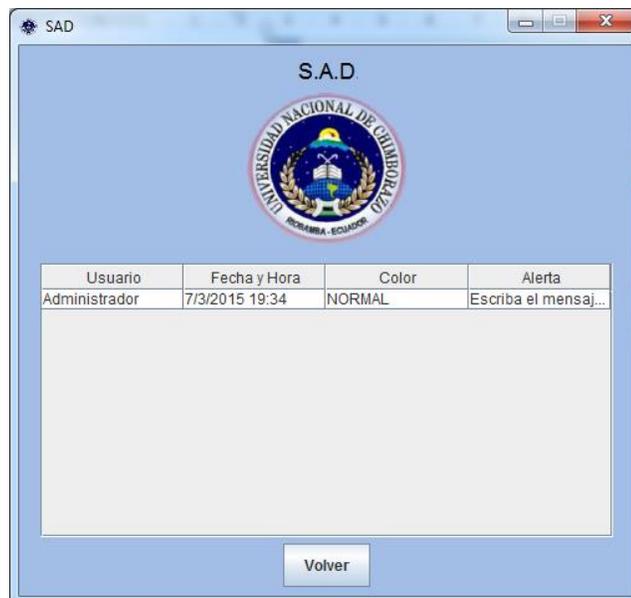
- En el cuadro de texto escriba toda la información sobre el motivo de la alerta.



- Presionar el botón enviar.



- Confirmar si va a enviar la alerta.
- Será enviada a todos los clientes que se encuentren conectados a la misma RED y que tengan la aplicación SAD cliente instalado.
- Observar las alertas enviadas presione en el botón reporte de alertas



Observar el usuario que ha ingresado, la fecha y hora, el tipo de alerta, y la información que se ha enviado.

## CLIENTE

- Para la instalación del cliente necesita que el dispositivo cuente con sistema operativo android.

- Tener habilitada la opción de Fuentes desconocidas en ajustes, seguridad.



- Pasar la aplicación SADcliente por bluetooth por cable de datos o retirando la memoria del celular y conectarla al equipo.
- Presionar abrir la aplicación y presionar instalar.
- Abrir la aplicación.



Escribir la IP del servidor y el Puerto, presionar en conectar.

- Una vez conectado se enciende la luz verde.

- Ahora dejar en segundo plano la aplicación presionando la tecla home del celular.
- En caso de recibir una alerta sonara el sonido de mensaje y notificara en la barra de notificaciones del celular.



- Abrir la aplicación desde la barra de notificación del celular o desde el menú de aplicaciones para mostrar la alerta.
- Se encenderá un botón con el tipo de alerta y se recibe la información correspondiente.



- Recibida la alerta y la información poner en segundo plano la aplicación.

## **4.13 METODOLOGÍA**

### **4.13.1 TIPO DE ESTUDIO**

#### **Estudio Comparativo**

El método comparativo suele ser popular en un estado temprano de la evolución de un campo de investigación, cuando se intenta salir del nivel inicial de los estudios de caso exploratorios a un nivel más avanzado de estructuras teóricas generales o leyes, como invariantes, causalidad o evolución.

El diseño de la investigación comparativa es simple. Se estudia ejemplares que pertenecen al mismo grupo pero que difieren en algunos aspectos. Estas diferencias llegan a ser el foco de la exanimación. La meta es descubrir por qué los casos son diferentes, para revelar la estructura subyacente general que genera o permite tal variación.

### **4.13.2 POBLACIÓN Y MUESTRA**

#### **4.13.2.1 POBLACIÓN**

Los 134 estudiantes Escuela de Ingeniería en Sistemas de la Universidad Nacional de Chimborazo.

#### **4.13.2.2 MUESTRA**

De los 134 estudiantes de la Escuela de Ingeniería en Sistemas de la Universidad Nacional de Chimborazo se seleccionan lo que en el momento de la encuesta disponen de la tecnología android.

Muestra= 36 estudiantes.

### 4.13.3 OPERACIONALIZACIÓN DE VARIABLES

En las tablas se presentan la Operacionalización conceptual y metodológica de las variables, las mismas que se han identificado de acuerdo a la hipótesis:

**TABLA N° 4 OPERACIONALIZACIÓN DE LAS VARIABLES**

VARIABLE	TIPO	DEFINICIÓN	DIMENSIÓN	INDICADORES
V1. Red Ad-Hoc Networks	Independiente	Una red "Ad-hoc", consiste en un grupo de ordenadores que se comunican cada uno directamente con los otros a través de las señales de radio sin usar un punto de acceso.	Conocimiento Profesional	Simulacros utilizando red Ad-Hoc Networks
V2. Mitigar y alertar desastres en la Universidad Nacional de Chimborazo.	Dependiente	Se refiere a una aplicación para dispositivos móviles que permita enviar una alerta en caso de algún desastre.	Desarrollo de Software	Encuestas

#### 4.13.4 PROCEDIMIENTOS

**TABLA N° 5 PROCEDIMIENTOS**

<b>N°</b>	<b>FASE</b>	<b>TÉCNICA DE INVESTIGACIÓN</b>	<b>INSTRUMENTO</b>	<b>DESCRIPCIÓN DEL INSTRUMENTO</b>
1	Estudio Introductorio y análisis de la red Ad-Hoc Networks	Estudio teórico	Libros, Manuales, Guías	Material bibliográfico que sirva de referencia para conocer y entender la red Ad-Hoc Networks
2	Análisis de tecnologías a integrar	Estudio teórico práctico	Libros, Manuales, Guías	Material bibliográfico de tecnologías que incorpora la red Ad-Hoc Networks y software libre para aplicaciones de dispositivos móviles.

3	Determinación de herramientas y tecnologías a incorporar en el proyecto	Estudio teórico práctico	Libros, Manuales, Guías	Análisis de herramientas y tecnologías a incorporar de las varias recopiladas anteriormente
4	Identificación de los principales requerimientos y funciones a cumplir por el sistema	Estudio experimental, práctico	Recursos tecnológicos	Análisis de funciones y requerimientos a cubrir en el proyecto.
5	Diseño de arquitectura e infraestructura	Estudio experimental, práctico	Recursos tecnológicos	Diseño y elaboración de la arquitectura de software.
6	Desarrollo del proyecto	Estudio experimental, práctico	Recursos tecnológicos	Desarrollo de la aplicación para dispositivos móviles.
7	Pruebas	Estudio experimental,	Recursos	Proceso de

		práctico	tecnológicos	validación
8	Identificación y corrección de errores	Estudio experimental, práctico	Recursos tecnológicos	Identificación y errores encontrados en ejecución de la aplicación.

Fuente: Estructura del proyecto

**CAPÍTULO V**  
**ANÁLISIS DE RESULTADOS**

Para el efecto se ha realizado una encuesta a 36 estudiantes de la Universidad Nacional de Chimborazo que disponen del sistema android en sus teléfonos celulares.

**5.1. PREGUNTA 3 Y 6 DE LA ENCUESTA TOMADAS PARA LA TOMA DE RESULTADOS**

**3. ¿Usted cree que se debería disponer de un sistema para alertar a la Universidad Nacional de Chimborazo sobre el riesgo de erupción del volcán Tungurahua?**

**TABLA N° 6 RESULTADOS DE LA PREGUNTA 3**

<b>INDICADOR</b>	<b>NÚMERO</b>	<b>PORCENTAJE</b>
NADA	15	42 %
POCO	2	6 %
MUCHO	19	52 %
<b>TOTAL</b>	<b>36</b>	<b>100 %</b>

**Fuente:** Encuesta a los estudiantes de la Escuela de Ingeniería en Sistemas  
**Elaborado por:** Jonny Vasco

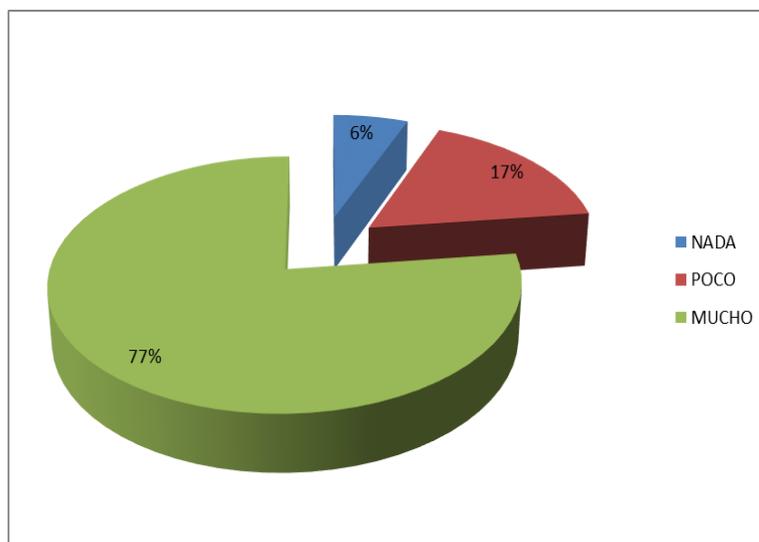


GRÁFICO N. 22 PREGUNTA 3 RESULTADOS  
Fuente: Investigador

### INTERPRETACIÓN

En la pregunta realizada a los estudiantes sobre si es necesario un sistema de alerta ante posible desastres naturales; 15 estudiantes que corresponden al 42 % manifiestan que NADA, 2 equivalente al 6 %, POCO, y 19 estudiantes que es igual al 52 % afirman que MUCHO.

### ANÁLISIS

Más de la mitad de los encuestados creen si se debe disponer de un sistema para alertar a la Universidad Nacional de Chimborazo sobre el riesgo de erupción del volcán Tungurahua.

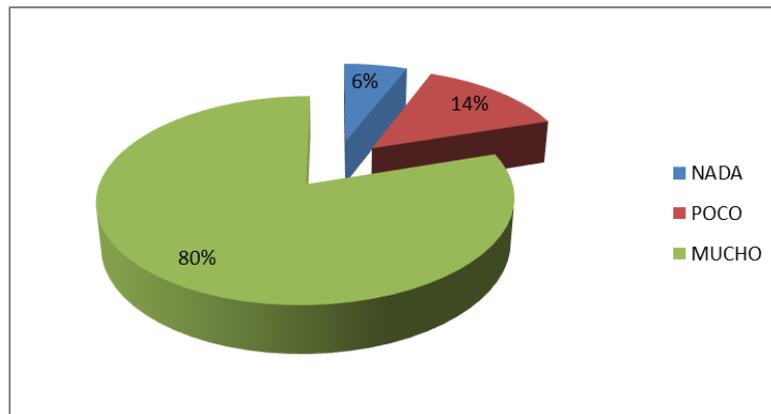
**6. ¿Usted cree que las alertas recibidas por redes móviles Ad-Hoc contribuirán con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo?**

**TABLA N° 7 RESULTADOS DE LA PREGUNTA 6**

INDICADOR	NÚMERO	PORCENTAJE
NADA	12	33 %
POCO	4	11 %
MUCHO	20	56 %
<b>TOTAL</b>	<b>36</b>	<b>100 %</b>

**Fuente:** Encuesta a los estudiantes de la Escuela de Ingeniería en Sistemas

**Elaborado por:** Jonny Vasco



**GRÁFICO N. 23 PREGUNTA 6 RESULTADOS**

**INTERPRETACIÓN**

En la pregunta realizada a los estudiantes sobre las alertas recibidas; 12 estudiantes que corresponden al 33 % manifiestan que NADA, 4 equivalente al 11 %, POCO, y 20 estudiantes que es igual al 56 % afirman que MUCHO.

## 5.1 ANÁLISIS

Más de la mitad de los encuestados creen que las alertas recibidas por la Universidad Nacional de Chimborazo, si ayudan a mitigar los riesgos de desastres naturales.

## 5.2. COMPROBACIÓN DE LA HIPÓTESIS

### 5.2.1. Hipótesis

Las alertas recibidas por redes móviles Ad-Hoc contribuyen con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo.

### 5.2.2 Comprobación

Para la comprobación de la hipótesis planteada se realizó una encuesta, y se implementará una la prueba de Chi-Cuadrado, para la comprobación de la hipótesis.

### 5.2.3 Prueba de la Hipótesis

Ho= Las alertas recibidas por redes móviles Ad-Hoc no contribuyen con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo.

H<sub>1</sub>= Las alertas recibidas por redes móviles Ad-Hoc contribuyen con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo.

### Chi Cuadrado

$$\chi_c^2 = \sum \frac{(f_o - f_e)^2}{f_e}$$

### Simbología

En el empleo de las diversas fórmulas se utilizó la siguiente simbología:

$\chi_c^2$  = “Chi” cuadrado calculado                       $f_o$  = frecuencia observada

$\chi_t^2$  = “Chi” cuadrado teórico                       $f_e$  = frecuencia esperada

$\Sigma$  = Sumatoria

$\alpha$  = nivel de significación

IC = intervalo de confianza

GL=grados de libertad

Donde:

IC= 95 %

e= 5 %

$\alpha = 0.05$

Nivel de Significancia  $\alpha = 0,05$  (Error permitido por defecto)

La hipótesis planteada se comprueba con las preguntas; 3 y 6.

Frecuencia observada

**TABLA N° 8 FRECUENCIA OBSERVADA**

<b>INDICADOR</b>	<b>NADA</b>	<b>POCO</b>	<b>MUCHO</b>	<b>TOTAL</b>
Usted cree que se debería disponer de un sistema para alertar a la Universidad Nacional de Chimborazo sobre el riesgo de erupción del volcán Tungurahua	15	2	19	36
Usted cree que las alertas recibidas por redes móviles Ad-Hoc contribuirán con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo	12	4	20	36
<b>TOTAL</b>	<b>27</b>	<b>6</b>	<b>39</b>	<b>72</b>

**Fe**= El total de la fila del número a buscar X el total de la columna del mismo número / para el gran total:

Ejemplo: Si busco la frecuencia esperada de 15 en la opción NADA de la pregunta 3, multiplico, total de filas es igual a = 36 X total de columnas es igual a 27, el resultado que es 972 dividido para 72 que es el gran total.

**Grados de libertad:**

Fórmula= (C-1) (F-1)

C= Columnas

F= Filas

Las filas y columnas se toman en cuenta las de los indicadores y las de las preguntas

Columnas= (3-1)= 2

Filas= (2-1) = 1

GL= (2X1) = 2

GL= 2

$\chi_c^2 = 9,48$

$\chi_i^2 = 5.99$

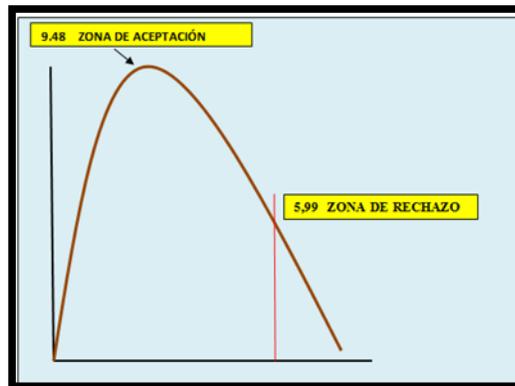
**TABLA N° 9 TABLA DE CONTINGENCIA DE LA HIPÓTESIS**

		<b>Fo</b>	<b>Fe</b>	<b>Fo-fe</b>	$\frac{(f_o - f_e)^2}{f_e}$
Usted cree que se debería disponer de un sistema para alertar a la Universidad Nacional de Chimborazo sobre el riesgo de erupción del volcán Tungurahua	<b>NADA</b>	15	13.5	1.5	0.16
	<b>POCO</b>	2	3	-1	0.3
	<b>MUCHO</b>	19	19.5	0.5	0.01

Usted cree que las alertas recibidas por redes móviles Ad-Hoc contribuirán con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo	<b>NADA</b>	12	13.5	-1.5	0.1
	<b>POCO</b>	4	1	3	9
	<b>MUCHO</b>	20	19.5	0.5	0.01
					<b>9.48</b>

### Verificación

Como  $\chi_c^2 = 9.48$  es mayor que  $\chi_t^2 = 5.99$ ; por lo que  $p$  es  $<$  que 0,05 % entonces se rechaza la hipótesis nula ( $H_0$ ) y se acepta la hipótesis específica 1 ( $H_1$ ) es decir: Las alertas recibidas por redes móviles Ad-Hoc contribuyen con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo.



## **CAPÍTULO VI**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **6.1. CONCLUSIONES**

- La red Ad-Hoc Networks necesitan de un algoritmo de enrutamiento para mantener la tabla de enrutamiento pequeña y que permita escoger la mejor ruta hacia el destino utilizando multicast.
  
- Las redes Multihop Ad-Hoc Networks son redes en donde los usuarios están distribuidos sobre una amplia área y que los nodos conectados deben colaborar para llevar mensajes a través de la red.
  
- Eclipse es un programa de software libre que permite el desarrollo de aplicaciones para dispositivos móviles con sistema operativo Android, las aplicaciones que permiten la comunicación entre una computadora y un dispositivo móvil con sistema operativo Android se lo conoce como socket.

## **6.2 RECOMENDACIONES**

- Para implementar una red Ad-Hot, se debe escoger de forma correcta el algoritmo de enrutamiento, en el caso de esta investigación se utiliza multicast ya que nos permite mantener actualizada y pequeña la tabla de enrutamiento.
- Al implementar multihop Ad-Hoc Networks para que los nodos colaboren en llevar mensajes formando una red multisalto el servidor debe contar con sistema operativo Linux para su fácil implementación.
- La aplicación desarrollada debe ser utilizada de forma adecuada ya que mediante la misma se brinda a la comunidad universitaria alertas de diferentes tipos de desastres.

## BIBLIOGRAFÍA

Escudero, A. (08 de Enero de 2007). *Prospección de sitio para la implementación de redes inalámbricas*. Recuperado el 12 de Septiembre de 2013, de <http://www.wilac.net/tricalcar>

Galeas, A. (2004). *Ad-Hoc, Network*. Recuperado el 08 de Enero de 2014, de <http://profesores.usfq.edu.ec/renej/Contenidos%20Comunicaciones%20Moviles/Exposiciones%202007/Ad%20Hoc%20Network.pdf>

López, J. (2004). *VENTAJAS DE USAR SUBREDES EN UNA RED AD-HOC* . Recuperado el 08 de Enero de 2014, de [http://research.ac.upc.edu/XARXES/CompNet/papers/paper\\_telecom\\_id.pdf](http://research.ac.upc.edu/XARXES/CompNet/papers/paper_telecom_id.pdf)

Staff, U. (2012). *Redes Avanzadas*. Recuperado el 20 de Septiembre del 2014, de <https://books.google.es/books?id=PK2s2YYGiokC&printsec=frontcover#v=onepage&q&f=false>

Mercado, A. (2009). *La computadora*.

Mercado, A. (2006). *Redes inalámbricas Ad-hoc*. Recuperado el 20 de Enero de 2014, de <http://facultad.bayamon.inter.edu/cgonzalezr/elen4618/adhoc.pdf>

In K. Al Agha, editor, *IFIP Ad-Hoc Networking*, pages 157–176. Boston: Springer, 2006.

Addison Wesley, 1998. *El Gran Libro de Android*. En *Proceedings of the 2008 Ambi-Sys workshop on Software Organisation and MonIToring of Ambient*

VARSHAVSKY, A., REID, B. y DE LARA, E. A cross-layer approach to service discovery and selection in MANETs. En Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on, páginas 8 pp.–466. 2005. (ref. pág. 36)

VERVERIDIS, C. N. y POLYZOS, G. C. Service discovery for mobile ad hoc networks: A survey of issues and techniques. IEEE Communications Surveys & Tutorials, vol. 10(3), páginas 30–45, 2008. (ref. pág. 28, 37)

VUKOVIC, M. y ROBINSON, P. Adaptive, planning based, web service composition for context awareness. En Proceedings of the Second International on Pervasive

# ANEXOS

## ANEXO I. HOJA DE RECEPCION DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO



UNIVERSIDAD NACIONAL DE CHIMBORAZO

---

Riobamba 18 de Diciembre del 2015

A petición verbal de la parte interesada, el Ing. Daniel Santillán, docente de la facultad de Ingeniería de la Universidad Nacional de Chimborazo.

### CERTIFICA QUE:

El señor **VASCO BARRENO JONNY EDUARDO** con C.I. **0603455270**, realizo para esta institución el sistema de Alerta de Desastres para dispositivos móviles.

Es todo cuanto puedo informar en honor a la verdad autorizando a los interesados hacer uso del presente documento cuando creyera conveniente.

Atentamente,

A handwritten signature in blue ink, appearing to read 'Daniel Santillán', is written over a faint circular stamp.

Ing. Daniel Santillán

Docente de la Facultad de Ingeniería.

## ANEXO II. ENCUESTA



### UNIVERSIDAD NACIONAL DE CHIMBORAZO

Seleccione con una X la opción que a su criterio es su respuesta:

1. **¿A su criterio la Universidad Nacional de Chimborazo está expuesta a algún desastre natural?**

NADA

POCO

MUCHO

2. **¿Le agradaría recibir información oportuna sobre riesgos que ocurran en la Universidad Nacional de Chimborazo?**

NADA

POCO

MUCHO

3. **¿Usted cree que se debería disponer de un sistema para alertar a la Universidad Nacional de Chimborazo sobre el riesgo de erupción del volcán Tungurahua?**

NADA

POCO

MUCHO

4. **¿Conoce la red mobile ad-hoc networks para dispositivos móviles?**

NADA

POCO

MUCHO

5. **Mediante la red mobile ad-hoc networks para dispositivos móviles, se podrá instalar en su teléfono celular una aplicación con la que podrá recibir la alerta si existe una inminente erupción del volcán Tungurahua. ¿Le gustaría instalarla?**

NADA

POCO

MUCHO

6. **¿Usted cree que las alertas recibidas por redes móviles Ad-Hoc contribuirán con la mitigación en caso de desastres en la Universidad Nacional de Chimborazo?**

NADA

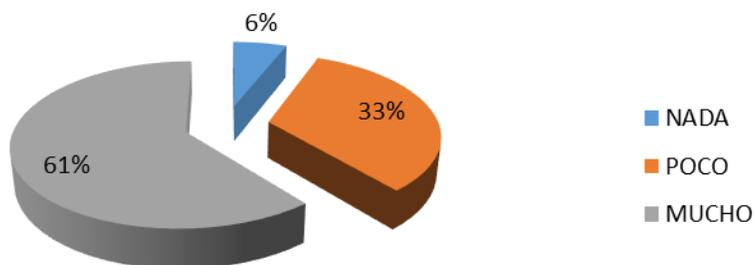
POCO

MUCHO

### ANEXO III. ANÁLISIS DE LAS PREGUNTAS 1, 2, 4, 5

#### 1. ¿A su criterio la Universidad Nacional de Chimborazo está expuesta a algún desastre natural?

INDICADOR	NÚMERO	PORCENTAJE
NADA	2	6 %
POCO	12	33 %
MUCHO	22	61 %
TOTAL	36	100 %



#### INTERPRETACIÓN

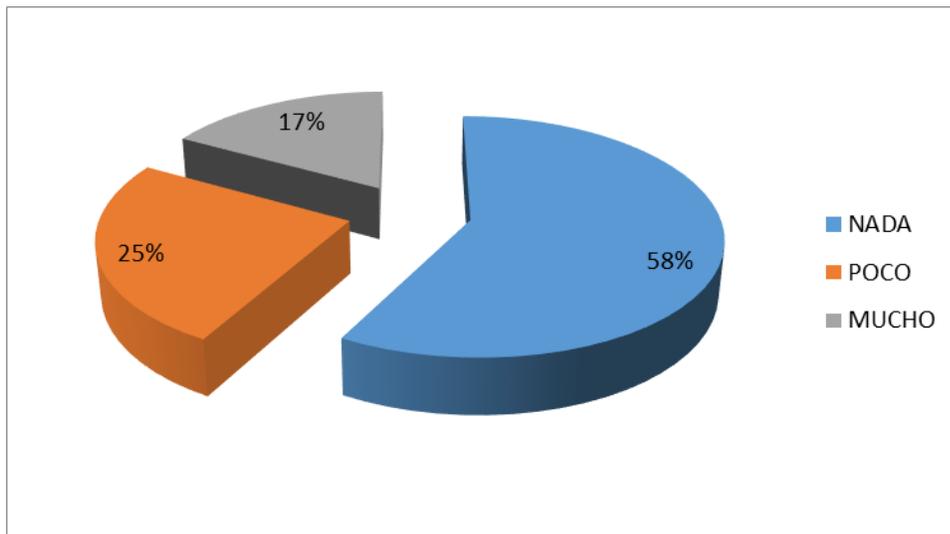
En la pregunta realizada a los estudiantes sobre los desastres naturales en la Universidad nacional de Chimborazo; 2 estudiantes que corresponden al 6 % manifiestan que NADA, 12 equivalente al 33 %, POCO, y 22 estudiantes que es igual al 61 % afirman que MUCHO.

#### ANÁLISIS

Más de la mitad de los encuestados creen que la Universidad Nacional de Chimborazo si está expuesta a Riesgos de desastres naturales.

## 2. ¿Le agradecería recibir información oportuna sobre riesgos que ocurran en la Universidad Nacional de Chimborazo?

INDICADOR	NÚMERO	PORCENTAJE
NADA	21	58 %
POCO	9	25 %
MUCHO	6	17 %
TOTAL	36	100 %



### INTERPRETACIÓN Y ANÁLISIS

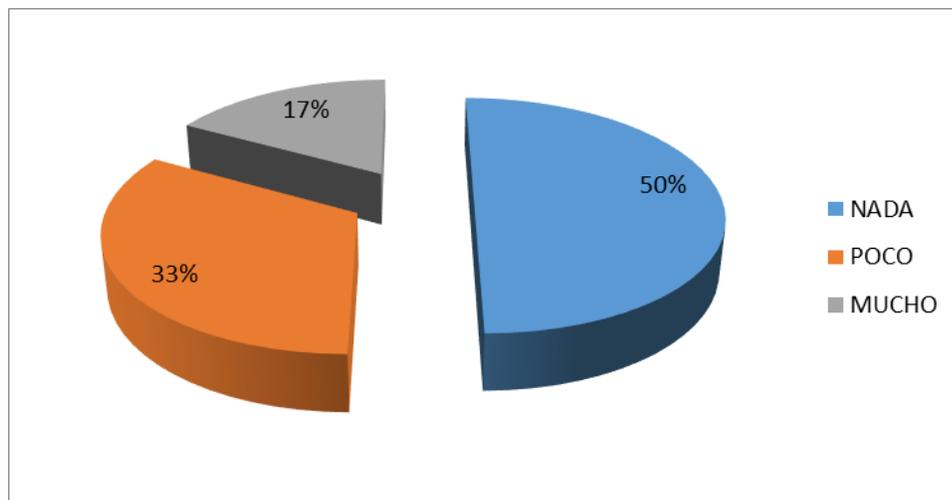
En la pregunta realizada a los estudiantes sobre si; le agradecería recibir información oportuna sobre riesgos que ocurran en la Universidad Nacional de Chimborazo; 21 estudiantes que corresponden al 58 % manifiestan que NADA, 9 equivalente al 25 %, POCO, y 6 estudiantes que es igual al 17 % afirman que MUCHO.

### ANÁLISIS

Más de la mitad de los encuestados creen no es necesario conseguir información sobre si la Universidad Nacional de Chimborazo esta expuesta a Riesgos de desastres naturales.

#### 4. ¿Conoce la red ad-hoc networks para dispositivos móviles?

INDICADOR	NÚMERO	PORCENTAJE
NADA	18	50 %
POCO	12	33 %
MUCHO	6	17 %
TOTAL	36	100 %



#### INTERPRETACIÓN Y ANÁLISIS

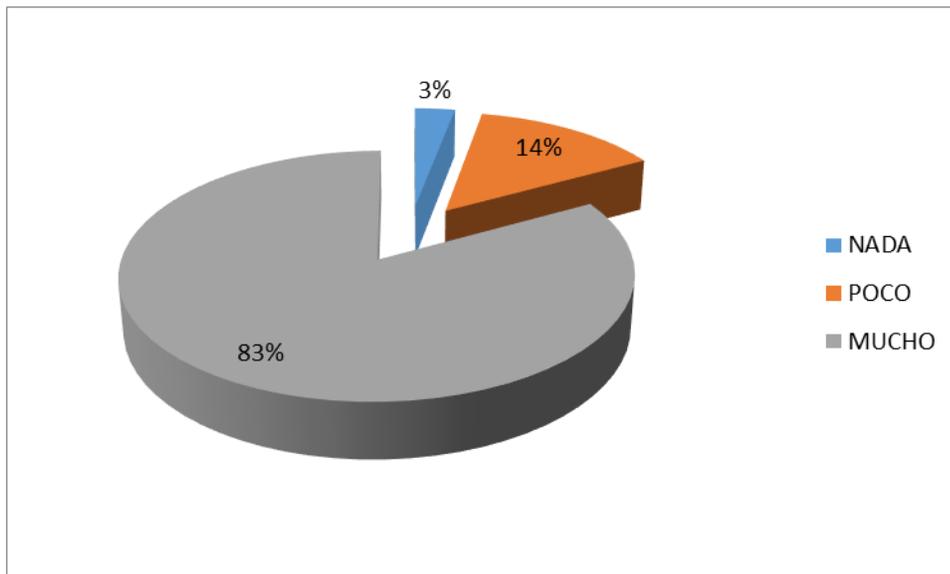
En la pregunta realizada a los estudiantes sobre si conoce la red ad-hoc networks para dispositivos móviles; 18 estudiantes que corresponden al 50 % manifiestan que NADA, 12 equivalente al 33 %, POCO, y 6 estudiantes que es igual al 17 % afirman que MUCHO.

#### ANÁLISIS

El 50 % de los encuestados no tienen conocimientos sobre la red ad-hoc networks para dispositivos móviles, que tienen el sistema Android.

**5. Mediante la red ad-hoc networks para dispositivos móviles, se podrá instalar en su teléfono celular una aplicación con la que podrá recibir la alerta si existe una inminente erupción del volcán Tungurahua. ¿Le gustaría instalarla?**

INDICADOR	NÚMERO	PORCENTAJE
NADA	1	3 %
POCO	5	14 %
MUCHO	30	83 %
TOTAL	36	100 %



#### INTERPRETACIÓN Y ANÁLISIS

En la pregunta realizada a los estudiantes sobre si le gustaría instalar la aplicación para dispositivos móviles; 1 estudiante que corresponden al 3 % manifiestan que NADA, 5 equivalente al 14 %, POCO, y 30 estudiantes que es igual al 83 % afirman que MUCHO.

## **ANEXO IV. SIMULACRO**

### **INTRODUCCIÓN**

El simulacro a realizar en la Universidad Nacional de Chimborazo para dar a conocer a los estudiantes sobre las redes Ad-Hoc y probar el sistema de alerta de desastres y la obtención de resultados a través de encuestas.

Pasos para el simulacro:

- Tomar unas fotos a los estudiantes sin saber del conocimiento de la alerta.
- Tomar unas fotos de la reacción de los estudiantes al recibir una alerta por parte de una persona desconocida.
- Informar sobre el simulacro.
- Instalar la Red Ad-Hoc y la aplicación para los dispositivos con sistema operativo Android.
- Tomar unas fotos al enviar una alerta a través del sistema de alerta de desastres.
- Realizar una encuesta para toma de datos.

### **ALERTA DADA POR VOZ**

Los estudiantes al escuchar alerta amarilla por el virus H1N1 se asustaron y salieron apurados como se muestra en las siguientes fotos:



Imagen de los estudiantes antes de recibir la alerta.



Salieron sin terminar de escuchar todo sobre la alerta.





Al revisar como dejaron su curso nos percatamos que por el susto dejaron olvidado sus cosas, a su vez prendidas las computadoras.

### **SISTEMA DE ALERTA DE DESASTRES**

Recibiendo la alerta en sus dispositivos Android





Como resultado la salida fue más ordenada y con más calma, recogiendo todas sus pertenencias y dejando apagados todos los equipos, es decir que les llegó toda la información que se les quería entregar del motivo de la alerta.

## ANEXO V. MANUAL DE USUARIO

### INFORMACIÓN GENERAL

Este producto está diseñado para alertar y mitigar desastres en la Universidad Nacional de Chimborazo.

Donde se cuenta con una aplicación servidor, de escritorio, que será la encargada de enviar a todos los clientes que se encuentren conectados en la red el tipo de alerta y el motivo de la misma y si se desea se puede incluir indicaciones de que se debe realizar.

Este producto también cuenta con una aplicación cliente, diseñada para trabajar sobre el sistema operativo android, la cual se conectara con el servidor.

A continuación detallamos paso a paso el manejo de este producto:

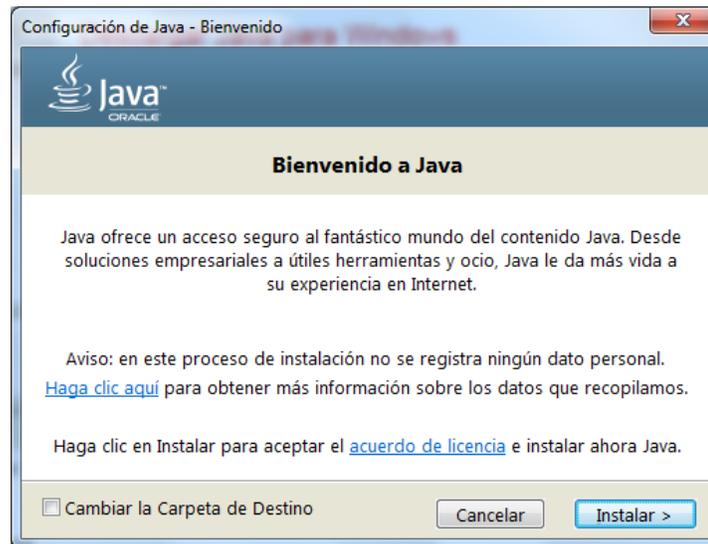
### SERVIDOR

#### COMPROBACION DEL SOFTWARE NECESARIO

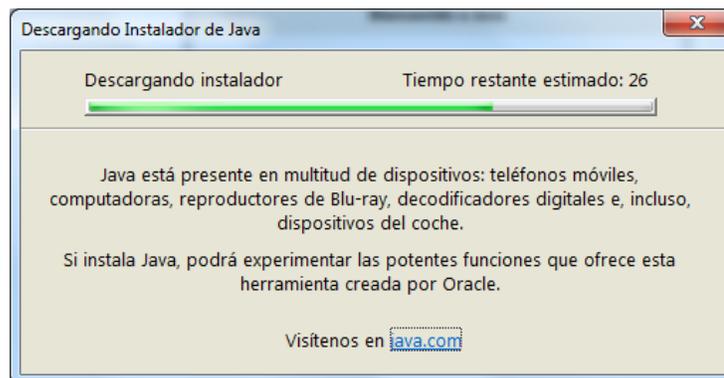
Para saber si el computador tiene instalado JAVA, se utiliza la utilidad ¿Tengo Java? de la propia página de Java: <http://www.java.com/es>, si aparece el mensaje de java instalado en el ordenador se puede seguir al siguiente paso, caso contrario aparece el mensaje Descarga gratuita de java, presionar en descargar e inicia el proceso de instalación.



Aparece la pantalla de bienvenida de JAVA.  
Presionar en instalar.



Aparece la ventana que indica que se está descargando lo necesario para la instalación.



Aparece ventana de ofertas al instalar JAVA

Desmarcar las opciones

Presionar en el botón siguiente

Aparece la ventana de instalación de JAVA



Aparece ventana de información de instalación



Aparece que Java se ha instalado correctamente y presionar cerrar.

Se ha terminado de instalar el software necesario para el funcionamiento del servidor. Copie la aplicación SAD en el computador que será el servidor, recordando que este debe estar conectado a la RED con la que van a conectarse también los dispositivos móviles.

Ejecute la aplicación

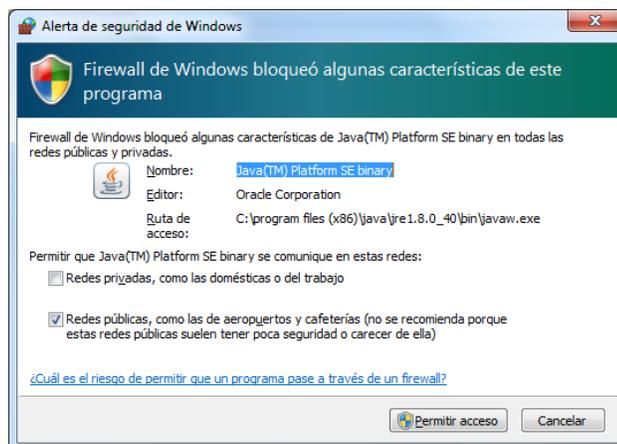


Ingrese el Usuario y Contraseña.

Hay dos usuarios un Administrador y su contraseña es AdminSAD, y el usuario emergencia y la contraseña SAD, los cuales solo deben ser entregados a las personas que están autorizadas a enviar alertas.

Una vez colocado el usuario y la contraseña correcta, aparecerá una alerta de seguridad de Windows.

Presionar Permitir acceso.



Utilizar nuestra aplicación servidor:



Observar que nos muestra una ip y un puerto el cual servirá para que los clientes puedan conectarse.

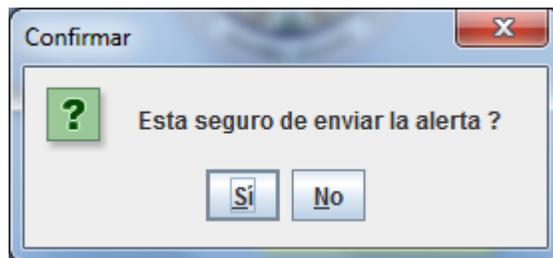
Para el uso de la aplicación seleccionar el tipo de alerta presionando en el botón que se requiera para notar que ha sido seleccionado se encenderá el botón.



En el cuadro de texto escribía toda la información sobre el motivo de la alerta.



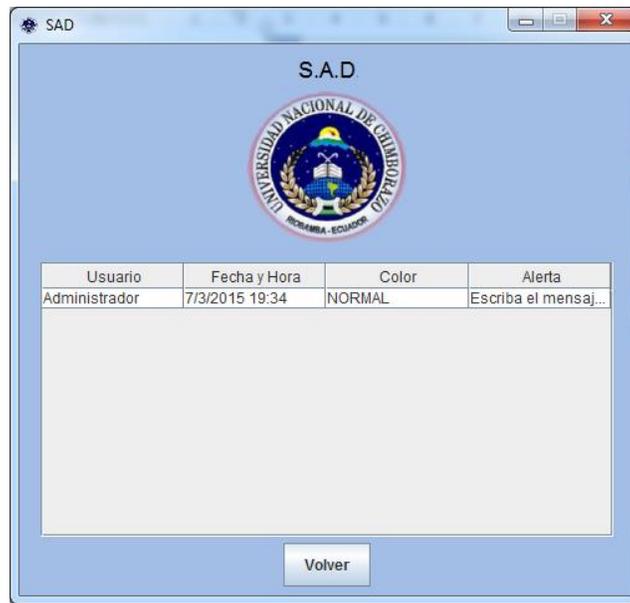
Presionar el botón enviar.



Confirmar si va a enviar la alerta.

Será enviada a todos los clientes que se encuentren conectados a la misma RED y que tengan la aplicación SAD cliente instalado.

Observar las alertas enviadas presione en el botón reporte de alertas



Observar el usuario que ha ingresado, la fecha y hora, el tipo de alerta, y la información que se ha enviado.

## CLIENTE

Para la instalación del cliente necesita que el dispositivo cuente con sistema operativo Android.

Tener habilitada la opción de Fuentes desconocidas en ajustes, seguridad.



Pasar la aplicación SADcliente por bluetooth por cable de datos o retirando la memoria del celular y conectarla al equipo.

Presionar abrir la aplicación y presionar instalar.

Abrir la aplicación.



Escribir la IP del servidor y el Puerto, presionar en conectar.

Una vez conectado se enciende la luz verde.

Ahora dejar en segundo plano la aplicación presionando la tecla home del celular.

En caso de recibir una alerta sonara el sonido de mensaje y notificara en la barra de notificaciones del celular.



Abrir la aplicación desde la barra de notificación del celular o desde el menú de aplicaciones para mostrar la alerta.

Se encenderá un botón con el tipo de alerta y se recibe la información correspondiente.



Recibida la alerta y la información poner en segundo plano la aplicación.

## **ANEXO VI. MANUAL TÉCNICO**

### **1 INTRODUCCION**

El manual Técnico es un documento que provee al usuario especificaciones sobre Propósito, Alcance, Restricciones del Sistema, Paquete Seguridad, Paquete de Fuentes, Interfaz de Usuario que servirá para conocer más sobre el producto SAD Versión 1.0

### **2 METAS**

- El Sistema SAD permitirá a los usuarios administradores acceder al Servidor desde una computadora que tenga JAVA instalado.
- El Sistema SAD permitirá a los usuarios clientes acceder al sistema desde un terminal con sistema operativo Android.
- Para poder acceder al Sistema SAD Servidor, se requiere de un código de usuario válido así como de una contraseña.

### **3 ESPECIFICACIÓN DE REQUERIMIENTO FUNCIONAL**

El sistema dispone de los siguientes requerimientos expuestos a continuación:

#### **3.1 Disponibilidad de software necesario para el Servidor**

La computadora donde se ejecutara el sistema SAD debe poseer las siguientes características mínimas:

Core 2 Duo,  
con al menos 2 GB de RAM y  
500 GB de espacio en el disco, con un S  
istema Operativo Windows 7, o Superior.

#### **3.2 Nivel de profundidad**

El acceso al servidor es de un nivel de profundidad de 1 a 2 clics desde la ventana de acceso de usuario del sistema SAD.

#### **3.3 Menú**

Tiene un Menú de botones de selección en el cual se escoge el tipo de alerta, también el botón para poder acceder a información sobre fechas, horas, tipo de alerta y motivo de la alerta.

### **3.4 Requisitos hardware necesarios**

Se necesitará un computador con acceso a la Red, con unas características mínimas tales como:

- Procesador AMD turion X2 Dual-Core Mobile RM-70. 2.0 GHz.
- 512 MB de RAM o superior
- Tarjeta de audio y video
- Resolución de pantalla 1280 x 800 (32bits) (60Hz)

### **3.5 Acceso**

El acceso al Servidor lo puede realizar personas que se encuentren encargadas del envío de alertas las cuales contarán con el usuario y contraseña.

### **3.6 Idioma Español**

El idioma que se aplica es español ya que está dirigida a los administradores del departamento de gestión de riesgos de la UNACH y es el lenguaje que se maneja en el medio.

### **3.7 Teclado**

Se acopla a los estándares de Windows.

### **3.8 Disponibilidad de software necesario para el Cliente**

El dispositivo móvil donde se ejecutara el sistema debe poseer las siguientes características mínimas: poseer sistema operativo androide 2.2 o superior. Tener activado la opción fuentes desconocidas en ajustes del dispositivo, seguridad, y activar fuentes desconocidas como muestra la imagen.



1: Es el botón de selección para activar Fuentes Desconocidas.

### **3.9 Nivel de profundidad**

El acceso a la aplicación es de un nivel de profundidad de 1 a 2 clics desde la ventana de acceso de usuario del sistema SAD.

### **3.10 Requisitos hardware necesarios**

Se necesitará un dispositivo móvil que soporte el sistema operativo androide.

### **3.11 Acceso**

El acceso a la aplicación lo puede realizar personas que cuenten con la aplicación instalada en su teléfono.

### **3.12 Idioma Español**

El idioma que se aplica es español ya que está dirigida a los estudiantes de la Universidad Nacional de Chimborazo.

## 4 INTERFACES

### 4.1 SERVIDOR



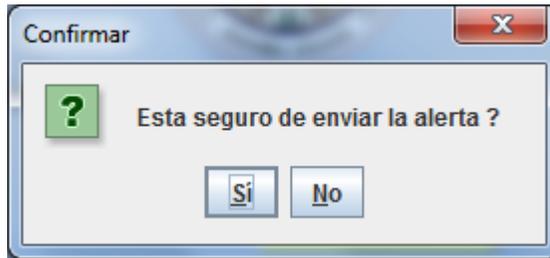
Pantalla de ingreso al sistema, ingreso de usuario y contraseña.



Patalla principal del sistema SAD.



Pantalla selección tipo de alerta amarilla.



Pantalla de confirmacion para enviar la alerta.



Ventana con información de alertas enviadas.

## 4.2 CLIENTE



Ventana de inicio de la aplicación.



Ventana de conexión al servidor



Ventana de recepción y notificación de mensaje del servidor.



Ventana de recepcion de alerta amarilla.

## 5 PRUEBAS FUNCIONALES

<b>FUNCIONES</b>	<b>FUNCIONA</b>	<b>NO FUNCIONA</b>
Disponibilidad de software necesario	SI	
Nivel de profundidad	SI	
Requisitos hardware necesarios	SI	
Idioma Español	SI	
Navegabilidad por teclado	SI	

ANEXO VII. TABLA DE CHI-CUADRADO

g.d.l	0,001	0,005	0,01	0,02	0,025	0,03	0,04	0,05	0,10	0,15	0,20	0,25	0,30	0,35	0,40	g.d.l
1	10,828	7,879	6,035	5,412	5,028	4,709	4,318	3,841	2,766	2,072	1,642	1,323	1,074	0,873	0,708	1
2	13,816	10,597	9,210	7,926	7,370	7,013	6,438	5,901	4,605	3,794	3,219	2,773	2,403	2,100	1,833	2
3	16,266	12,838	11,345	9,937	9,348	8,947	8,311	7,815	6,251	5,217	4,602	4,108	3,665	3,320	2,996	3
4	18,467	14,860	13,277	11,668	11,143	10,712	10,026	9,600	7,779	6,745	5,989	5,385	4,878	4,438	4,045	4
5	20,515	16,750	15,086	13,308	12,833	12,375	11,644	11,070	9,236	8,115	7,200	6,626	6,064	5,573	5,132	5
6	22,458	18,548	16,812	15,033	14,489	14,068	13,198	12,582	10,645	9,446	8,528	7,941	7,321	6,825	6,211	6
7	24,322	20,278	18,475	16,622	16,013	15,509	14,703	14,067	12,017	10,798	9,803	9,207	8,583	7,806	7,283	7
8	26,124	21,955	20,090	18,168	17,535	17,010	16,171	15,507	13,362	12,027	11,020	10,219	9,524	8,909	8,251	8
9	27,877	23,589	21,666	19,679	19,023	18,480	17,608	16,919	14,684	13,288	12,242	11,389	10,656	10,006	9,414	9
10	29,588	25,180	23,209	21,161	20,483	19,922	19,021	18,307	15,987	14,534	13,442	12,549	11,781	11,097	10,472	10
11	31,264	26,737	24,725	22,618	21,920	21,342	20,412	19,675	17,275	15,767	14,631	13,701	12,899	12,184	11,520	11
12	32,908	28,260	26,217	24,054	23,337	22,742	21,795	21,026	18,589	16,989	15,912	14,995	14,011	13,266	12,584	12
13	34,528	29,810	27,688	25,472	24,736	24,125	23,142	22,362	19,812	18,202	16,985	15,984	15,119	14,345	13,636	13
14	36,123	31,319	29,141	26,673	26,119	25,493	24,495	23,685	21,064	19,406	18,151	17,117	16,222	15,421	14,685	14
15	37,697	32,801	30,578	28,259	27,488	26,948	25,816	24,996	22,307	20,603	19,311	18,245	17,322	16,494	15,720	15
16	39,252	34,267	32,030	29,633	28,845	28,191	27,136	26,296	23,542	21,793	20,465	19,269	18,418	17,565	16,780	16
17	40,780	35,710	33,489	30,995	30,101	29,523	28,445	27,587	24,769	22,977	21,615	20,489	19,511	18,633	17,826	17
18	42,291	37,156	34,895	32,346	31,526	30,845	29,745	28,868	25,989	24,155	22,760	21,605	20,601	19,699	18,868	18
19	43,780	38,582	36,191	33,687	32,852	32,158	31,037	30,144	27,204	25,329	23,900	22,718	21,689	20,764	19,910	19
20	45,251	39,997	37,566	35,020	34,170	33,462	32,321	31,419	28,412	26,498	25,038	23,828	22,775	21,826	20,951	20
21	46,707	41,401	38,932	36,343	35,479	34,759	33,597	32,671	29,615	27,662	26,171	24,925	23,858	22,988	21,991	21
22	48,148	42,796	40,289	37,659	36,781	36,049	34,867	33,928	30,813	28,822	27,301	26,039	24,939	23,947	23,021	22
23	49,578	44,181	41,638	38,968	38,076	37,332	36,131	35,172	32,007	29,979	28,429	27,141	26,018	25,006	24,069	23
24	51,179	45,559	42,980	40,270	39,364	38,609	37,389	36,415	33,196	31,132	29,553	28,241	27,096	26,063	25,106	24
25	52,620	46,928	44,314	41,566	40,646	39,880	38,642	37,652	34,362	32,282	30,675	29,339	28,172	27,118	26,143	25
26	54,052	48,290	45,642	42,856	41,923	41,146	39,889	38,885	35,563	33,429	31,795	30,425	29,246	28,173	27,179	26
27	55,476	49,645	46,963	44,140	43,195	42,407	41,132	40,113	36,741	34,574	32,912	31,528	30,319	29,227	28,214	27
28	56,892	50,993	48,278	45,419	44,461	43,662	42,370	41,337	37,916	35,715	34,027	32,620	31,391	30,279	29,249	28
29	58,301	52,336	49,588	46,693	45,722	44,913	43,604	42,557	39,087	36,854	35,139	33,711	32,461	31,321	30,282	29
30	59,703	53,672	50,892	47,962	46,979	46,160	44,834	43,773	40,256	37,990	36,259	34,800	33,530	32,382	31,316	30
31	61,098	55,003	52,191	49,226	48,222	47,402	46,059	44,985	41,422	39,124	37,359	35,887	34,598	33,431	32,349	31
32	62,487	56,328	53,486	50,487	49,480	48,641	47,292	46,194	42,585	40,256	38,666	36,973	35,665	34,680	33,381	32
33	63,870	57,648	54,776	51,743	50,725	49,876	48,500	47,400	43,745	41,386	39,572	38,058	36,731	35,529	34,413	33
34	65,247	58,964	56,061	52,995	51,966	51,107	49,716	48,602	44,903	42,514	40,676	39,141	37,795	36,576	35,444	34
35	66,619	60,275	57,342	54,244	53,203	52,335	50,928	49,802	46,059	43,640	41,770	40,223	38,859	37,623	36,475	35
36	67,982	61,581	58,611	55,493	54,491	53,564	52,151	51,011	47,216	44,888	42,919	41,300	39,916	38,496	37,406	36
37	69,340	62,882	59,874	56,742	55,780	54,793	53,364	52,220	48,463	46,137	44,068	42,421	41,041	39,527	38,337	37
38	70,693	64,179	61,131	58,000	57,069	56,024	54,567	53,483	49,648	47,386	45,347	43,522	41,782	40,658	39,268	38
39	72,042	65,472	62,381	59,259	58,358	57,264	55,742	54,698	50,823	48,645	46,616	44,623	42,523	41,369	40,199	39
40	73,400	66,762	63,621	60,518	59,656	58,508	56,946	55,728	51,895	49,844	47,885	45,616	43,264	42,040	41,222	40
60	89,607	81,952	80,379	84,580	81,288	82,225	80,882	79,082	74,397	71,341	68,972	66,981	65,227	63,628	62,125	60
80	129,839	116,321	112,329	108,069	106,629	105,422	103,459	101,879	96,578	93,106	90,405	88,130	86,120	84,284	82,566	80
100	137,208	128,289	124,116	119,648	118,136	116,869	114,806	113,145	107,565	103,904	101,054	98,650	96,524	94,581	92,761	100
120	140,440	140,169	135,807	131,142	129,561	128,237	126,079	124,342	118,468	114,659	111,667	109,141	106,906	104,862	102,986	120
140	143,617	143,648	138,950	133,918	132,211	130,780	128,447	126,507	120,233	116,062	112,806	110,055	107,616	105,383	103,289	140
160	147,451	148,847	143,840	138,471	136,648	134,118	131,824	129,613	122,827	118,252	114,854	111,884	109,269	106,883	104,604	160

Distribución ji cuadrada - Pág. 1