



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Trabajo Terminal

“Sistema de Administración de Perfiles para Redes MANET”

No. 2014B084

Presentan

Aguilar Zúñiga Daniel Eduardo
Arroyo Zamora José Manuel
Yañez Anaya Uriel Yael.

Directores

M. en C. Chadwick Carreto Arellano
Dr. Mario Eduardo Rivero Ángeles

Diciembre 2015



ESCOM
Escuela Superior de Cómputo



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA



No. de TT: 2014-B084

18-Diciembre-2015

Documento Técnico

“Sistema de Administración de Perfiles para Redes MANET”

Presentan

Aguilar Zúñiga Daniel Eduardo
Arroyo Zamora José Manuel
Yañez Anaya Uriel Yael.

Directores

M. en C. Chadwick Carreto Arellano Dr. Mario Eduardo Rivero Ángeles

Resumen

El presente documento describe el desarrollo de un sistema de transferencia de información y administración de perfiles aplicado en redes móviles basado en nodos (MANET). Con el sistema se busca eliminar la gran dependencia a Internet al momento de poder compartir información, por lo tanto, el sistema tendrá la capacidad de transferir información pertinente al usuario en base a su perfil e identificar a otros usuarios con los cuales pueda obtener o compartir información, todo esto siendo transparente para el cliente.

Palabras clave.

Redes de Computadoras, Redes Ad-Hoc, Dominios de Servicio, Cómputo Móvil.

josemarroyoz7@gmail.com
daniel.aguilar.zuniga@gmail.com
urielya@gmail.com



ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA



DEPARTAMENTO DE FORMACIÓN INTEGRAL E
INTITUCIONAL

COMISIÓN ACADÉMICA DE TRABAJO TERMINAL

México D.F. 13 de enero del 2016

DR. FLAVIO ARTURO SÁNCHEZ GARFIAS
PRESIDENTE DE LA COMISIÓN ACADÉMICA
DE TRABAJO TERMINAL
P R E S E N T E

Por medio del presente, se informa que los alumnos que integran el **TRABAJO TERMINAL: 2014-B086**, titulado "Sistema de Administración de perfiles para redes MANET" concluyeron satisfactoriamente su trabajo.

Los discos (DVDs) fueron revisados ampliamente por sus servidores y corregidos, cubriendo el alcance y el objetivo planteado en el protocolo original y de acuerdo a los requisitos establecidos por la Comisión que Usted preside.

ATENTAMENTE

M. en C. Chadwick Carreto Arellano
Director del Trabajo Terminal

Dr. Mario Eduardo Rivero Angeles
Director del Trabajo Terminal

Advertencia

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.” La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen. Información adicional sobre este reporte técnico podrá obtenerse en: La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000.

Índice

1 Capítulo I	9
1.1 Resumen.....	9
1.2 Palabras clave.....	9
1.3 Introducción.....	9
1.4 Objetivo.....	10
1.5 Justificación.....	10
1.6 Estado del Arte.....	13
1.6.1 Protocolos Proactivos.	14
1.6.2 Protocolos Reactivos.	14
1.6.3 Protocolos de Enrutamiento Híbridos.....	15
1.6.4 Protocolos-Clúster.	16
1.6.5 Protocolos de Nodos Centrada.	16
1.6.6 Protocolos MANET de Alto Nivel.....	16
1.6.7 Diferentes Tipos de Redes.....	16
1.7 - Problemática	20
1.8 - Soluciones Existentes	21
1.8.1 Implementación del protocolo HLMP en Android. ^[7]	21
1.8.2 Servel Mesh (Aplicación). ^[8]	22
1.8.3 FireChat(Aplicación) ^[9]	23
2 Capítulo II	25
2.1 Propuesta de Solución.	25
2.1.1 Hardware.	25
2.1.2 Software.....	25
2.2 Arquitectura del Sistema	26
2.2.1 Módulo de Interoperabilidad.....	27
2.2.2 Módulo de Red.....	27
2.2.3 Módulo de Comunicación.....	27
2.2.4 Características	28
2.2.5 Ventajas.....	28
2.2.6 Desventajas.....	29

.....	30
2.3 Estudio de Viabilidad	30
2.3.1 Viabilidad técnica ^[15]	30
2.3.2 Viabilidad económica	32
2.3.3 Viabilidad Legal.....	36
2.3.4 Ley de Transparencia y Libre Acceso a la Información.	37
2.3.5 Ley de Registro y Derechos de Autor.....	39
2.4 Metodología	40
2.4.1 Planeación	42
3 Capítulo III.	43
3.1 Análisis y Diseño.	43
3.2 Estructura Básica de Android	43
3.3 Diseño de HLMP para Android	43
3.4 Introducción.....	44
3.5 Requerimientos no funcionales y Restricciones.....	44
3.5.1 Desempeño.	44
3.5.2 Disponibilidad.	44
3.5.3 Escalabilidad.....	45
3.5.4 Facilidad de Uso e Ingreso de Información.	45
3.5.5 Facilidad para las Pruebas.....	45
3.5.6 Flexibilidad.....	45
3.6 Requerimientos funcionales	50
3.7 Casos de uso	53
3.7.1 Descripción de casos de uso	53
3.8 Requerimientos de Sistema.....	73
3.8.1 Diagrama de Casos de uso.....	75
3.9 Diagrama de Secuencia	78
3.9.1 Interacción general entre nodos.....	80
3.9.2 Registro	80
3.9.3 Agregar Grupo	82
3.9.4 Invitación a Grupo.....	83

3.9.5 Contestar Invitación	84
3.9.6 Administrar información	84
3.9.7 Actualizar nodos	86
.....	87
3.9.8 Notificar	87
3.9.9 Gestionar usuarios.....	88
3.9.10 Salir	89
3.10 Diagrama de Clases.....	90
3.10.1 Descripción de Clases:	90
3.11 Diagrama de Actividades.....	96
3.11.1 Registro de Usuario	96
3.11.2 Agregar Grupo	96
3.11.3 Identificar nodos cercanos.	97
3.11.4 Mandar solicitud de conexión a nodo receptor.	98
3.11.5 Seleccionar Invitaciones a Grupo.....	99
.....	99
3.11.6 Ingresar a Gestor de Información.....	101
.....	101
3.12 Diagrama de Estados	101
3.12.1 Registro al Sistema.....	101
3.12.2 Agregar Grupo.	102
3.13 Administrador	104
.....	109
3.14 Diagramas de Colaboración.	111
3.14.1 Agregar Grupo.	111
3.14.2 Baja Usuario	113
3.14.3 Alta de Usuario	114
3.14.4 Acceso al Sistema.	115
3.14.5 Desconexión del Sistema.....	116
3.14.6 Consulta Usuario.	117
3.14.7 Invitación a Grupo.....	117

3.14.8 Registro de Usuario.	119
3.14.9 Notificar.....	120
3.14.10 Interacción entre nodos.....	121
3.14.11 Actualizar nodos.	123
3.14.12 Administrar información	124
3.14.13 Contestar información	125
3.15 Diagrama Relacional de Base de Datos.	127
3.16 Diagrama Arquitectura final	127
3.16.1 Descripción del Diagrama de Arquitectura Final.....	128
7. Bibliografía.	147

Lista de Tablas

Tabla 1: Principales Protocolos Proactivos y Reactivos

Tabla 2: Ventajas y Desventajas de una red Manet

Tabla 3: Comparación de Soluciones existentes

Tabla 4: Características del sistema

Tabla 5: Ventajas del sistema

Tabla 6: Desventajas del sistema

Tabla 7: Recursos Humanos

Tabla 8: Recursos Tecnológicos

Tabla 9: Recursos Materiales

Tabla 10: Flujo de Pago

Tabla 11: Roles SCRUM

Tabla 12: Requerimientos no funcionales

Tabla 13: Requerimientos funcionales

Tabla 14: Descripción CU1

Tabla 15: Descripción CU2

Tabla 16: Descripción CU3

Tabla 17: Descripción CU4

Tabla 18: Descripción CU5

Tabla 19: Descripción CU6

Tabla 20: Descripción CU7

Tabla 21: Descripción CU8

Tabla 22: Descripción CU9

Tabla 23: Descripción CU10

Tabla 24: Descripción CU11

Tabla 25: Descripción CU12

Tabla 26: Roles del Sistema

Tabla 27: Métodos Diagrama de Secuencia

1 Capítulo I

1.1 Resumen.

El presente documento describe el desarrollo de un sistema de transferencia de información y administración de perfiles aplicado en redes móviles basado en nodos (MANET). Con el sistema se busca eliminar la gran dependencia a Internet al momento de poder compartir información, por lo tanto, el sistema tendrá la capacidad de transferir información pertinente al usuario en base a su perfil e identificar a otros usuarios con los cuales pueda obtener o compartir información, todo esto siendo transparente para el cliente.

Es bien sabido que al pensar en dispositivos móviles, tenemos como idea inherente a las redes WiFi para obtener acceso a Internet y de esta forma consultar la información que requerimos. Sin embargo, existe otro tipo de conexión cuya característica radica en comunicar una red de dispositivos sin necesidad de un punto centralizado de acceso.

Es así como nace High Level MANET Protocol (HLMP), un protocolo que tiene como objetivo proveer comunicación entre estos dispositivos móviles, generando una red interconectada que utiliza los mismos equipos disponibles en la red como antenas receptoras y emisoras. Cabe destacar que este sistema se implementará en ambientes con sistema operativo Android.

1.2 Palabras clave.

Redes de Computadoras, Redes Ad-Hoc, Dominios de Servicio, Cómputo Móvil.

1.3 Introducción

La tecnología móvil ha abierto la puerta a un mundo de posibilidades que ha permitido que hoy en día todos podamos conectarnos sin importar donde nos encontremos – de camino al autobús, de compras en una tienda, etc. y no sólo podamos escucharnos, si no también vernos, gracias a teléfonos móviles, tabletas, relojes e incluso gafas. [1]

Además, con la computación ubicua, todas las personas pueden acceder a servicios de información adecuados sea cual sea la situación en la que se encuentren.

Con esto, se pretende desaparecer a los dispositivos computacionales haciéndolos situarse en segundo plano para así crear dispositivos que se mezclen en la cotidianidad hasta que lleguen a ser indistinguibles.

Un factor determinante para el auge de los dispositivos móviles es la sencillez con la que podemos acceder a diversas aplicaciones en cuestión de segundos, sin embargo, la mayoría de estas dependen en gran medida de tener una conexión a internet. Con este sistema buscamos eliminar en cierta forma la necesidad de acceder a la red para transferir información.

Este sistema en un futuro, podrá ser implementado en hospitales, escuelas, oficinas, etc. pretendiendo dar solución de forma rápida y sencilla a tareas que pueden ir desde la transferencia de un plan de estudios, presentación en una junta de negocios, recibir una notificación para atender una emergencia en un hospital e inclusive ayudar a salvar vidas en una catástrofe.

1.4 Objetivo

Analizar, diseñar y desarrollar un sistema el cual permite administrar de manera eficiente la información de los usuarios según su perfil y poder transferir esta de manera automática en una red MANET simulada.

1.5 Justificación

Las redes inalámbricas convencionales, convencionales en el sentido de que como requisitos previos por ejemplo las redes 2G o 3G las cuales han alcanzado un grado de penetración alta y las redes LAN (Red de Área local), requieren una infraestructura de red fija con la administración centralizada para su funcionamiento, lo

que potencialmente consume una gran cantidad de tiempo y dinero para la configuración y el mantenimiento.

Además, un número cada vez mayor de dispositivos tales como ordenadores portátiles, asistentes digitales personales (PDA), ordenadores de bolsillo, tablet PCs, teléfonos inteligentes, reproductores de MP3, cámaras digitales, etc. se proporcionan con interfaces inalámbricas de corto alcance.

Por consiguiente, estos dispositivos son cada vez más pequeños, más baratos, más fáciles de usar y más potentes. Esta evolución está impulsando una nueva alternativa para la comunicación móvil, en el que los dispositivos móviles de auto-creación, las redes inalámbricas de auto-administración y de autoorganización, se llaman red móvil ad hoc.

El incremento y diversificación del uso de dispositivos móviles ha fortalecido la experiencia del usuario mexicano con las plataformas digitales. De acuerdo a la 2ª edición del "Estudio de Usos y Hábitos de Dispositivos Móviles en México", realizada por IAB México y Millward Brown, 84% de los mexicanos mayores de 13 años cuentan con algún dispositivo móvil y 6 de cada 10 navegan en internet de forma activa a través de ellos.

De acuerdo con el estudio, la penetración de smartphones en 2013 es de 39%, incrementando 17 puntos con respecto al año anterior (22% en 2012), mientras que la posesión de teléfonos celulares -no smartphones- pasó de 87% a 78%, lo cual indica una migración de dispositivos básicos a otros más avanzados. Además, la posesión de tablets pasó de 6% en 2012 a 24% en 2013.

Para los usuarios en México, es esencial tener siempre una disponibilidad en sus dispositivos móviles, un 87% dice que casi nunca salen de su casa sin ellos y un 44% declara que no les gusta estar incomunicados, por lo tanto procuran salir de su casa siempre con sus dispositivos, es decir, tenerlos al alcance en todo momento, otro 60% declara nunca apagar sus dispositivos móviles.

La población mexicana averigua cada vez más las ventajas al tener un dispositivo móvil a su alcance en todo momento, y esas ventajas se pueden resumir en cuatro, las cuales son, medios de comunicación, dentro del trabajo, conexión a Internet para entretenimiento y la comodidad, los respectivos porcentajes se dividen en un 72%,41%, 34% y 25%.

En el estudio destacan que los mexicanos realizan ocho actividades distintas con su dispositivo móvil en promedio. Cerca del 88% los utiliza para actividades básicas a comparación con el 91% del 2012, como comunicarse por teléfono o el envío de mensajes de texto. Las actividades que se consideran lúdicas como escuchar música, los juegos o tomar fotografías son realizadas por el 79% de los usuarios a comparación con el 70% en 2012 mientras que otras actividades que tienen relación con el internet como la redes sociales, correo electrónico, tienda en línea, entre otros, se incrementaron significativamente gracias a los usuarios ya que pasó de un 38% del 2012 a un 61%.

Más de la mitad de personas que utilizan sus medios móviles son para actividades lúdicas o hacer actividades que tengan relación con Internet

En base a todos los datos presentados, podemos deducir que la migración a dispositivos móviles continúa en ascenso y a pesar de ofrecernos una capacidad ilimitada de funciones que nos facilitan la vida, aún requieren que las tareas sean dictadas por los usuarios.

Con este sistema pretendemos hacer automática la transferencia de información a usuarios que comparten una red en común; dependiendo del perfil que se tenga y la cercanía a otros dispositivos.

El sistema usará algoritmos de Inteligencia Artificial al momento de brindar los servicios para que los usuarios tengan la información indicada en el momento adecuado y no haya información repetida en su dispositivo. Cabe destacar, el dispositivo emisor será capaz de seleccionar el o los archivos que se transmitirán así como los usuarios que los recibirán.

Si bien es cierto que acceder a internet es un recurso indispensable para las actividades cotidianas en estos días, también es cierto que este servicio no siempre está disponible al alcance de todos o en cualquier lugar o momento. Transacciones de archivos entre equipos de trabajo, compañeros de oficina, etc. depende en gran medida de Internet. Con este sistema se pretende eliminar esta dependencia puesto que la información será transmitida de un dispositivo a otros en cuestión de segundos y de manera transparente para los usuarios.

1.6 Estado del Arte.

El periodo de indagación que se ha llevado a cabo se ha delimitado a información desde el 2005 al 2014.

Hoy en día las redes inalámbricas están a la par de popularidad que los dispositivos móviles, esto se debe a la facilidad con la que cualquier usuario puede acceder a la información en cualquier momento. Generalmente, las redes inalámbricas que se usan hoy en día necesitan una arquitectura y gestión que sean fijos y centralizados, respectivamente. En cambio las redes MANET que se tratan aquí no es necesario que cuenten con una arquitectura fija, ni tampoco de los elementos o gestión centralizada con que contaría una red inalámbrica. Como punto de partida para hacer la investigación de las redes AD Hoc se puede comenzar a partir del año de 1995, es decir, en cierta conferencia de la IETF (Internet Engineering Task Force). En estos primerizos debates su tema central y principal fueron las redes inalámbricas satelitales, redes gubernamentales y redes auto-configurables.

El tema central fue la adaptación de los protocolos que existen para trabajar en Internet junto con las redes inalámbricas y con un dinamismo muy alto. Después cerca del año de 1997, se formó un grupo de trabajo que se centró en las MANET's de la IETF, el cual fue llamado, el MANET-WG, por sus siglas en inglés, que significa MANET-WORKGROUP. Este grupo de trabajo tenía varias tareas principales las cuales una o varias de ellas eran la especificación de interfaces y protocolos que soportarán el Internet que estaba basado en IP sobre las MANET. Pero la mayoría de las investigaciones y adelantos que se hacían sobre estos tópicos son gracias a este grupo de desarrollo o trabajo, y otra de sus tareas es que actúa como organismo principal para normalizar.

Otros grupos fueron creados, uno de ellos fue el que se creó en Japón alrededor del año 2003, y se llama Ad Hoc Consortium, cuyo objetivo es la investigación de unir los intereses y los esfuerzos de la industria, así como del mundo académico y gubernamental para el uso de la tecnología y conocimiento en la redes Ad Hoc en cualquier tipo de aplicación.

En el ámbito de los tipos de protocolos existentes para crear redes MANETs se pueden diferenciar tres grandes grupos: protocolos proactivos, reactivos e híbridos. En el primer grupo, los nodos mantienen información de encaminamiento hacia todos los dispositivos de la red. En el segundo grupo, los nodos actualizan las tablas de

encaminamiento solamente en caso de necesidad. Por último, el tercer grupo intenta incorporar las ventajas de los protocolos proactivos y reactivos^[2].

1.6.1 Protocolos Proactivos.

Dentro de este grupo se encuadran aquellos protocolos que basan su funcionamiento en el intercambio periódico de mensajes para actualizar sus tablas de rutas. La característica principal es que, aunque no haya movilidad entre los dispositivos que componen la red o no cambie la topología, siempre va a haber un número mínimo de mensajes mandados para mantener la conexión y tener constancia de los nodos que forman la red^[16].

Se actualiza constantemente información de la topología de la red y se asegura de que esté disponible para todos los nodos. Reducen la latencia de red (o tiempo de retardo del sistema), pero los gastos generales de datos aumentan mediante la constante actualización de la información de enrutamiento.

En general, la ventaja de los protocolos proactivos radica en la constante actualización de las tablas de enrutamiento, sin embargo la desventaja es el costo de ancho de banda debido a la alta carga de transmisión de información.

A continuación se describen algunos ejemplos significativos de estos protocolos proactivos: DSDV, WRP y OLSR.

1.6.2 Protocolos Reactivos.

Dentro de este grupo, se encuadran aquellos protocolos que basan su funcionamiento en la actualización de las rutas solamente cuando sean necesarias. La característica principal es que, si hay movilidad o cambio en la topología, los dispositivos no actualizan sus tablas de encaminamiento a menos que lo requieran. De hecho, no se conoce cómo es la red completa porque no se sabe cuáles son las tablas de los demás nodos, por lo que únicamente hay disponible información sobre las rutas activas^[16].

Son los cuales usan algoritmos de enrutamiento que crean rutas solamente cuando un nodo de origen desea mandar información hacia algún nodo destino. También denominados bajo demanda, estos tienen mayores retardos de

descubrimientos de ruta pero usan más eficientemente el ancho de banda que los protocolos proactivos.

Algunos ejemplos representativos son los protocolos AODV, DSR y ABR, que se explican a continuación.

Se pueden clasificar de la siguiente manera:

- Basados en la fuente: En la cual en la cabecera se transporta la ruta completa de la fuente al destino.
- Punto a punto: Solamente en la cabecera de paquetes de datos se encuentra la dirección destino y la dirección del próximo salto.

La siguiente tabla comparativa muestra los diversos protocolos empleados para redes Ad-Hoc así como la categoría a la que pertenecen:

Proactivos	DSDV, OLSR, CGSR, WRP, TBRPF, B.A.T.M.A.N (En desarrollo)
Reactivos	AODV, LMR, TORA DSR, LQSR

Tabla 1: Principales Protocolos Proactivos y Reactivos

1.6.3 Protocolos de Enrutamiento Híbridos.

Tomando lo anterior, los protocolos de enrutamiento híbridos basados en MANET integran los protocolos anteriores los cuales son proactivos y reactivos, el resultado que se consigue es un mayor rendimiento en sus componentes y se puede ajustar de forma dinámica a condiciones diferentes de la red. Se combinan las ventajas de los protocolos que integran a un híbrido. Falta decir que son de peso ligero, simple y diseñado para evitar una carga muy pesada de control. Los protocolos híbridos se pueden clasificar en nodo-céntrico y cluster-centrica.

Una red como cualquier otra se puede organizar en grupos y esto a su vez puede que en subgrupos dentro una red de clúster-céntrica. De esta manera al agrupar se optimiza el uso de los recursos y el tamaño de las tablas de enrutamiento tenderá a reducirse.

1.6.4 Protocolos-Clúster.

Se centra en diferentes reglas para el envío y recibimiento de datos dentro del clúster y envío y recibimiento dentro del mismo, es decir, que se comunique con otros clúster.

1.6.5 Protocolos de Nodos Centrada.

Son protocolos que por su simplicidad se incorporan al emisor y receptor de la información, dentro de estos se incluyen los protocolos como el de enrutamiento de ojo de pez de la zona, así como el de dos zonas.

1.6.6 Protocolos MANET de Alto Nivel.

Este tipo de protocolo el cual es HLMP, consiste en la automatización de procesos que se ven involucrados cuando se establecen conexiones inalámbricas entre dispositivos móviles, estos pueden hacer tráfico de mensajes a través de las redes móviles.

Tiene varias funciones entre las cuales es la transmisión de un único datagrama de red, el cual consta de dos partes, el encabezado y el cuerpo, estos se explican enseguida:

- Encabezado.- Indica el tamaño de los datos internos.
- Cuerpo.- Indica el mensaje de comunicación ya sea para enviar o recibir.

1.6.7 Diferentes Tipos de Redes.

1.6.7.1 Red de Infraestructura.

Esta es una red basada en cliente-servidor, donde las computadoras personales son los clientes personales que se comunican con el servidor, el cual es el punto de acceso.

Para que sea posible una comunicación entre dos computadoras personales es necesario que estén conectadas ambas al mismo punto de acceso,, cuando un cliente cualquiera quiera entablar comunicación con otro cliente, el mensaje enviado es para el Punto de Acceso, y finalmente este lo envía al destinatario, es pocas palabras, es un sistema central.

1.6.7.2 Red Manet.

Una red móvil ad hoc es una red formada sin ninguna administración central o no hay un nodo central, sino que consta de nodos móviles que usan una interfaz inalámbrica para enviar paquetes de datos. Los ordenadores están en igualdad de condiciones.

Según su aplicación, las redes ad hoc pueden ser clasificadas de la siguiente manera ^[3]:

➤ **Red Mesh.**- Este tipo de redes que comúnmente son malladas, es decir son acopladas, también se les puede llamar de malla inalámbrica, y de ese tipo de arquitectura son, en pocas palabras, son las redes que mezclan las dos topologías de redes inalámbricas, la AD-HOC y la Infraestructura. A nivel básico son redes con topología de infraestructura pero que pueden unirse a la red por medio de dispositivos, aunque estos no tengan al alcance los puntos de acceso debido a que no están en una rango de cobertura, estarán dentro del rango de alguna tarjeta que indirecta o directa está en el rango de al menos un punto de acceso.

➤ **Red de sensores.**- Esta red se compone de pequeños nodos los cuales están equipado con sensores, y estos a su vez ayudan con la tarea común para la que fueron encomendados. Este tipo de redes están hechas por cierto conjunto de sensores con capacidades sensitivas y a la vez con una comunicación que es inalámbrica, con los cuales a estas redes pueden hacer redes ad hoc sin una arquitectura física que ya esté establecida ni con la e central mencionada anteriormente.

➤ **Red Vehicular (VANET).**- Vehicular Ad-Hoc Network es una red en la cual sus nodos están conformados por automóviles por tal motivo es un tipo

específico de redes MANET, funciona a base de sensores para mantener la conectividad, incluso

En la imagen siguiente se muestran las redes de Infraestructura y Ad-hoc así como la forma en la que operan.

Imagen 1: Redes de infraestructura y ad-hoc



Como se puede ver en la imagen anterior, una de las grandes ventajas de la red Manet que se ilustra es que ya no hay dispositivos que centralizan la red, es decir, todos intercambian información entre sí, en pocas palabras, cada dispositivo estaría actuando como un ente centralizado de envío de información ^[4].

A continuación se muestra una tabla con ventajas y desventajas para la red Manet ^[5].

Ventajas	Desventajas
Las redes pueden establecerse en cualquier lugar y momento.	Los recursos limitados y la seguridad física limitada.
Independencia de la administración central de la red.	La falta de instalaciones de autorización.

Auto-configuración, los nodos son también routers.	Topología de la red volátil hace que sea difícil de detectar nodos maliciosos.
Auto-sanación a través de la re-configuración continua.	Los protocolos de seguridad para las redes de cable no pueden trabajar para redes ad hoc.
Facilitar el acceso a información y servicios independientemente de la posición geográfica.	No se puede contar con la infraestructura.
Las personas que pueden acceder a Internet desde muchos lugares diferentes.	Cada nodo debe tener pleno rendimiento.
Redes funcionan sin ninguna infraestructura preexistente.	El rendimiento se ve afectado por la carga del sistema.
	Fiabilidad requiere un número suficiente de nodos disponibles.
	Redes dispersas pueden tener problemas.
	Las grandes redes pueden tener latencia excesiva, que afecta a algunas aplicaciones.

Tabla 2: Ventajas y Desventajas de una red Manet

Problemas importantes

Los factores críticos que afectan el rendimiento de una red ad hoc.

- **Número de nodos en la red:** Si se incrementa el número de nodos, se vuelve más compleja la tarea de administración de la red, actualmente las redes ad hoc eliminan este problema con un número muy limitado de nodos para una red de área local.
- **Tamaño de la red y la densidad:** Si la red incrementa de tamaño los nodos tienen más carga de trabajo como router, ya que una de esas tareas sería el incrementar la tabla de enrutamiento para las rutas, previamente inundando la red para la búsqueda de la ruta más óptima.
- **Patrón de tráfico y de carga:** Al tener más carga de datos y/o paquetes que enviarse entre nodo y nodo, los algoritmos de enrutamiento convencionales usadas en las redes tradicionales son insuficientes para una red que cuenta con dinamismo y movilidad.
- **Enlaces unidireccionales:** Los nodos nunca estarán en una sola dirección, se deben mover cambiando constantemente sus rangos de alcance, así como sus nodos vecinos con los cuales se comunican.
- **Limitaciones terrestres:** Las redes ad hoc están pensadas solo para áreas locales, por lo que su implementación en una área más grande, se volvería un trabajo más arduo.
- **El equilibrio de carga:** Al momento de actualizar la tabla en un protocolo proactivo o al encontrar una ruta en un protocolo reactivo, se ocupa ancho de banda, la cual es necesaria equilibrar con la información que es enviada entre los nodos que forman la red.
- **Modelo de movilidad y la velocidad:** Debido a la movilidad de los nodos, constantemente se cambiará la carga en los enlaces de los nodos afectando la velocidad dependiendo de la aplicación a la que se adapte la red ad hoc

1.7 - Problemática

El problema a resolver es el ámbito en la implementación de Redes Manet en dispositivos móviles, aunque hay algunos trabajos previos dentro de esta área donde existe la transferencia de archivos, pero no de manera automática que es lo que se busca en este trabajo, sino también definir una solución reutilizable tomando como base aspectos significativos de los trabajos que se tienen en la actualidad.

Los trabajos previos también carecen de una administración en base a un perfil de cada usuario, lo cual es muy importante dentro de un sistema para que los usuarios sólo tengan acceso a la información que corresponde a su perfil dependiendo de los diferentes permisos concedidos.

Uno de los puntos a resolver es la implementación de un protocolo similar a HLMP, para el sistema operativo Android con el cual se va a trabajar.

Otro de los puntos a tratar son las diferentes versiones del sistema operativo Android, ya que se tendrá que investigar que sus diferentes versiones no afecten el funcionamiento del sistema dentro de una red AD-HOC que se quiere desarrollar.

También se tiene que ver que dicho protocolo de comunicación sea soportada por los diferentes dispositivos móviles con sistema operativo Android, teniendo en cuenta temas de ruteo de mensajes y de comunicación en red para dispositivos móviles, así mismo dentro de la misma Red AD-HOC y el propio sistema a desarrollar.

Finalmente lo que se debe hacer es poder comunicar mediante un protocolo similar a HLMP dos Smartphones con sistema operativo Android para una comunicación directa entre los dispositivos.

1.8 - Soluciones Existentes

En la investigación realizada se encontraron diversos sistemas ^{[6][5]} que trabajan bajo redes Ad-Hoc, sin embargo para intereses del presente trabajo, se han destacado tres de ellos, los cuales son:

1.8.1 Implementación del protocolo HLMP en Android. ^[7]

Este Sistema se desarrolló en el Departamento de Ciencias de la Computación de la Universidad de Chile en 2012. Dicho sistema consiste en el desarrollo del protocolo HLMP (High Level Manet Protocol) enfocado a Smartphones con sistema operativo Android y tiene como objetivo proveer comunicación entre estos dispositivos móviles, generando una red interconectada que utiliza los mismos equipos disponibles en la red como antenas receptoras y emisoras con la finalidad de transferir archivos aunque no se cubre esto último.

Principales Características:

- 1) Se logró la conexión entre dos dispositivos, hasta tres dispositivos utilizando un intermediario
- 2) Conexión Ad-Hoc automatizada, el usuario sólo deberá configurar su nombre de usuario
- 3) Utiliza como base el sistema Operativo Android.
- 4) Se trató la transferencia de archivos, pero no se logró.
- 5) Se logra la interoperabilidad con otros sistemas operativos como Windows.
- 6) Logra utilizar software de terceros para auxiliarse en ciertas funciones.
- 7) Deja una importante base para poder implementar trabajos futuros como seguridad en el envío de datos.

1.8.2 Serval Mesh (Aplicación). [8]

Es una aplicación desarrollada por Serval Mesh con ayuda de la Cruz Roja de Nueva Zelanda en el año 2014 para solucionar la falla de redes celulares durante algún desastre cuando estas se encuentran bloqueadas. Permite comunicar a los celulares inteligentes aun cuando hay alguna falla catastrófica en la red celular, esto gracias a que usa comunicaciones Mesh las cuales conectan a los celulares a una misma red mediante el Wi-Fi y así permite llamadas celulares privadas, mandar mensajes de textos seguros y compartir archivos en diferentes ambientes.

Principales Características:

- 1) Se logró la conexión entre dos dispositivos o más.
- 2) Cada dispositivo móvil se convierte en un servidor, incluso sin una red de celulares.
- 3) Se logró la transferencia de archivos.
- 4) Se logró emplear mensajes de texto desde lugares remotos

5) Se logró la interoperabilidad con Android.

1.8.3 FireChat(Aplicación)^[9]

Aplicación creada por Open Garden Inc actualizada por última vez en el año 2015, fue lanzada primeramente para enfrentar a la amenaza de restricciones de Internet y de cobertura celular. Esta aplicación promete mantener al usuario conectado dondequiera que se encuentre, todo esto gracias a la tecnología de las redes Mesh las cuales permiten tener a sus múltiples nodos conectados a Internet aun siendo uno sólo el que logre estar conectado.^[10]

Principales características:

- 1) Se logró la conexión entre dos dispositivos o más.
- 2) Cada dispositivo móvil se convierte en un servidor, incluso sin una red de celulares.
- 3) Se logró la transferencia de archivos.
- 4) Se logró emplear mensajes de texto desde lugares remotos
- 5) Se logró la interoperabilidad con diferentes Sistemas Operativos, los cuales incluyen IOs y Android.

En base a la siguiente tabla comparativa se considera que el sistema de administración de perfiles destaca de los anteriores debido a que será un servicio que no se encuentra disponible en el país e inclusive en América Latina, si bien por el momento no se pretende alcanzar interoperabilidad entre sistemas operativos, este sistema ofrece un gran avance que es la transferencia de información, además de, hacerlo de forma automática con sólo detectar el tipo de perfil a quien se transmitirá.

	Implementación del Protocolo HLMP	Serval Mesh	FireChat
Administración en base a Perfil	X	X	X
Conexión entre dos o más	✓	✓	✓

dispositivos			
Se utiliza el sistema Operativo Android	✓	✓	✓
Se logra interoperabilidad con otros Sistemas Operativos	✓	X	✓
Utiliza software de terceros para auxiliarse	✓	X	X
Transferencia de información	X	✓	✓
Importante Base para el futuro	✓	✓	✓
Sistema Mexicano	X	X	X
Transferencia Automática de Información	X	X	X

Tabla 3: Comparación de Soluciones existentes

2 Capítulo II

2.1 Propuesta de Solución.

La solución requiere especificar dos aspectos: hardware y software. A continuación se describe cada una de estas y además se comenta acerca de los requisitos, factibilidad y otros temas relacionados sobre la evaluación previa al desarrollo.

2.1.1 Hardware.

Acerca de requisitos en cuanto a Hardware, se nombran los siguientes:

- Procesador 1GHz
- 1 Gb de Ram
- 2 Gb de memoria interna

2.1.2 Software

Para poder dar solución a la problemática antes mencionada, se propone desarrollar una aplicación la cual está desarrollada en el lenguaje de programación Android, este sistema lo que busca es poder conectar dos o más usuarios y que pueda haber comunicación y envío de información entre estos.

Con este sistema se busca que haya una clasificación de información mediante un perfil, el cual ayudará a tener contacto con los demás usuarios. Con todo esto, el sistema por sí mismo, mantendrá actualizada la información del usuario de una manera transparente para el mismo.

Como requisito para poder realizar esta solución se deberá tener:

- Android 4.0 en adelante
- Acceso a Fotos/Medios/Archivos
- Acceso completo a la red
- Acceso a Identidad y Contactos
- Ver conexiones de red

- Ver conexiones Wi-Fi

2.2 Arquitectura del Sistema

El sistema cuenta con una arquitectura la cual facilitará la comunicación entre todos los dispositivos o nodos que posean instalada la aplicación, en este se observa cómo habrá interacción entre cada uno de estos siempre y cuando tengan el mismo perfil así como también se logra destacar que un nodo puede tener uno o más perfiles y así mismo podría obtener información de otro medio, el cual, puede ser una base de datos, internet o algún otro repositorio de información para así, al final, compartir todo esto con los demás dispositivos.

Se tendrá en cuenta un router como punto de acceso, desde donde los nodos móviles, es decir, los diferentes dispositivos podrán acceder a la información de los nodos disponibles en la red, de esta manera se simulara una red MANET con todas sus características anteriormente señaladas.

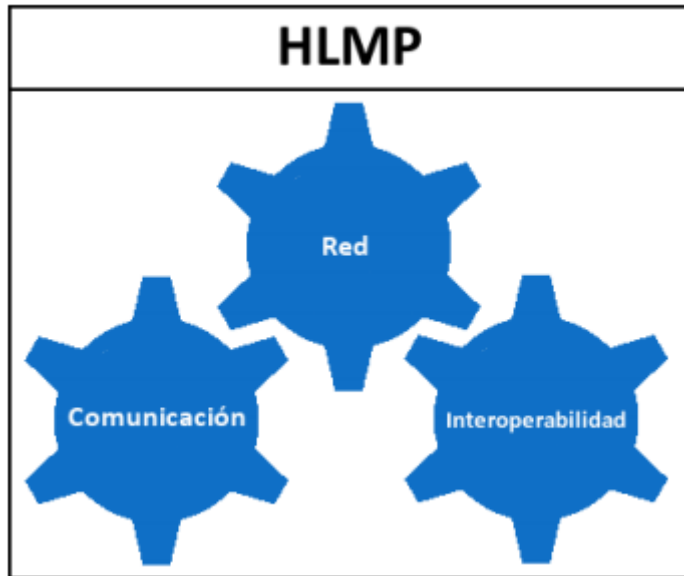
Este desarrollo se constituye de dos componentes de software, tomando como referencia la biblioteca HLMP para ser usada en Android y la otra es una aplicación que hace uso de esta biblioteca que permite reconocer a los dispositivos de la red, transferir archivos. Para el caso de la biblioteca HLMP.

Este se compone de tres módulos:

- 1. Comunicación.**
- 2. Red.**
- 3. Interoperabilidad.**

Se puede observar a continuación en la imagen cómo interactúan los anteriores módulos mencionados.

Imagen 2: Interacción entre módulos



2.2.1 Módulo de Interoperabilidad.

Esta capa está a cargo de procedimientos que son delegados al sistema operativo que tienen relación con el proceso de conexión a la MANET. Algunos de ellos son: configuración del perfil de WLAN Ad-Hoc, control del dispositivo de red inalámbrico, configuración de dirección IP y sub-máscara de red, lectura de notificaciones del sistema operativo, entre otros.

2.2.2 Módulo de Red.

Esta componente implementa los servicios TCP y UDP, necesarios para el intercambio de datos en la MANET. Al momento de ser recibidos, los mensajes son validados y encolados, para ser luego atendidos por la capa de Comunicación. Esta capa tiene también la responsabilidad de administrar las conexiones TCP, con dispositivos remotos que se encuentran dentro de la vecindad del usuario.

2.2.3 Módulo de Comunicación.

Este módulo implementa la API o interfaz que los desarrolladores pueden usar para dar soporte a la comunicación en aplicaciones colaborativas móviles. Administra

también los servicios de organización de archivos y empaquetamiento de datos, entre otros.

2.2.4 Características

El sistema tendrá todas las características que debe tener todo sistema de información, las cuales son ^[11]:

# Característica	Descripción
1	Flexibilidad la cual facilitará cualquier modificación o actualización.
2	Robustez lo cual hará que no haya actualizaciones periódicamente, ya que cumplirá con todos los requerimientos necesarios para soportar en el mercado.
3	Claridad lo que quiere decir que tendrá todos los términos bien definidos para su futura lectura y análisis.

Tabla 4: Características del sistema

Además contará con otras características, las cuales se definirán como ventajas o desventajas comparadas con las características de los sistemas mencionados en el capítulo anterior.

2.2.5 Ventajas.

# Ventaja	Descripción
1	Se puede establecer en cualquier lugar y momento una conexión

2	Estará soportado para el sistema operativo Android 4.0 en adelante.
3	El sistema contará con una administración de la información de cada usuario en base a su(s) perfil(es).
4	Se podrá comunicar hasta con “n” cantidad de usuarios en un momento determinado.
5	Tendrá transferencia automática de información a los nodos que tengan el mismo perfil que el nodo emisor.

Tabla 5: Ventajas del sistema

2.2.6 Desventajas.

# Desventaja	Descripción
1	Tendrá un número limitado de dispositivos móviles que se podrán conectar.
2	No tendrá interoperabilidad entre sistemas operativos en las primeras versiones del mismo.
3	No tendrá un punto de acceso para salida a Internet, ya que se simulara la red MANET.

Tabla 6: Desventajas del sistema

En la siguiente imagen se puede apreciar gráficamente la arquitectura.

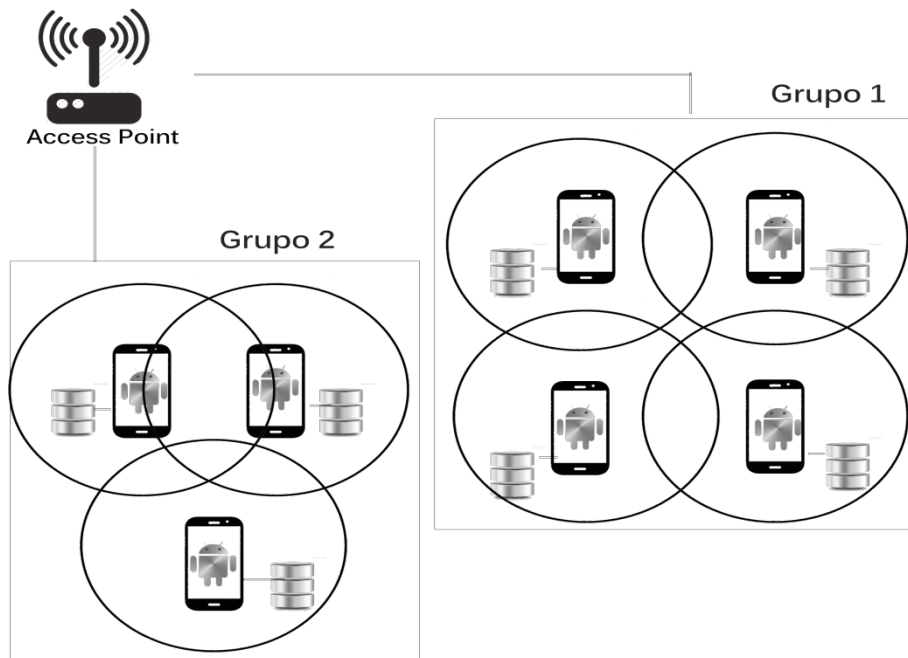


Figura 3.- Arquitectura del Sistema

2.3 Estudio de Viabilidad

Para lograr saber si el proyecto tendrá éxito, es necesario disminuir al mínimo el riesgo asociado al tomar una decisión de inversión. ^[12]

Lo que se realizará en este subcapítulo será una recopilación, análisis y evaluación de los diferentes tipos de información los cuales determinan si el proyecto conlleva o no riesgos en los temas técnicos, económicos y legales.

Como conclusión se buscará saber si es deseable realizar el proyecto con base en el rendimiento que se obtendría del mismo.

2.3.1 Viabilidad técnica ^[15]

La investigación de viabilidad técnica ayuda a identificar la idea del sistema, la cual se hace a partir de la información que ya se posee, al juicio común y la opinión que da la experiencia.

Antes de comenzar a desarrollar de este sistema, se verificó lo siguiente para estar seguros de un próspero sistema.

2.3.1.1 Hardware:

Dos a tres modelos de dispositivos móviles que cuenten con un sistema operativo Android con una versión 4.0 en adelante, así como también que posean un procesador de 1GHz, 512MB de Ram y 1GB de memoria interna. Por otra parte se requerirá un router que servirá como punto de acceso, y que puede ser un camino a futuro para hacer una red MANET verdadera.

2.3.1.2 Software:

Se requerirá una versión 4 o posterior de Android, esto por las ventajas que se describen a continuación ^[13]:

- WiFi-Direct.
- Nueva API para los desarrolladores.
- Ajustes en MMS, Email/Exchange, Calendario, Contactos, DSP, IPv6 y VPN.
- Tap and Go permite a los usuarios migrar rápidamente a un nuevo dispositivo Android, el uso de NFC y Bluetooth para transferir Detalles de la cuenta Google, ajustes de configuración de datos del usuario y las aplicaciones instaladas.

Las anteriores características afectan directamente o indirectamente en el desarrollo del sistema debido a que están involucradas en la configuración de los redes tanto dentro como fuera del dispositivo

2.3.1.3 Comunicación.

No se tenía un avance significativo de esta biblioteca en este tipo de dispositivos. Entonces siempre existió la duda sobre lo complicado y el tiempo que tomaría. Pero por otro lado, se tienen trabajos existentes que aplicaron bibliotecas y aplicaciones similares (ver Soluciones Existentes), por lo tanto, se tiene un respaldo de gran importancia y con resultados favorables sobre la comunicación con diferentes tipos de MANET en Android.

2.3.1.4 Soporte de Software.

La implementación del sistema debe ser sobre el lenguaje Java porque toda aplicación en Android se construye sobre este lenguaje, por lo tanto, para utilizar la biblioteca de manera expedita se ocupa el mismo lenguaje.

2.3.1.5 Usabilidad sobre la aplicación.

El grueso de este desarrollo se centró en desarrollar un sistema basado en HLMP, entonces, sabiendo los anteriores desafíos e incertidumbres se desarrolla un sistema lo más usable posible en el tiempo disponible. .

Por consiguiente, este sistema está basado en soluciones existentes, por lo tanto nuevos retos o nuevos objetivos no se podrán tener contemplados y podrían pasar desapercibidos en el desarrollo, de otra manera seguramente aparecerán problemas técnicos y de aprendizaje.

En conclusión según los recursos técnicos que se requieren para el desarrollo del proyecto, estos se pueden conseguir en el país, por tal razón, el proyecto es factible técnicamente.

2.3.2 Viabilidad económica.

Para efectuar el estudio de viabilidad económica se ha considerado determinar el presupuesto de costos de los tres recursos fundamentales, los cuales son recursos humanos, recursos tecnológicos y recursos materiales.

Además, ayudará a realizar el análisis costo-beneficio del sistema, el mismo que permitirá determinar si es factible el desarrollar económicamente el proyecto.

A continuación se desarrolla cada uno de los recursos fundamentales:

2.3.2.1 Recursos humanos:

Para el presente trabajo terminal se identificaron los puestos que se necesitarían para el completo desarrollo del sistema:

- Jefe de Proyecto.
- Analista.

- Diseñador
- Programador de Redes de Comunicación.
- Administrador de BD.

Cabe mencionar que para la práctica en realidad se cuenta con 3 personas para la elaboración del sistema por lo que cada persona debe de adquirir distintos roles durante el análisis, diseño y desarrollo del sistema.

Para la implementación se planea que cada uno de los 3 desarrolladores trabaje durante 4 horas diarias durante un lapso de 3 meses, lo cual genera 240 horas por cada desarrollador.

En el presente año, se cotiza la hora de trabajo a un desarrollador junior por \$100^[14], lo cual genera un total de \$8,000 al mes y por los tres meses tendremos un costo de \$24,000 por cada desarrollador.

Para el diseño se calculan otros dos meses por cada uno de los desarrolladores, lo cual daría un total de \$16,000. Juntando diseño e implementación, el total por los 5 meses trabajando en el proyecto sería de \$40,000 por desarrollador.

Resumiendo lo anterior, se vería de la siguiente manera:

Recursos Humanos			
Desarrollador	Tiempo(Hrs) x desarrollador	Precio x hora	Precio unitario
3	400	\$100	\$40,000
Total			\$120,000

Tabla 7: Recursos Humanos

2.3.2.2 Recursos tecnológicos:

Se hará uso de las siguientes tecnologías para el desarrollo del sistema:

Tecnología	Precio - MXN
Hardware	
Smartphones que cuenten con las características antes mencionadas. (2)	\$10000
Computadora (7/Hora). (3)	\$ 8400
Software	
Licencia Android	\$ 500
Java	Libre
HLMP	Libre
Android Studio	Libre
SQLite	Libre
Star UML	Libre

Total	\$ 18900
--------------	----------

Tabla 8: Recursos Tecnológicos

2.3.2.3 Recursos materiales:

Durante el desarrollo del Sistema se requiere:

Objeto	Precio
Cartuchos para impresora (2)	\$850
Papelería	\$150
Total	\$1000

Tabla 9: Recursos Materiales

2.3.2.4 Flujo de Pago

Flujo de Pago	
Recursos	Costos
Recursos Humanos.	\$120000

Recursos Tecnológicos.	18900
Recursos Materiales.	1000
Imprevistos (10%)	13990
Total	\$153890

Tabla 10: Flujo de Pago

2.3.3 Viabilidad Legal

La viabilidad legal nos permite determinar los derechos que tienen los autores sobre la documentación realizada por estos en este proyecto, la cual es exclusividad de los desarrolladores de este sistema.

2.3.3.1 Ley Orgánica de Protección de Datos.

Es una ley que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

Su objetivo principal es regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el cual sean tratados, los derechos de los ciudadanos sobre ellos y las obligaciones de aquellos que los crean o tratan.

2.3.3.2 Títulos de propiedad

Se recomienda adquirir las licencias para el software a emplearse en la implantación de un sistema informático de manera auténtica, con la finalidad de no tener inconvenientes legales a futuro.

Además, la viabilidad legal permite saber si hay inconvenientes legales que puedan impedir la implementación del proyecto. Entonces, se considerará lo siguiente:

- **Derechos de autor:** En dicho programa todo el contenido estará protegido en base a derecho autor y en caso de requerir información de un tercero se hará referencia según las normas IEEE para referenciar el artículo, documento, libro original.
- **Uso de licencias para el software a emplearse:** Como se ha estado mencionando en el presente documento, el software en mayoría que se usará será de forma gratuita y en algún caso excepcional se hará el uso de licencia para su uso.

2.3.4 Ley de Transparencia y Libre Acceso a la Información.

La información es un elemento imprescindible en el desarrollo del ser humano, ya que aporta elementos para que éste pueda orientar su acción en la sociedad.

El acceso a la información es una instancia necesaria para la participación ciudadana y la protección de los derechos civiles, dado que sin información adecuada, oportuna y veraz, la sociedad difícilmente se encuentre en condiciones óptimas para participar en la toma de decisiones públicas. A efecto de lo anterior ha surgido la necesidad de utilizarla de manera racional y productiva en beneficio del individuo y de la comunidad, para lo cual nuestra constitución la ha establecido como una garantía individual de todo gobernado y, la legislación secundaria en la Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental, ha incorporado reglas que buscan garantizar el ejercicio efectivo de este derecho.

El derecho a la información instituido en el último párrafo del artículo 6o. constitucional, adicionado mediante reforma publicada el 6 de diciembre de 1977, estaba limitado por la iniciativa de reformas y los dictámenes legislativos correspondientes, a constituir, solamente, una garantía electoral subsumida dentro de la reforma política de esa época, que obligaba al Estado a permitir que los partidos políticos expusieran ordinariamente sus programas, idearios, plataformas y demás características inherentes a tales agrupaciones, a través de los medios masivos de comunicación. Sin embargo, actualmente y gracias a la Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental, el derecho a la información y el libre acceso a la misma constituyen un garantía que se encuentra estrechamente vinculada con el respeto a la verdad, elemento fundamental para garantizar una verdadera democracia participativa, así como de mecanismos de gobernabilidad. A su vez contribuye a la disminución de la corrupción y fomenta la transparencia en la toma de las decisiones públicas. Asimismo, ese derecho del individuo, con la adición al

contenido original del artículo 6o., quedó también equilibrado con el derecho que tiene la sociedad a estar veraz y objetivamente informada, para evitar que haya manipulación.

Así, el Estado asume la obligación de cuidar que la información que llega a la sociedad en general, refleje la realidad y tenga un contenido que permita y coadyuve al acceso a la información veraz y oportuna, para que todo ciudadano que así lo requiera, pueda recibir en forma fácil y rápida conocimientos en la materia, ciencia o asunto que sea de su interés. Con ello se trata de propiciar, una participación informada para la solución de los grandes problemas nacionales y particulares, para evitar que se deforme el contenido de los hechos que pueden incidir en la formación de opinión.

El derecho a la Información, actualmente se encuentra consignado en la Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental, la cual tiene en cuenta aspectos tales como la veracidad de la información proporcionada, fijación de los plazos para la entrega de la información; vías administrativas, jurisdiccionales y judiciales para actuar frente a la denegatoria o entrega parcial de información; imposición de sanciones, régimen de excepciones, etc. O sea, que el derecho a la información, estrechamente vinculado con el derecho a conocer la verdad, exige que las autoridades se abstengan de dar a la comunidad información manipulada, incompleta o falsa, so pena de incurrir en violación grave a las garantías individuales en términos de lo dispuesto en el artículo 108 constitucional, Dicha Ley en su artículo 3, fracción XIV, inciso e), contempla como sujeto obligado al cumplimiento de la misma, al Tribunal Federal de Justicia Fiscal y Administrativa; en tanto que el artículo 61 de la misma, se obliga a este Tribuna a establecer mediante reglamentos o acuerdos de carácter general, los órganos, criterios y procedimientos institucionales para proporcionar a los particulares el acceso a la información, de conformidad con los principios y plazos establecidos en esa ley.

De lo anterior, surge la necesidad de la emisión de este reglamento, cuya finalidad es la dar cumplimiento a las disposiciones contenidas en ley que inciden en las obligaciones de este Tribunal para garantizar y facilitar el derecho de acceso a la información de todo ciudadano que acude al mismo en busca de información; propiciando con ello determinar los órganos encargados de proporcionar la información que le sea solicitada; así como las obligaciones y facultades del órgano en cuestión; ordenar los instrumentos, técnicas, actualizaciones y medios de la información en él contenidos para ponerlos al servicio de la comunidad, de modo que cumplan con la finalidad esencial de promover el desarrollo individual y colectivo de hombres y mujeres contribuyendo así al bienestar social.

En este reglamento se encuentran los lineamientos y procedimientos para permitir a cualquier ciudadano acceder a toda la documentación preparatoria o antecedente documental que contribuyó a la redacción del acto generado en este

órgano jurisdiccional, pero que no se encuadra en la categoría de documentación reservada o confidencial, y que por ende no conlleva la obligación de ser dada a la publicidad (escritos, oficios, pruebas, actuaciones de autoridad, dictámenes periciales, diligencias procedimentales, expediente en general, etc.). Ello en virtud de que la Ley de Transparencia y Acceso a la Información Pública Gubernamental, en su artículo 14 fracción IV, establece, entre otros supuestos, como información reservada los expedientes judiciales o de los procedimientos administrativos seguidos en forma de juicio, en tanto no hayan causado estado; o que aún causándolo, continúen con esa calificación en los términos establecidos por la propia ley.

Para los fines de este reglamento se toma en cuenta que la propia Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental, contempla limitaciones específicas del ejercicio del derecho a la información, tendientes a equilibrar el derecho del individuo frente a terceros y a la sociedad, puesto que el ejercicio de ese derecho no debe menoscabar la moral, los derechos de tercero, que implica el honor, la dignidad y el derecho a la intimidad de éste, en su familia y decoro; así como tampoco puede provocar algún delito o perturbar el orden público.

2.3.5 Ley de Registro y Derechos de Autor

Podemos definir al derecho de autor como "... la facultad exclusiva que tiene el creador intelectual para explotar temporalmente, por sí o por terceros, las obras de su autoría (facultades de orden patrimonial), y en la de ser reconocido siempre como autor de tales obras (facultades de orden moral), con todas las prerrogativas inherentes a dicho reconocimiento.

El artículo 13 cataloga las clases de obras que son objeto de protección, a continuación el listado:

- Literaria;
- Musical, con o sin letra;
- Dramática;
- Danza;
- Pictórica o de dibujo;
- Escultórica y de carácter plástico;
- Caricatura e historieta;
- Arquitectónica;

- Cinematográfica y demás obras audiovisuales;
- Programas de radio y televisión;
- Programas de cómputo;
- Fotográfica;
- Obras de arte aplicado que incluyen el diseño gráfico o textil, y
- De compilación, integrada por las colecciones de obras, tales como las enciclopedias, las antologías, y de obras u otros elementos como las bases de datos, siempre que dichas colecciones, por su selección o la disposición de su contenido o materias, constituyan una creación intelectual.

Las demás obras que por analogía puedan considerarse obras literarias o artísticas se incluirán en la rama que les sea más afín a su naturaleza.

Se entiende por programa de computación la expresión original en cualquier forma, lenguaje o código, de un conjunto de instrucciones que, con una secuencia, estructura y organización determinada, tiene como propósito que una computadora o dispositivo realice una tarea o función específica. Los programas de computación se protegen en los mismos términos que las obras literarias. Dicha protección se extiende tanto a los programas operativos como a los programas aplicativos, ya sea en forma de código fuente o de código objeto. Se exceptúan aquellos programas de cómputo que tengan por objeto causar efectos nocivos a otros programas o equipos.

2.4 Metodología

Se propone emplear la metodología SCRUM, la cual es una metodología de desarrollo ágil la cual nos va a permitir un proceso iterativo e incremental. Se establecerá cada Sprint con una duración de 3 semanas, se irá ajustando en cada uno de estos Sprints la funcionalidad así como se irá añadiendo nuevas prestaciones las cuales se priorizarán según la cual aporte mayor valor al proyecto.

Entre los beneficios que se tendrán al usar esta metodología serán:

- Flexibilidad a cambios
- Mayor productividad
- Predicciones de tiempo
- Reducción de riesgos

En cuanto a los roles, se tendrán los siguientes:

Rol	Nombre
Maestro de SCRUM	Chadwick Carreto Arellano
Dueño del producto	Mario Eduardo Rivero Ángeles
Equipo	Aguilar Zúñiga Daniel Eduardo Arroyo Zamora José Manuel Yáñez Anaya Uriel Yael

Tabla 11: Roles SCRUM

A continuación se muestra el ciclo de vida de SCRUM para poder entender cómo se van ir llevando los roles y los respectivos trabajos dentro del proyecto:

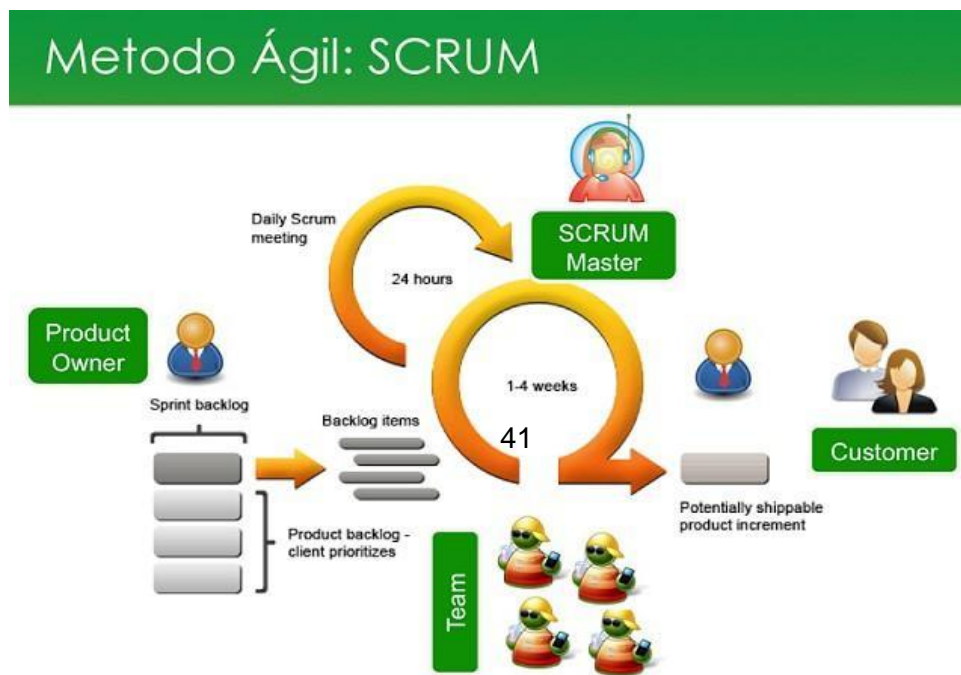


Figura 4.- Ciclo de vida de SCRUM

2.4.1 Planeación

Se contará con 3 sprints para la primera sección del proyecto, en los cuales se producirán los siguientes entregables:

- ❖ Sprint 1: Capítulo 1
 Fechas inicial: 1 de abril del 2015
 Fecha final: 13 de abril del 2015
- ❖ Sprint 2: Capítulo 2
 Fechas inicial: 14 de abril del 2015
 Fecha final: 1 de mayo del 2015
- ❖ Sprint 3: Capítulo 3
 Fechas inicial: 2 de mayo del 2015
 Fecha final: 22 de mayo del 2015

Los temas de cada capítulo están descritos en el índice del presente documento.

Ahora se mostrará cómo se ve el cronograma general de trabajo.

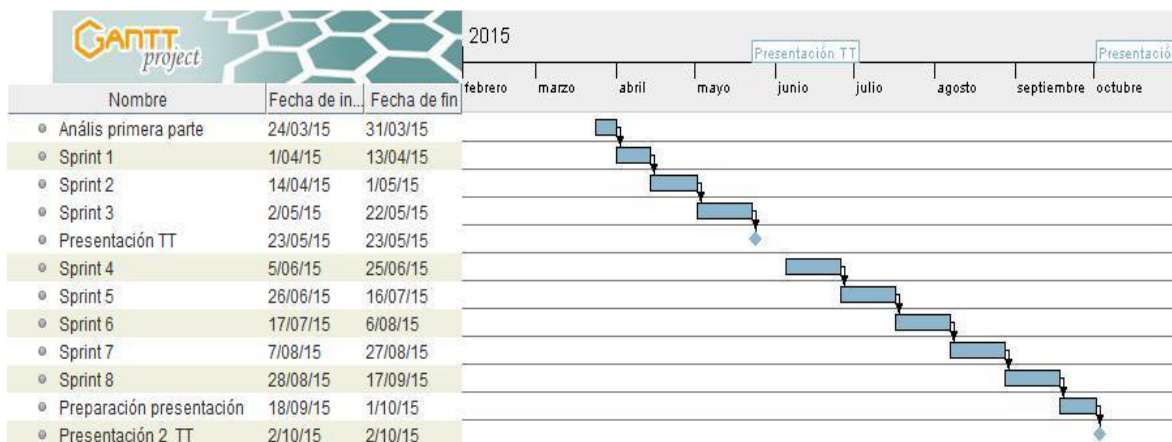


Figura 5.- Cronograma de actividades

3 Capítulo III.

3.1 Análisis y Diseño.

El diseño de la biblioteca HLMP está pensado en un ambiente de desarrollo en Android, por consiguiente, en esta parte se expondrán en breve algunos elementos importantes para un buen desarrollo dentro de este sistema operativo, con el único objetivo de entender y comprender las componentes y funciones reutilizadas y creadas en el diseño de la biblioteca HLMP para Android, al término se expondrá el diseño de la solución.

3.2 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.

3.3 Diseño de HLMP para Android

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:

1. **Android-AdHoc:** Encargado de la administración del WiFi en el dispositivo.
2. **HLMP-Java:** Relacionado al comportamiento del protocolo HLMP.

3. **HLMPConnect:** Es el esqueleto de la aplicación que administra los recursos anteriores y permite la comunicación con el usuario.

3.4 Introducción

En la siguiente sección se busca describir cuáles serán los servicios proporcionados por el sistema y sus posibles restricciones operativas.

3.5 Requerimientos no funcionales y Restricciones.

A continuación se especifican los requisitos no funcionales del proyecto. A diferencia de los funcionales, estos requisitos no exigirán ningún tipo de funcionalidad pero serán indispensables a la hora de realizar el sistema.

3.5.1 Desempeño.

- Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios a nivel local. En este sentido la información almacenada podrá ser consultada y actualizada permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
- El sistema debe estar en capacidad de dar respuesta al acceso de todos los usuarios con tiempo de respuesta aceptable y uniforme, en la medida de las posibilidades tecnológicas de la red, en períodos de alta, media y baja demanda de uso del sistema.

3.5.2 Disponibilidad.

- Estar disponible 100% o muy cercano a esta disponibilidad durante el horario hábil laboral de la escuela a nivel local (Ejemplo: de lunes a viernes de de 7:00 a.m. a 9:00 p.m., con excepción de los días festivos).
- Operar de la misma manera para todos los usuarios (profesores y alumnos) en la red.

3.5.3 Escalabilidad.

- El sistema debe ser construido sobre la base de un desarrollo evolutivo e incremental, de manera tal que nuevas funcionalidades y requerimientos relacionados puedan ser incorporados afectando el código existente de la menor manera posible; para ello deben incorporarse aspectos de reutilización de componentes.
- El sistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

3.5.4 Facilidad de Uso e Ingreso de Información.

- El sistema debe ser de fácil uso y entrenamiento por parte de los usuarios deL SISTEMA, así como de fácil adaptación de la entidad con el mismo.
- El sistema no debe permitir el cierre de una operación hasta que todos sus procesos, subprocesos y tareas relacionados, hayan sido terminados y cerrados satisfactoriamente.
- El ingreso de información al sistema debe diseñarse con transacciones que permitan el ingreso de los datos de forma parcial; es decir, que el tamaño de las páginas de registro (o formularios) de información sean adecuadas de acuerdo con la estabilidad de la red.

3.5.5 Facilidad para las Pruebas.

- El sistema debe contar con facilidades para la identificación de la localización de los errores durante la etapa de pruebas y de operación posterior.

3.5.6 Flexibilidad.

- El sistema debe ser diseñado y construido con los mayores niveles de flexibilidad en cuanto a la parametrización de los tipos de datos, de tal manera que la administración del sistema sea realizada por un administrador funcional del sistema.

3.5.7 Instalación.

- El sistema debe ser fácil de instalar en todas las plataformas de hardware y software de base definidas en este trabajo.

3.5.8 Mantenibilidad.

- Toda el sistema deberá estar completamente documentado, cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados tanto en el código fuente como en los manuales de administración y de usuario.
- El sistema debe contar con una interfaz de administración que incluya: Administración de usuarios, Administración de módulos y Administración de parámetros. En cada una de éstas secciones deberá ofrecer todas las opciones de administración disponibles para cada uno.
- El sistema debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema.

3.5.9 Operatividad.

- El sistema debe ser de fácil operación, y que demande un bajo nivel de soporte de los usuarios del sistema.
- El sistema deberá poder ser administrado por las personas encargadas o designadas.

3.5.10 Seguridad.

- Se programarán las partes específicas del protocolo de red intentando que sean lo más independientes posibles de la aplicación creada, con el fin de proporcionar la mayor extensibilidad posible.
- El acceso al Sistema debe estar restringido por el uso de claves asignadas a cada uno de los usuarios. Sólo podrán ingresar al Sistema las personas que estén registradas, estos usuarios serán clasificados en varios tipos de usuarios (o roles) con acceso a las opciones de trabajo definidas para cada rol.
- El control de acceso implementado debe permitir asignar los perfiles para cada uno de los roles identificados.
- Respecto a la confidencialidad, el sistema debe estar en capacidad de rechazar accesos o modificaciones indebidos (no autorizados) a la información y proveer los servicios requeridos por los usuarios legítimos del sistema.
- El sistema deberá contar con mecanismos que permitan el registro de actividades con identificación de los usuarios que los realizaron.
- El sistema debe contar con pistas de auditoría de las actividades que se realizan sobre el sistema con niveles razonables para su reconstrucción e identificación de los hechos.

3.5.11 Validación de Información.

- El sistema debe validar automáticamente la información contenida en los formularios de ingreso. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como obligatoriedad de campos, longitud de caracteres permitida por campo, manejo de tipos de datos, etc.

3.5.12 Arquitectura.

- La solución debe ser 100% basada en móvil y toda la parametrización y administración debe realizarse desde el propio nodo.
- La solución debe tener interfaces gráficas de administración y de operación en idioma español y en ambiente 100% Móvil.

- La información de los formularios que corresponda a listas de selección deberá ser parametrizada y administrable.
- Garantizar que el diseño de las consultas no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

3.5.13 Backups.

- El sistema deberá proveer mecanismos para generar backup's periódicamente de la información que se mantiene en el sistema. Los backup's deben ser responsabilidad del administrador del sistema quien deberá crearlos, almacenarlos y recuperar la información en el caso que se pierda información.

En la tabla siguiente se muestran numerados los requerimientos no funcionales del sistema más importantes, además se les da una escala de importancia que va de Baja a Alta según que lo indispensable que sea, además también esta misma escala será usada pero para su complejidad en la hora del desarrollo del sistema.

Requerimientos No Funcionales			
ID	Requerimiento	Importancia	Complejidad
RNF1	La interfaz de red para la interconexión de los dispositivos será WiFi.	Alta	Alta
RNF2	El sistema operativo para el que se realizará la aplicación será Android. Debido a esto, el lenguaje de programación utilizado será JAVA, puesto que es el utilizado en su	Alta	Moderado

	plataforma de desarrollo.		
RNF3	La interfaz de usuario debe ser intuitiva y de fácil manejo.	Alta	Alta
RNF4	El sistema debe ser P2P (peer-to-peer); esto es, debe ser simétrico para cualquier nodo dentro de las MANET's que forman los usuarios, de este modo ningún nodo posee una posición privilegiada por sobre otro (aparte de los privilegios que tenga asociado su rol). En este trabajo esta simetría es alcanzada parcialmente, puesto que de todos modos los usuarios deben acudir a un administrador centralizado para "inscribirse" en el sistema.	Alta	Alta
RNF5	El acceso a los archivos debe ser rápido para archivos de cierto tamaño. El acceso debe demorar menos de 20 segundos en archivos con tamaño inferior a 5MB.	Alta	Alta
RNF6	El sistema contará con un protocolo de información el cual facilita simular o desarrollar una red MANET.	Alta	Moderado

RNF7	Se debe realizar un control y manejo de todas las excepciones posibles para asegurar la fiabilidad del sistema	Alta	Alta
RNF8	La entrega final del sistema debe estar documentada y cubrir los requerimientos especificados	Alta	Alta
RNF9	Se debe cumplir con las actividades establecidas en el cronograma en tiempo y forma, durante el desarrollo del proyecto	Alta	Alta
RNF10	Realización de pruebas unitarias	Alta	Moderado
RNF11	Realización de pruebas de integración	Alta	Moderado
RNF12	Realización de pruebas de desempeño	Alta	Moderado

Tabla 12: Requerimientos no funcionales

3.6 Requerimientos funcionales

En este apartado se definirán los servicios que proporcionará el sistema, la manera en que debe reaccionar a entradas particulares y cómo se debe comportar en situaciones específicas. ^[16]

Requerimientos Funcionales			
ID	Requerimiento	Importancia	Complejidad
RF1	El usuario deberá colocar sus datos para registrarse dentro del sistema.	Alta	Baja
RF2	El usuario será capaz de agregar 1 o más grupos para compartir información.	Moderada	Baja
RF3	El usuario será capaz de invitar a 1 o más usuarios a un(o) grupo(s) para compartir información.	Moderada	Baja
RF4	El usuario será capaz de aceptar/negar invitaciones de grupo	Moderada	Baja
RF5	El usuario podrá operar funciones dentro de su perfil único	Alta	Moderada
RF6	El usuario contará con la opción de administrar toda la información que se encuentre dentro de su dispositivo en base a un	Alta	Moderada

	grupo o perfil.		
RF7	Cada uno de los nodos deberá poder interactuar con los nodos cercanos a este.	Alta	Alta
RF8	Cada uno de los nodos tendrá que mantener actualizados a los nodos cercanos en base a la información más reciente	Alta	Alta
RF9	El sistema tendrá la capacidad de transferir la información entre los nodos para así mantener actualizado a cada uno de estos.	Alta	Alta
RF10	El sistema informará al usuario al momento que su información haya sido actualizada.	Baja	Baja
RF11	El administrador será capaz de llevar un control de los usuarios que están dentro del sistema.	Moderado	Moderado
RF12	El usuario será capaz de salir del sistema	Alto	Baja

Tabla 13: Requerimientos funcionales

3.7 Casos de uso

3.7.1 Descripción de casos de uso

3.7.1.1 Caso de Uso 1- Registro de Usuario

	Descripción
Función	Registro usuario
Descripción	El usuario registrará sus datos en un formulario para poder acceder al sistema
Entradas	Datos del usuario
Fuente	Formulario de registro
Salidas	Mensaje de "Registro exitoso"
Destino	Ventana principal
Acción	Datos guardados dentro del sistema
Requerimientos	-
Precondición	-
Postcondición	El usuario quedará registrado dentro del sistema

Tabla 14: Descripción CU1

Trayectoria del Caso de uso

Trayectoria Principal

- 1- El actor selecciona la opción "Registro de Usuario"
- 2- El sistema muestra la interfaz para ingresar los "Datos de Usuario"
- 3- El actor ingresa en la interfaz sus datos de usuario
- 4- El sistema valida los datos ingresados por el actor
- 5- El actor finaliza procesos de Registro

- 6- Almacena los datos de usuario y muestra mensaje "Registro exitoso"
-- Fin Trayectoria Principal

Trayectoria Alternativa A: Cerrar Interfaz de Registro de Usuario

Condición: El ACTOR decide salir de la interfaz de registro

Viene del punto 1,3,5 de la Trayectoria Principal

- 1- El sistema muestra mensaje "Actualmente se está realizando alguna acción en el área de trabajo, ¿Está seguro de salir del área de trabajo?"
 - 2- El actor confirma el cierre de la interfaz
 - 3- El sistema cierra la interfaz de registro y muestra la ventana principal
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Formato de los datos Incorrectos

Condición: Los datos ingresados por el ACTOR, no tienen el formato correcto para ser almacenado en el sistema

Viene del punto 4 de la Trayectoria Principal

- 1- El sistema muestra en pantalla el mensaje "Formato Incorrecto"
 - 2- El actor cierra el mensaje
 - 3- Vuelve al punto (3) de la Trayectoria Principal, donde el actor se encontraba ingresando los datos
- Fin de la Trayectoria Alternativa

Trayectoria Alternativa C: Campos requeridos sin llenar

Condición: El ACTOR no ha ingresado la información requerida por el sistema para realizar el registro del usuario

Viene del punto 4 de la Trayectoria Principal

- 1- El sistema muestra en pantalla el mensaje "Aún faltan campos por llenar, Verifique que todos los campos marcados con un asterisco se hayan capturado"
 - 2- El actor cierra el mensaje
 - 3- Vuelve al punto (3) de la Trayectoria Principal, donde el actor se encontraba ingresando los datos
- Fin de la Trayectoria Alternativa

3.7.1.2 Caso de uso 2- Agregar grupo

	Descripción
Función	Agregar grupo
Descripción	El usuario será capaz de agregar 1 o más grupos para compartir información.
Entradas	Datos del usuario Nombre del grupo
Fuente	Formulario de grupo
Salidas	Mensaje de registro exitoso
Destino	Usuario
Acción	Nuevo grupo guardado en el dispositivo
Requerimientos	RF1
Precondición	Haber agregado sus datos anteriormente
Postcondición	El usuario pertenecerá correctamente a un(os) grupo(s)

Tabla 15: Descripción CU2

Trayectoria del Caso de uso

Trayectoria Principal

- 1- El actor selecciona la opción "Agregar grupo"
 - 2- El sistema muestra la interfaz de "Nuevo grupo"
 - 3.- El actor ingresa en la interfaz el nombre del grupo
 - 4.- El sistema valida información ingresada por el actor
 - 5.- El actor finaliza el proceso nuevo Grupo
 - 6.- El sistema almacena el nuevo grupo y manda mensaje de "Registro exitoso"
- Fin Trayectoria Principal

Trayectoria Alternativa A: Cerrar Interfaz

Condición: El ACTOR decide salir de la interfaz de registro
Viene del punto 1,3,5 de la Trayectoria Principal

- 1- El sistema muestra mensaje "Actualmente se está realizando alguna acción en el área de trabajo, ¿Está seguro de salir del área de trabajo?"
 - 2- El actor confirma el cierre de la interfaz
 - 3- El sistema cierra la interfaz de registro y muestra la ventana principal
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Formato de los datos Incorrectos

Condición: Los datos ingresados por el ACTOR, no tienen el formato correcto para ser almacenado en el sistema

Viene del punto 4 de la Trayectoria Principal

- 1- El sistema muestra en pantalla el mensaje "Formato Incorrecto"
 - 2- El actor cierra el mensaje
 - 3- Vuelve al punto (3) de la Trayectoria Principal, donde el actor se encontraba ingresando los datos
- Fin de la Trayectoria Alternativa

Trayectoria Alternativa C: Campos requeridos sin llenar

Condición: El ACTOR no ha ingresado la información requerida por el sistema para realizar el registro del usuario

Viene del punto 4 de la Trayectoria Principal

- 1- El sistema muestra en pantalla el mensaje "Aún faltan campos por llenar, Verifique que todos los campos marcados con un asterisco se hayan capturado"
 - 2- El actor cierra el mensaje
 - 3- Vuelve al punto (3) de la Trayectoria Principal, donde el actor se encontraba ingresando los datos
- Fin de la Trayectoria Alternativa

Trayectoria Alternativa D: Nombre de grupo ya existente

Condición: El ACTOR ingresa un nombre de grupo ya existente

Viene del punto 4 de la Trayectoria Principal

- 1- El sistema muestra en pantalla el mensaje "Nombre de grupo ya se ha registrado anteriormente, favor de cambiar nombre"
 - 2- El actor cierra el mensaje
 - 3- Vuelve al punto (3) de la Trayectoria Principal, donde el actor se encontraba ingresando los datos
- Fin de la Trayectoria Alternativa

3.7.1.3 Caso de Uso 3- Invitar a grupo

	Descripción
Función	Invitar a grupo
Descripción	El usuario será capaz de invitar a un grupo a uno o más usuarios.
Entradas	Datos del usuario emisor Datos del usuario receptor Nombre del grupo
Fuente	Formulario de “Invitación a grupo”
Salidas	Mensaje de Invitación a grupo exitoso
Destino	Usuario receptor
Acción	Mandar mensaje para unirse a un grupo a todos los usuarios seleccionados que se encuentren en la caché del dispositivo emisor o que estén en el área de red.
Requerimientos	RF2
Precondición	Haber agregado o haberse unido a un grupo para poder compartir información
Postcondición	Invitación de grupo a usuarios seleccionados por el creador del grupo

Tabla 16: Descripción CU3

Trayectorias de Caso de Uso

Trayectoria Principal

- 1.- El usuario entra a la interfaz de “Invitar a grupo”
- 2.- El sistema muestra en la interfaz los usuarios que se encuentran en la cache, los usuarios que se encuentran cercanos y los grupos en los cuales se ha unido.
- 3.- El usuario selecciona uno o más usuarios para uno o más grupos
- 4.- El sistema valida la información

- 5.- El usuario finaliza la acción
 - 6.- El sistema manda mensaje a los usuarios seleccionados y muestra mensaje de “Envío de invitación exitoso”
- Fin Trayectoria Principal

Trayectoria Alternativa A: Cerrar Interfaz

- Condición: El ACTOR decide salir de la interfaz de registro
Viene del punto 1,3,5 de la Trayectoria Principal
- 1- El sistema muestra mensaje “Actualmente se está realizando alguna acción en el área de trabajo, ¿Está seguro de salir del área de trabajo?”
 - 2- El actor confirma el cierre de la interfaz
 - 3- El sistema cierra la interfaz de registro y muestra la ventana principal
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Conexión fallida

- Condición: El SISTEMA no puede conectar con el(los) usuario(s)
Viene del punto 2,6 de la Trayectoria Principal
- 1- El sistema muestra mensaje “Conexión fallida con el usuario”
 - 2- El actor cierra el mensaje
 - 3- El sistema intenta enviar de forma automática el mensaje más tarde
- Fin Trayectoria Alternativa

3.7.1.4 Caso de Uso 4- Aceptar/Negar invitación de grupo

	Descripción
Función	Aceptar/Negar invitación de grupo
Descripción	El usuario será notificado que ha recibido un

	mensaje de invitación a un grupo por parte de cuál usuario y podrá decidir si unirse o no a este.
Entradas	Datos del usuario emisor Nombre del grupo
Fuente	Mensaje de notificación
Salidas	Mensaje de unión a grupo exitoso
Destino	Grupo receptor
Acción	Aceptar/Negar invitación de usuario a grupo por medio de un mensaje de petición, con lo cual podrá compartir o recibir algún servicio que se encuentre disponible.
Requerimientos	RF3
Precondición	Haber registrado la información solicitada por parte del sistema en la aplicación
Postcondición	Usuario agregado o no a un grupo creado previamente.

Tabla 17: Descripción CU4

Trayectorias de Caso de Uso

Trayectoria Principal

- 1.- El usuario entra a la interfaz de "Invitaciones a grupo"
- 2.- El sistema muestra en la interfaz las solicitudes de grupo que tiene sin respuesta hasta el momento.
- 3.- El usuario selecciona una solicitud de grupo
- 4.- El sistema muestra la petición a grupo que ha recibido y por parte de qué usuario.
- 5.- El usuario decide si aceptar dicha petición
- 6.- El sistema guarda información y manda mensaje "Unión de grupo exitoso"
- 7.- El sistema manda respuesta al usuario remitente del mensaje de petición.

--Fin Trayectoria Principal

Trayectoria Alternativa A: Cerrar Interfaz

Condición: El ACTOR decide salir de la interfaz de registro

Viene del punto 1,3,5 de la Trayectoria Principal

- 1- El sistema muestra mensaje "Actualmente se está realizando alguna acción en el área de trabajo, ¿Está seguro de salir del área de trabajo?"
 - 2- El actor confirma el cierre de la interfaz
 - 3- El sistema cierra la interfaz de registro y muestra la ventana principal
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Conexión fallida

Condición: El SISTEMA no puede conectar con el(los) usuario(s)

Viene del punto 7 de la Trayectoria Principal

- 1- El sistema muestra mensaje "Conexión fallida con el usuario"
 - 2- El actor cierra el mensaje
 - 3- El sistema intenta enviar de forma automática el mensaje más tarde
- Fin Trayectoria Alternativa

Trayectoria Alternativa C: Petición denegada

Condición: El ACTOR decide a no unirse al grupo

Viene del punto 5 de la Trayectoria Principal

- 1- El sistema muestra mensaje "Invitación rechazada correctamente"
 - 2- El actor cierra el mensaje
 - 3- El sistema regresa al usuario a la Ventana Principal
- Fin Trayectoria Alternativa

3.7.1.5 Caso de Uso 5- Operar funciones dentro de un perfil único

	Descripción
Función	Operar funciones dentro de un perfil único
Descripción	El usuario podrá llevar a cabo operaciones dentro de su perfil
Entradas	Perfil escogido
Fuente	Lista de opciones
Salidas	Mensaje de función realizada correctamente

Destino	Usuario receptor
Acción	Modificación de información personal del usuario según el perfil del mismo.
Requerimientos	RF1, RF2
Precondición	Perfil
Postcondición	Datos actualizados

Tabla 18: Descripción CU5

Trayectorias de Caso de Uso

Trayectoria Principal

- 1.- El usuario entra al menú de funciones del sistema
- 2.- El sistema muestra las funciones a las que pueda acceder
- 3.- El usuario selecciona una función para modificar su información
- 4.- El sistema muestra campos de dicha función
- 5.- El usuario completa nuevamente dichos campos con información actualizada
- 6.- El sistema almacena la información mandada del usuario.
- 7.- El sistema manda mensaje de "Información almacenada correctamente"

Trayectoria Alternativa A: Cerrar Interfaz

Condición: El ACTOR decide salir de la interfaz de registro

Viene del punto 1,3,5 de la Trayectoria Principal

- 1- El sistema muestra mensaje "Actualmente se está realizando alguna acción en el área de trabajo, ¿Está seguro de salir del área de trabajo?"
 - 2- El actor confirma el cierre de la interfaz
 - 3- El sistema cierra la interfaz de registro y muestra la ventana principal
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Formato de los datos Incorrectos

Condición: Los datos ingresados por el ACTOR, no tienen el formato correcto para ser almacenado en el sistema

Viene del punto 5 de la Trayectoria Principal

- 1- El sistema muestra en pantalla el mensaje "Formato Incorrecto"

- 2- El actor cierra el mensaje
- 3- Vuelve al punto (3) de la Trayectoria Principal, donde el actor se encontraba ingresando los datos
- Fin de la Trayectoria Alternativa

Trayectoria Alternativa C: Campos requeridos sin llenar

Condición: El ACTOR no ha ingresado la información requerida por el sistema para realizar el registro del usuario

Viene del punto 5 de la Trayectoria Principal

- 1- El sistema muestra en pantalla el mensaje “Aún faltan campos por llenar, Verifique que todos los campos marcados con un asterisco se hayan capturado”
- 2- El actor cierra el mensaje
- 3- Vuelve al punto (3) de la Trayectoria Principal, donde el actor se encontraba ingresando los datos
- Fin de la Trayectoria Alternativa

3.7.1.6 Caso de Uso 6- Administrar información del dispositivo

	Descripción
Función	Administrar información del dispositivo
Descripción	El usuario contará con la opción de administrar toda la información que se encuentre dentro de su dispositivo.
Entradas	Información del usuario
Fuente	Gestor de información
Salidas	Tabla “Servicio/Grupo” Mensaje exitoso
Destino	Nodo receptor
Requerimientos	RF2
Precondición	Perfil escogido Estar dentro del área de red

Poscondición	Tabla que contenga
--------------	--------------------

Tabla 19: Descripción CU6

Trayectoria del Caso de Uso.

Trayectoria Principal.

- 1.- El usuario entra al Gestor de Información del sistema
 - 2.- El sistema muestra los servicios e información que dispone y los grupos en los cuales se encuentra
 - 3.- El usuario selecciona un servicio para un(os) grupo(s)
 - 4.- El sistema valida la acción del usuario
 - 5.- El usuario finaliza el proceso “Administrar información”
 - 6.- El sistema guarda dicha acción en una tabla “Servicio/Grupo” y muestra un mensaje de acción exitosa
- Fin trayectoria principal

Trayectoria Alternativa A: Cerrar Interfaz

Condición: El ACTOR decide salir de la interfaz de registro

Viene del punto 1,3,5 de la Trayectoria Principal

- 1- El sistema muestra mensaje “Actualmente se está realizando alguna acción en el área de trabajo, ¿Está seguro de salir del área de trabajo?”
 - 2- El actor confirma el cierre de la interfaz
 - 3- El sistema cierra la interfaz de registro y muestra la ventana principal
- Fin Trayectoria Alternativa

3.7.1.7 Caso de Uso 7- Cada uno de los nodos podrá interactuar con los nodos cercanos a este.

	Descripción
Función	Interactuar con nodos cercanos
Descripción	Cada nodo podrá ver los dispositivos

	cercanos e interactuar con ellos por medio de servicios para transferencia de información
Entradas	Petición de conexión
Fuente	Nodo fuente
Salidas	Solicitud de conexión, transferencia de información.
Destino	Nodos concurrentes.
Acción	Transferencia de Información
Requerimientos	R5
Precondición	Estar registrado en el sistema, estar dentro del área de red.
Postcondición	Identificar si cuentan con el mismo perfil, transferencia de información.

Tabla 20: Descripción CU7

Trayectoria del Caso de Uso.

Trayectoria Principal.

- 1.- El nodo emisor manda solicitud de conexión al nodo receptor
 - 2.- El nodo receptor manda información de conexión al nodo servidor
 - 3.- El nodo servidor realiza búsqueda de contacto en su repositorio de información
 - 4.- El nodo servidor devuelve información de contacto al nodo receptor
 - 5.- El nodo receptor guarda información de contacto en su repositorio de información para futuras consultas
 - 6.- El nodo receptor manda confirmación de conexión al nodo emisor
- Fin trayectoria principal

Trayectoria Alternativa A: Conexión fallida

Condición: El NODO EMISOR no puede conectar con el NODO RECEPTOR
Viene del punto 1 de la Trayectoria Principal

- 1- El nodo emisor cierra conexión con el nodo receptor
 - 2.- El nodo receptor cierra conexión con el nodo emisor
 - 3.- El nodo emisor se prepara para reintentar solicitud de conexión
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Conexión fallida Servidor

Condición: El NODO RECEPTOR no puede conectar con el SERVIDOR
Viene del punto 2 de la Trayectoria Principal

- 1- El nodo receptor cierra conexión con el servidor
 - 2.- El servidor cierra conexión con el nodo receptor
 - 3.- En nodo receptor se prepara para reintentar solicitud de conexión
- Fin Trayectoria Alternativa

Trayectoria Alternativa B-1: Conexión fallida por periodo de tiempo Servidor

Condición: El NODO RECEPTOR no puede conectar con el SERVIDOR después de múltiples intentos en un periodo de tiempo de 10 segundos

- 1.- El nodo receptor manda mensaje de conexión fallida al nodo emisor
 - 2.- El nodo emisor cierra conexión con el nodo receptor
 - 3.- El nodo receptor cierra conexión con el nodo emisor
- Fin Trayectoria Alternativa

Trayectoria Alternativa D: Nodo receptor tiene información de contacto

Condición: El NODO RECEPTOR en su repositorio de información contiene la información de contacto

Viene del punto 2 de la Trayectoria Principal

- 1- El nodo receptor manda confirmación de conexión al nodo emisor.
- Fin Trayectoria Alternativa

3.7.1.8 Caso de Uso 8- Cada uno de los nodos tendrá que mantener actualizados a los nodos cercanos en base a la información más reciente

	Descripción
Función	Mantener a los nodos con la información actualizada

Descripción	Cada nodo origen debe enviar a los demás todos los servicios que hayan sido modificados con la finalidad de mantener todo actualizado.
Entradas	Tablas de última modificación.
Fuente	Nodos fuente
Salidas	Información modificada Mensaje de acción exitosa
Destino	Nodos cercanos
Acción	Transferencia de Información
Requerimientos	R6, R5
Precondición	Haber modificado algún archivo, permanecer dentro de la red

Tabla 21: Descripción CU8

Trayectoria del Caso de Uso.

Trayectoria Principal.

- 1.- El usuario ingresa al sistema previamente.
- 2.- El sistema, ya con una sesión activa, empieza a identificar los nodos cercanos a este dentro de un radio determinado.
- 3.- El sistema empieza a enviar peticiones para identificar a los nodos.
- 4.- El sistema selecciona al nodo(s) dentro de su radio.
- 5.- El sistema empieza el envío de todos los servicios que hayan sido modificados con la finalidad de mantener todo actualizado basándose en las tablas de modificación.
- 6.- Termina la transferencia modificando las tablas de última modificación para que se utilicen más adelante.
- 7.- El sistema muestra un mensaje de acción exitosa
-Fin trayectoria principal

Trayectoria Alternativa A: Conexión fallida

- Condición: El NODO EMISOR no puede conectar con el NODO RECEPTOR
Viene del punto 3,4,5,6 de la Trayectoria Principal
- 1- El nodo emisor cierra conexión con el nodo receptor
 - 2.- El nodo receptor cierra conexión con el nodo emisor
 - 3.- El nodo emisor se prepara para reintentar solicitud de conexión
- Fin Trayectoria Alternativa

3.7.1.9 Caso de Uso 9- El sistema tendrá la capacidad de transferir la información entre los nodos

	Descripción
Función	Transferir Información
Descripción	Implementando un protocolo de red Manet, el sistema podrá transferir toda la información que sea solicitada por cualquier nodo desde otros.
Entradas	Nodos fuente
Fuente	Nodos concurrentes
Salidas	Información enviada a otros nodos.
Destino	Nodos destino
Acción	Transferencia de información entre nodos que se encuentren en el área de red del nodo emisor de información
Requerimientos	R5, R6, R7
Precondición	Nodo origen y destino se encuentren dentro de la red. Ambos nodos pertenezcan al mismo grupo. Petición de conexión
Postcondición	Información de ambos nodos actualizada

Tabla 22: Descripción CU9

Trayectoria de Caso de Uso

Trayectoria Principal

- 1.- El nodo principal manda petición de información de protocolo con el cual se va a transferir información al nodo secundario.
- 2.- El nodo secundario envía protocolo al nodo principal
- 3.- El nodo principal manda información de grupo al que pertenece
- 4.- El nodo secundario comprueba información y manda tablas de última modificación de la información que puede compartir
- 5.- El nodo principal compara tablas de última modificación y envía petición de actualización
- 6.- El nodo secundario transfiere información al nodo principal
- 7.- El nodo principal cierra conexión
- 8.- El nodo secundario cierra conexión

Trayectoria Alternativa A: Conexión fallida

Condición: El NODO EMISOR no puede conectar con el NODO RECEPTOR

Viene del punto 5 de la Trayectoria Principal

- 1- El nodo emisor cierra conexión con el nodo receptor
 - 2.- El nodo receptor cierra conexión con el nodo emisor
 - 3.- El nodo emisor se prepara para reintentar solicitud de conexión
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Conexión perdida nodo Emisor

Condición: El nodo RECEPTOR pierde la conexión con el nodo EMISOR

Viene del punto 2,4,6 de la Trayectoria Principal

- 1.- El nodo receptor cancela proceso de conexión
 - 2.- El nodo emisor cierra conexión
 - 3.- El nodo receptor cierra conexión
 - 4.- El nodo emisor reintentará conexión
- Fin Trayectoria Alternativa

Trayectoria Alternativa C: Conexión perdida nodo Receptor

Condición: El nodo EMISOR pierde la conexión con el nodo RECEPTOR y transcurre el tiempo de temporizador

Viene del punto 3,5 de la Trayectoria Principal

- 1.- El nodo emisor deja transcurrir el temporizador
 - 2.- El nodo receptor cancela proceso
 - 3.- El nodo emisor cancela proceso
 - 4.- El nodo receptor cierra conexión
 - 5.- El nodo emisor cierra conexión y reintenta conexión
- Fin Trayectoria Alternativa

3.7.1.10 Caso de Uso 10- El sistema informará al usuario al momento que su información haya sido actualizada.

	Descripción
Función	Notificar actualización
Descripción	Cada vez que el sistema haga una actualización de información de los nodos concurrentes a este, habrá una notificación de todas las acciones que se llevaron a cabo para que los usuarios estén sincronizados.
Entradas	Servicio confirmado Información actualizada.
Fuente	Tabla de última actualización
Salidas	Notificación de acción realizada
Destino	Barra de notificación del dispositivo
Acción	Notificación de cualquier acción en la cual se haya modificado información del mismo nodo.
Requerimientos	RF9
Precondición	Registro de usuario Obtener grupo(s) Interacción entre nodos

Postcondición	Usuario notificado de las acciones previamente realizadas.
---------------	--

Tabla 23: Descripción CU10

Trayectoria de Caso de Uso

Trayectoria Principal

- 1.- El sistema al finalizar una transferencia de información, manda mensaje de notificación al dispositivo.
- 2.- El dispositivo muestra notificación
- 3.- El sistema actualiza tabla de última actualización

Trayectoria Alternativa A: Notificación fallida

Condición: El sistema le es imposible mandar mensaje de notificación
Viene del punto 1 de la Trayectoria Principal

- 1.- El sistema muestra mensaje de "Notificación Fallida"
 - 2.- El usuario cierra ventana de mensaje
 - 3.- El sistema muestra mensaje de las acciones realizadas
- Fin Trayectoria Alternativa

3.7.1.11 Caso de Uso 11- El administrador será capaz de llevar un control de los usuarios que están dentro del sistema.

	Descripción
Función	Gestionar Usuarios.
Descripción	El administrador será capaz de llevar un control de los usuarios que están dentro del sistema
Entradas	Información de usuario Fecha última conexión
Fuente	Relación de usuario y fecha de última

	conexión
Salidas	Mensaje de usuario eliminado correctamente Mensaje de actualización a nodos cercanos
Destino	Repositorio de información
Acción	Eliminar usuario que no haya sincronizado su información en un periodo largo de tiempo
Requerimientos	R1
Precondición	Estar registrado en el sistema. Estar dentro del área de red.
Postcondición	Usuarios disponibles para recibir o dar algún servicio dentro del sistema

Tabla 24: Descripción CU11

Trayectoria de Caso de Uso

Trayectoria Principal

- 1.- El administrador entra a la sección "Administrar Usuarios".
- 2.- El sistema muestra en la interfaz una relación de usuario y fecha de última sincronización
- 3.- El administrador selecciona a los usuarios que no se han sincronizado en un periodo de tiempo largo y escoge la opción eliminar
- 4.- El sistema coloca al usuario en estado eliminado de su repositorio de información
- 5.- El sistema muestra mensaje de "usuario eliminado correctamente" y emite mensaje de actualización a los nodos cercanos.

--Fin Trayectoria Principal

Trayectoria Alternativa A: Conexión fallida

Condición: El NODO EMISOR no puede conectar con el NODO RECEPTOR
Viene del punto 5 de la Trayectoria Principal

- 1- El nodo emisor cierra conexión con el nodo receptor

- 2.- El nodo receptor cierra conexión con el nodo emisor
- 3.- El nodo emisor se prepara para reintentar solicitud de conexión
- Fin Trayectoria Alternativa

Trayectoria Alternativa B: Cerrar Interfaz

Condición: El ACTOR decide salir de la interfaz de registro

Viene del punto 1,3 de la Trayectoria Principal

- 1- El sistema muestra mensaje “Actualmente se está realizando alguna acción en el área de trabajo, ¿Está seguro de salir del área de trabajo?”
- 2- El actor confirma el cierre de la interfaz
- 3- El sistema cierra la interfaz de registro y muestra la ventana principal
- Fin Trayectoria Alternativa

3.7.1.12 Caso de Uso 12- Salir del sistema

	Descripción
Función	Salir del sistema.
Descripción	El usuario será capaz de cerrar la aplicación para dejar de interactuar con los demás usuarios.
Entradas	Información de usuario Información de conexión
Fuente	Datos del dispositivo del usuario
Salidas	Mensaje de salida del sistema.
Destino	Usuario
Acción	Eliminar usuario que no haya sincronizado su información en un periodo largo de tiempo
Requerimientos	R1

Precondición	Tener abierta la aplicación.
Postcondición	Sistema cerrado

Tabla 25: Descripción CU12

Trayectoria de Caso de Uso

Trayectoria Principal

- 1.- El usuario pulsa la opción salir del menú.
 - 2.- El sistema apaga la red con la cual intercambia información.
 - 3.- El sistema muestra mensaje de “Regresa pronto”.
 - 4.- El usuario cierra el mensaje
 - 5.- El sistema cierra la aplicación
- Fin Trayectoria Principal

Trayectoria Alternativa A: El sistema está interactuando con otro nodo

CONDICIÓN: El sistema está dando o pidiendo algún servicio a un nodo cercano.

Viene del punto 2 de la Trayectoria Principal.

- 1.- El sistema muestra mensaje “Intercambiando información”
 - 2.- El usuario espera hasta que finalice el proceso.
 - 3.- El sistema reintenta el punto 2 de la Trayectoria Principal.
- Fin Trayectoria Alternativa

3.8 Requerimientos de Sistema.

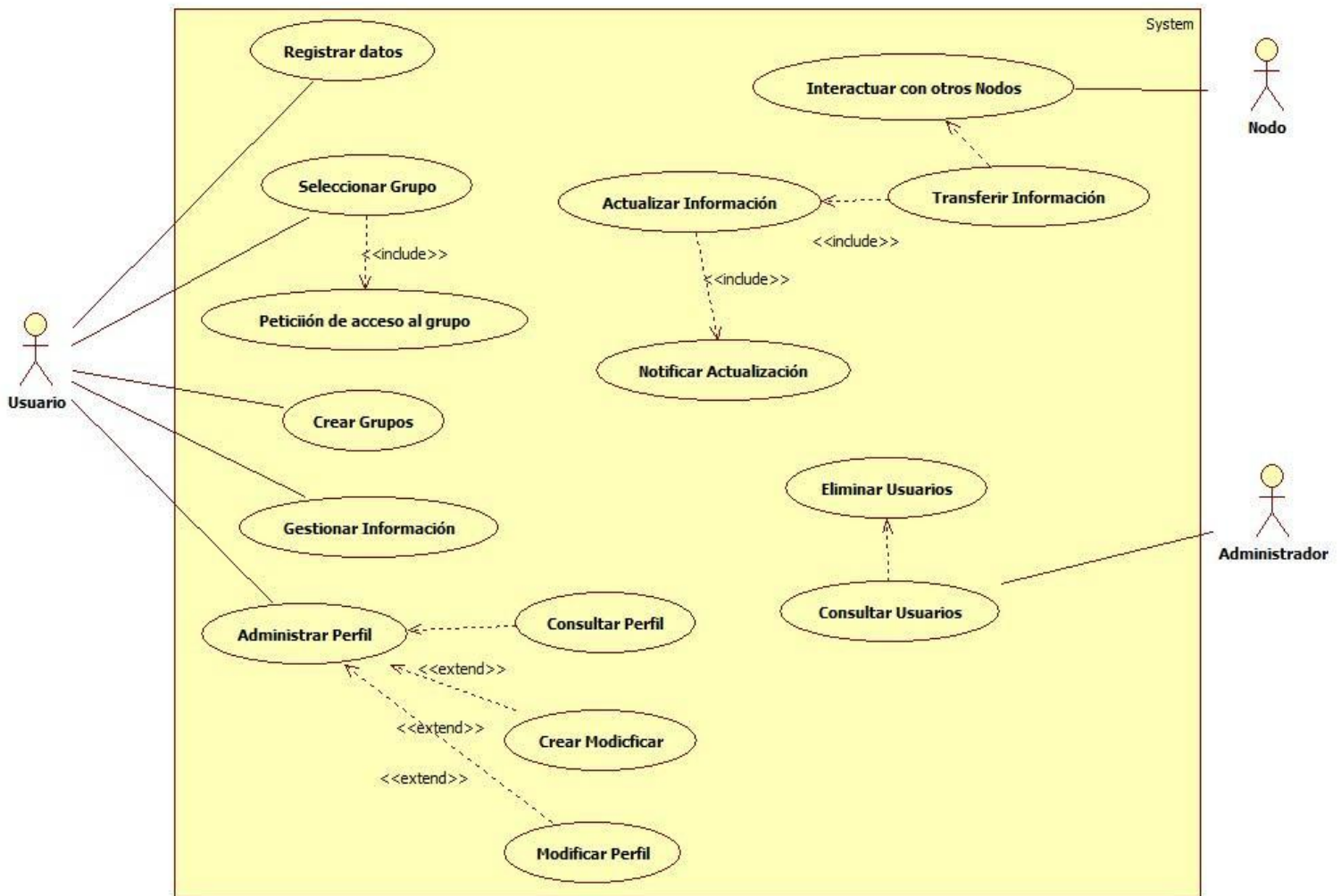
Para el correcto funcionamiento del sistema se consideran las siguientes características:

- ❖ Router
- ❖ Smartphone con Procesador de 1GHz, 1Gb de Ram y 2Gb de memoria interna con Sistema Operativo Android versión 4.0 o superior.

Definición de los roles.

Rol	Descripción
Administrador	El administrador del sistema lleva control de los roles del sistema. Es un nodo especial dentro del sistema, sus características especiales son gestionar a los diferentes usuarios para que haya una mayor interacción entre estos.
Usuario	Es un usuario directo que interactúa con el sistema cumpliendo funciones respectivas a su perfil, como la transferencia de información.

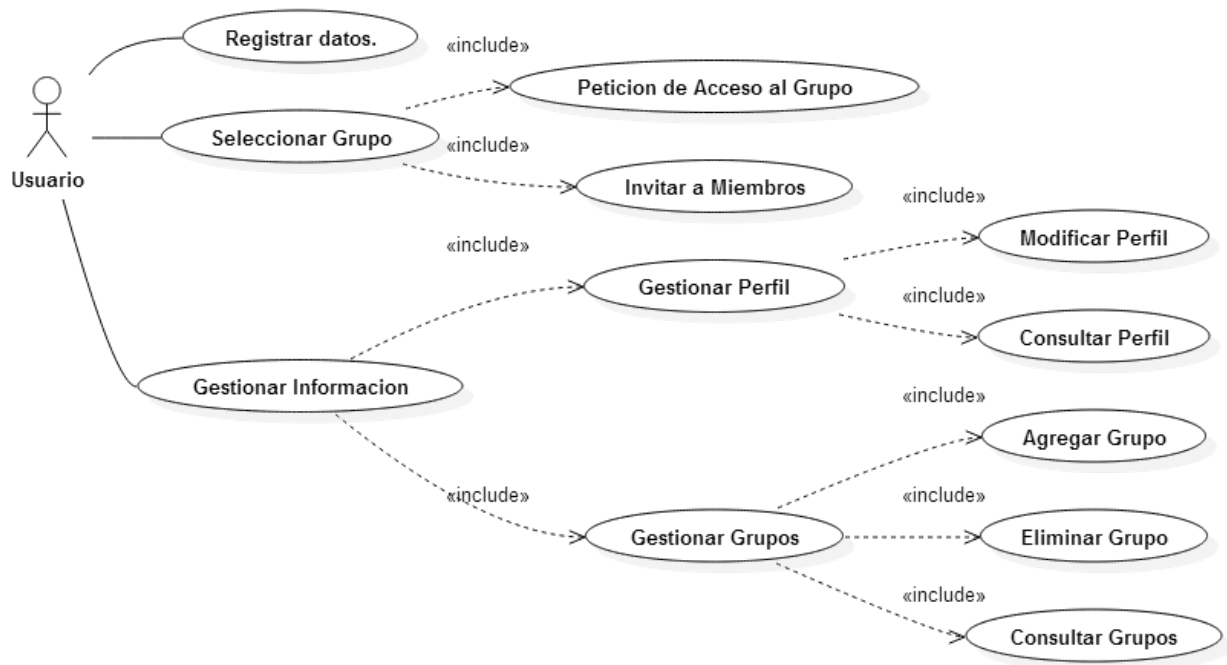
Tabla 26: Roles del Sistema



3.8.1 Diagrama de Casos de uso

En este diagrama se muestran de manera general las principales de los actores que están dentro del sistema. Se puede ver que son acciones más generalizadas, sin embargo, más adelante se explicaran a detalle posteriormente con un diagrama de casos de uso por actor, solo para especificar de manera más detallada las acciones de los actores dentro del sistema.

3.8.1.1 Diagrama de Casos de Uso de Usuario



Registrar datos: El usuario registra sus datos correspondientes.

Seleccionar Grupo: El usuario selecciona un grupo que se encuentre en esos momentos.

1. **Petición de Acceso al Grupo:** Se envía una demanda para pertenecer a dicho grupo seleccionado previamente.
2. **Invitar a Miembros:** El usuario envía invitaciones a usuarios para que se unan a cierto grupo.

Gestionar Información: Corresponde a una serie de acciones que podrá llevar a cabo el usuario.

Gestionar Perfil: Corresponde a una serie de acciones solo para el perfil por parte del usuario.

Gestionar Grupos: Corresponde a una serie de acciones solo para los grupos por parte del usuario

- Agregar Grupo.
- Borrar Grupo.
- Consultar Grupo.

3.9 Diagrama de Secuencia

Los diagramas de secuencia muestran la forma en que un grupo de objetos se comunican entre sí a lo largo del tiempo.

A continuación se describirán los métodos más usados para el desarrollo del sistema:

Nombre método	Descripción
Pedir(Servicio/Formulario)	El método que se encarga de realizar una solicitud de servicio/formulario.
Procesa(Servicio/Formulario/Petición)	El método tiene como salida el servicio/formulario/petición que fue solicitada por el método pedir.
Enviar(Información/Respuesta)	El método se encarga de hacer llegar a otro objeto los parámetros que este contenga.
Almacenar(Información/Respuesta)	El método tiene como objetivo guardar en el dispositivo lo que se encuentre como parámetro
Finalizar(Proceso/solicitud)	El método se encarga de concluir la acción que estaban llevando los objetos
Eliminar(usuario)	Método que realiza la solicitud de eliminar algún usuario del sistema
Solicitar_Conexión(información_conexión)	Se solicita conectarse al sistema por medio del nodo que este a su

	disposición
Buscar(Información)	Se encarga de investigar si dentro del objeto se encuentra la información que le solicitaron dentro del parámetro
Validar(información)	Método que verifica que la información previamente sea la correcta y cumpla con las características correspondientes que solicita este módulo del sistema
Conexión()	Método que se encarga de establecer una conexión segura con la base de datos correspondiente haciendo uso de un servidor
Cerrar Conexión ()	Método que se encarga de finalizar la conexión con la base de datos, y de esa manera evitar operaciones posteriores
Obtención_Servicio()	Se hacen los preparativos correspondientes para establecer una conexión exitosa entre los dos actores que lo requieran.
Registro(respuesta_conexión)	Se registran los datos de conexión los cuales toma de los parámetros ingresados previamente por el usuario.

Tabla 27: Métodos Diagrama de Secuencia

3.9.1 Interacción general entre nodos

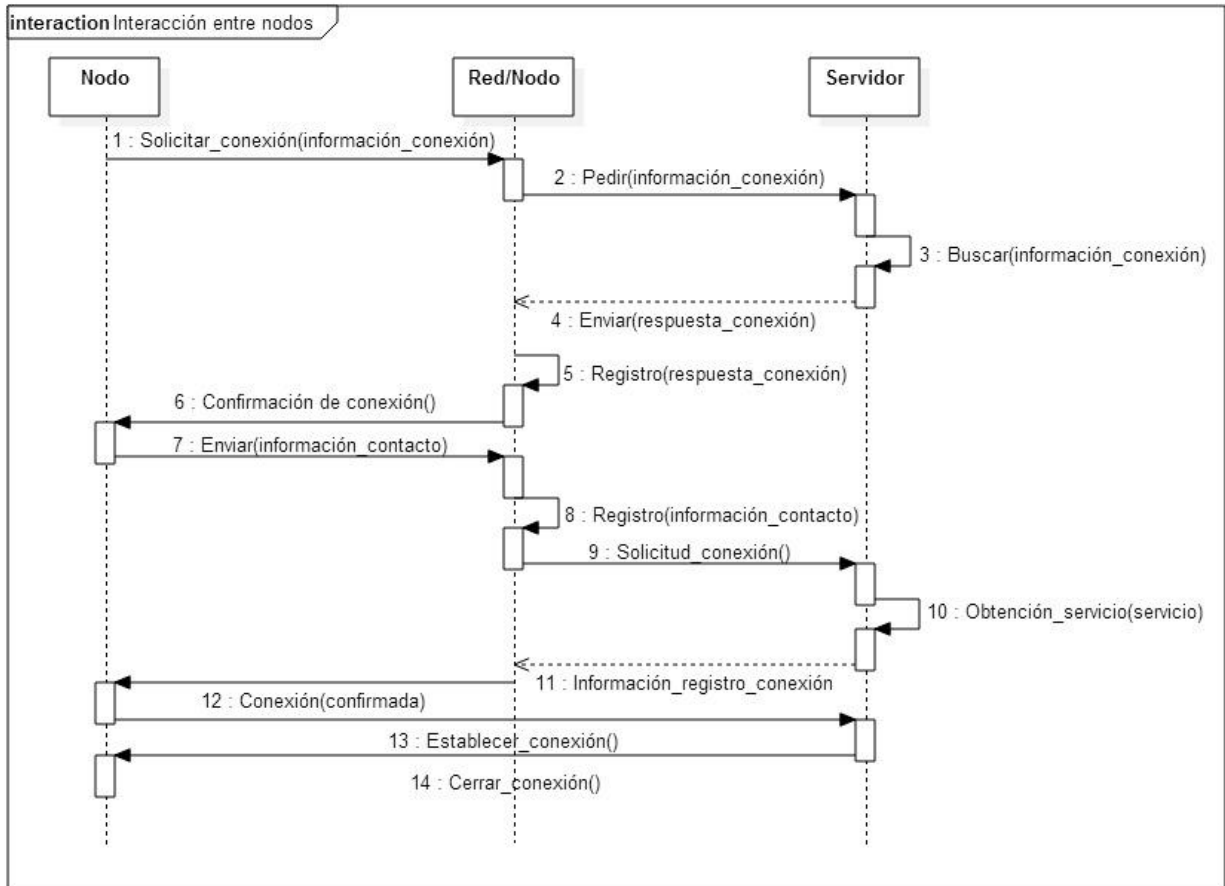


Figura 6.- Interacción entre nodos DS1

3.9.2 Registro

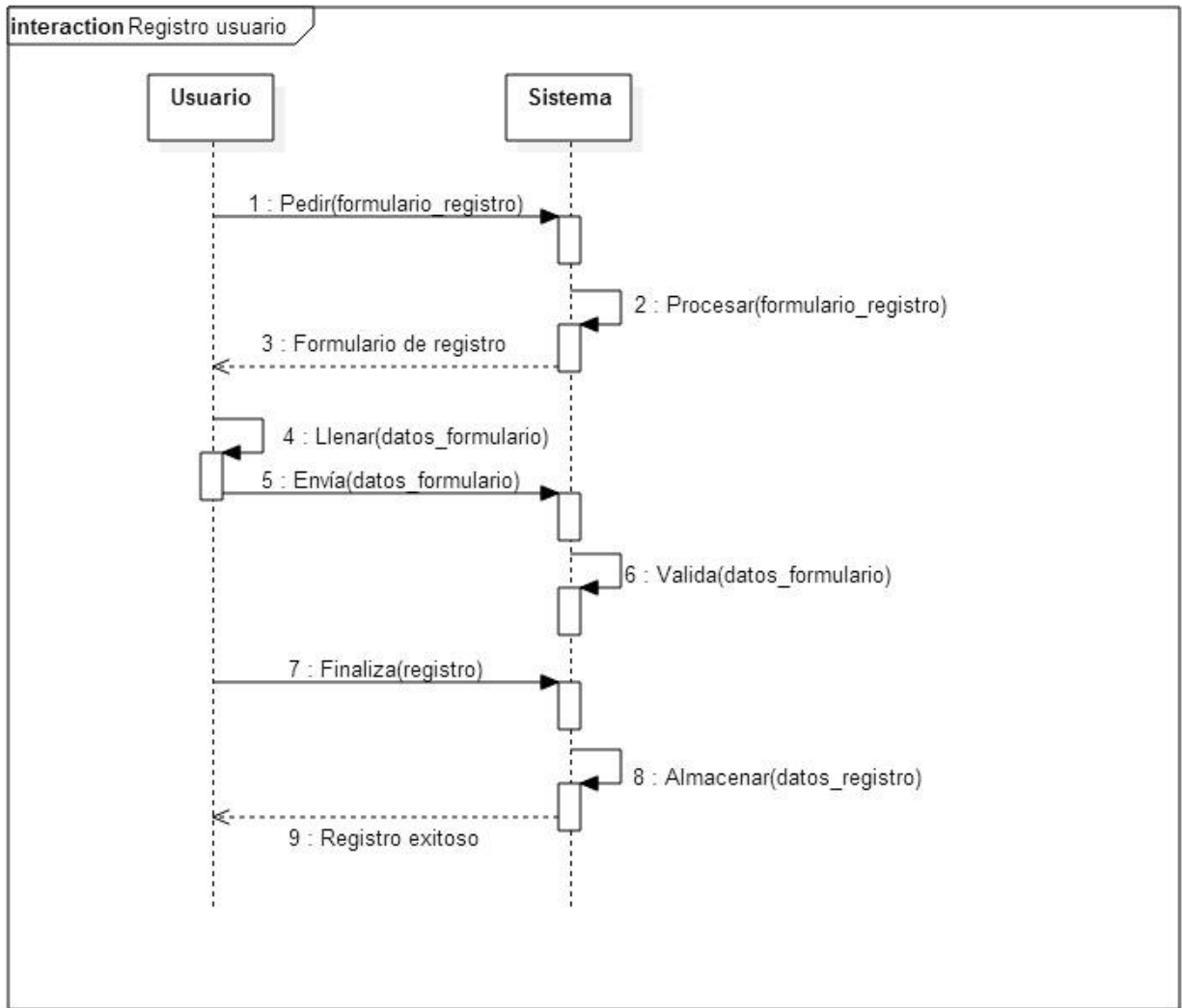


Figura 7.- Registro DS2

3.9.3 Agregar Grupo

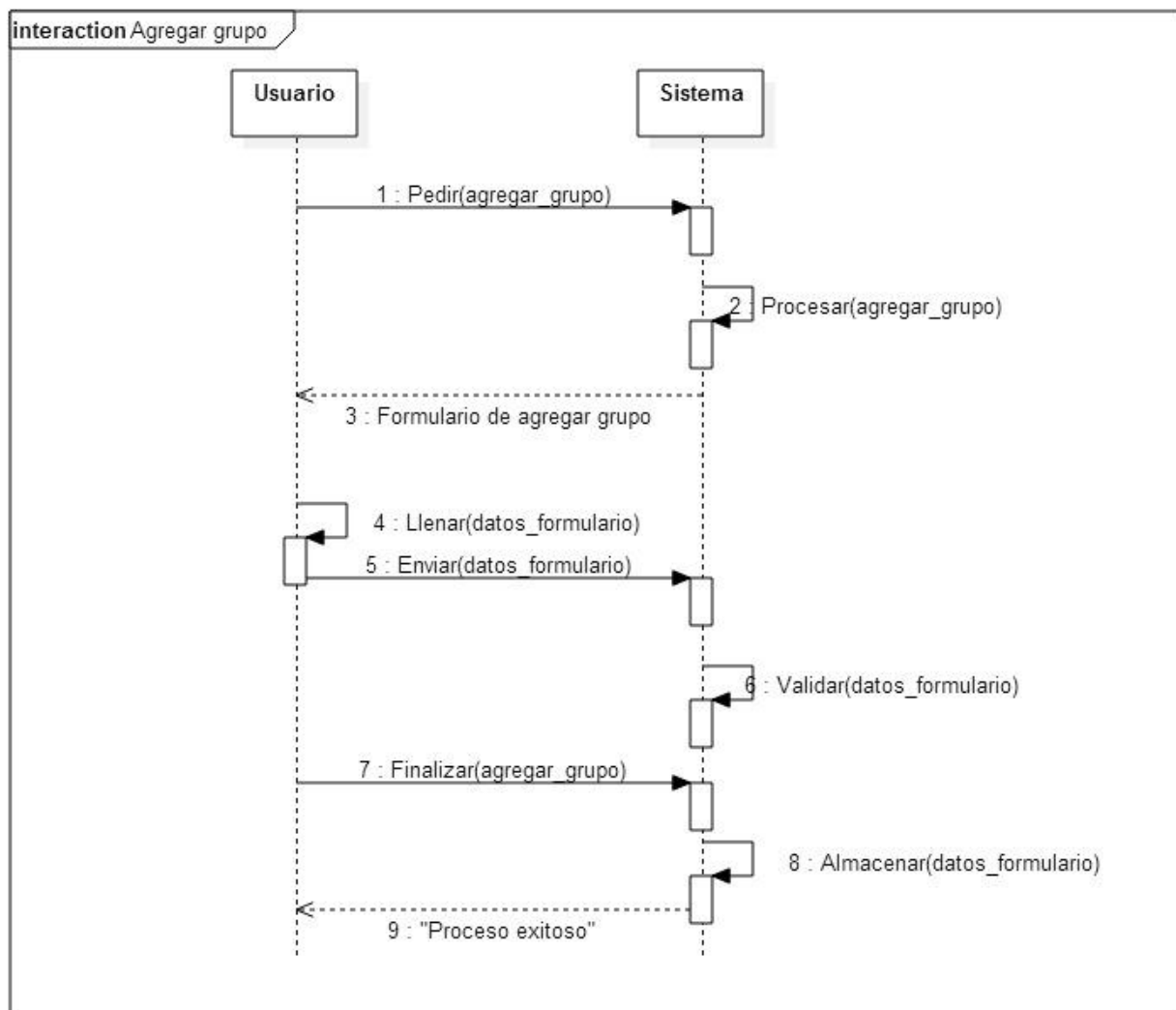


Figura 8.- Agregar grupo DS3

3.9.4 Invitación a Grupo.

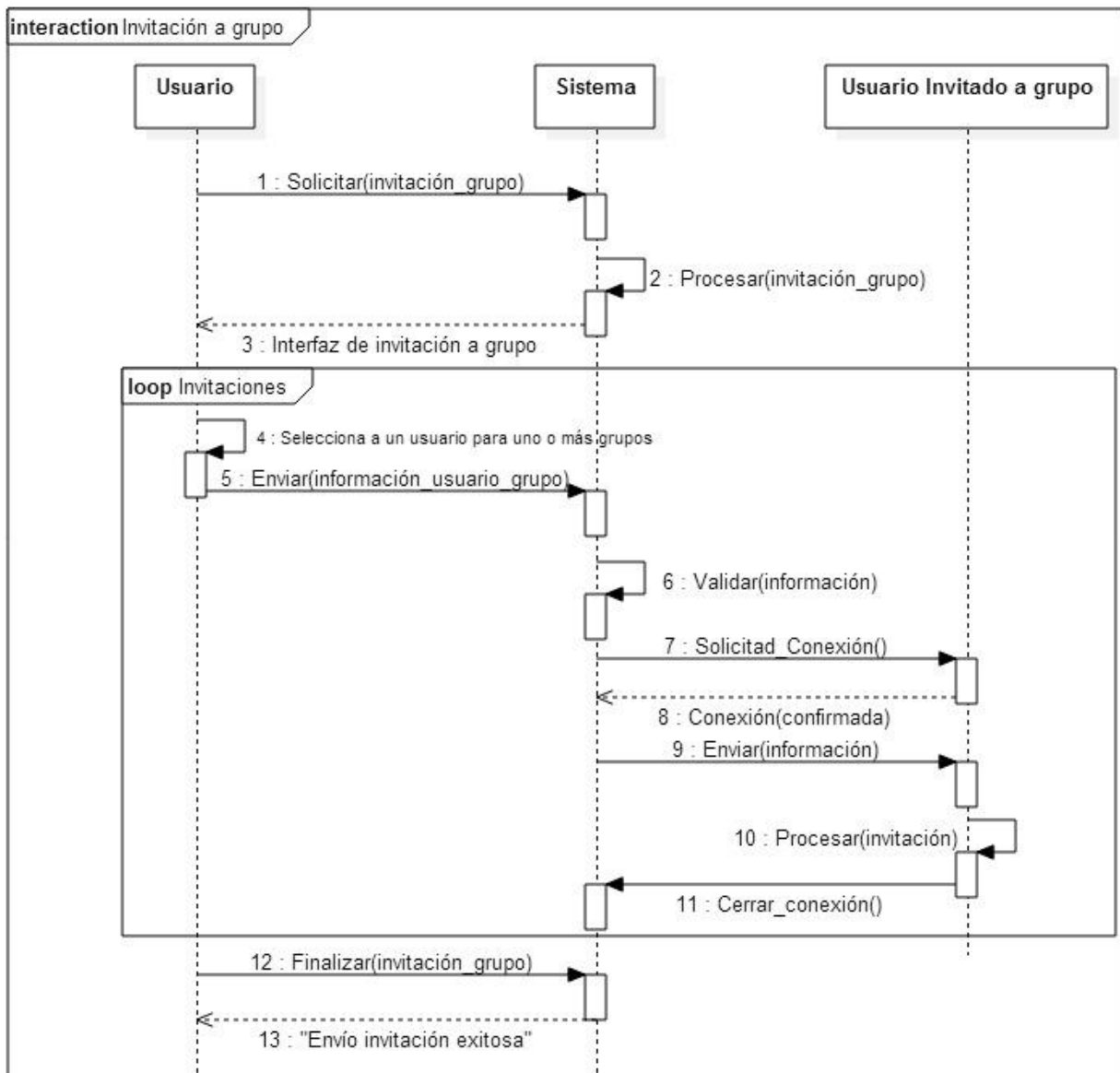


Figura 9.- Invitación a grupo DS4

3.9.5 Contestar Invitación

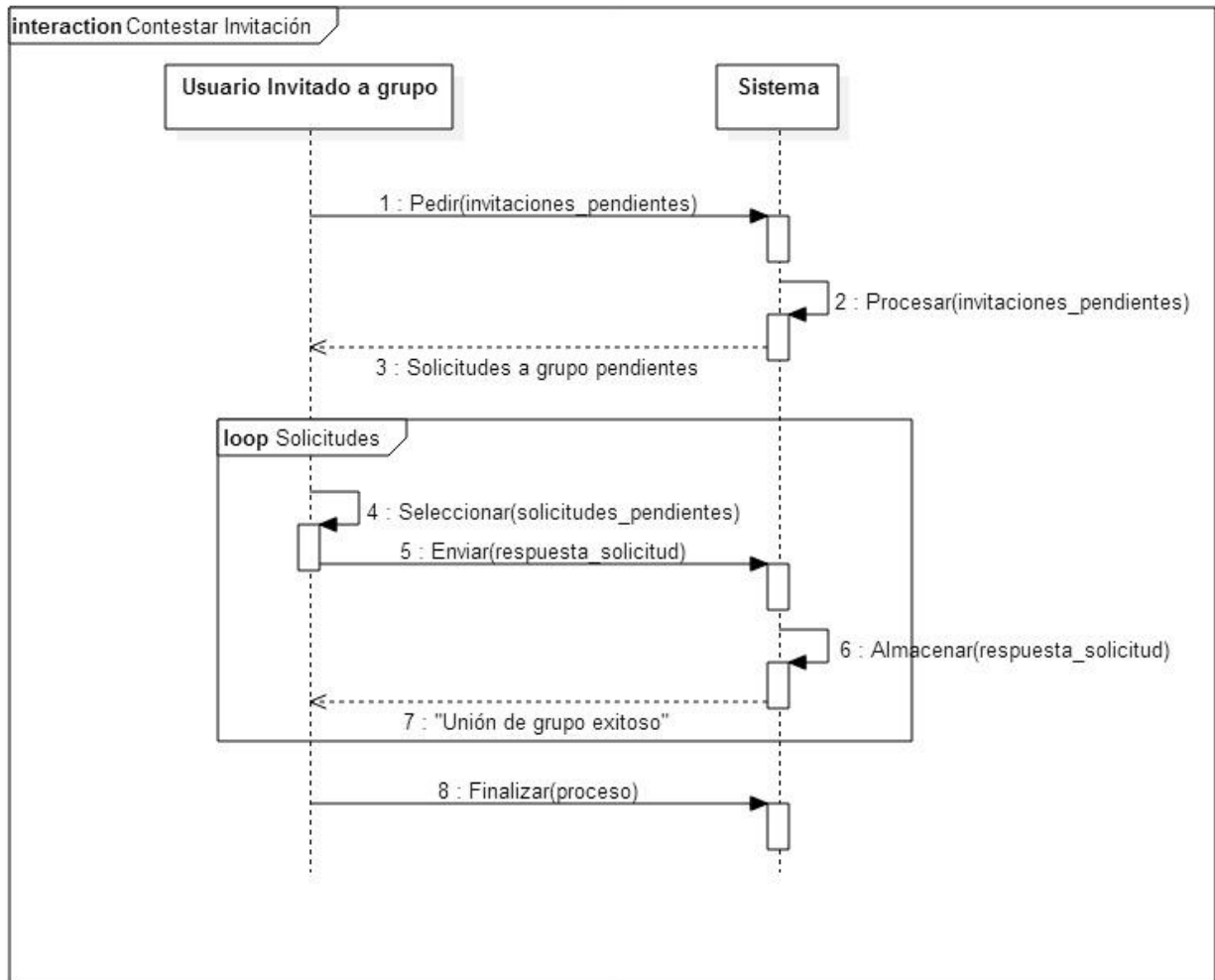


Figura 10.- Contestar invitación DS5

3.9.6 Administrar

información

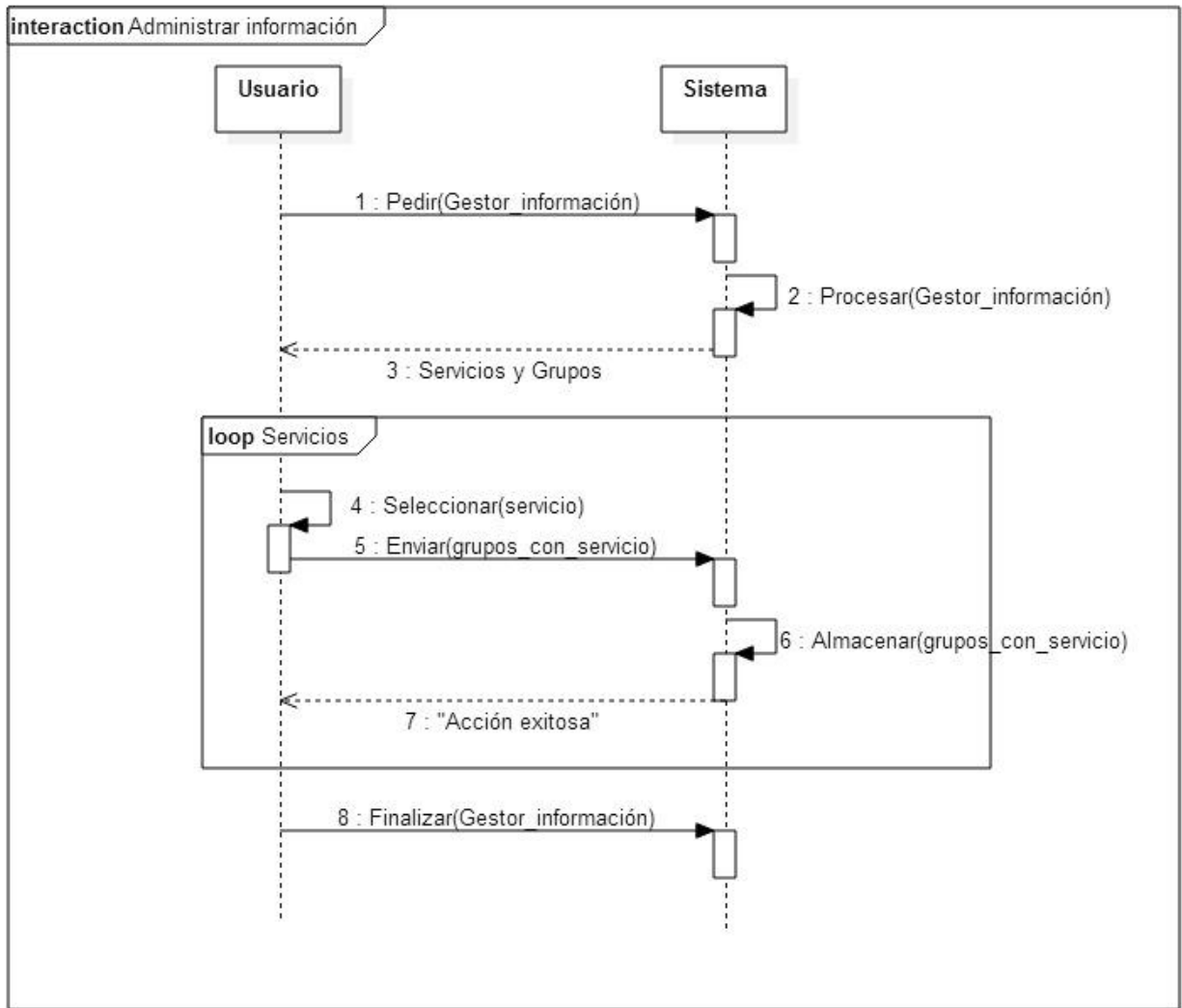


Figura 11.- Administrar información DS6

3.9.7 Actualizar nodos

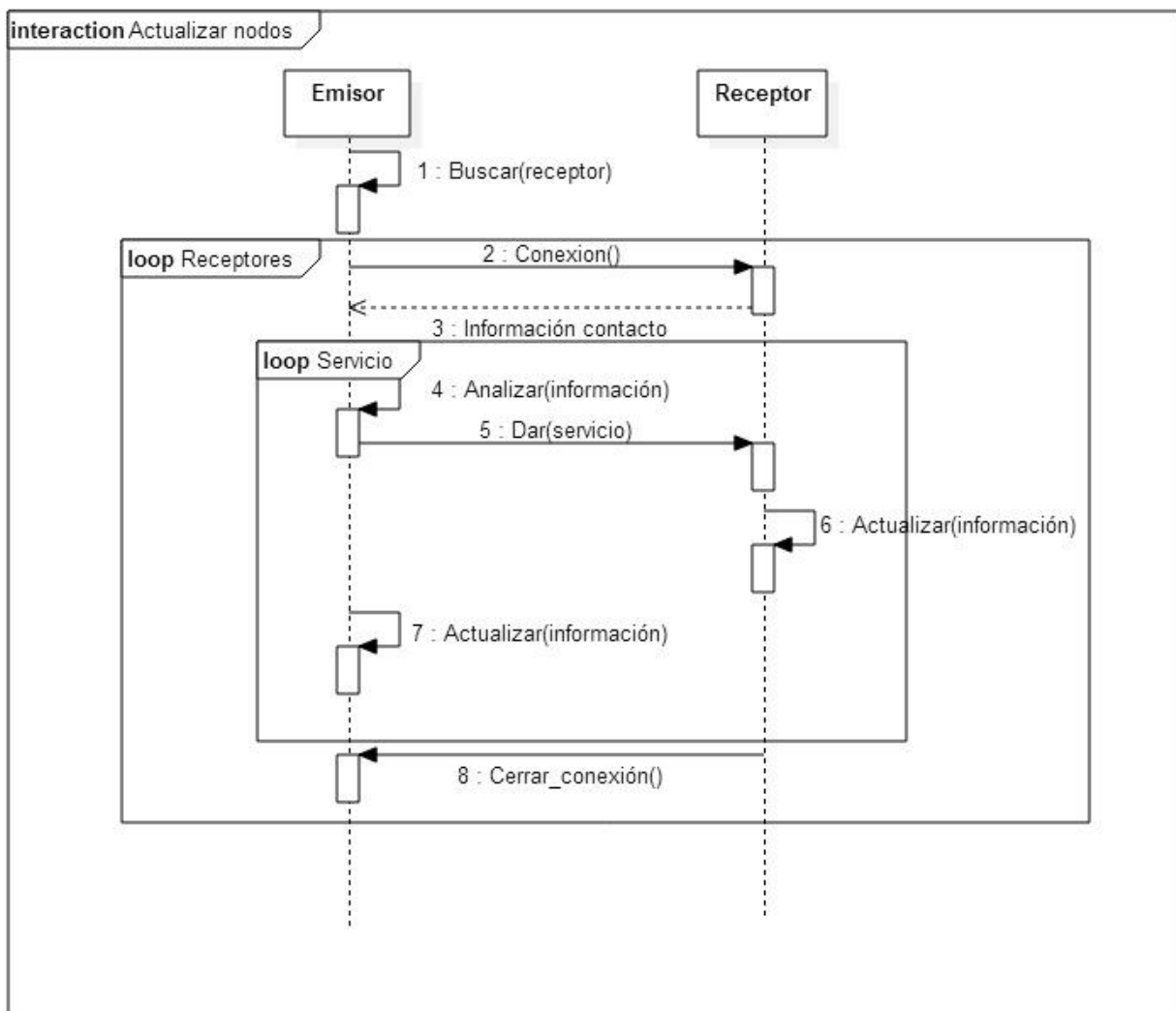
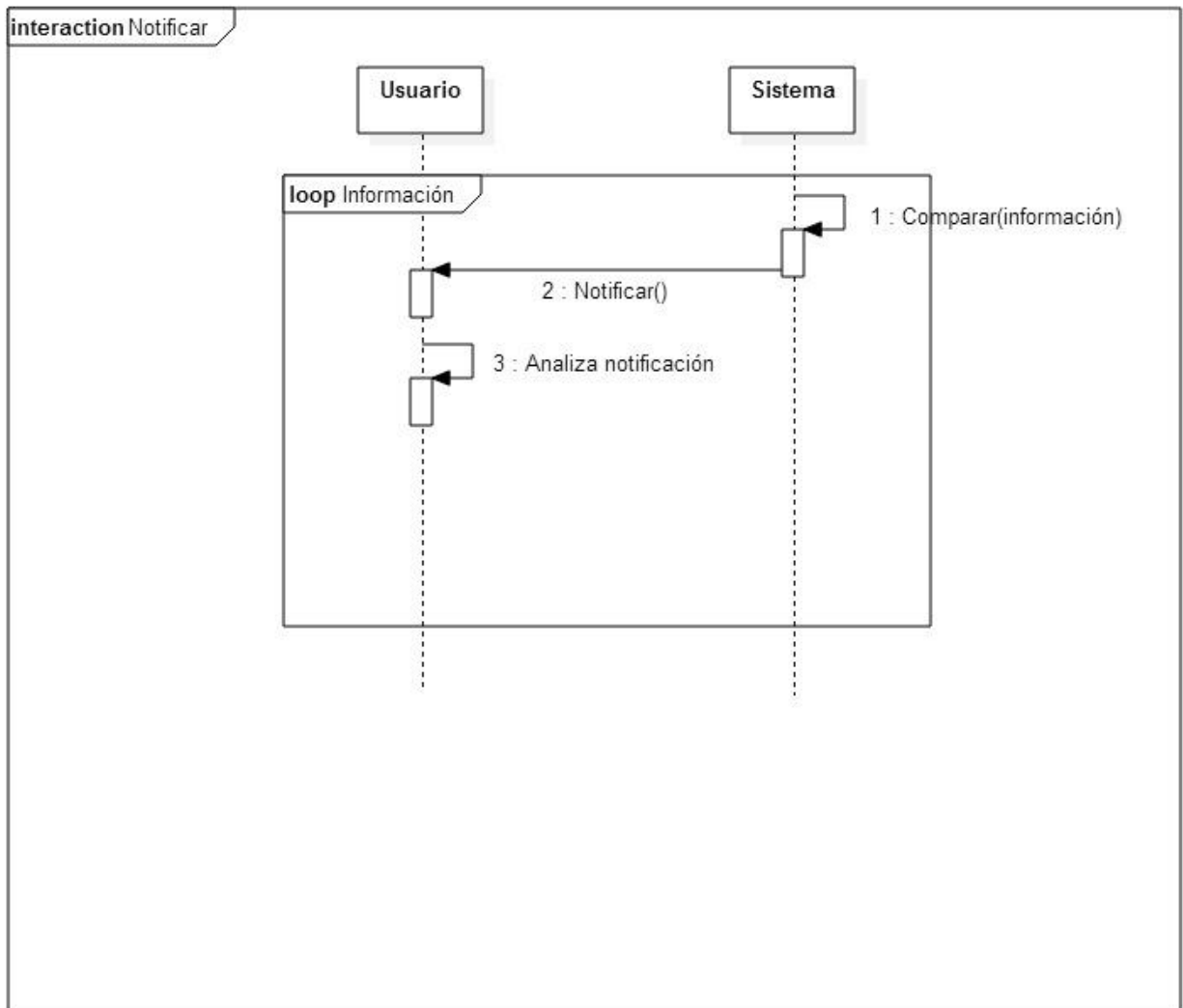


Figura 12.- Actualizar nodos DS7

3.9.8 Notificar



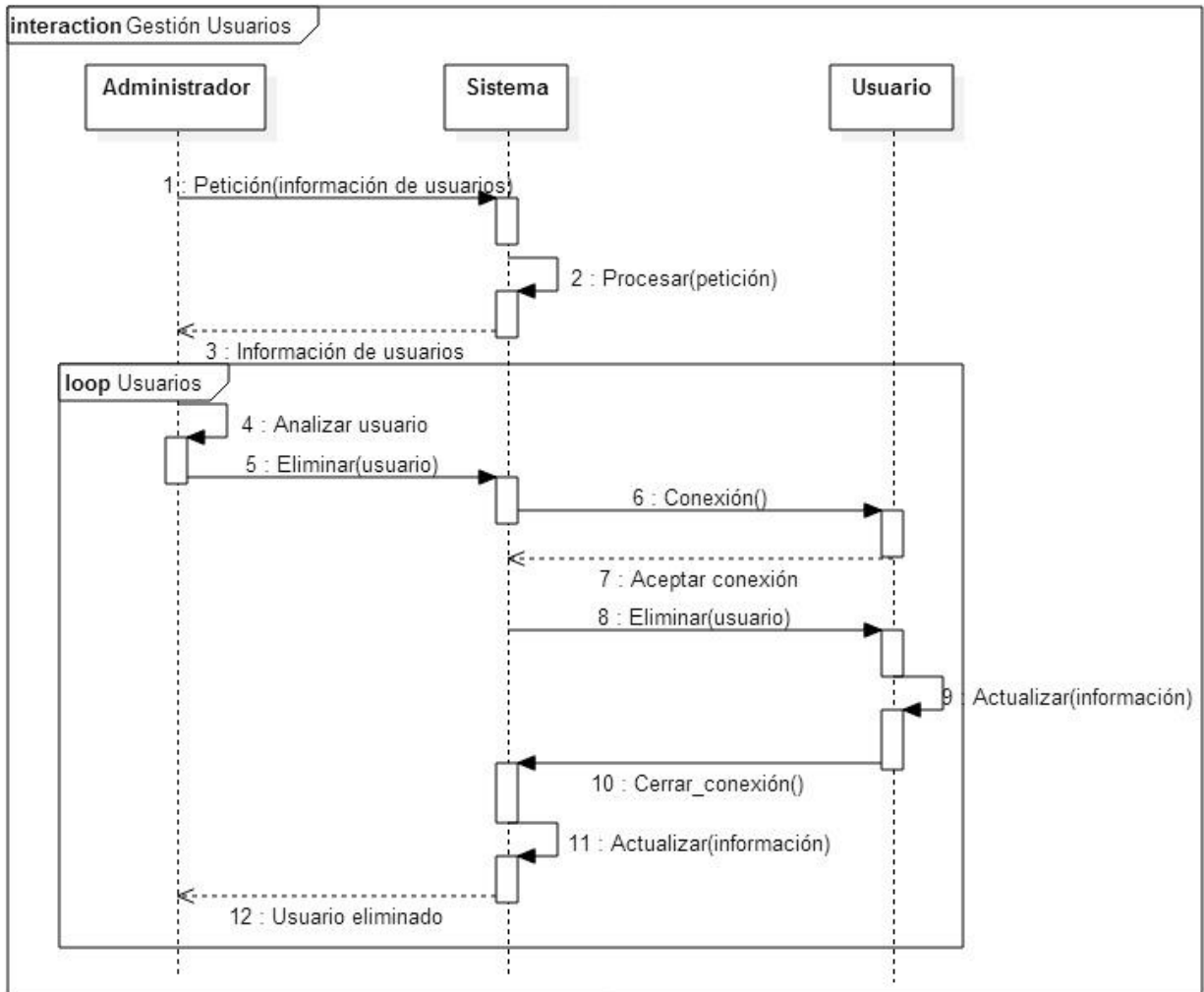


Figura 14.- Gestión de usuarios DS9

3.9.10 Salir

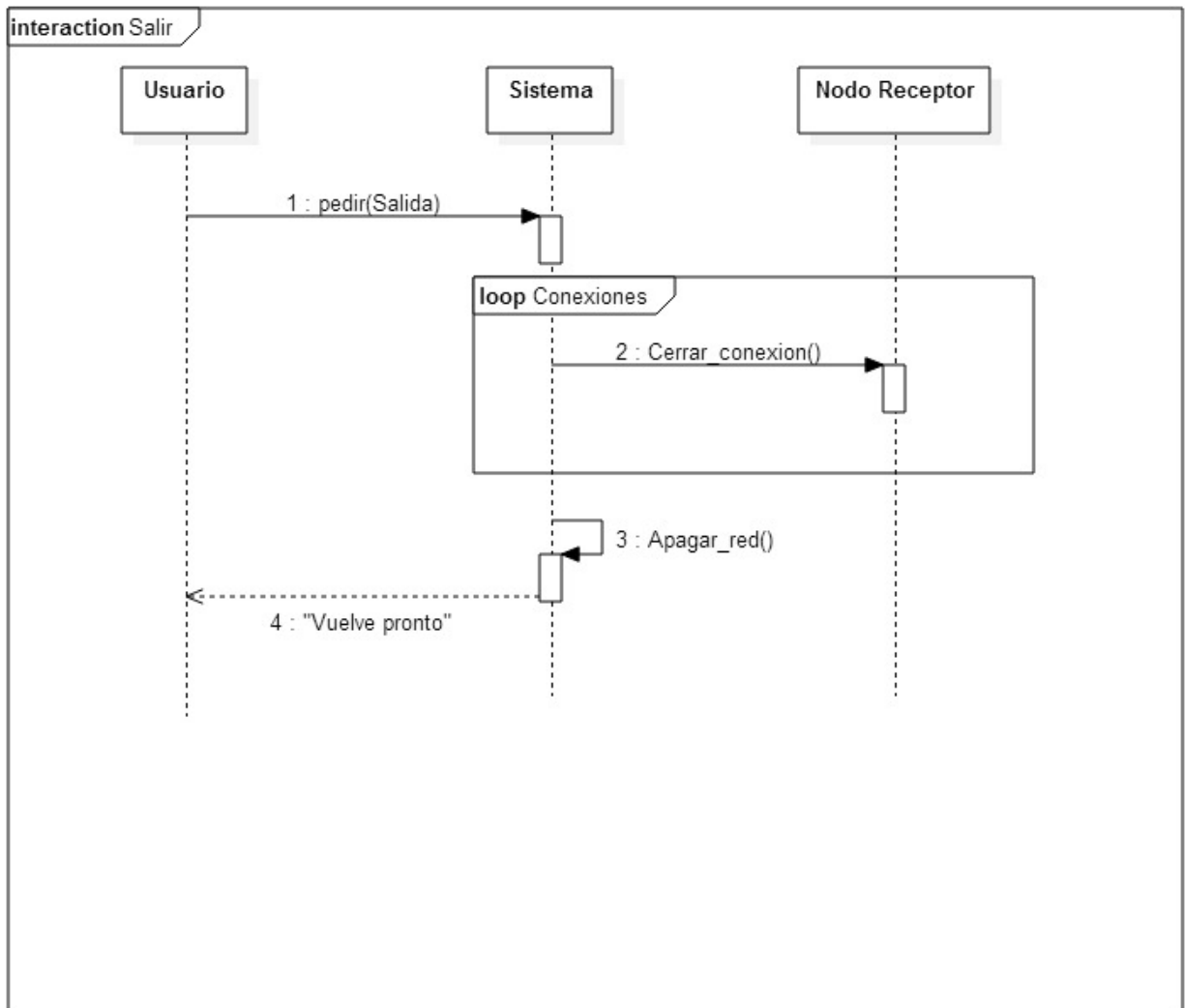


Figura 15- Salir DS10

3.10 Diagrama de Clases.

En el diagrama siguiente se mostrará la interacción y la descripción de las clases con las cuales cuenta el sistema, como se logra observar se cuenta con 5 clases: Perfil, Servicio, Nodo, Servidor y Conexión. Cada una de éstas cuenta con características en específico, los cuales son sus variables y/o métodos; los cuales servirán para el desarrollo del Sistema de Administración de Perfiles para redes MANET.

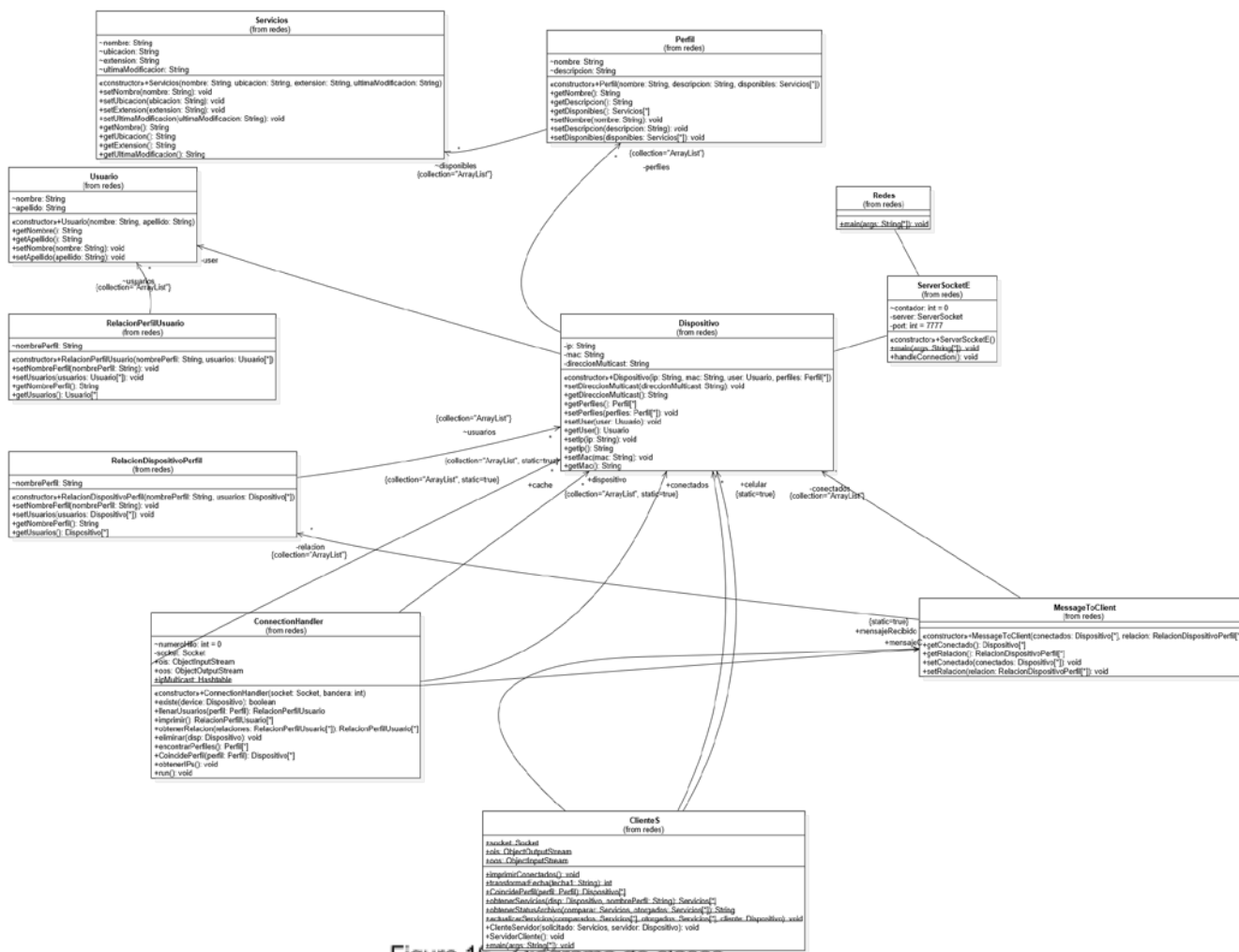


Figura 16.- Diagrama de clases

3.10.1 Descripción de Clases:

3.10.1.1 Perfil:

Perfil
-nombre: String -descripcion: String
+getNombre(): String +setNombre(nombre: String): void +getDescripcion(): String +setDescripcion(descripcion: Strin

Figura 17.- Clase Perfil

- **Variables:**

- **nombre:** Private, String. Nombre del usuario
- **descripcion:** Private, String. Descripción del usuario

- **Métodos:**

- **getNombre():** Método público que sirve para obtener el nombre.
- **setNombre():** Método público que sirve para colocar el nombre
- **getDescripcion():** Método público que sirve para obtener la descripción.
- **setDescripcion():** Método público que sirve para colocar la descripción.

3.10.1.2 Nodo:

Nodo
-mac_Address: String -nombre: String -protocolo: String
+getMAC(): String +setMAC(mac_Address: String): void +getNombre(): String +setNombre(nombre: String): void +getProtocolo(): String +pedirServicio(perfil: Perfil): Servicio

Figura 18.- Clase Nodo

- **Variables:**

- **mac_Address:** Private, String. Dirección Mac del nodo, el cual servirá como identificador único de usuario.
- **nombre:** Private, String. Nombre del nodo, el cual servirá para identificar a los diferentes usuarios entre sí.
- **protocolo:** Private, String. Protocolo con el cual cuenta el nodo y podrá interactuar con los demás nodos.
-

- **Métodos:**

- **getMAC():**Método público para obtener la dirección MAC.
- **setMAC():**Método público para colocar una dirección MAC.
- **getNombre():**Método público para obtener el nombre.
- **setNombre():**Método público para colocar el nombre.
- **getProtocolo():**Método público para obtener el protocolo fijado en el nodo.
- **pedirServicio():**Método público para llevar a cabo el servicio de transferencia por parte del nodo.

3.10.1.3 Servicio

Servicio
-nombre: String
-direccion: String
-count: int
-ultimaConsulta: Date
+getNombre(): String
+setNombre(nombre: String): void
+getDireccion(): String
+setDireccion(direccion: String): void
+getCount(): int
+iniciarCount(): void
+contar(): void
+getUltimaConsulta(): Date
+setUltimaConsulta(ultimaConsulta: Date): void

Figura 19.- Clase Servicio

- **Variables:**
 - **nombre:** Private, String. Nombre con el cual los diferentes usuarios van a identificar al servicio.
 - **dirección:** Private, String. Dirección en la cual va a estar alojado el servicio para su uso en el sistema.
 - **count:** Private, Int. Variable en el cuál se sabrá cuántas veces se ha llamado el servicio.
 - **ultimaConsulta:** Private, Date. Fecha en la que fue solicitada la última vez dicho servicio.
- **Métodos:**
 - **getNombre():** Método público para obtener el Nombre del Servicio.
 - **setNombre():** Método público para colocar un nombre al Servicio.
 - **getDireccion():** Método público para obtener la dirección en la cual está alojado el servicio
 - **setDireccion():** Método público para colocar la dirección en la cual va a estar alojado el servicio.
 - **getCount():** Método público para obtener la variable count que contiene las veces que se ha llamado dicho servicio
 - **iniciarCount():** Método público en el cual se le da valor inicial a la variable count.
 - **contar():** Método público en el cual se le suma uno al contador
 - **getUltimaConsulta():** Método público para obtener la variable fecha en la cual fue última vez consultado el servicio.

- **setUltimaConsulta():** Método público para colocar la fecha en la cual fue última vez consultado el servicio.

3.10.1.4 Servidor

Servidor
+obtenerServicio(nombreServicio: String): boolean
+ofrecerServicio(perfilCliente: Perfil): boolean

Figura 20.- Clase Servidor

- **Variables**
- **Métodos:**
 - **obtenerServicio():** Método público que obtiene un servicio en específico del Nodo Servidor.
 - **ofrecerServicio():** Método público que manda un servicio específico a un cliente específico, el cual esté en el área de red del Nodo Servidor.

3.10.1.5 Conexión

Conexion
-fecha: Date
+obtenerConexion(nodo: Nodo, seccion: Seccion): boolean
+cerrarConexion(): boolean
+getFecha(): Date
+setFecha(fecha: Date): void

Figura 21.- Clase Conexión

- **Variables:**
 - **fecha:** Private, Date. Variable fecha la cual contiene el valor de fecha en la cual fue establecida la conexión entre el Nodo y el Servidor
- **Métodos:**
 - **obtenerConexion():** Método público el cual establece la conexión entre el Nodo y el Servidor especificado en el método.
 - **cerrarConexion():** Método público el cual finaliza la conexión establecida entre el Nodo y el Servidor.

- **getFecha():** Método público que obtiene el valor fecha en la cual fue establecida la conexión.
- **setFecha():** Método público para colocar la fecha en la que fue establecida la conexión entre un Nodo y un Servidor.

3.11 Diagrama de Actividades.

3.11.1 Registro de Usuario

Descripción de Actividades:

- 1.- **Opción Registrarse:** Se elige la opción de Registro entre una gama de posibilidades.
- 2.- **Ingresar Datos:** Se ingresan los datos que se requieran.
- 3.- **Validar datos:** Se validan los datos introducidos en la actividad anterior.
- 4.- **Confirmar:** Se confirman los datos para un registro exitoso en el sistema.

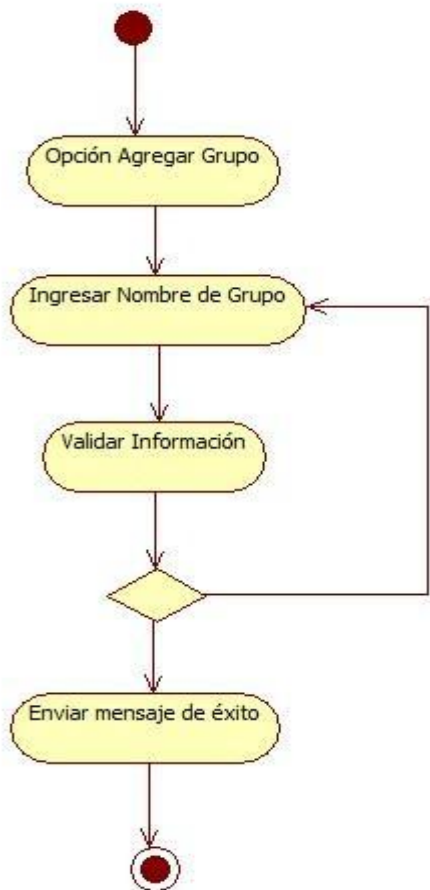


Figura 22.- Registro Usuario DA1

3.11.2 Agregar Grupo

Descripción de Actividades:

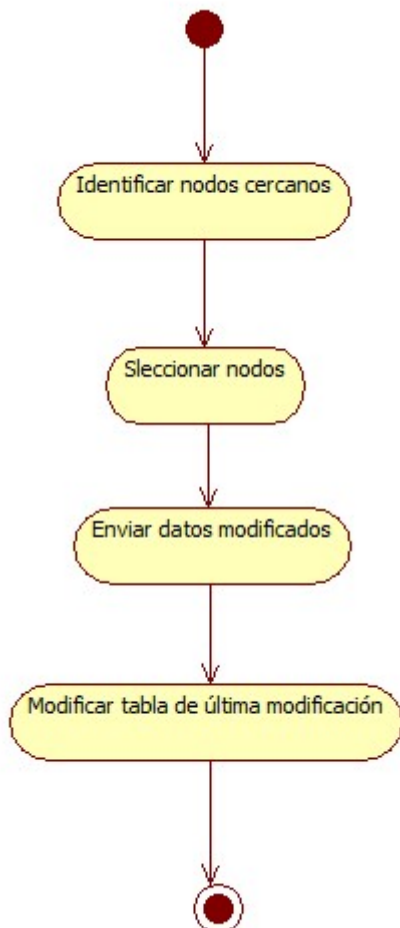
- 1.- **Opción Agregar Grupo:** Se elige la opción de Agregar Grupo entre una gama de posibilidades.

2.- Ingresar Nombre de Grupo: Se ingresa el Nombre del Grupo que se quiera agregar.

3.- Validar Información: Se validan los datos introducidos en la actividad anterior.

4.- Enviar mensaje de texto: Se envía un mensaje de éxito una vez que los datos introducidos son válidos en la actividad anterior (Valida Información).

Figura 23.- Agregar grupo DA2



3.11.3 Identificar nodos cercanos.

Descripción de Actividades:

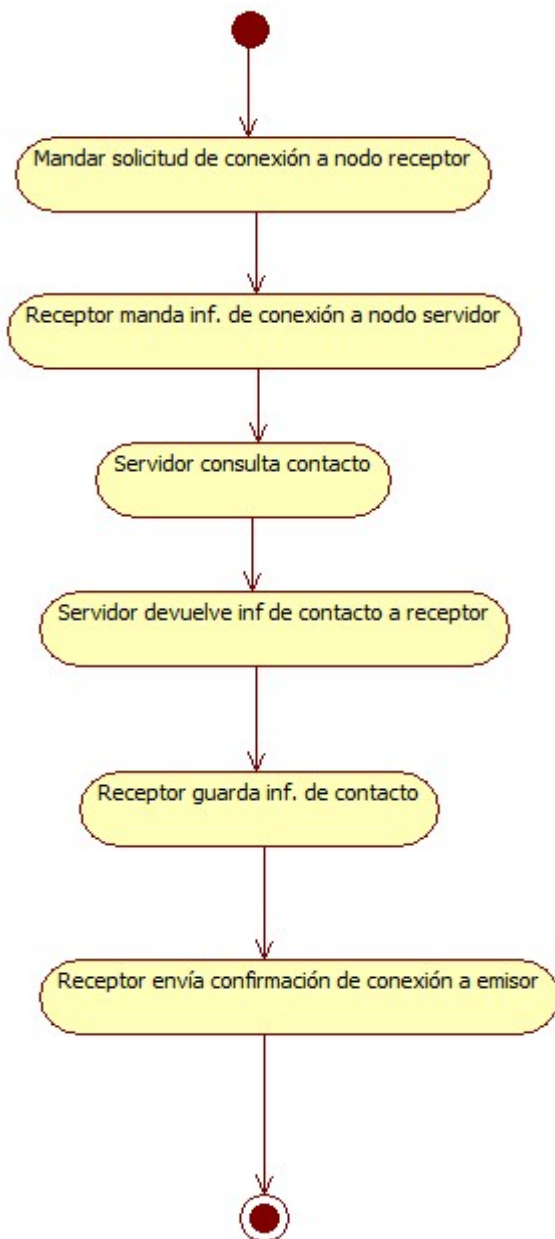
1.- Identificar nodos cercanos: Se hace una debida inspección que nos permita encontrar los nodos cercanos tomando como base a un nodo “central”, para llevar operaciones conjuntas posteriores.

2.- Seleccionar Nodos: Se seleccionan los nodos que cumplan ciertos requisitos y características que participaran con el nodo “central” que tomó como base para seleccionar a los nodos cercanos.

3.- Enviar datos modificados: Se hace un intercambio de información, así como una transferencia también, sobre los datos que se han modificado hasta el último momento.

4.- Modificar tabla de última modificación: Se modifica la tabla con la información actualizada del punto anterior, de esa manera, la información siempre está disponible y al día.

Figura 23.- Identificar nodos DA3



3.11.4 Mandar solicitud de conexión a nodo receptor.

Descripción de Actividades:

1.- Mandar solicitud de conexión a nodo receptor: Se hace la conexión a un nodo vecino del cual se quiera disponer de su información.

2.- Receptor manda información de conexión a nodo servidor: Se hace la transferencia de información de conexión.

3.-Servidor consulta contacto: Se hace la consulta del contacto del nodo receptor con la información del nodo servidor.

4.-Servidor devuelve información de contacto a receptor: El nodo servidor

devuelve la información que encontro del contacto al nodo receptor correspondiente.

5.-Receptor guarda información de contacto:El nodo receptor actualiza su información de contacto con la que recibió del nodo servidor.

6.- Receptor envía confirmación de conexión a emisor: El nodo receptor envía una confirmación de conexión a un tercer nodo que se quiera conectar o pidió la conexión.

3.11.5 Seleccionar Invitaciones a Grupo.

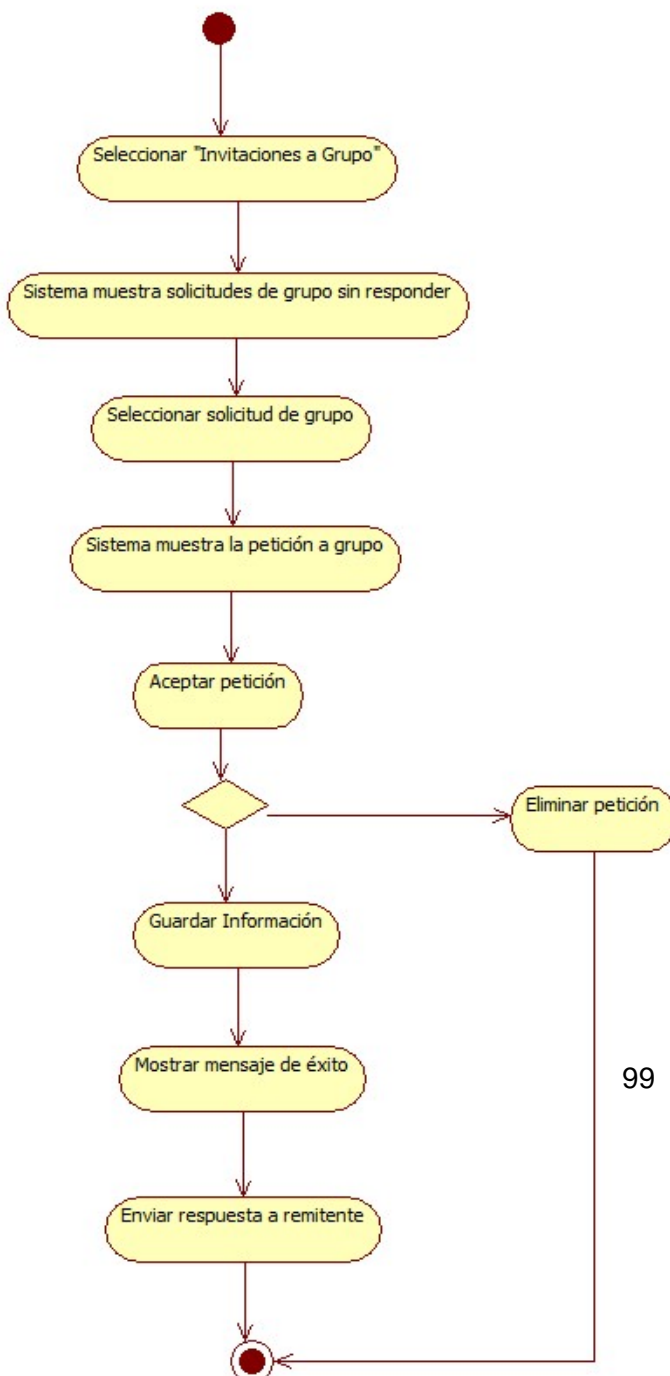
Descripción de Actividades:

1.- Seleccionar “Invitaciones a Grupo”: Se selecciona la opción desde el sistema “Invitaciones a Grupo”.

2.- Sistema muestra solicitudes de grupo sin responder: Se hace la muestra de todas las solicitudes que no están respondidas y/o contestadas.

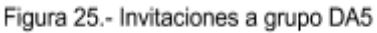
3.- Seleccionar solicitud de grupo: Se selecciona la solicitud que se requiera en ese momento por parte del usuario.

4.- Sistema muestra la petición a grupo: Se muestra la información que contiene esa selección, como el grupo al que va dirigida.



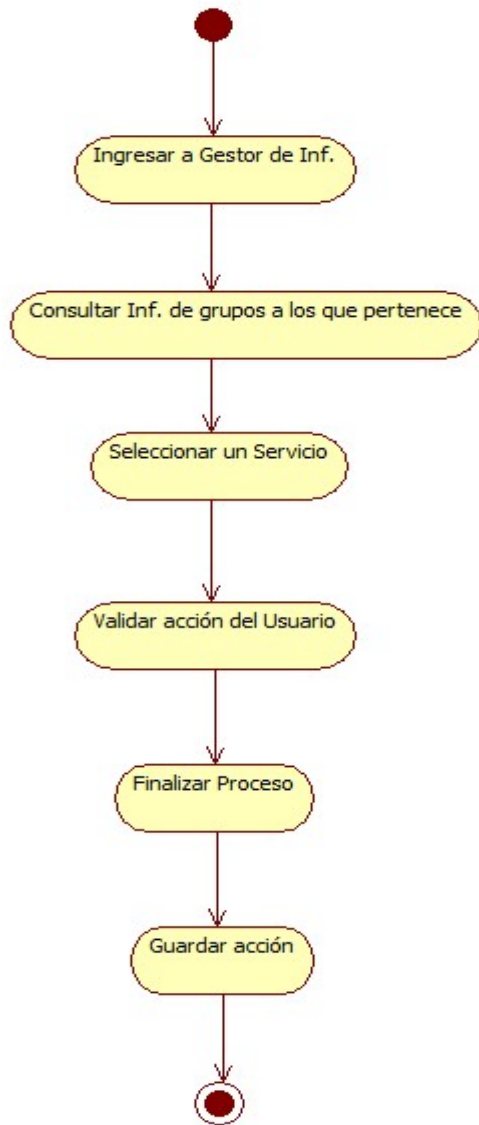
5.- Aceptar petición: Se selecciona si se acepta o no la petición de invitación a grupo.

6.- Eliminar petición: Se elimina la petición de la lista de invitaciones.

7.-  **Guardar información:** Se guarda la información de la invitación para que posteriormente sea aceptada.

8.- Mostrar mensaje de éxito: Se muestra un mensaje que la invitación ha sido aceptada por lo que se procederá a que el usuario tenga un nuevo grupo.

9.- Envía respuesta a remitente: Se envía una respuesta al que dio de alta la invitación para que esté enterado que un nuevo usuario ha ingresado al grupo.



3.11.6 Ingresar a Gestor de Información.

Descripción de Actividades:

1.- Ingresar a Gestor de Información: Se ingresa al gestor para operaciones de búsqueda y validación.

2.- Consultar Información de grupos a los que pertenece: Se consulta la información de grupos de un usuario si este tiene al menos uno.

3.- Seleccionar un Servicio: Se elige un servicio que posteriormente será utilizado.

4.- Validar acción del Usuario: La acción del usuario es verificado por el sistema y es aceptada o no.

5.- Finalizar Proceso: Se finaliza las acciones que está llevando a cabo el usuario o el administrador en este módulo.

6.- Guardar Acción: Se guardan las modificaciones que el usuario hizo en este apartado.

3.12 Diagrama de Estados.

3.12.1 Registro al Sistema.

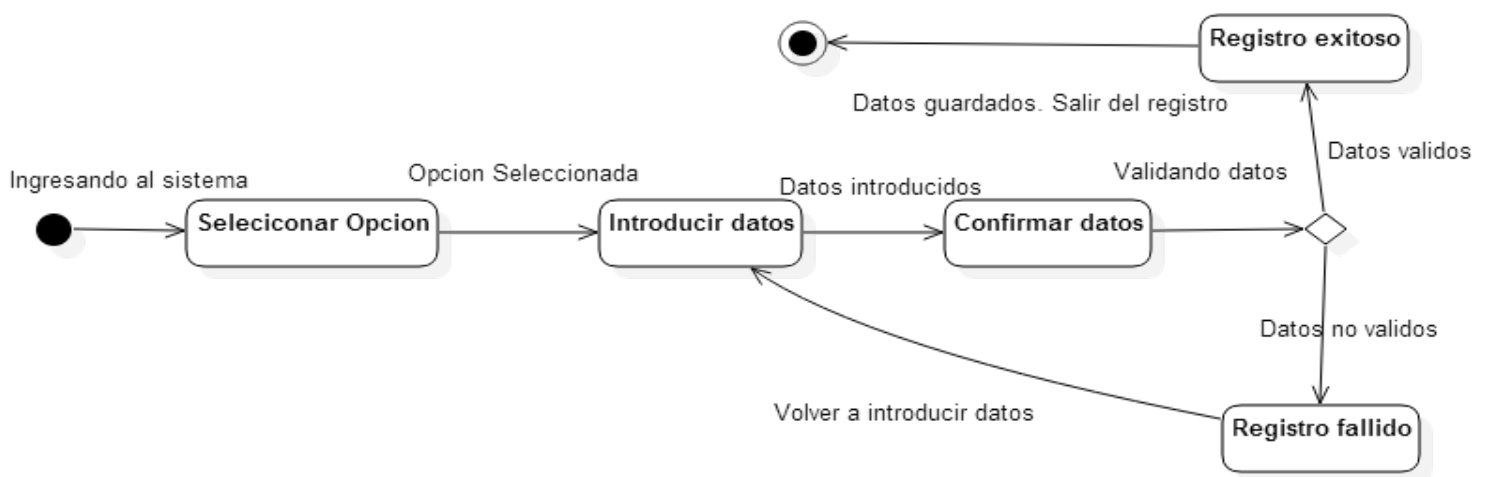


Figura 26.- Registro de usuario DE1

Descripción de Estados:

- 1.- **Selecionar Opción:** El usuario selecciona la opción de registro al sistema entre una lista de opciones.
- 2.- **Introducir datos:** El usuario introduce los datos que el sistema requiera para un registro exitoso.
- 3.- **Confirmar datos:** El usuario confirma los datos que introdujo sean correctos y válidos.
- 4.- **Registro Exitoso:** El usuario se registró satisfactoriamente al sistema al introducir datos válidos.
- 5.- **Registro Fallido:** El usuario no se registró satisfactoriamente al sistema al introducir datos no válidos. Por lo tanto, vuelve al estado Introducir datos.

3.12.2 Agregar Grupo.

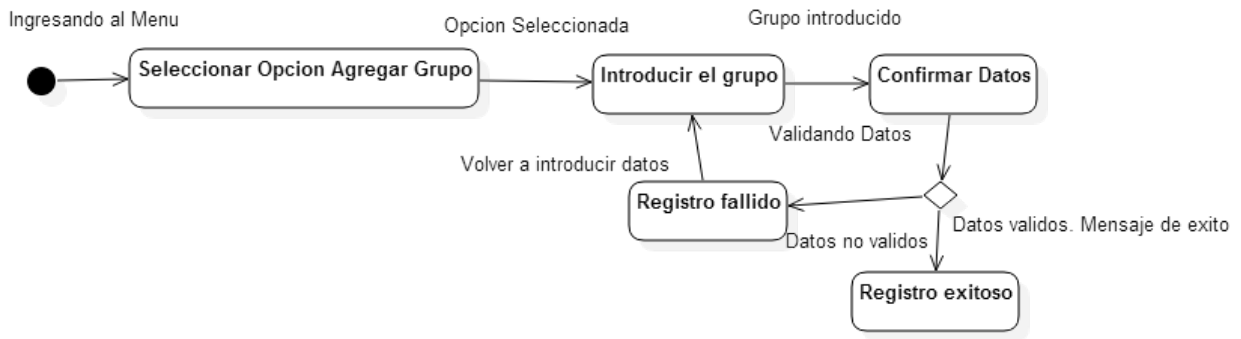


Figura 27.- Agregar grupo DE2

Descripción de Estados:

- 1.- **Seleccionar Opción Agregar Grupo:** El usuario selecciona la opción de agregar grupo al sistema entre una lista de opciones.
- 2.- **Introducir el grupo:** El usuario introduce el grupo que el sistema requiera para un registro exitoso.
- 3.- **Confirmar datos:** El usuario confirma los datos que introdujo sean correctos y válidos.
- 4.- **Registro Exitoso:** El grupo se registró satisfactoriamente al sistema al introducir datos válidos.
- 5.- **Registro Fallido:** El grupo no se registró satisfactoriamente al sistema al introducir datos no válidos. Por lo tanto, vuelve al estado Introducir el grupo.

3.13 Administrador

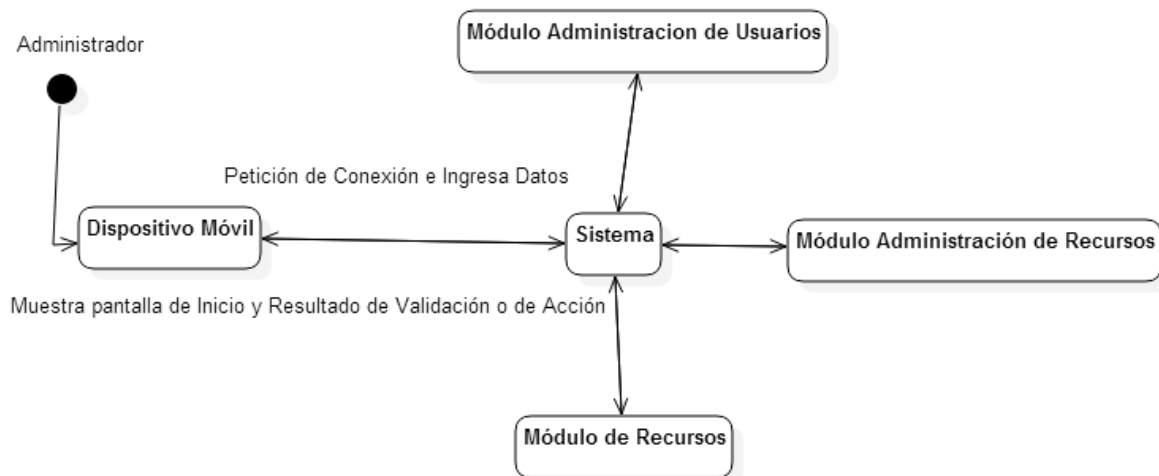


Figura 28.- Administrador DE3

Descripción de Estados:

- 1.- **Dispositivo Móvil:** Estado que indica que el administrador hará uso de un dispositivo móvil para poder acceder al sistema y realizar las acciones que convenga necesarias.
- 2.- **Módulo Administración de Usuarios:** Estado que indica la manipulación de toda la administración de usuarios.
- 3.- **Módulo de Recursos:** Estado que indica la manipulación de la base de datos, así como del ser servidor.
- 4.- **Sistema:** Estado que indica que el administrador hará uso del sistema por medio del dispositivo móvil que usará.

5.- **Módulo Administración de Recursos:** Estado que indica la manipulación de toda la administración de recursos.

3.12.4 Usuario

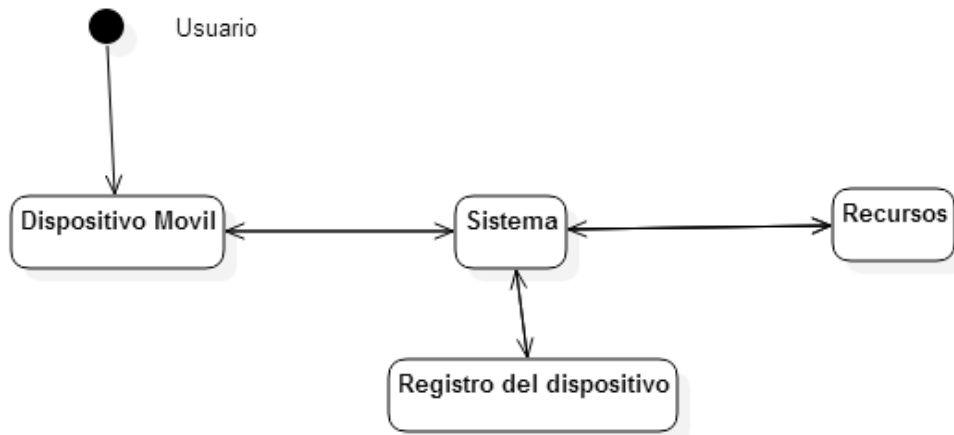


Figura 29.- Usuario DE4

Descripción de Estados:

- 1.- **Dispositivo Móvil:** Estado que indica que el usuario hará uso de un dispositivo móvil para poder acceder al sistema y realizar las acciones que convenga necesarias.
- 2.- **Sistema:** Estado que indica que el usuario hará uso del sistema por medio del dispositivo móvil que usará.
- 3.- **Recursos:** Estado que indica la manipulación de la base de datos, así como del ser servidor.
- 4.- **Registro del dispositivo:** Estado que indica que se requerirá dar de alta un dispositivo con el cual el usuario estará interactuando dentro del sistema

3.12.5 Aplicaciones

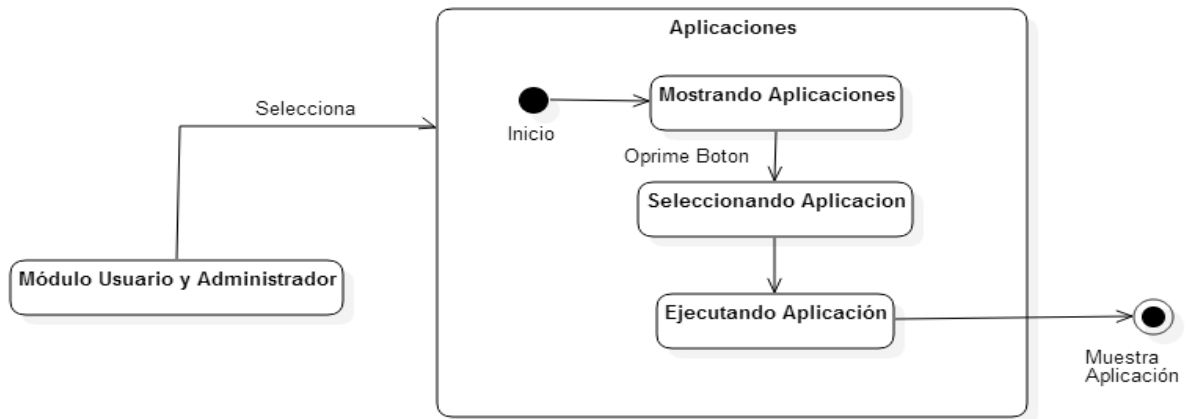


Figura 30.- Aplicaciones DE5

Descripción de Estados:

- 1.- **Módulo Usuario y Administrador:** Estado que indica que el usuario y/o administrador jugarán un papel importante dentro de este proceso, al mismo tiempo que seleccionan apartado de Aplicaciones.
- 2.- **Mostrando Aplicaciones:** Estado que indica que se desplegaran las aplicaciones que hay en ese momento.
- 3.- **Seleccionando Aplicación:** Estado que indica que se elegirá un aplicación para su posterior ejecución.
- 4.- **Ejecutando Aplicación:** Estado que indica que se está ejecutando la aplicación seleccionada por parte del usuario o administrador.

3.12.6 Cierre de Sesión

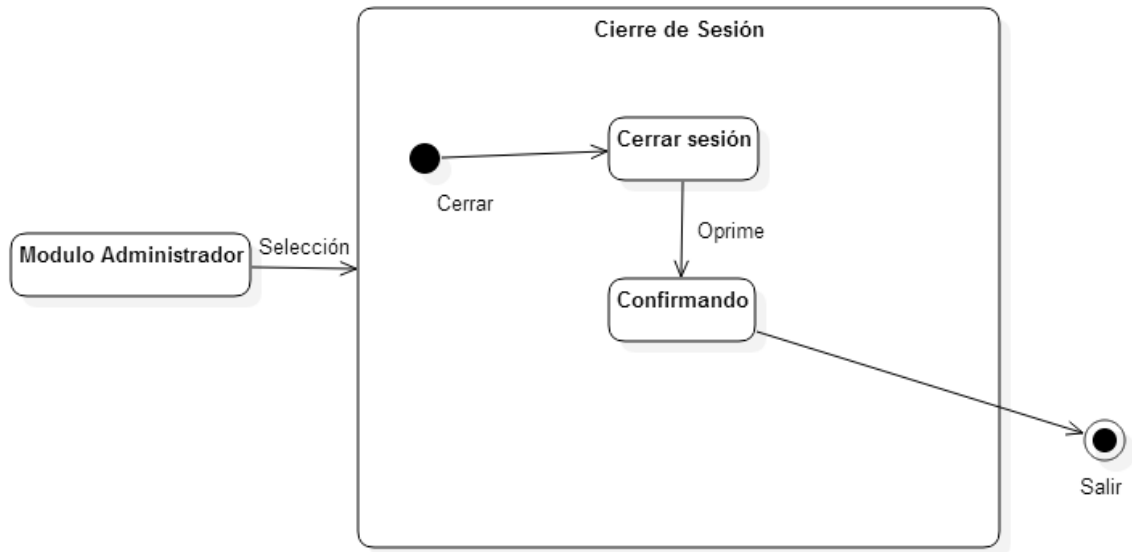


Figura 31.- Cierre de Sesión DE6

Descripción de Estados:

- 1.- **Módulo Administrador:** Estado que indica que el administrador jugará un papel importante dentro de este proceso, al mismo tiempo que seleccionan apartado de Cierre de Sesión.
- 2.- **Cerrar Sesión:** Estado que indica que finaliza la sesión activa que el administrador tiene dentro del sistema.
- 3.- **Confirmando:** Estado que indica que se confirma la finalización de dicha sesión y se prepara todo para salir del sistema.

3.12.7 Registro Dispositivo de Usuario

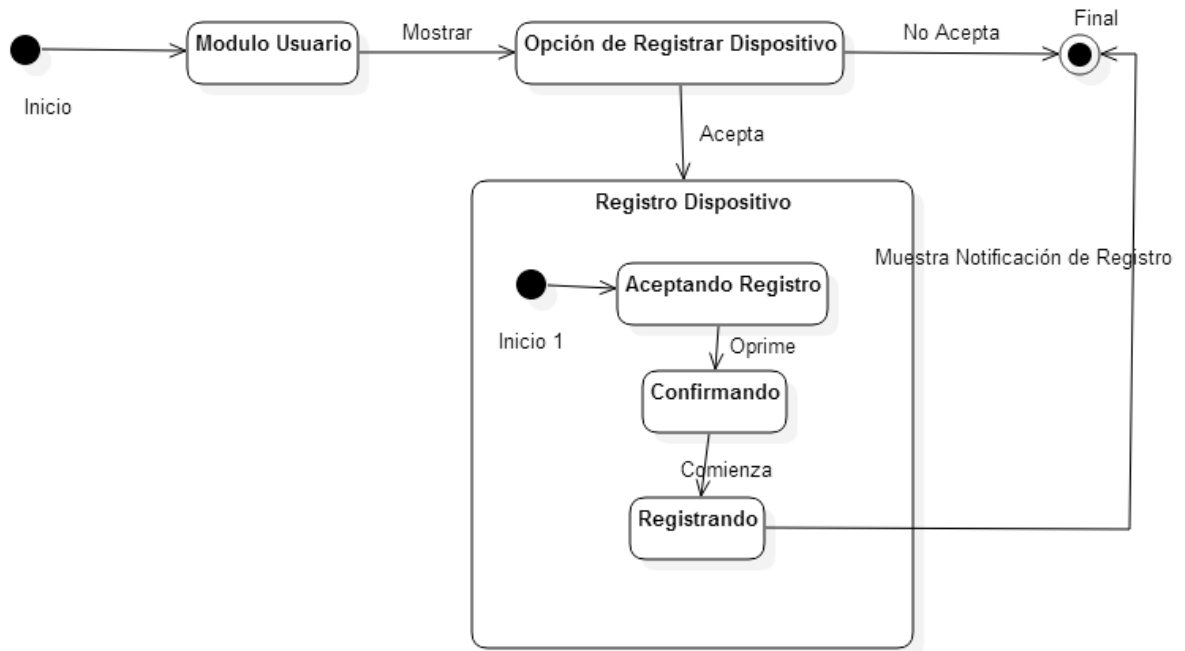


Figura 32.- Registro dispositivo de Usuario DE7

Descripción de Estados:

- 1.- **Módulo Usuario:** Estado que indica que el usuario jugará un papel importante dentro de este proceso.
2. **Opción de Registrar Dispositivo:** Estado que indica que se procede con la alta del dispositivo que usará el usuario dentro del sistema.
- 3.- **Aceptando Registro:** Estado que indica que se da un visto bueno al registro del dispositivo.
- 4.- **Confirmando:** Estado que indica que se confirma que la alta del dispositivo es correcta.
- 5.- **Registrando:** Estado que indica que una vez que la confirmación ha sido positiva se procederá con el registro.

3.12.8 Modificar Perfil

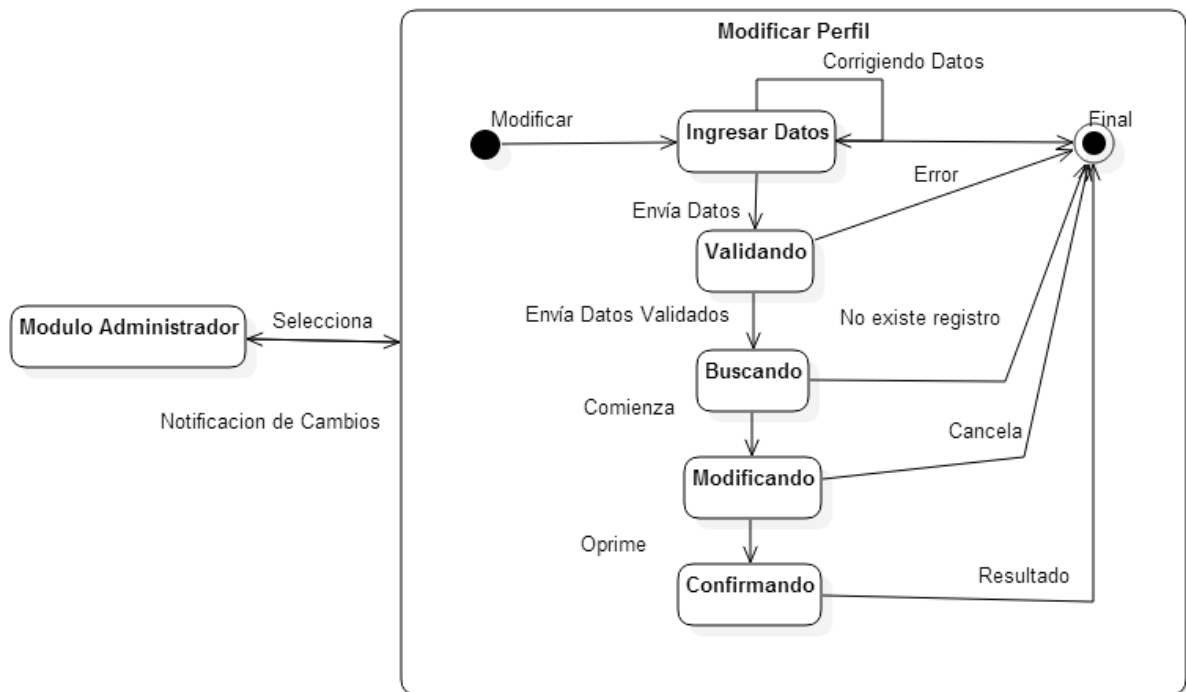


Figura 33.- Modificar Perfil DE8

Descripción de Estados:

- 1.- **Módulo Administrador:** Estado que indica que el administrador jugará un papel importante dentro de este proceso.
- 2.- **Ingresar Datos:** Estado que indica que se registran los datos correspondientes.
- 3.- **Validando:** Estado que indica que se validan los datos ingresados.
- 4.- **Buscando:** Estado que indica que buscan los datos ingresados.
- 5.- **Modificando:** Estado que indica que se modifican los datos ingresados.
- 6.- **Confirmando:** Estado que indica la confirmación de todo lo que se llevó a cabo anteriormente.

3.12.9 Búsqueda Usuario/Recurso

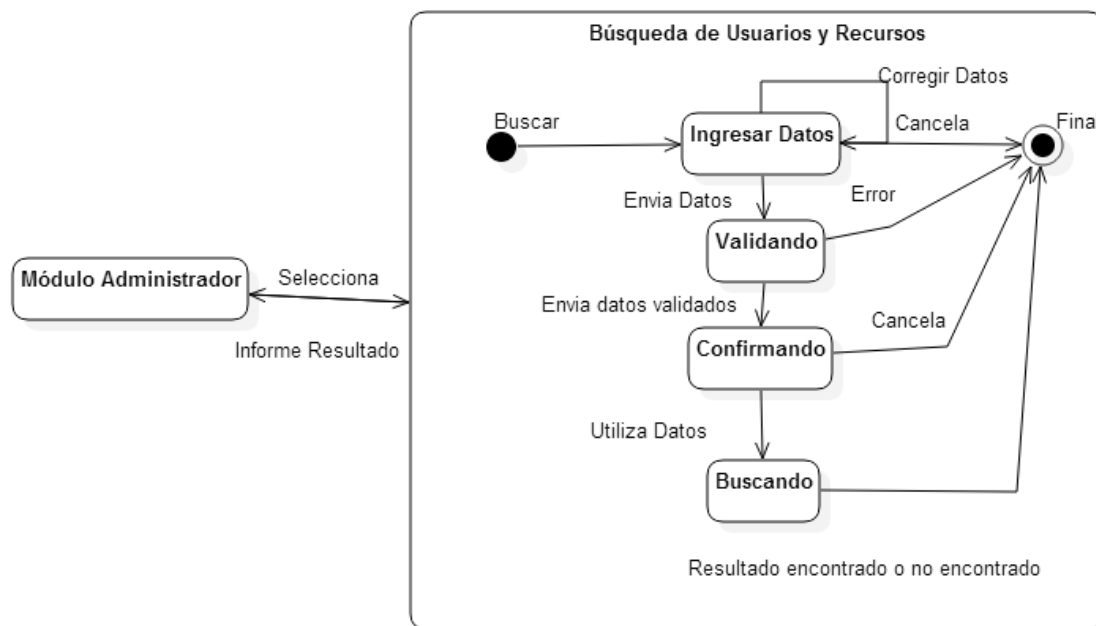


Figura 34.- Búsqueda usuario DE9

Descripción de Estados:

- 1.- **Módulo Administrador:** Estado que indica que el administrador jugará un papel importante dentro de este proceso.
- 2.- **Ingresar Datos:** Estado que indica que se registran los datos correspondientes.
- 3.- **Validando:** Estado que indica que se validan los datos ingresados.
- 4.- **Confirmando:** Estado que indica la confirmación de todo lo que se llevó a cabo anteriormente.
- 5.- **Buscando:** Estado que indica que buscan los datos ingresados.

3.14 Diagramas de Colaboración.

3.14.1 Agregar Grupo.

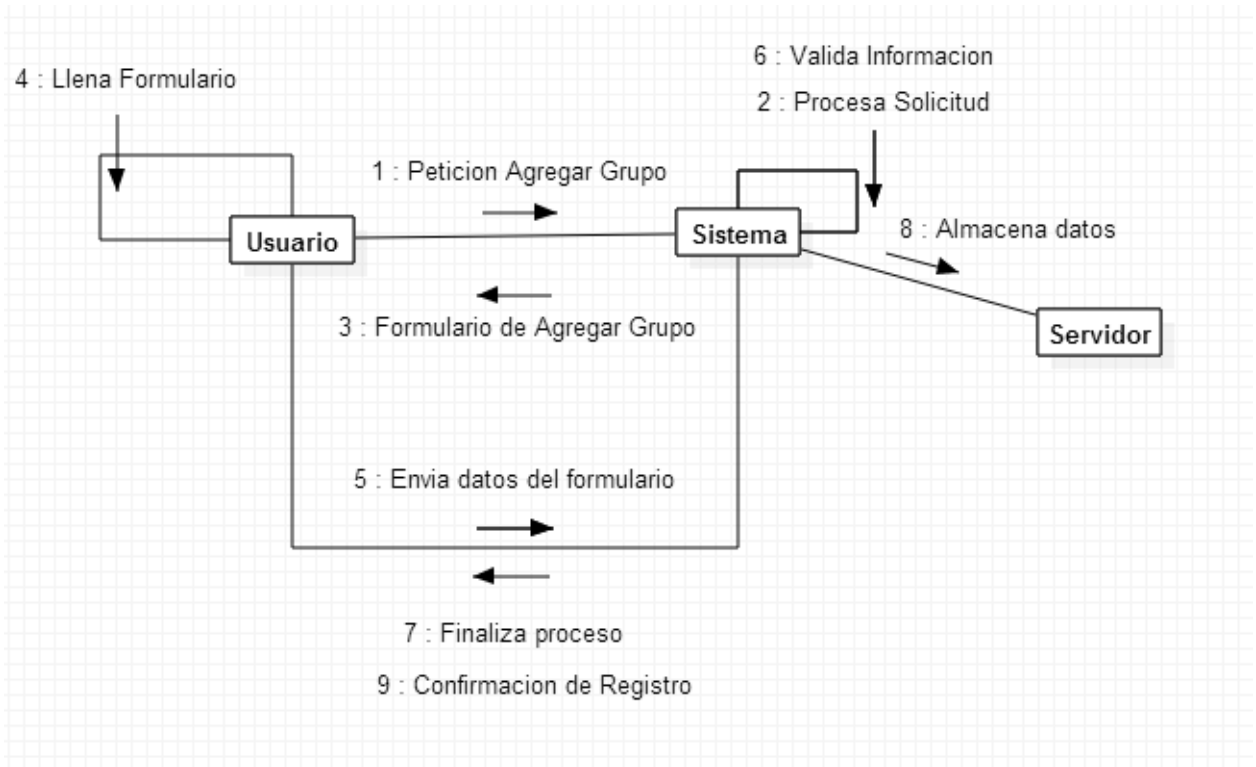


Figura 35.- Agregar grupo DC1

Descripción de Enlaces:

- 1.- **Petición Agregar Grupo:** Enlace que se encarga de transmitir que se quiere agregar un nuevo grupo en el sistema.
- 2.- **Procesa Solicitud:** Enlace que se encarga de procesar la solicitud anterior, haciendo uso de la base de datos y el sistema en sí.
- 3.- **Formulario de Agregar Grupo:** Enlace que se encarga de enviar el formulario para que sea llenado por el usuario.

4.- **Llenar Formulario:** Enlace que se encarga de presentar los campos correspondientes al usuario para que este efectúa las operaciones correspondientes.

5.- **Envía datos del Formulario:** Enlace que envía los datos del agregado de grupo a la clase correspondiente, la cual se encargará de tratarlos de la manera correcta.

6.- **Valida Información:** Enlace que verifica que la información previamente sea la correcta y cumpla con las características correspondientes que solicita este módulo del sistema.

7.- **Finaliza Proceso:** Enlace que da por finalizado el proceso de agregar el grupo correspondiente, y da entrada a que se guarden los datos correctamente sin intervención alguna del usuario.

8.- **Almacena los datos del Formulario:** Enlace que almacena los datos ya validados en la base de datos del servidor de nuestro sistema, para posteriormente estos sean utilizados en operaciones siguientes.

9.- **Proceso Exitoso:** Enlace que confirma que tanto el agregado del grupo y el guardado de los datos fueron exitosos y esto se refleja mediante un mensaje al usuario.

3.14.2 Baja Usuario

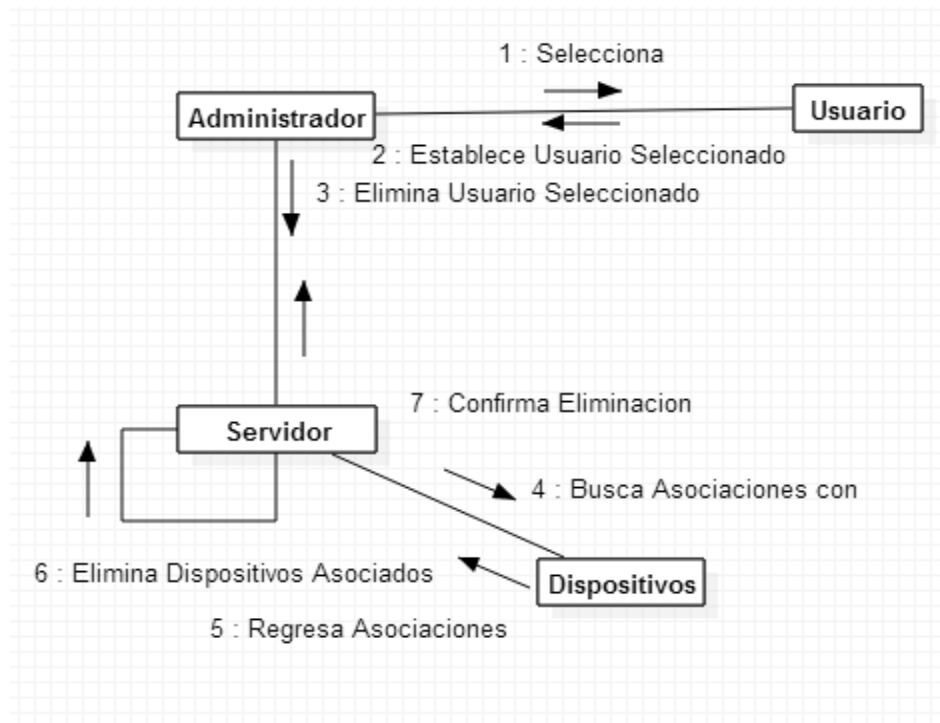


Figura 36.- Baja de Usuario DC2

Descripción de Enlaces.

- 1.- **Selecciona:** Enlace que indica que el administrador selecciona un usuario para realizar operaciones posteriores con el.
- 2.- **Establece Usuario Seleccionado:** Enlace que confirma que el administrador puede manipular al usuario seleccionado.
- 3.- **Eliminar Usuario Seleccionado:** Enlace que indica que el usuario seleccionado ha sido eliminado del sistema por parte del administrador.
- 4.- **Busca Asociaciones:** Enlace que se encarga que después de eliminar al usuario busque dispositivos asociados con el.

5.- **Regresa Asociaciones:** Enlace que se encarga de regresar todas las asociaciones que tenga el usuario con diferentes dispositivos.

6.- **Elimina Dispositivos Asociados:** Enlace que se encarga de eliminar las asociaciones así como también los dispositivos del usuario.

7.- **Confirma Eliminación:** Enlace que confirma que la operación de eliminación ha sido llevada con éxito.

3.14.3 Alta de Usuario

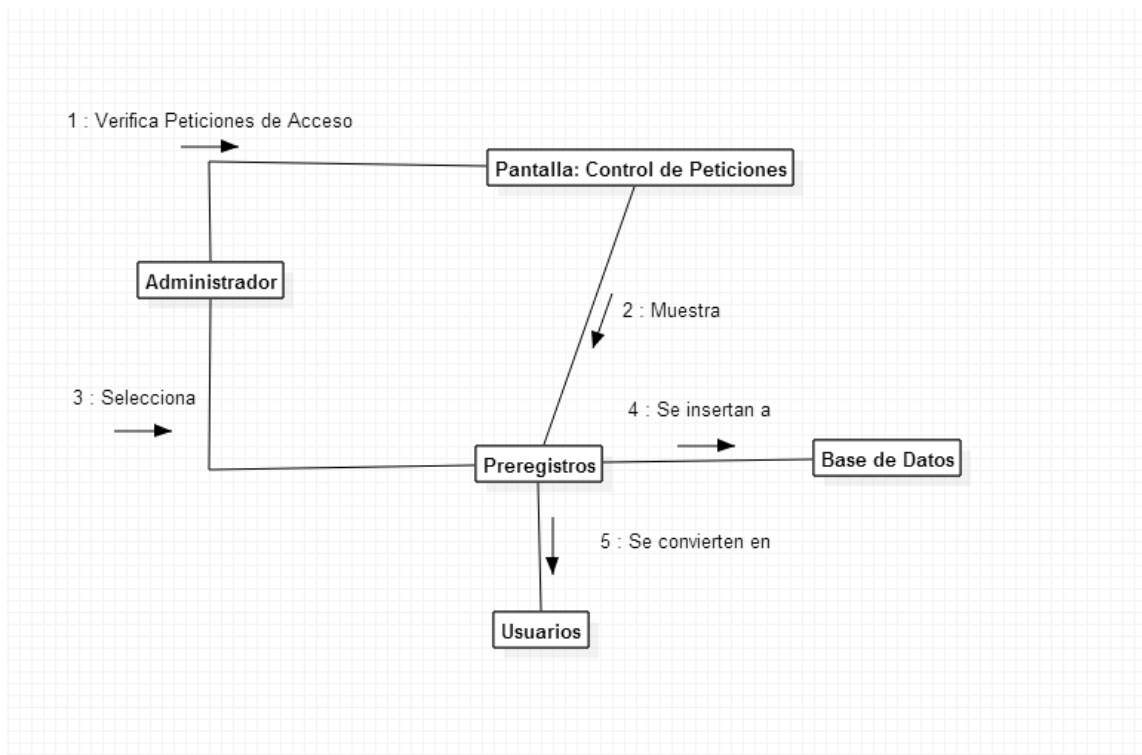


Figura 37.- Alta de usuario DC3

Descripción de Enlaces.

1.- **Verifica Peticiones de Acceso:** Enlace que indica que el administrador verifica todos los usuarios que se han registrado para poderlos finalmente ingresar de manera definitiva al sistema.

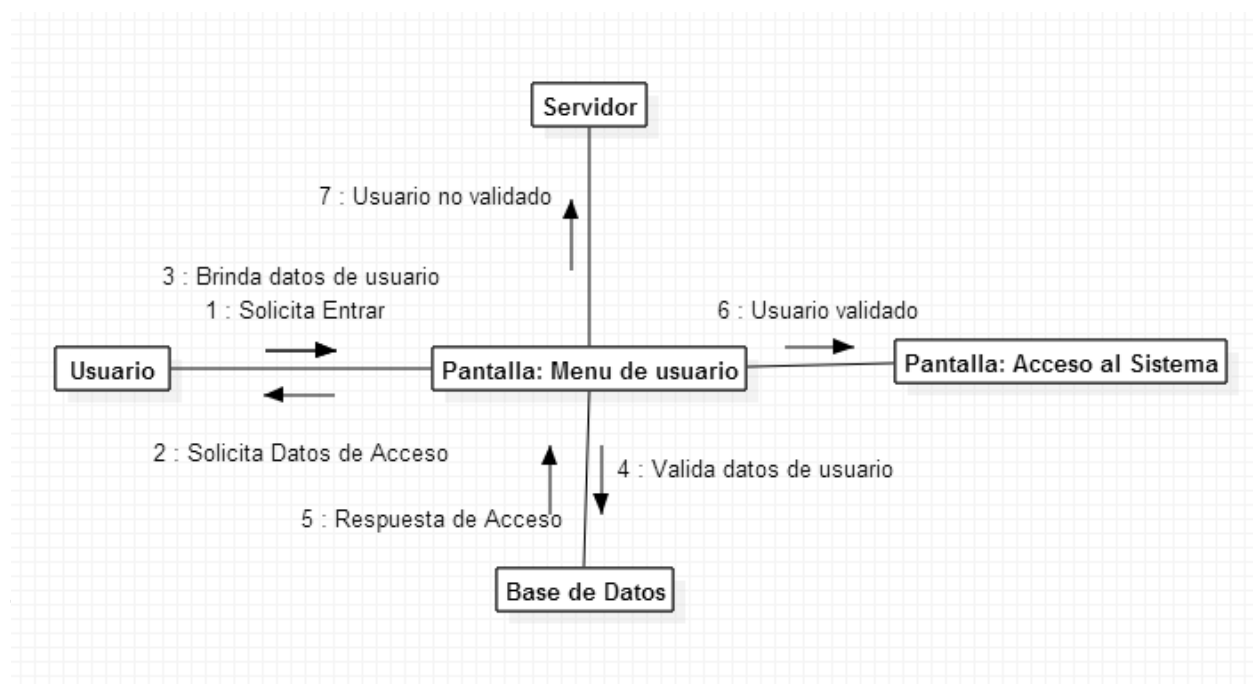
2.- **Muestra:** Enlace que indica que se van a mostrar todos los usuarios que se han registrado pero les hace falta la validación por parte del administrador.

3.- **Selecciona:** Enlace que indica que el administrador seleccionará de uno a más usuarios para validarlos y corroborar los datos.

4.- **Se insertan a:** Enlace que indica que los usuarios después de validarlos e ingresaran en una base de datos para posteriores operaciones.

5.- **Se convierten en:** Enlace que indica que finalmente se convierten en usuarios totales del sistema.

3.14.4 Acceso al Sistema.



5.- **Respuesta de Acceso:** Enlace que indica que una vez validados los datos se aceptará o restringirá el acceso al sistema.

6.- **Usuario validado:** Enlace que indica que la respuesta de acceso fue exitosa y por lo tanto obtiene acceso al sistema.

7.- **Usuario no validado**: Enlace que indica que la respuesta de acceso fue fallida y por lo tanto no obtiene acceso al sistema.

3.14.5 Desconexión del Sistema

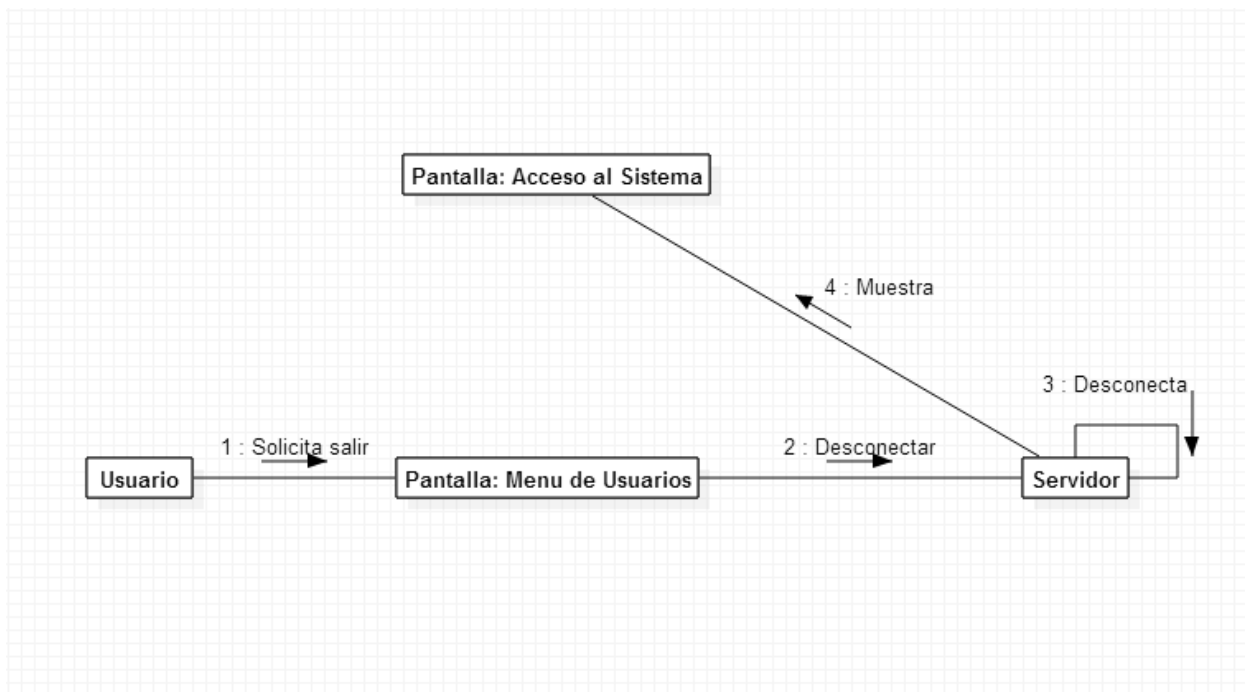


Figura 39.- Desconexión al sistema DC5

Descripción de Enlaces:

- 1.- **Solicita salir**: Enlace que indica que el usuario desea salir del sistema.
- 2.- **Desconectar**: Enlace que indica que se procederá a desconectar al usuario dentro del sistema.
- 3.- **Desconecta**: Enlace que indica que se está ejecutando la desconexión.
- 4.- **Muestra**: Enlace que indica que al tener éxito la desconexión se mostrará la pantalla de acceso al sistema.

3.14.6 Consulta Usuario.

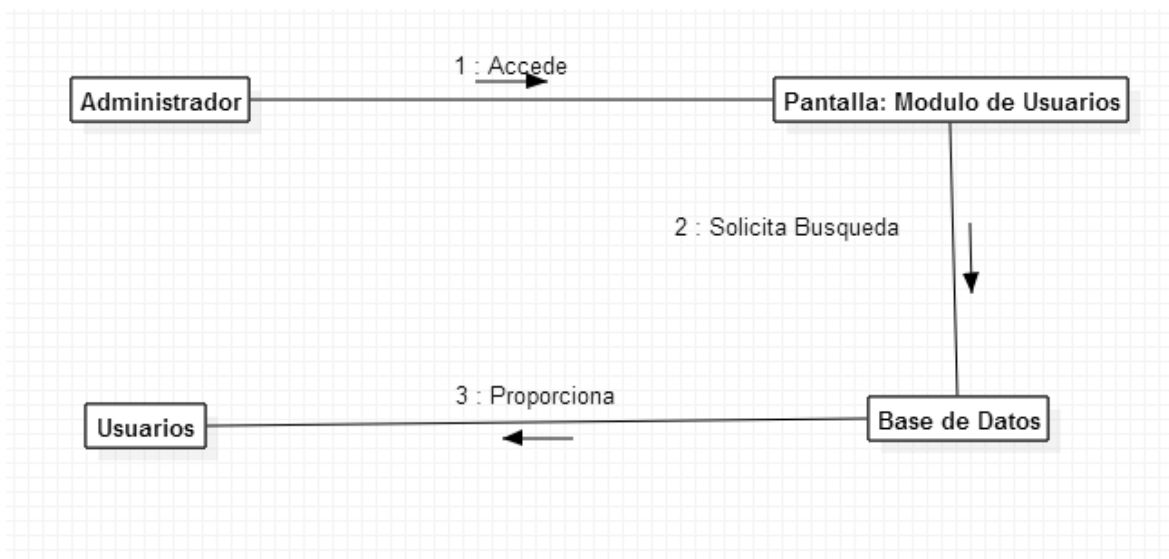


Figura 40.- Consulta de Usuario DC6

Descripción de Enlaces:

- 1.- **Accede:** Enlace que indica que el administrador accede al módulo de usuarios.
- 2.- **Solicita Búsqueda:** Enlace que indica que se está solicitando una búsqueda de usuarios.
- 3.- **Proporciona:** Enlace que indica que el resultado de la búsqueda será información sobre usuarios.

3.14.7 Invitación a Grupo

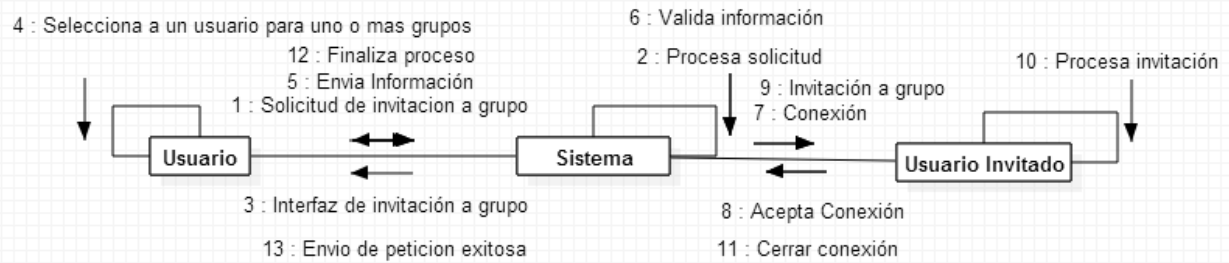


Figura 41.- Invitación a grupo DC7

Descripción de Enlaces:

- 1.- **Solicitud de invitación a grupo:** Enlace que indica que se hace una solicitud de invitación a un grupo en el cual un usuario envía de una a muchas.
- 2.- **Procesa solicitud:** Enlace que indica que se procesan las solicitudes enviadas en el sistema
- 3.- **Interfaz de invitación a grupo:** Enlace que indica que se le presenta al usuario los diferentes grupos que están dados de alta en el sistema.
- 4.- **Selecciona a un usuario para uno o más grupos:** Enlace que indica que se seleccionan de una a más usuarios para cada una de las invitaciones.
- 5.- **Envía información:** Enlace que indicia que se envía la información correspondiente a los usuarios seleccionados.
- 6.- **Valida información:** Enlace que indica que se validan los datos seleccionados.
- 7.- **Conexión:** Enlace que indica que se procederá a pedir una conexión con el invitado.
- 8.- **Acepta conexión:** Enlace que indica que se aceptó una conexión entre remitente y destinatario.
- 9.- **Invitación a grupo:** Enlace que indica que se procederá a enviarle la invitación a cada uno de los usuarios seleccionados.

- 10.- **Procesa invitación:** Enlace que indica que el usuario destino procesa dicha invitación.
- 11.- **Cerrar conexión:** Enlace que indica que se finaliza la conexión.
- 12.- **Finaliza proceso:** Enlace que indica que el proceso ha terminado.
- 13.- **Envío de petición exitosa:** Enlace que indica un mensaje que el proceso fue llevado a cabo con éxito.

3.14.8 Registro de Usuario.

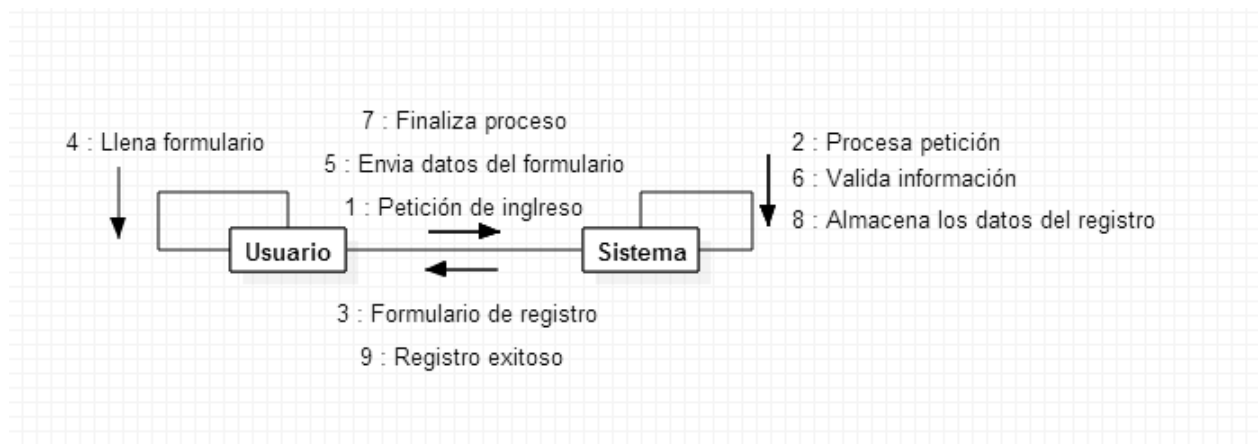


Figura 42.- Registro de Usuario DC8

Descripción de Enlaces:

- 1.- **Petición de Ingreso:** Enlace que indica que el usuario pide un a petición de registro al sistema.

- 2.- **Procesa petición:** Enlace que indica que se procesa la solicitud de registro.
- 3.- **Formulario de registro:** Enlace que indica el envío del formulario de registro.
- 4.- **Llena formulario:** Enlace que indica la acción del llenado del formulario.
- 5.- **Envía datos del formulario:** Enlace que indica que se envía a los datos del formulario llenado.
- 6.- **Valida información:** Enlace que indica que se validan los datos del formulario.
- 7.- **Finaliza proceso:** Enlace que indica que se terminó el proceso del registro.
- 8.- **Almacena los datos del registro:** Enlace que indica que los datos se están guardando.
- 9.- **Registro exitoso:** Enlace que indica que el registro ha sido de una manera exitosa.

3.14.9 Notificar

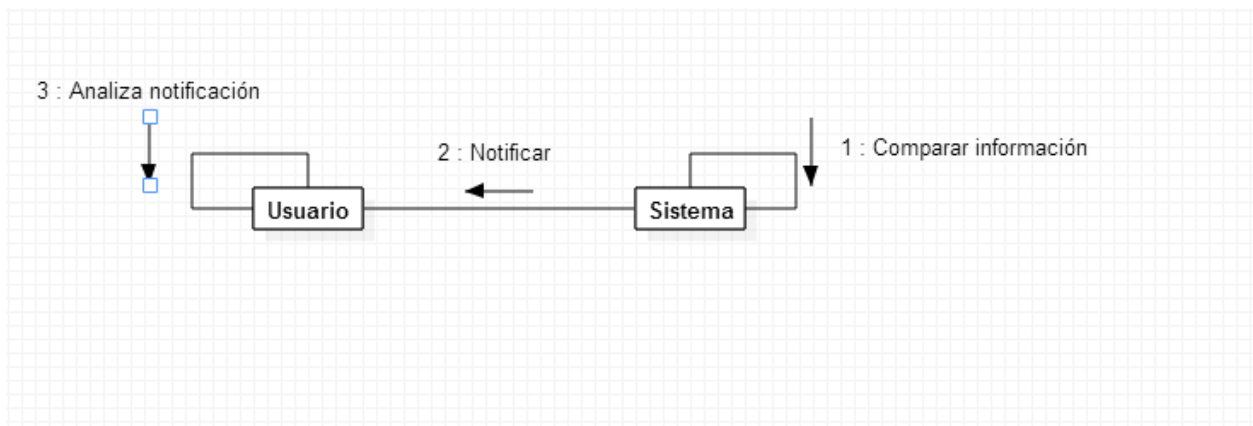


Figura 43.- Notificar DC9

Descripción de Enlaces:

1.- **Comparar información:** Enlace que indica que se visualizará la información hasta ese momento y si hay algún cambio se le notificará al usuario.

2.- **Notificar:** Enlace que indica que se le envía la notificación al usuario destino por medio del sistema.

3.- **Analiza notificación:** Enlace que indica que el usuario analiza la notificación que le acaba de llegar y enterarse de lo que sucedió.

3.14.10 Interacción entre nodos.

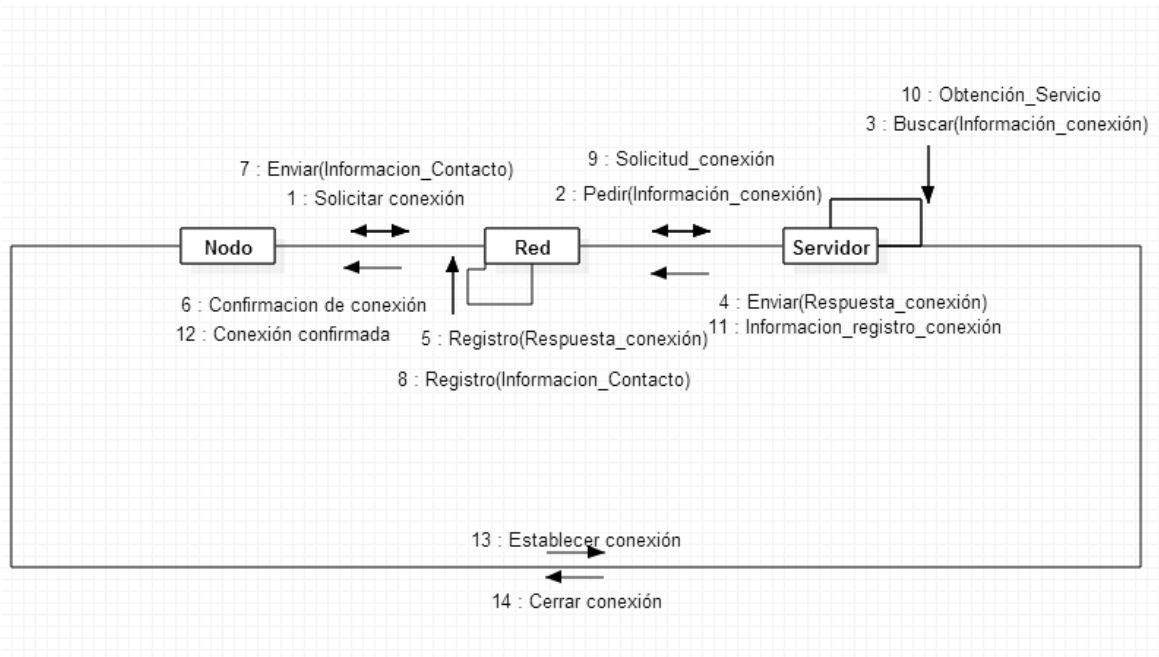


Figura 44.- Interacción entre nodos DC10

Descripción de Enlaces:

- 1.- **Solicitar conexión:** Enlace que indica que se solicita una conexión entre dos.
- 2.- **Pedir(información conexión):** Enlace que indica la petición de la conexión con todo y su información.
- 3.- **Buscar (información conexión):** Enlace que indica la búsqueda de los parámetros asignados.
- 4.- **Enviar (Respuesta conexión):** Enlace que indica la respuesta a la petición de la conexión.
- 5.- **Registro (Respuesta conexión):** Enlace que indica que se registra la respuesta de conexión.
- 6.- **Confirmación de conexión:** Enlace que indica la confirmación de la conexión.
- 7.- **Enviar (Información Contacto):** Enlace que indica que se envía la información de contacto.
- 8.- **Registro (Información Contacto):** Enlace que indica el registro de la información de contacto.
- 9.- **Solicitud Conexión:** Enlace que indica la solicitud de una nueva conexión.
- 10.- **Obtención Servicio:** Enlace que indica la obtención del servicio requerido mediante la conexión solicitada.
- 11.- **Información registro conexión:** Enlace que indica que se transfiere la información de la conexión.
- 12.- **Conexión confirmada:** Enlace que indica que la conexión ha sido confirmada.
- 13.- **Establecer conexión:** Enlace que indica que se establece una conexión.
- 14.- **Cerrar conexión:** Enlace que indica que se cierra la conexión.

3.14.11 Actualizar nodos.

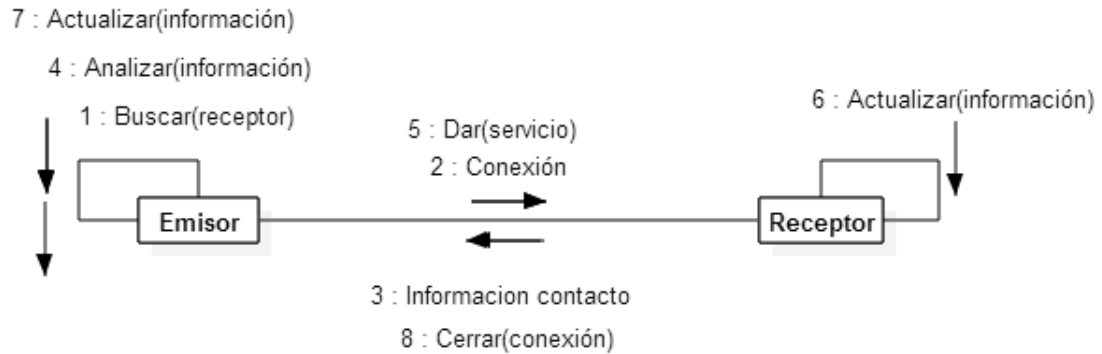


Figura 45.- Actualizar nodo DC12

Descripción de Enlaces:

- 1.- **Buscar (receptor):** Enlace que indica la búsqueda de un receptor dentro de un radio asignado.
- 2.- **Conexión:** Enlace que indica que una vez encontrado al receptor se establecerá una conexión.
- 3.- **Información contacto:** Enlace que indica el envío de información de contacto del receptor al emisor.
- 4.- **Analizar:** Enlace que indica que el emisor analiza la información de contacto del receptor.
- 5.- **Dar (servicio):** Enlace que indica que comenzara el servicio del emisor con el receptor.

6.- **Actualizar (información):** Enlace que indica que se actualiza la información del receptor.

7.- **Actualizar (información):** Enlace que indica que se actualiza la información del emisor.

8.- **Cerrar (conexión):** Enlace que indica el cierre de la conexión.

3.14.12 Administrar información



Figura 46.- Administrar información DC13

Descripción de Enlaces:

1.- **Pedir (gestor información):** Enlace que indica que el usuario pide si no hacen falta asignaciones de grupo por asignar al usuario.

2.- **Procesar (invitaciones pendientes):** Enlace que indica que el sistema está verificando si existe alguna asignación faltante por parte del usuario.

3.- **Solicitudes a grupo pendientes:** Enlace que indica el resultado de la búsqueda por parte del sistema al procesar las invitaciones pendientes.

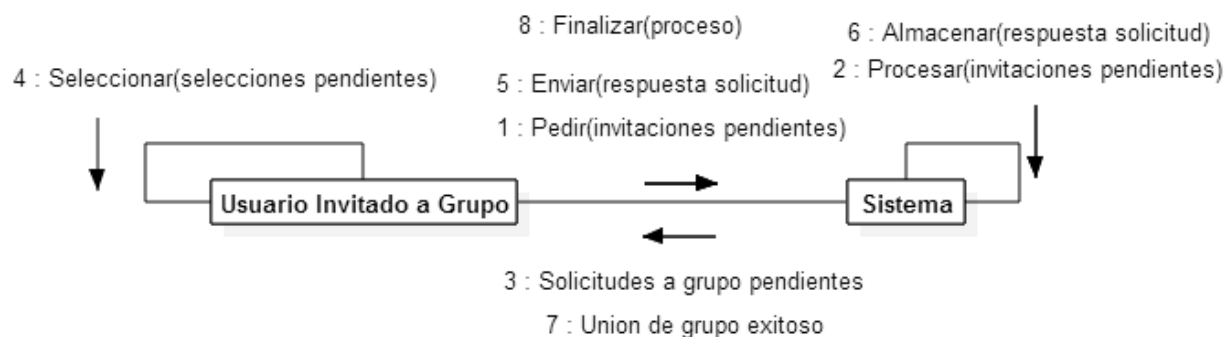
4.- **Seleccionar (invitaciones pendientes):** Enlace que indica que el usuario selecciona una de sus invitaciones pendientes.

5.- **Enviar (respuesta solicitud):** Enlace que indica que se envía la invitación seleccionada.

6.- **Almacenar (respuesta solicitud):** Enlace que indica que se hace la unión entre el usuario y el grupo.

7.- **Unión de grupo exitoso:** Enlace que indica que se envía un mensaje que el usuario está en el grupo seleccionado.

8.- **Finalizar (proceso):** Enlace que indica que el proceso ha sido finalizado.



3.14.13 Contestar Figura 47.- Contestar información DC14 información

Descripción de Enlaces:

1.- **Pedir (invitaciones pendientes):** Enlace que indica que el usuario pide si no hacen falta asignaciones de grupo por asignar al usuario.

2.- **Procesar (invitaciones pendientes):** Enlace que indica que el sistema está verificando si existe alguna asignación faltante por parte del usuario.

3.- **Solicitudes a grupo pendientes:** Enlace que indica el resultado de la búsqueda por parte del sistema al procesar las invitaciones pendientes.

4.- **Seleccionar (invitaciones pendientes):** Enlace que indica que el usuario selecciona una de sus invitaciones pendientes.

5.- **Enviar (respuesta solicitud):** Enlace que indica que se envía la invitación seleccionada.

6.- **Almacenar (respuesta solicitud):** Enlace que indica que se hace la unión entre el usuario y el grupo.

7.- **Unión de grupo exitoso:** Enlace que indica que se envía un mensaje que el usuario está en el grupo seleccionado.

8.- **Finalizar (proceso):** Enlace que indica que el proceso ha sido finalizado.

3.15 Diagrama Relacional de Base de Datos.

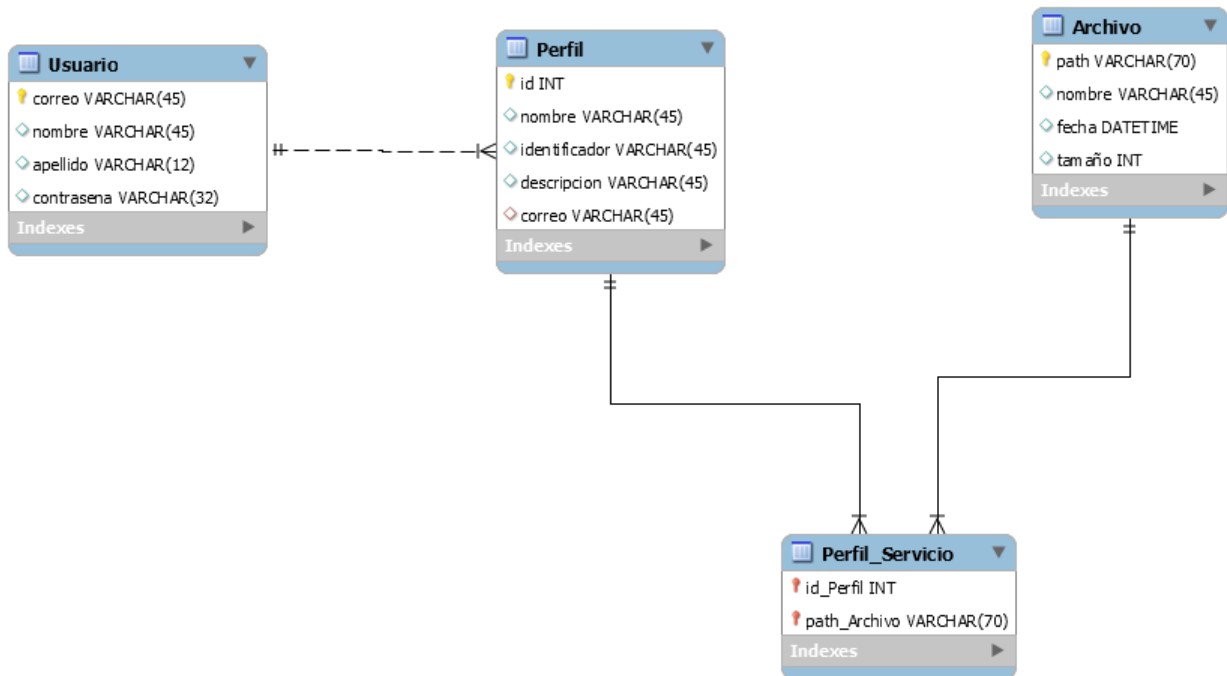


Figura 48.- Diagrama de Base de Datos

3.16 Diagrama Arquitectura final

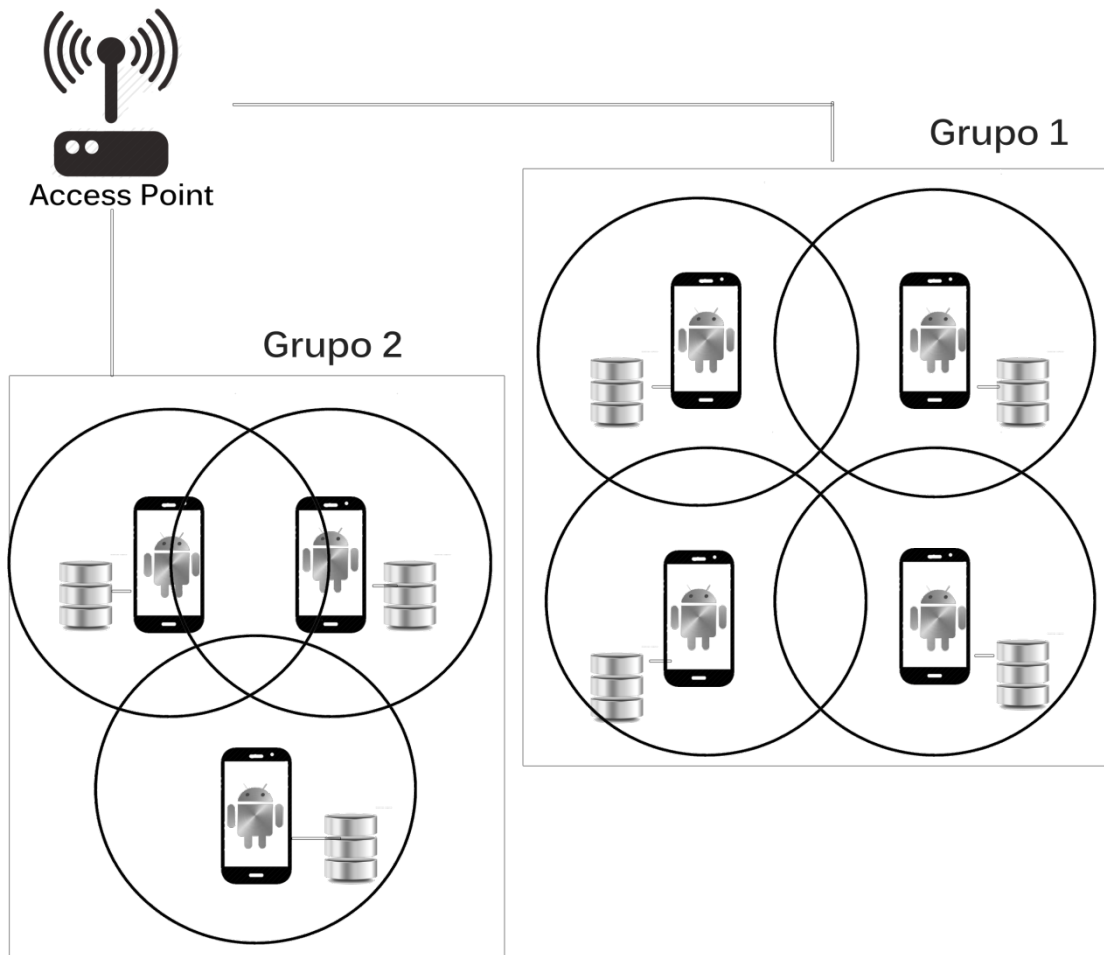


Figura 49.- Diagrama de Arquitectura Final

3.16.1 Descripción del Diagrama de Arquitectura Final.

El sistema está basado en dispositivos móviles, los cuales, serán conectados mediante Wi-Fi ya que se hará una simulación de una red Manet, para ello se empleará un Access Point para tener control de los usuarios que estarán conectados a esta. Cada nodo tendrá asignada su propia base de datos desarrollada en SQLite

4.0 Desarrollo/Implementación

Se describirá a continuación los métodos más importantes a la elaboración del Proyecto.

El primero a mostrar es el método que el Servidor usa para leer los dispositivos que se encuentran conectados y regresarle esta información a los dispositivos para que estos sigan la comunicación.

```
Timer temp;
temp = new Timer();
TimerTask task = new TimerTask() {
@Override
public void run(){

    try
    {
        dispositivo=(Dispositivo)ois.readObject();

        obtenerIPs();

        if(!existe(dispositivo)){
            conectados.add(dispositivo);
        }
        imprimirConectados();
        oos.writeObject(conectados);

    } catch (IOException e) {
        eliminar(dispositivo);
        temp.cancel();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
};temp.schedule(task, 10, 10000);
```

Como se observa, se lee un Objeto de la clase Dispositivo el cual contiene toda la información de cada Dispositivo y la almacena en una Lista, la cual manda a los usuarios; repite el proceso de recibir y mandar la lista cada 10 segundos en un hilo independiente por cada usuario que se encuentre en la red.

Ahora, por parte del usuario, se mostrará cómo obtiene su información y la transmite al servidor para así obtener a los usuarios que se encuentran conectados.

```
WifiManager wm = (WifiManager) context.getSystemService(context.WIFI_SERVICE);
String ip_device= Formatter.formatIpAddress(wm.getConnectionInfo().getIpAddress());
Log.d("Ip ->", "es: "+ip_device);

AdminSQLiteOpenHelper ash=new AdminSQLiteOpenHelper(context,"redManet", null, 1);
celular= new Dispositivo(ip_device,serial,ash.consultarUsuarios(context),ash.obtener_perfiles(context));

ArrayList<Perfil> perfiles=new ArrayList<>();
perfiles=ash.obtener_perfiles(context);

Socket cliente = new Socket(ip, port);
if (cliente.isConnected()) {
    ObjectOutputStream oos = new ObjectOutputStream(cliente.getOutputStream());
    ObjectInputStream ois = new ObjectInputStream(cliente.getInputStream());
    oos.writeObject(celular);
    conectados = (ArrayList<Dispositivo>)ois.readObject();
    obtenerInformacion();
}
```

Obtiene la IP actual del dispositivo, la información de usuario y los perfiles como servicios con los que cuenta cada uno y los almacena en un objeto de clase Dispositivo, después, se la manda al servidor y recibe la lista de usuarios conectados.

Ya con la lista, se puede realizar el proceso de actualización de información.

```
for(int i=0;i<perfiles.size();i++){
    ArrayList<Dispositivo> coincidencias=CoincidePerfil(perfiles.get(i));
    for(int k=0;k<coincidencias.size();k++){
        servicios=obtenerServicios(coincidencias.get(k),perfiles.get(i).getNombre());
        actualizarServicios(servicios,perfiles.get(i).getDisponibles(),coincidencias.get(k),perfiles.get(i));
    }
}
```

Lo que realiza, es leer de los dispositivos conectados en base a los perfiles del usuario y así poder obtener sus servicios y en caso de que existan servicios nuevos o modificados los actualiza.

Para actualizar los servicios, se crea un hilo el cual es el encargado de solicitar el servicio al Dispositivo que lo tenga y es destruido hasta que finaliza la transferencia del servicio.

```

ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream());

oos.writeObject(mensaje);

//Se realiza el proceso en el cual se recibirá el archivo
//"/storage/sdcard0/" + mensaje.getArchivo().getNombre()

DataOutputStream output;
BufferedInputStream bis;
BufferedOutputStream bos;

byte[] receivedData;
int in;
String file;

receivedData = new byte[1024];
bis = new BufferedInputStream(socket.getInputStream());
DataInputStream dis=new DataInputStream(socket.getInputStream());
//Para guardar fichero recibido
bos = new BufferedOutputStream(new FileOutputStream("/storage/sdcard0/" + mensaje.getArchivo().getNombre()));
while ((in = bis.read(receivedData)) != -1){
    bos.write(receivedData,0,in);
}
bos.close();
dis.close();

Log.d("Archivo", "El archivo fue pasado correctamente");
AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(context, "redManet", null, 1);
admin.compararArchivo(context, "/storage/sdcard0/" + mensaje.getArchivo().getNombre(), mensaje.getArchivo().getNombre(),
notificacion(0,"por parte de: "+cliente.getUser().getNombre(),mensaje.getArchivo().getNombre() + " actualizado",null));

```

Como se muestra se manda al Dispositivo un objeto de clase Mensaje el cual viene la información del servicio que se está solicitando y se recibe el servicio, inmediatamente se guarda en la base de datos el nuevo servicio que está disponible y se lanza una notificación para que el usuario esté informado de la nueva actualización que se ha hecho.

Cabe mencionar que cada Dispositivo abre un hilo servidor el cual es el encargado de recibir la petición de los servicios y poder mandarlos a los demás dispositivos.

```

Mensaje mensaje = (Mensaje)ois.readObject();
if(mensaje.getBandera()==true){
    DataInputStream input;
    BufferedInputStream bis;
    BufferedOutputStream bos;
    int in;
    byte[] byteArray;
    //Fichero a transferir
    final String filename = mensaje.getArchivo().getPath();
    final File localFile = new File( filename );

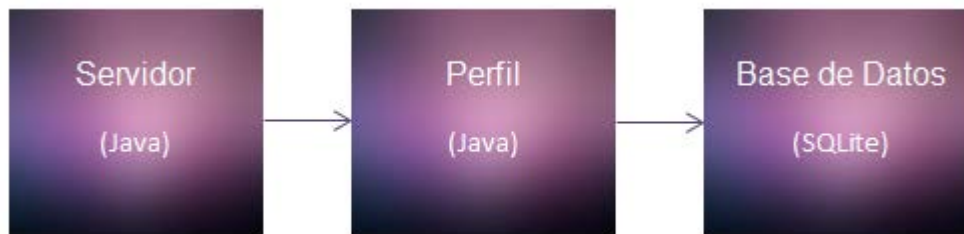
    bis = new BufferedInputStream(new FileInputStream(localFile));
    bos = new BufferedOutputStream(socket.getOutputStream());
    //Enviamos el fichero
    byteArray = new byte[8192];
    while ((in = bis.read(byteArray)) != -1){
        bos.write(byteArray,0,in);
    }
    bis.close();
    bos.close();
}else{
    Log.d("Servidor", "Se le mando una invitacion");
    AdminSQLiteOpenHelper ash=new AdminSQLiteOpenHelper(context,"redManet", null, 1);
    SQLiteDatabase sqldb = ash.getWritableDatabase();
    ash.onUpgrade(sqldb,5,6);
    ash.compararInvitaciones(context, mensaje.getInvitacion().getUsuario(), mensaje.getInvitacion().getPerfil());
    sqldb.close();
    ash.close();
}

```

Se recibe el objeto de clase Mensaje, el cual contiene si se le manda una solicitud de servicio, el cual, si es el caso se le transmite y termina el proceso, o si no, recibe una invitación para la unión de grupo.

Lo descrito anteriormente fue lo más relevante para mantener los servicios de los dispositivos actualizados.

4.1 Módulos que integran el sistema.



4.1.1 Servidor

Este módulo se encarga de mantener actualizados a los dispositivos de quienes se encuentran en la red y quienes no, además, se encarga de ofrecerles la información relevante de los mismos. Al darles esta información a los dispositivos, está seguro que se puedan comunicar entre ellos y se pueda mantener sus servicios actualizados y gestionados.

4.1.2 Perfil

Dentro de este bloque se efectúa todo el proceso del sistema, la base de datos es alojada en cada nodo, los procesos de comunicación se efectúan en este módulo una vez que el servidor ha concedido la dirección IP del nodo a quien se le brinda el servicio.

4.1.2.1 Módulo de Interoperabilidad.

Esta capa está a cargo de procedimientos que son delegados al sistema operativo que tienen relación con el proceso de conexión a la MANET. Algunos de ellos son: configuración del perfil de WLAN Ad-Hoc, control del dispositivo de red inalámbrico, configuración de dirección IP y sub-máscara de red, lectura de notificaciones del sistema operativo, entre otros.

4.1.2.2 Módulo de Red.

Esta componente implementa los servicios TCP y UDP, necesarios para el intercambio de datos en la MANET. Al momento de ser recibidos, los mensajes son validados y encolados, para ser luego atendidos por la capa de Comunicación. Esta capa tiene también la responsabilidad de administrar las conexiones TCP, con dispositivos remotos que se encuentran dentro de la vecindad del usuario.

4.1.3 Base de Datos

El desarrollo de este módulo se desarrolló en SQLite debido a que a diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica si no que la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos.

4.2 Infraestructura de Conexión

Para la implementación del sistema se ha trabajado con un Router marca TP-Link WR841N el cual es el encargado del área de red y así mismo, de ofrecer a cada dispositivo que se conecte una IP, la cual servirá de identificador para que cada usuario sepa con quien se debe comunicar.

4.3 Implantación

El sistema de Administración de Perfiles para redes Manet establece una red Ad-Hoc simulando una red MANET para brindar servicios de manera automática a los usuarios que se encuentren dentro de una red establecida dentro de un radio máximo de 25 m,

Requisitos del Sistema e Instalación

Requisitos Mínimos

Sistema Operativo	Android 4.0
Espacio en Memoria	30 MB (Necesarios para la Instalación)
RAM	512 MB
Procesador	Qualcomm a 1 GHz, GPU Adreno 205

4.3.1 Instalación

Para instalar el sistema sólo basta con copiar el archivo APK en el dispositivo y aceptar los puntos que requiere.

4.3.2 Funcionamiento

Aspectos Básicos

Para acceder a las opciones que brinda el sistema, en la parte lateral derecha se encuentra un menú desplegable como se muestra en la figura

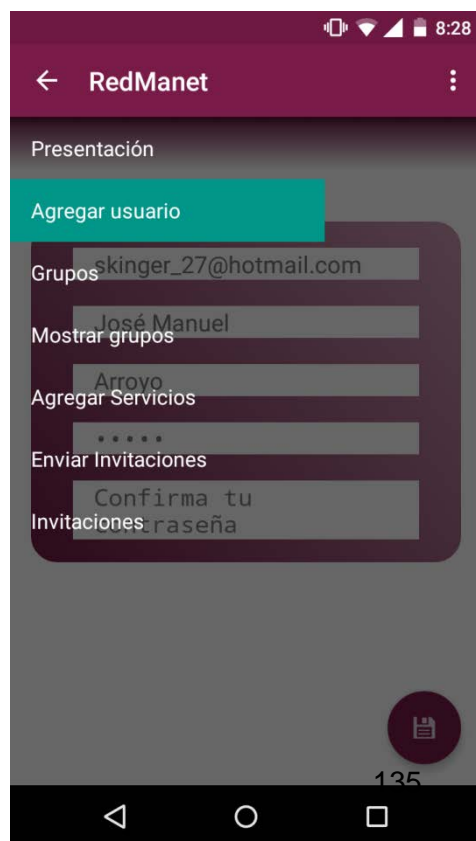


Figura 54.-Menú Principal.

4.3.2.1 Agregar usuario

El sistema sólo podrá registrar un usuario una única vez para de esta forma otorgará seguridad respecto a quien usa la aplicación.



Figura 55.-Registro de Usuario.



Figura 56.-Registro de Usuario.

Una vez llenado el formulario, se da click en el botón inferior derecho para registrar al usuario. A continuación se mostrará un mensaje advirtiendo que el usuario ha sido registrado (Figura 3).

Estos datos no podrán ser modificados del sistema es por eso que se requiere atención en este punto.

4.3.2.2 Crear Grupos

Cuando se requiera crear un grupo para poder ofrecer un servicio, basta con acceder a dicha sección en el menú y rellenar los campos correspondientes.

De igual forma, después de llenar los campos, se da clic al botón inferior para agregar el grupo.

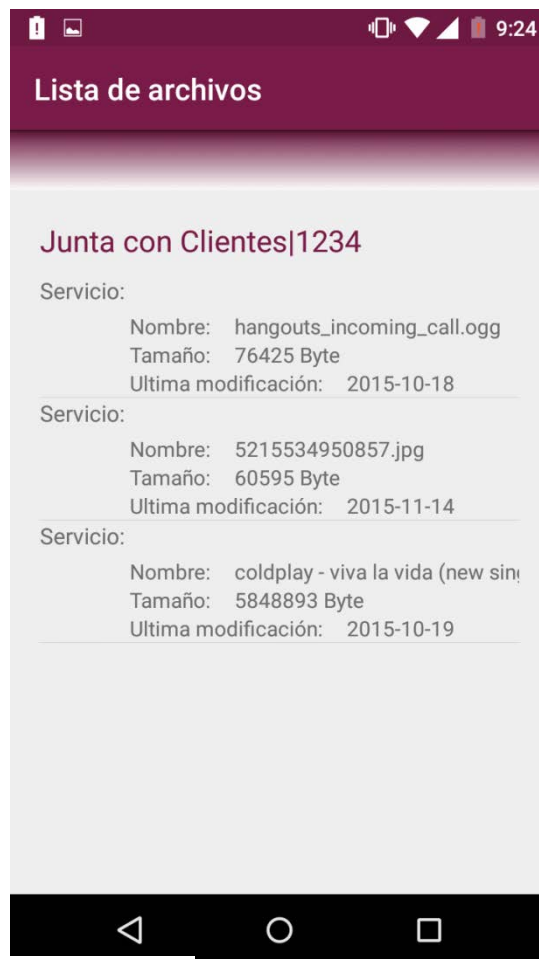


Figuras 57 y 58 .-Crear Grupos.

4.3.2.3 Mostrar grupos

Una vez que se cuente con al menos un grupo, en el menú se accede a “Mostrar Grupos” donde se podrá visualizar una breve descripción de los servicios contenidos en este, la cual incluye el nombre del servicio, fecha de última modificación y tamaño del archivo.

Para consultar todos los servicios que tiene un grupo, basta con dar clic sobre el mismo.



Figuras 59 y 60 .-Consulta de Grupos

4.3.2.4 Agregar Servicios

En este apartado se seleccionan los servicios que estarán disponibles para compartir con los demás nodos pertenecientes al mismo grupo.

En el menú se selecciona la opción “Agregar Servicios” donde se visualizará un menú desplegable con la lista de los grupos disponibles y un campo de texto, al dar clic sobre este, se abrirá un explorador de archivos para seleccionar los que se deseen compartir.

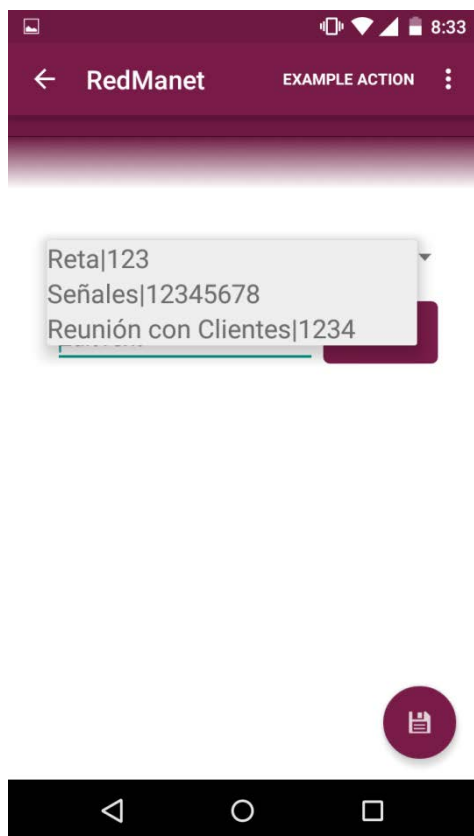


Figura 61.- Seleccionar grupo para agregar servicio.

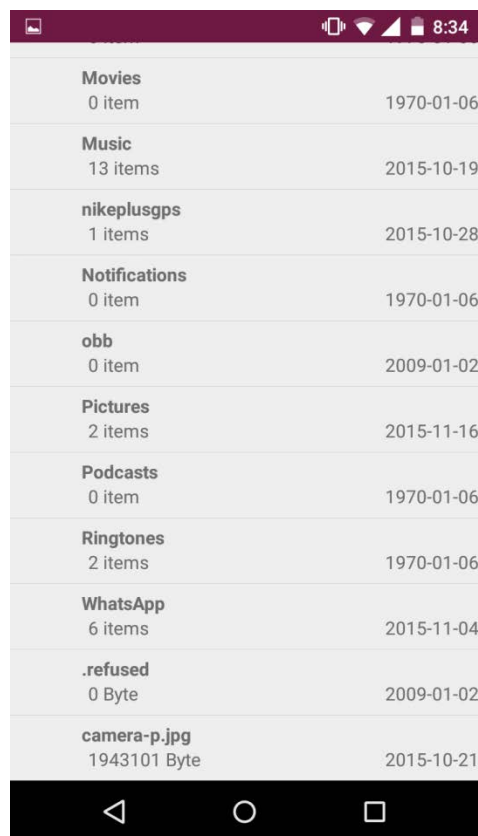


Figura 62 .-Explorador de Archivos.



Figura 63 .-Añadir Servicio.

5.0 Pruebas.

5.1 Pruebas del sistema.

Para esta sección, se buscó la apropiada navegación dentro del sistema, inserción de datos, el procesamiento y recuperación de la misma. Se probó con diferentes usuarios en diferentes dispositivos móviles, los cuales fueron:

- Moto x 1ra Generación
- Samsung Galaxy S4 mini
- Moto G Ferrari
- Samsung Note 3
- Sony Xperia M2
- Samsung Galaxy Grand Prime
- Samsung Galaxy S3 mini

Se observó que hay una correcta navegación, así como, un correcto procesamiento de la información, esto se debe a que se usa una base de datos SQLite la cual garantiza guardar y recuperar la información de manera sencilla y segura dentro de la aplicación. Otro de los aspectos importantes que se tomaron en cuenta fue cumplir con los requisitos de los casos de uso y reglas de negocio antes mencionadas en el actual documento, lo cual es un aspecto muy importante, se mencionarán los puntos más importantes que se cumplieron para el correcto funcionamiento de la aplicación:

- Agregar usuario
- Agregar perfil
- Invitar a perfil
- Interacción entre nodos
- Actualización de servicios entre nodos
- Notificación de información actualizada

Al cumplir con lo anterior, se concluye que las pruebas de sistema se finalizan de manera correcta ya que cumple con el objetivo de las mismas al realizar una navegación apropiada, gestión de información y la finalización de los requisitos mencionados en los casos de uso y reglas de negocio.

Para garantizar esta prueba se hizo uso de la herramienta Monkey para efectuar pruebas de estrés sobre la aplicación móvil simulando 1000 acciones y como se muestra en la siguiente imagen, los resultados fueron exitosos.

```
C:\Windows\system32\cmd.exe
:Sending Touch (ACTION_DOWN): 0:(577.0,742.0)
:Sending Touch (ACTION_UP): 0:(573.18085,745.1263)
:Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.manetn.redmanet/.MainActivity;end
// Allowing start of Intent < act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.manetn.redmanet/.MainActivity > in package com.manetn.redmanet
:Switch: #Intent;action=android.intent.action.MAIN;category=android.intent.category.LAUNCHER;launchFlags=0x10200000;component=com.manetn.redmanet/.MainActivity;end
// Allowing start of Intent < act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.manetn.redmanet/.MainActivity > in package com.manetn.redmanet
//[calendar_time:2015-11-23 10:44:01.236 system_uptime:100109422]
// Sending event #900
//[calendar_time:2015-11-23 10:44:01.237 system_uptime:100109422]
// Sending event #900
:Sending Touch (ACTION_DOWN): 0:(119.0,887.0)
:Sending Touch (ACTION_UP): 0:(110.15957,963.2467)
:Sending Touch (ACTION_DOWN): 0:(226.0,503.0)
:Sending Touch (ACTION_UP): 0:(211.24239,552.2216)
:Sending Touch (ACTION_DOWN): 0:(84.0,750.0)
:Sending Touch (ACTION_UP): 0:(91.53712,746.8497)
// Rejecting start of Intent < act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.android.settings/.DevelopmentSettings > in package com.android.settings
:Sending Trackball (ACTION_MOVE): 0:(-1.0,1.0)
:Sending Touch (ACTION_DOWN): 0:(495.0,817.0)
:Sending Touch (ACTION_UP): 0:(500.3671,813.82495)
:Sending Trackball (ACTION_MOVE): 0:(-4.0,-1.0)
:Sending Touch (ACTION_DOWN): 0:(36.0,939.0)
:Sending Touch (ACTION_UP): 0:(35.851727,945.9633)
:Sending Trackball (ACTION_MOVE): 0:(-4.0,-4.0)
:Sending Touch (ACTION_DOWN): 0:(2.0,1118.0)
// Rejecting start of Intent < act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.android.settings/.DevelopmentSettings > in package com.android.settings
:Sending Touch (ACTION_UP): 0:(0.0,1174.3241)
:Sending Touch (ACTION_DOWN): 0:(311.0,278.0)
:Sending Touch (ACTION_UP): 0:(304.86288,263.78223)
:Sending Touch (ACTION_DOWN): 0:(300.0,943.0)
:Sending Touch (ACTION_UP): 0:(310.77338,986.869)
:Sending Touch (ACTION_DOWN): 0:(360.0,890.0)
:Sending Touch (ACTION_UP): 0:(362.00604,888.5889)
// Rejecting start of Intent < act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.android.settings/.DevelopmentSettings > in package com.android.settings
:Sending Touch (ACTION_DOWN): 0:(280.0,1000.0)
Events injected: 1000
:Sending rotation degree=0, persist=false
:Dropped: keys=0 pointers=0 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=3831ms (0ms mobile, 0ms wifi, 3831ms not connected)
// Monkey finished
C:\Users\Arroyo\AppData\Local\Android\sdk\platform-tools>
```

Figura 64 .- Prueba de estrés con Monkey.

5.2 Pruebas de Integridad de Datos y Base de datos

El objetivo de estas pruebas fue asegurar que los métodos de acceso y proceso funcionarán correctamente y sin ocasionar corrupción de datos.

Para esto, se ejecutó cada uno de los métodos en donde se hace una solicitud a la base de datos para observar el resultado. Se hará mención a los métodos más usados y su resultado:

Método	Resultado	Respuesta - Porcentaje
consultaUsuarios(?)	Un objeto de la clase usuario con base en la tabla usuarios.	Correcta - 100%
obtener_perfiles(?)	Lista de objetos de la clase Perfil con base en la tabla perfiles	Correcta - 100%
obtener_perfil(?,?,?)	Un objeto de la clase Perfil con base a un nombre e identificador de perfil en la tabla de perfiles.	Correcta - 100%
altaPerfiles(?,?,?,?)	Un booleano verdadero para la inserción de los datos a la tabla perfiles con base a un nombre, identificador y descripción del perfil.	Correcta - 100%
altaArchivos(?,?,?,?,?,?)	Un booleano verdadero para la inserción de los datos a la tabla archivos con base a la información de un objeto de clase Archivo.	Correcta - 100%
obtenerArchivos(?,?,?)	Lista de objetos de la clase Archivo de la tabla archivos con base a un nombre e identificador de perfil.	Correcta - 100%

Se hizo mención de sólo algunos métodos que acceden a la base de datos, como se observa la respuesta fue correcta con un porcentaje del 100%, esto gracias a que SQLite ayuda mantener consistencia de los datos y permite que varios procesos o hilos puedan acceder a la misma de base de datos sin problemas, además, es importante destacar que la información que se guarda en la misma es reducida y esto facilita su acceso y procesamiento con un 100% de efectividad.

5.3 Pruebas de usabilidad

Para la realización de estas pruebas se usaron 10 dispositivos con las características solicitadas para corroborar que la aplicación llevará el proceso de comunicación correctamente, el primer punto a analizar fue que cada dispositivo fuera capaz de ver los usuarios conectados en la red, lo cual arrojó los siguientes resultados:

# de dispositivos	Cantidad de dispositivos que vieron conectados	% de visibilidad ([Total de dispositivos - Dispositivos no vistos]/100)
7 dispositivos	10 dispositivos	100%
2 dispositivos	9 dispositivos	90%
1 dispositivo	8 dispositivos	80%

Analizando los resultados de la tabla anterior, se observa que el 70% de los dispositivos pudieron ver a los 10 dispositivos que se encontraban en la red (se cuenta también al mismo dispositivo) mientras que el otro 30% tuvieron un promedio de 8.5 dispositivos que pudieron ver conectados en la red.

Otro de los aspectos que se analizaron fue que las invitaciones de perfil llegarán correctamente a su destinatario por parte del remitente, lo cual arrojó los siguientes resultados:

# de prueba	Invitaciones enviadas	Invitaciones recibidas	% de efectividad
1	5	5	100%
2	5	4	80%
3	10	9	90%
4	10	10	100%
5	10	9	90%

La tabla anterior muestra que no todas las invitaciones que son enviadas son recibidas, que se cuenta con un 92% de efectividad al momento de enviar las invitaciones de perfil a los usuarios conectados, esto debido a problemas con la red.

También se hizo la prueba de enviar 2 veces la misma invitación a un mismo dispositivo, lo cual resultó en un 100% de los casos se llegaba la invitación al destinatario, lo cual, resulta efectivo pero a un costo mayor de trabajo para el usuario.

En cuanto a los servicios de transferencia de información ofrecidos por los usuarios, se observó que aunque el usuario no observe a todos los dispositivos conectados en la red, en cuanto ofrezca el servicio antes mencionado a algún usuario que tenga el mismo perfil, éste podrá al mismo tiempo ofrecer el servicio hasta llegar a todos los usuarios que tengan el mismo perfil en la red aunque el primer usuario no lo haya podido ver en un inicio.

Se validó que los servicios de transferencia por parte de un usuario que se ofrecían llegarán completos a los demás usuarios, esto se aseguró gracias a que cada

dispositivo que solicita un servicio genera un hilo independiente el cual finaliza hasta que la transferencia del servicio se haya finalizado, al mismo tiempo, el dispositivo al cual se le solicita el servicio genera un hilo en el cual sólo se comunica con el solicitante hasta finalizar el proceso.

También es importante mencionar que cada usuario sólo solicita un servicio a la vez, esto garantiza que no se repita o que solicite el mismo servicio a dos dispositivos diferentes. En cuanto a los dispositivos que se les solicita el servicio, a lo máximo pueden atender a 5 usuarios, lo cual garantiza que no genere una cantidad de hilos excesivos lo cual haga terminar la aplicación de manera inesperada.

Como conclusión de este apartado de pruebas, se asegura que la red permite mantener actualizado a los usuarios de todos los servicios de transferencia de información de manera directa o indirecta, siempre y cuando se tenga el mismo perfil.

6 Conclusiones y Trabajo a Futuro

6.1 Conclusiones

En este documento se ha presentado el desarrollo del sistema de Administración de Perfiles para Redes MANET teniendo como objetivo principal la administración de servicios en ambientes móviles simulando una red MANET, en caso particular, el servicio de transferencia de información a través de Smartphones.

Para conseguir este objetivo se hizo un análisis de algunos sistemas similares. En el mercado se han encontrado aplicaciones que proporcionan servicios muy parecidos inclusive con mayor alcance sin embargo hasta ahora no existe algún sistema mexicano que brinde las características que este proyecto ofrece. La manera en que este sistema maneja la administración de servicios es práctica ya que basta con crear un grupo, añadir los servicios que podrá brindar y el sistema se encarga del resto.

En cuanto a seguridad, el proyecto nos garantiza que la persona es quien dice ser puesto que se ha establecido un solo usuario por nodo y esta información no podrá ser modificada una vez creada, de esta forma al mantenerse interactuando dentro de la red, los nodos serán identificados con su respectiva información. Por otro lado, se asegura integridad de la información ya que al momento de brindar un servicio solamente existe comunicación entre los nodos interesados sin pasar por intermediarios que puedan captar la información.

En el aspecto de confidencialidad, el sistema exclusivamente comparte la información que el nodo responsable desea hacer pública dentro de cada grupo y es responsabilidad de cada usuario el contenido que se podrá publicar.

A lo largo del desarrollo se implementaron diferentes soluciones manejando otras tecnologías consiguiendo los objetivos propuestos.

Como ya se mencionó, en esta primera etapa, el sistema es capaz de a través de la creación de grupos, poder intercambiar información (archivos de cualquier formato: .pdf, .mp3, .doc, .jpg, .png, etc.) entre nodos pertenecientes a los mismos.

Al realizar las pruebas se asegura que el sistema

Con base en lo anterior se pueden destacar algunos puntos del sistema, los cuáles son:

Es un sistema que funciona bien en cualquier versión de Android a partir de la 4.0

Tolerante a fallos.

Sistema Responsivo.

Garantiza seguridad contra modificación de mensajes, no repudio y repetición (ante usuarios promedio).

Proporciona confidencialidad de datos.

Manejo de integridad de datos.

6.2 Trabajo a Futuro.

El proyecto desarrollado e implementado cumple con los puntos acordados al inicio sin embargo, este trabajo aún puede mejorarse en algunos aspectos, los más destacables son:

Interoperabilidad entre Sistemas Operativos.

Implementación del Protocolo HLMP.

Sistema totalmente descentralizado.

Añadir más Servicios al Sistema, por ejemplo implementar una alerta para situaciones de emergencia.

7. Bibliografía.

[1] Marc Vicente, “La tecnología móvil ya no sólo está en el dispositivo móvil”, [Fecha de consulta 27 de agosto de 2014]. Disponible en: <http://www.puromarketing.com/12/19297/tecnologia-movil-solo-esta-dispositivo-movil.html>.

[2] “Encaminamiento en Redes Ad-Hoc”. Departamento de Sistemas Telemáticos y Computación (GSyC) Universidad Rey Juan Carlos. Noviembre de 2013 [Fecha de consulta 6 de abril de 2015]. Disponible en: http://gsyc.es/~mortuno/rom/06-encaminamiento_adhoc.pdf

[3] Dr Farid Farahmand “Delay Tolerant Networks: Challenges and Applications” [Fecha de consulta 11 de abril de 2015]. Disponible en: http://www.engr.uconn.edu/~tehrani/teaching/CE_Seminar/Farahmand.pdf

[4] Drs. Baruch Awerbuch & Amitabh Mishra. “Introduction to Ad hoc Networks CS-647: Advanced Topics in Wireless Networks”. Department of Computer Science Johns Hopkins University [Fecha de consulta 12 de abril de 2015]. Disponible en: http://www.cs.jhu.edu/~cs647/intro_adhoc.pdf.

[5] José Ignacio Ruiz Núñez “Configuración DHCP en Redes MANET Subordinadas” Departamento de Ingeniería del Software e Inteligencia Artificial Universidad Complutense de Madrid [Fecha de consulta 14 de marzo de 2015]. Disponible en: http://eprints.ucm.es/10066/1/Jos%C3%A9_Ignacio_Ruiz_N%C3%BA%C3%B1ez.pdf.

[6] Ing. Jairo Orlando Maldonado Alarcón “Sistema inteligente de monitoreo para el consumo de energía de redes inalámbricas ad hoc” Universidad Nacional de Colombia Departamento de Ingeniería de Sistemas e Industrial Bogotá, Colombia 2013 [Fecha de consulta 14 de marzo de 2015]. Disponible en: <http://www.bdigital.unal.edu.co/39515/1/300221.2014.pdf>

[7] Felipe Ignacio Valverde Campos, “Implementación del protocolo HLMP en Android”, [Fecha de consulta 12 de abril de 2015]. Disponible en: http://repositorio.uchile.cl/bitstream/handle/2250/111920/cf-valverde_fc.pdf?sequence=1&isAllowed=y.

[8] Serval Project, The Serval Mesh, [Fecha de consulta 14 de abril de 2015]. Disponible en: https://play.google.com/store/apps/details?id=org.servalproject&hl=es_419

[9] Richard Taylor, "Mesh networks chat apps to the rescue", [Fecha de consulta 15 de abril de 2015]. Disponible en: <http://www.bbc.com/news/technology-27225869>.

[10] "FireChat, Open Garden Inc", [Fecha de consulta 14 de abril de 2015]. Disponible en: <https://itunes.apple.com/mx/app/firechat/id719829352?mt=8>.

[11] Mario Orestes Prado Martínez, Sistema de Información de apoyo a la toma de decisiones en el proceso de monitoreo para la producción, [Fecha de consulta 20 de abril de 2015]. Disponible en: <http://www.eumed.net/libros-gratis/2011b/973/caracteristicas%20y%20definiciones%20de%20sistemas%20de%20informacion.html>.

[12] José I. Vega, Los estudios de viabilidad para los negocios, [Fecha de consulta 23 de abril de 2015] Disponible en: http://www.uprm.edu/cde/public_main/Informes_Articulos/articulos/ArticuloViabilidad.pdf

[13] Mil Aplicaciones, Versiones de Android, [Fecha de consulta 26 de abril de 2015] Disponible en: <http://milaplicaciones.com/versiones-de-android/>

[14] Everac99, Sueldos IT en México 2015: de startups y desarrollo móvil, [Fecha de consulta el 26 de abril de 2015]. Disponible en: <https://everac99.wordpress.com/2015/03/20/sueldos-it-en-mexico-2015-de-startups-y-desarrollo-movil/>

[15] Viabilidad operativa, técnica y económica de un proyecto. [Fecha de consulta el 26 de abril de 2015]. Disponible en: <http://proyctocomputojmas.blogspot.mx/2007/05/ejemplo-viabilidad-operativa-tnica-y.html>.

[16] José Manuel Pérez Hueso, Protocolos de encaminamiento IP en dispositivos móviles para crear redes MANET's, [Fecha de consulta el 26 de abril de 2015]. Disponible en: http://dtstc.ugr.es/it/pfc/proyectos_realizados/downloads/Memoria2012_JoseManuelPer ezHueso.pdf

[17] Ian Sommerville, "Requerimientos del software", en *Ingeniería de Software*, Pearson Addison Wesley: 200. 691 pp.

Glosario.

ANDROID.- Sistema Operativo basado en el núcleo Linux.

HLMP.- High Level MANET Protocol.

MANET.- Mobile Ad Hoc Network.

RED AD HOC.- Red inalámbrica descentralizada.

WIFI-DIRECT.- Protocolo implementado para la transferencia de datos entre dos dispositivos a través de la red Wi-Fi.

Near Field Communication (NFC).- Tecnología tipo wireless, de corto alcance para la transferencia de datos entre dispositivos.