



CIC IPN



INSTITUTO POLITÉCNICO
NACIONAL

CENTRO DE INVESTIGACIÓN EN
COMPUTACIÓN

LABORATORIO DE CÓMPUTO DISTRIBUIDO Y
PARALELO
AMBIENTE UBICUO PARA COLABORACIÓN
OPORTUNISTA

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A
Emiliano Gutiérrez Arias

Director de tesis: Mc. Rolando Menchaca Méndez
Codirector: Mc. Rolando Quintero

México, D.F., 2006.



INSTITUTO POLITECNICO NACIONAL SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 23 del mes de Junio de 2006 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"AMBIENTE UBICUO PARA COLABORACIÓN OPORTUNISTA"

GUTIÉRREZ

Apellido paterno

ARIAS

materno

TELÉSFORO EMILIANO

nombre(s)

Con registro:

B	0	2	0	8	9	2
---	---	---	---	---	---	---

aspirante al grado de: **MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACIÓN DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente

Dr. Jesús Manuel Olivares Ceja

Secretario

M. en C. Sergio Sandoval Reyes

Primer vocal
(Director de Tesis)

M. en C. Rolando Menchaca Mondéz

Segundo vocal

M. en C. Rolando Quintero Téllez

Tercer vocal

M. en C. Rubén Peredo Valderrama

Suplente

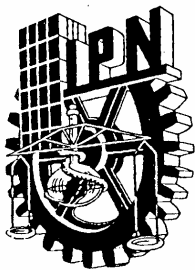
Marco Antonio Moreno Ibarra

EL PRESIDENTE DEL COLEGIO



Dr. Hugo César Coyote Estrada

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION



INSTITUTO POLITECNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la Ciudad de México, D.F., el día 13 del mes Octubre del año 2006 , el (la) que suscribe Ing. Gutierrez Arias Emiliano alumno (a) del Programa de Maestría en ciencias de la computación con número de registro B020892 , adscrito a la Sección de Estudios de Posgrado e Investigación de Investigación del CIC unidad Zacatenco , manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Mc. Rolando Menchaca Méndez y cede los derechos del trabajo intitulado Ambiente Ubicuo para Colaboración Oportunista , al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección eeggaa19@hotmail.com . Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Ing. Emiliano Gutiérrez Arias
Nombre y firma

Un Agradecimiento a:

El consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo recibido durante mis estudios de maestría en el Centro de Investigación en computación del Instituto Politécnico Nacional.

El Programa Institucional de Formación de Investigadores (PIFI) por el apoyo recibido para la realización de la presente tesis.

Así también al Centro de Investigación en Computación del IPN por el apoyo recibido durante mi estancia en sus instalaciones.

Agradezco profundamente a toda mi familia por enseñarme el valor del amor, la amistad y el trabajo.

Quiero dedicar este trabajo a mi Mamá, a mi Papá, a Victoria, a Juan, a Marcos, a Ricardo, a Georgina.

A mi asesor de Tesis el Mc. Rolando Menchaca Méndez, y a todos mis sinodales.

Y quiero expresar un gran sentimiento de gratitud a las siguientes personas.

Marivel, Felix, Anabel, Azael por ser mis compañeros y amigos durante la maestría

Sinceramente Gracias.

RESUMEN

Ambiente Ubicuo para Colaboración Oportunista

Para las personas envueltas en una sesión de edición colaborativa resulta importante un sistema que facilite la colaboración oportunista sobre el documento, donde puedan llevar a cabo la edición del documento por dispositivos móviles o por computadoras de escritorio, donde cada una de las personas involucradas en el documento conozca el estatus hacia el documento de los demás colaboradores. La importancia radica en que el documento podría terminarse en un tiempo más corto y con una mayor participación de las personas involucradas.

En esta Tesis presentamos el diseño y la implementación de un sistema para asistir en la edición colaborativa oportunista de documentos compartidos. Este sistema está basado en el paradigma de mensajería instantánea, debido a que utiliza el concepto de grupo para definir una lista de usuarios que trabajan en un documento compartido en particular y conocen el estado de los usuarios con respecto al documento (Editando, Leyendo, Sin usar). El sistema está implementado usando una arquitectura *Peer to Peer* (P2P) y puede ser accedido por dispositivos móviles tales como computadoras de bolsillo o teléfonos celulares, así también desde computadoras de escritorio. La arquitectura proporciona útiles propiedades al sistema como la tolerancia a fallas, la posibilidad de usar una infraestructura de llave pública para las transacciones seguras, escalabilidad, un repositorio P2P para almacenar los documentos compartidos y un servicio de percepción distribuida.

ABSTRACT

Ubiquitous Environment for Opportunistic Interaction

For people involved in a collaborative edition session, a system that supports opportunistic collaboration over a document is an important tool because it can foster relevant author interactions during crucial stages of the edition process. On the other hand, in order to maximize the chance of an opportunistic collaboration the tool has to be able to allow collaborative edition not regarding the type of device that an author is currently using. In this line of ideas the system must provide to every author with enough information to become aware of good opportunities to interact.

In this Thesis we present the design and implementation of a ubiquitous system that supports opportunistic collaborative edition of shared documents. Our system is based on the instant messaging metaphor in the sense that it uses concepts and interfaces similar to those used in instant messaging systems. We employ the concept of group to define a cluster of users that work on a particular shared document and use awareness to convey the state use awareness to convey the state of users with respect to the shared documents (editing, reading, not using it). The system is implemented using a peer-to-peer (P2P) architecture and can be accessed by means of mobile devices such as handheld computers or cellular phones as well as from desktop computers. The P2P architecture provides the system with useful properties such as fault tolerance, the possibility of using public key infrastructures to implement secure transactions, scalability, a P2P repository to store shared documents and a distributed awareness service.

ÍNDICE

RESUMEN	1
ABSTRACT	2
ÍNDICE.....	3
ÍNDICE DE TABLAS Y FIGURAS.....	6

Capítulo 0

0.1 INTRODUCCIÓN.....	9
0.1.1 Redes <i>Peer to Peer</i>	10
0.1.2 Computación ubicua.....	11
0.2 ANTECEDENTES.....	12
0.2.1 Sistema COARSY	12
0.2.2 Proyecto UARC.....	13
0.2.3 PIÑAS: Un Middleware para el trabajo cooperativo para la Web.	13
0.2.4 El sistema Quilt	14
0.2.5 El Sistema GROVE.....	15
0.2.6 ShrEdit.....	15
0.2.7 AIDA con DoMo.....	16
0.3 PLANTEAMIENTO DEL PROBLEMA.....	18
0.3.1 Escenario	20
0.4 JUSTIFICACIÓN.....	23
0.5 OBJETIVOS.....	25
0.5.1 Objetivo general	25
0.5.2 Objetivos Específicos.....	25
0.6 Resumen	26
0.7 REFERENCIAS	27

Capítulo 1 Estado del arte de computación ubicua

1.1 ESTADO DEL ARTE DE COMPUTACIÓN UBICUA	30
1.1.2 Computación Ubicua – Computación Móvil.....	31
1.1.3 Computación de bolsillo.....	34
1.4.1 Ejemplos:.....	39
1.1.4.2 Operaciones	39
1.1.4.3 Ejemplos:.....	40
1.1.4.4 Privacidad.....	41
1.1.5 Administrando Datos en servicios de información dependiente de la localización	41
1.1.5.1 Tipos de consultas	43
1.1.5.2 Consulta local contra no local	43
1.1.5.3 Consulta simple	43
1.1.5.4 Consulta general	43
1.2 Resumen	45
1.3 REFERENCIAS	46

Capítulo 2 Colaboración Oportunista

2.1 COLABORACION OPORTUNISTA.....	48
2.1.1 Actividades simultáneas	49
2.1.2 Movilidad	50
2.1.3 ¿Compartición de recursos?	50
2.2 Mensajería Instantánea (MI) para Colaboración	51
2.3 Casos de Uso	54
2.3.1 Caso de uso de un ambiente ubicuo de colaboración.	54
2.3.2 Caso de uso de colaboración oportunist entre Revisor-Autor.	55
2.3.3 Caso de uso de colaboración oportunist entre Revisores.....	56
2.4 Resumen	57
2.5 REFERENCIAS	58

Capítulo 3 Redes *Peer to Peer*

3.1 REDES <i>PEER TO PEER</i> (P2P).....	61
3.1.1 El proyecto JXTA como red virtual.	61
3.1.2 Los JXTA ID's	62
3.1.3 Los anuncios en JXTA	62
3.1.4 Los peer Rendezvous.....	63
3.1.5 Redes <i>Rendezvous</i> de baja consistencia	64
3.1.6 La vista de los <i>peers</i> del <i>Rendezvous</i>	64
3.1.7 Conexión de los <i>peers edge</i> a los <i>Rendezvous</i>	64
3.1.8 Propagación en los <i>Rendezvous</i>	65
3.1.9 Los <i>peers Relay</i>	66
3.1.10 Los mensajes en JXTA.....	67
3.1.11 Los <i>PeerGroups</i>	68
3.1.12 Los Pipes	69
3.1.13 Seguridad en el Proyecto JXTA	70
3.2 JXME.....	71
3.2.1 Principales Objetivos JXME	71
3.2.3 JXTA Relays	71
3.2.3.1 Proporcionar interoperabilidad con los protocolos JXTA.....	72
3.2.3.2 Los <i>Relays</i> no comprometen la visión P2P.	72
3.3 Resumen	73
3.4 REFERENCIAS	74

Capítulo 4 Arquitectura de ECO

4.1 ARQUITECTURA DE ECO.....	76
4.1.1 Requerimientos para sistemas de computación ubicuos para colaboración.....	78
4.1.2 Aspectos de Implementación.....	79
4.1.2.1 ECO-Repositorio de archivos (FR)	79
4.1.2.2 ECO - Servicio de percepción Distribuida (DA).....	81
4.1.2.3 ECO Herramienta colaborativa	83
4.1.2.4 El Espacio de Colaboración Potencial (ECO-IM).....	84

4.1.2.5 El Espacio de Colaboración Real (ECO-CE)	86
4.2 Resumen	88
4.3 REFERENCIAS	89

Capítulo 5 Diseño

5.1 DISEÑO	92
5.2 IMPLANTACION Y PRUEBAS	100
5.2.1 Creando un documento de colaboración	101
5.2.2 Enviar el documento al repositorio de Archivos	102
5.2.3 Abrir el documento en WS	103
5.2.3.1 En línea	103
5.2.3.2 Inactivo	103
5.2.3.3 Editando	103
5.2.3.4 En Lectura	103
5.2.4 Las pruebas realizadas	104
5.2.5 Trabajando con dispositivos móviles.	110
5.2.5.1 Encontrar a un peer móvil	110
5.2.5.2 Enviar el documento a un peer móvil	110
5.3 Resumen	112

Capítulo 6 Conclusiones y trabajos futuros

6.1 CONCLUSIONES Y TRABAJOS FUTUROS	114
6.1.1 Escalabilidad	114
6.1.2 Acceso independientemente del dispositivo	115
6.1.3 Asistencia a la movilidad y a la desconectividad	115
6.1.4 Percepción del contexto	115
6.1.5 Mecanismos de seguridad	115
6.1.6 Protocolos abiertos	115
6.1.7 Tolerancia a fallas	116
6.1.2 Limitaciones	116

GLOSARIO

APÉNDICE A

MANUAL DE INSTALACIÓN

ÍNDICE DE TABLAS Y FIGURAS

Tabla 0.1. Comparaciones entre algunos sistemas de edición colaborativa.	17
Figura 0.1a. Se despliega una interfaz de mensajería instantánea	20
Figura 0.1b. La lista de usuario también representa el estatus sobre el documento.....	20
Figura 0.2a. La interfaz del autor del documento, en la parte derecha un cuadro donde asigna permisos y en la parte de abajo una ventana de chat.....	21
Figura 0.2b. La interfaz de un revisor del documento, en la parte derecha un cuadro donde solicita permisos y en la parte de abajo una ventana para enviar mensajes cortos.....	21
Figura 0.3. La interfaz cuando un usuario solicita ingresar a la colaboración sobre un documento	22
Figura 1.1. Arquitectura de un sistema de adaptación del servicio por contenido.....	32
Figura 1.2. Tiempo que tardaría en descargarse.....	33
Figura 1.3. Ajuste del color al tiempo de descarga	33
Figura 1.4. Ajuste del ancho de banda minimizando el tiempo de descarga.....	33
Figura 1.5. Muestra una interacción entre personas a través de su computadora de bolsillo	35
Figura 1.6. Formando comunidades generales a partir de particulares	38
Figura 1.7 Sistema ContChat.....	38
Figura 2.1 Interfaz de un mensajero instantáneo.....	51
Figura 2.2 El espacio potencial de colaboración	52
Figura 2.3 Caso de uso de la edición de un revisor en un ambiente ubicuo.....	54
Figura 2.4 Caso de uso de colaboración entre un autor y un revisor.....	55
Figura 2.5 Caso de uso de colaboración oportunista entre el autor y más de un revisor ...	56
Figura 3.1 La red virtual del proyecto JXTA	62
Figura 3.2. Ejemplo de un Anuncio de un <i>Peergroup</i>	63
Figura 3.3. Los super <i>peers Rendezvous</i> del proyecto JXTA.....	63
Figura 3.4. Una DHT consistente	66
Figura 3.5. Los super <i>peers Relay</i> del proyecto JXTA.....	67
Figura 3.6. Los <i>PeerGroups</i> en el Proyecto JXTA	68
Figura 3.7. Los <i>Pipes</i> en JXTA	69
Figura 3.8. Dispositivos inalámbricos que son miembros de una red JXTA conectados a través de <i>peerRelay</i> , y pueden comunicarse independientemente de los protocolos de red.	72
Figura 4.1 La red virtual del proyecto JXTA	79
Figura 4.3 Diagrama de secuencias de la búsqueda de los coautores para mantenerlos informados del documento.	81
Figura 4.4. Diagrama de secuencias de la solicitud del documento para participar en una sesión colaborativa.	82
Figura 4.5. Diagrama de secuencias de la aceptación de la participación del Cliente2.	82
Figura 4.3. Interfaz principal de ECO (ECO-IM). a) ECO-IM corriendo sobre PDA. b) ECO-IM corriendo sobre una computadora de escritorio. En ECO el estado de un usuario está relacionado al documento compartido, es decir, si un usuario está editando o leyendo o no está en línea.	83
Figura 4.4. Diagrama de transiciones de actividades entre un trabajo individual y colaborativo en ECO	84

Figura 4.5. Implementación de una sesión de votos en ECO-IM para decidir si un nuevo miembro es admitido en la presente sesión colaborativa entre autores.....	85
Figura 4.6. ECO-CE que implementa el espacio de colaboración real, (a) sección de edición, (b) sección del panel de administración en la interfaz del usuario con más alta prioridad, (c) sección de <i>chat</i> , (d) la interfaz usada por un usuario particular para preguntar por un permiso en particular sobre el documento.....	86
Figura 5.1. Diagrama de clase de la conexión a la red JXTA	93
Figura 5.2. Diagrama de clases de los mensajes para la comunicación en la red JXTA....	95
Figura 5.3. Pantalla para la identificación del nuevo peer.	95
Figura 5.4. Definición de la clase Buddy.	96
Figura 5.5. Definición de la clase ProfileAdvertisement.	97
Figura 5.6. Información necesaria para la administración de un nuevo documento	98
Figura 5.7. Definición de la clase GuiFile.....	98
Figura 5.8. Definición de la clase ChatGuiEdition.....	99
Figura 5.9. Interfaz de colaboración entre los miembros de la red P2P	100
Figura 5.10. Interfaz para crear un documento de colaboración	101
Figura 5.11. Pantalla para elegir un documento para la colaboración.....	102
Figura 5.12. Pantalla de propiedades sobre el documento.	104
Figura 5.13. Pantalla de confirmación de los datos de administración.	105
Figura 5.14. Cuadro de diálogo donde indica que encontró un Repositorio para el archivo.	106
Figura 5.15. Pantalla de los colaboradores después de que les llega el documento.....	106
Figura 5.16. Pantalla de edición, de asignación de permisos y de envío de mensajes instantáneos para un colaborador con la máxima prioridad.	108
Figura 5.17. Pantalla de edición, de asignación de permisos y de envío de mensajes instantáneos para un colaborador con una prioridad menor.	109
Figura 5.18. Pantalla donde un colaborador solicita hacer algunas observaciones sobre el documento.	110
Figura 5.19. Pantalla de un dispositivo móvil donde aparece el documento para que pueda ser leído.	111

- Introducción
- Antecedentes
- Planteamiento del problema
- Justificación
- Objetivos



En la siguiente sección se presenta una introducción al Trabajo Colaborativo asistido por Computadora (CSCW). Así también, se presentan los antecedentes a nuestro trabajo, es decir, sistemas que tomamos de base para la realización del presente sistema de colaboración. Después planteamos el problema que vamos a resolver con algunos escenarios de colaboración utilizando nuestro sistema. Finalmente, la justificación y objetivos de nuestro trabajo.

0.1 INTRODUCCION

La rápida y continua evolución de la información y los nuevos mecanismos de comunicación entre las personas han sido de gran importancia para la trascendencia de cualquier organización, los principales aspectos son el aumento de disponibilidad de las redes de computadoras y el énfasis en los sistemas distribuidos para proporcionar asistencia al trabajo en equipo. Esto forma un dominio de actividades llamado Trabajo Colaborativo Asistido por Computadora (*Computer Supported Cooperative Work CSCW*) [0.1]. Por su parte los *Groupware* son “Sistemas basados en la tecnología computacional que apoyan a un grupo de personas involucradas en una tarea en común, con una interfaz para un ambiente de trabajo compartido” [0.1]. Eso significa que la noción de CSCW es el estudio de técnicas y herramientas de un *Groupware*, así como el estudio de los efectos psicológicos, sociales y organizacionales.

Un *Groupware* ofrece ventajas significativas sobre los sistemas de un solo usuario. Algunas de las razones principales para su aceptación son:

- Proporcionar facilidad de comunicación: más fácil, más clara y más penetrante (*pervasing*).
- Permite la comunicación, que de otra manera no sería posible.
- Reduce los costos de comunicación.
- Reúne múltiples experiencias y perspectivas.
- Forma grupos con intereses comunes
- Ahorra tiempo y costos en un trabajo de grupo coordinado.
- Facilita la solución de problemas en grupo.
- Propone nuevas formas de comunicación, tales como el intercambio anónimo o la interacción estructurada.

Por esto el área de investigación de Trabajo Colaborativo Asistido por Computadora ha experimentado un gran crecimiento y aceptación en los últimos años, con el enfoque principal en la comprensión de las características de comunicación y cooperación entre los miembros de un grupo de trabajo que se unen para resolver un problema por medio de las herramientas computacionales.

El diseño de un *groupware* envuelve la comprensión de trabajo en grupo y cómo las personas se comportan trabajando en grupo. A su vez, involucra la comprensión de la tecnología de computadoras trabajando en red y cómo los aspectos de esta tecnología afectan las experiencias de los usuarios. Todas esas ideas relacionadas al diseño de interfaces de usuario tradicional son relevantes, puesto que la tecnología todavía involucra a las personas.

Así también, varios de los aspectos de las herramientas para trabajo en grupo requieren de especial consideración. Por ejemplo, un millón de personas trabajando en grupo se comportarán diferente a un grupo que lo integran cinco personas, por lo tanto, se debe considerar parámetros de desempeño de las tecnologías para soportar a una gran cantidad

de participantes y diferentes variedades de grupos. Los sistemas deben ser sensibles y confiables, los diseñadores deberán de entender el grado de homogeneidad de los usuarios, los roles posibles que las personas desempeñan en un trabajo cooperativo y por quién son hechas las decisiones claves y cómo influyen sobre los demás.

De ahí que los avances tecnológicos y el trabajo en redes de computadoras que se caracteriza en la actualidad, determinan un incremento en el uso de la tecnología de la información para permitir el trabajo cooperativo de los miembros de un equipo.

0.1.1 Redes *Peer to Peer*

Por otra parte, el creciente desarrollo de sistemas distribuidos *peer to peer* (P2P), ha hecho posible la implementación de nuevas aplicaciones para trabajo colaborativo, como son por ejemplo, la compartición de información y los sistemas de almacenamiento de gran magnitud [0.2].

Estas redes ofrecen varias ventajas sobre los sistemas distribuidos tradicionales tales como el balanceo de carga automática y la auto organización. Pero la característica más importante de estos sistemas, son que debido a la naturaleza simétrica de par a par, el sistema puede ajustarse a las propiedades deseables cuando nuevos nodos son añadidos a la red. Esas propiedades incluyen desempeño, disponibilidad y tolerancia a fallas.

En contraste con el modelo cliente servidor, los sistemas P2P intentan ser descentralizados, auto organizados con respecto a un contexto externo, y con un enfoque de recursos balanceados. Por lo tanto, cada nodo en el sistema puede proporcionar ambas funciones tanto de cliente como de servidor.

Las redes *peer to peer*, en esencia, proporcionan una manera de unir PC's de forma conjunta sin la necesidad de un servidor central. Cuando, por ejemplo, Napster apareció (el sistema que permite que las personas busquen y recuperen información de las computadoras personales en todo el mundo), se pronosticó que transformaría la Internet. Pero *peer to peer* está ganando popularidad y atracción principalmente como una técnica glamorosa para hacer procesos de negociación más eficientes. Comúnmente, las conexiones *peer to peer* son usadas para la compartición de archivos, en ambos entornos de negocios y personales como ya se mencionó al principio. Algunos ejemplos de este tipo de canales de comunicación son: Napster, Gnutella, Morpheus, BearShare y KaZaA.

Las conexiones *peer to peer* son generalmente anónimas, por que la gente sin conocerse puede compartir información. De igual forma, un usuario puede tener acceso fácilmente a una red *peer to peer* desde una computadora de escritorio o desde un dispositivo móvil sin alguna complejidad extra e independientemente de la localización física. De ahí la integración de P2P a aplicaciones de trabajo en grupo donde los participantes sin dificultad puedan interactuar con el sistema y con los demás participantes.

Otras tecnologías y disciplinas podrían ser integradas contribuyendo desde una perspectiva y metodología diferente para adquirir conocimiento sobre grupos de personas y sugerir cómo el trabajo en equipo debería ser apoyado [0.3]. El punto es que CSCW intenta integrar la investigación de trabajo cooperativo y organizacional no sólo desde la perspectiva de tecnología sino también de la perspectiva de otras disciplinas [0.4].

0.1.2 Computación ubicua

En la visión de computación ubicua (*ubiquitous computing*) se tendrán computadoras en cualquier lugar, podrán ser accedidas en cualquier tiempo y usadas por cualquier persona [0.5], de ahí la importancia de la computación ubicua para asistir a las aplicaciones colaborativas.

En este trabajo se incluye la disciplina de computación ubicua (*computing pervasing*), porque en computación ubicua, una óptima herramienta es aquella que no acapare toda la atención de la persona, es decir, lo importante es enfocarse sobre la tarea no sobre la herramienta.

A largo plazo el uso de las computadoras de escritorio y estaciones de trabajo decaerá porque el acceso a las computadoras estará por todas partes: por las paredes, en las manos, en los ojos, en cualquier lugar físico [0.5]. La computación ubicua tiene como idea mejorar el uso de la computadora haciendo varias computadoras disponibles a través de un entorno físico, pero haciéndolo lo menos intrusivo al usuario.

Este documento tiene la finalidad de presentar el ambiente ubicuo de una herramienta de trabajo colaborativo, donde el grupo de trabajo o grupo de personas involucradas en la herramienta serán un conjunto de revisores y autores. Las actividades que tienen en común este grupo de trabajo son la edición o revisión de documentos, toda la infraestructura para el manejo de las comunicaciones y de la información será a través de las redes P2P por las características antes mencionadas.

Para el desarrollo del presente trabajo se consideraron sistemas colaborativos que sirvieron como base, es decir, se buscó aquellos sistemas con características y funcionalidades similares a los objetivos de nuestro sistema. A continuación se presentan los Antecedentes de nuestro trabajo, donde encontraremos una lista de sistemas colaborativos así como una tabla de comparación entre ellos.

0.2 ANTECEDENTES

Hablar de edición colaborativa, es hablar de la actividad que involucra a un conjunto de personas que participan en cada fase de la creación o revisión de un documento. Es decir, las personas se convierten en autores y revisores. Algunos de ellos participan directamente planeando o escribiendo el documento, mientras que otros lo revisarán y lo aprobarán. Por lo tanto, se puede decir que editar documentos es una actividad colaborativa que involucra a un sistema colaborativo [0.6].

En el área de investigación de Trabajo Colaborativo Asistido por Computadora se han desarrollado una gran variedad de sistemas colaborativos para la elaboración de documentos.

A continuación se describen algunos de los sistemas colaborativos más conocidos y que pueden servir como referencia para la elaboración de un sistema de edición colaborativo. Así como una breve comparación entre ellos.

0.2.1 Sistema COARSY

El sistema COARSY es un sistema colaborativo de edición y revisión [0.6]. Este sistema tiene una arquitectura Cliente/Servidor. El cliente es la aplicación que el usuario usa para revisar y modificar un documento. Cada cliente trabaja independiente de los demás y no tiene conocimiento de cómo comunicarse con los otros.

Este sistema usa dos servidores: un servidor de base de datos y un servidor de aplicación. El servidor de datos introduce y extrae información de la base de datos. El servidor de aplicación es usado para comunicarse con el servidor de la base de datos y así obtener o proporcionar la información requerida y generada hacia el cliente para su operación.

El servidor de aplicación recibe mensajes del cliente y determina si la información es para algún documento o es información para la base de datos.

Este sistema fué desarrollado utilizando el lenguaje de programación Java, el cual permite a las aplicaciones ser ejecutadas en varias plataformas. El sistema es ejecutado en computadoras que están conectadas al Internet.

El proyecto incluye varias características que determinan el tipo de interacción que pueden tener los participantes. Las características son la identificación del autor con los revisores, identificación entre los revisores, identificación de los revisores con el autor y el acceso de los revisores a las contribuciones hechas por otros revisores.

Entonces una vez dentro del sistema cada participante puede revisar o editar algún documento. El documento y las contribuciones son desplegados en una interfaz, la cual puede ser vista por cada participante. Se está pensando en desarrollar un cliente COARSY sobre dispositivos móviles como por ejemplo Asistentes Personales Digitales (PDA's).

0.2.2 Proyecto UARC

El proyecto UARC [0.7] se enfoca a la creación de una interfaz para el trabajo colaborativo. Esta interfaz es implementada como un sistema distribuido orientado a objetos sobre el Internet y proporciona un entorno colaborativo, donde una comunidad geográficamente extensa realiza experimentos científicos en conjunto con datos atmosféricos en tiempo real de varios lugares de estudio.

El sistema puede soportar ambas modalidades, trabajo síncrono y asíncrono, los usuarios pueden requerir múltiples secciones para lograr sus objetivos y a veces no pueden planear un tiempo común entre los participantes para sincronizar la colaboración.

Este sistema fomenta la colaboración dinámica en multigrupo. Una colaboración puede involucrar la compartición de trabajo a través de un grupo limitado. En UARC, un grupo de científicos puede estar más interesado en datos de instrumentos de radar y otro grupo en datos de imágenes satelitales. El sistema está preparado para estos casos donde se da la colaboración de grupos con diferentes intereses. Inclusive, si un grupo necesita cambiar rápidamente, los datos se mueven dinámicamente de un grupo a otro.

En un sistema CSCW los sistemas son usados por un gran número de personas, esto incrementa la cantidad de información, y las modificaciones de la información son muy constantes. Por lo tanto, se necesitan mecanismos para filtrar la información y presentar la información minimizada de los datos compartidos, así como actualizada.

0.2.3 PIÑAS: Un Middleware para el trabajo cooperativo para la Web.

Esta plataforma proporciona un medio para asistir el trabajo colaborativo entre autores de archivos de edición sobre la *Web*, con esta plataforma diferentes usuarios pueden crear documentos *Web* compartidos de una manera consistente y controlada [0.8]. PIÑAS tiene algunas características importantes tales como identificación de los autores, identificación de recursos y de documentos, un administrador de archivos, replicación de documentos, sistemas de consistencia y almacenamiento.

El diseño de la infraestructura *web* PIÑAS es guiado por los siguientes requerimientos:

- Autenticación del autor.
- Disponibilidad y replicación de la información.
- Consistencia de la información.
- Trabajo cooperativo

La infraestructura PIÑAS propone una arquitectura híbrida con el sistema COARSY (presentado anteriormente) y con algunos entornos experimentados en el proyecto ALLIANCE [0.9] en el cual diseñan un entorno para el *World Wide Web*, que consiste en un navegador/editor que permite a los usuarios distribuidos con ayuda del Internet producir documentos HTML y XML de una manera consistente.

Entonces, usando las aplicaciones ALLIANCE o el sistema COARSY los usuarios pueden acceder a documentos editables en PIÑAS. Es posible establecer enlaces entre documentos PIÑAS, y todos esos documentos son tratados de la misma forma. Se extiende la identificación y la funcionalidad del direccionamiento para permitir a los servidores *http* distinguir entre diferentes tipos de documentos, así como la naturaleza del cliente que solicita documentos (o servicios) que pueden ser del proyecto ALLIANCE o COARSY. De hecho, la identificación del cliente es solicitada para poder satisfacer de manera apropiada su petición. Para definir prioridades, el contenido cambia dependiendo de las características del cliente que lo solicite.

Por lo tanto, PIÑAS ofrece un sistema cooperativo abierto con otros sistemas de las mismas características, una completa integración, una base de datos homogénea de documentos, una identificación de documentos y recursos, así como una óptima replicación de documentos.

0.2.4 El sistema Quilt

Quilt [0.10]. Este es un sistema colaborativo asíncrono desarrollado por Bellcore. Maneja los aspectos de producción de documentos, Quilt esencialmente forma una superestructura la cual maneja específicamente los aspectos de colaboración de un grupo de autores. Para Quilt la identidad es la principal preocupación ya que esto es importante en el aspecto de coordinación, la cual asegura que el progreso del trabajo sea conocido por todos y que no exista información redundante. Compartir trabajo incluye información del trabajo, manejo del trabajo y la relación interpersonal en el grupo.

Quilt proporciona un rango de facilidades explícitas que ayudan a soportar esos aspectos de procesos colaborativos. Los principales mecanismos disponibles a los usuarios Quilt son un sistema *hipermedia*¹ que representa todo el documento con anotaciones, una grabación del rastro de las personas participantes, un correo electrónico integrado y un sistema de conferencias. Las anotaciones proporcionan un mecanismo por el cual el usuario puede comentar con cualquier otro acerca del material del documento. El sistema de rastreo permite a los colaboradores revisar las actividades de cualquier participante. El correo electrónico y el de conferencias proporcionan mecanismos con los cuales el usuario puede discutir documentos relacionados y actividades relacionadas.

Este mecanismo es un sistema de asignación de roles configurable, lo cual controla el grado de acceso que los usuarios tienen al documento. El uso de un rol explícito proporciona información acerca del carácter de las actividades de los participantes; si se conoce lo que tus colegas están “revisando”, entonces el grado de incertidumbre de las actividades es reducido.

¹ Hipermedia, una colección de documentos que contiene referencias o enlaces, la cual con la ayuda de un navegador permite la lectura y la facilidad para navegar de un documento a otro.

0.2.5 El Sistema GROVE

GROVE [0.11] es un sistema sincronizado. Este sistema está compuesto de un editor multiusuario para la creación y edición de documentos textuales. GROVE está diseñado para usar ambos, colaboración “cara a cara” y situaciones donde la colaboración es remota. GROVE toma las ventajas de la comunicación de audio para soportar conocimiento informal entre los participantes, la representación de las actividades de los otros usuarios está implícita en el sistema. Una ventana GROVE mostrará a cada usuario el texto de entrada introducido por alguno de los otros participantes, a cada nodo se le asigna una marca de activo, inactivo ó sin conexión. Las partes del documento son presentadas a cada usuario a través de una vista, la cual puede ser pública, privada o compartida. De hecho, se usa un mecanismo de control de interfaces en cada nodo.

La noción de vista sirve para diferenciar la información presentada a cada usuario, GROVE puede manejar cadenas de texto simples así como documentos fuera de línea. La estructura de GROVE, también implica servidores, de algún modo, para estructurar las actividades en las cuales el grupo trabaja. GROVE restringe algunas tareas en un esfuerzo para prevenir conflictos y proporcionar conocimiento mutuo de las actividades.

0.2.6 ShrEdit

ShrEdit [0.12] es un editor de texto multiusuario y es sincronizado. Fué desarrollado como una herramienta para explorar el soporte a diseño de documentos en una reunión entre participantes.

ShrEdit permite a un grupo de usuarios editar un conjunto de documentos colaborativamente. Cada usuario puede tener un número de ventanas privadas y compartidas. Una ventana compartida presenta una vista sobre el documento compartido; cada usuario tiene un cursor de edición con cada ventana compartida, permitiendo editar texto concurrentemente. Las vistas de los documentos son únicas a cada usuario; cada ventana del usuario puede ser de diferente tamaño así como la alineación del documento. Las ventanas privadas contienen documentos en los cuales solo un usuario puede ver y editar, y puede ser usada para hacer notas o crear documentos los cuales pueden pasar a ser en un tiempo posterior un documento compartido.

En ShrEdit la compartición de una ventana es a nivel de selección de texto, ningún usuario puede editar texto el cual ha sido seleccionado por otro usuario. Similarmente, no es posible para dos usuarios tener su cursor de edición en el mismo punto del documento. Sin embargo, las acciones de edición de los otros usuarios son desplegadas en las pantallas distintas a cada uno de los usuarios conectados.

Existe un control de ventanas asociado con cada ventana de edición el cual despliega el nombre de los participantes de la sección, con esto los usuarios pueden encontrar a los demás. También permite rastrear a los demás, lo que significa que otros usuarios pueden ver lo que los otros usuarios están observando.

En ShrEdit todos los participantes tienen igual acceso a la ventana del documento compartido y pueden editarlo en cualquier tiempo. Esto permite a los usuarios adoptar diferentes estilos de trabajo.

0.2.7 AIDA con DoMo

AIDA [0.13] es un mensajero instantáneo con el conocimiento de la presencia de un cliente con un dispositivo móvil, más específicamente para PDA's (Asistente Personal Digital). Utiliza un servidor para obtener la información del estatus de los demás usuarios. La información en el dispositivo móvil se sincroniza con el servidor cada vez que el recurso es conectado al punto de acceso.

AIDA tiene una interfaz que despliega el nombre de la persona así como su estatus, el estatus puede ser: En línea, Desconectado, Vuelvo enseguida, Ocupado y Alrededor, este último, es cuando el usuario está en tránsito de un lugar a otro usando su dispositivo móvil y puede establecer una conexión por un periodo corto.

AIDA también implementa el concepto de conocimiento de presencia asociada al usuario en una aplicación de un mensajero instantáneo, esto hace posible compartir recursos. De esta forma, soporta coordinación de actividades de edición colaborativa introduciendo el concepto de presencia de documentos.

Ofrece también nuevas oportunidades de encuentro casual en una comunidad de autores. Por ejemplo, cuando un usuario nota que el documento ha sido abierto, él podría encontrar esto como una oportunidad para enviar un mensaje relevante a la persona que tiene activo el documento ó más aún para unir a dos colegas en una sesión de revisión colaborativa sincronizada.

Se extiende la arquitectura y funcionalidad de AIDA para soportar el entorno de computación penetrante llamado DoMo. La arquitectura DoMo contiene los siguientes componentes:

- Un dispositivo móvil con AIDA.
- Un servidor o una computadora de escritorio. Esta computadora es usada para acceder a ambos, a la red de información y a los dispositivos móviles.
- Un servidor que contiene la información de las personas participantes como el nombre y el estatus.

Los agentes que representan los recursos. Estos agentes pueden estar en cualquier otra computadora con acceso al punto de acceso. Un agente implementa un protocolo para registrarse en el directorio de agentes del servidor, una interfaz para interactuar con un recurso desde el dispositivo móvil o una computadora de escritorio.

Como se puede ver, existe una gran diversidad de trabajos realizado al respecto, aquí se trató de redactar los más importantes y los más parecidos al sistema que se desea implementar, así como los más referenciados en la literatura.

A continuación se exhibe un cuadro comparativo presentado en [0.13]:

Tabla 0.1. Comparaciones entre algunos sistemas de edición colaborativa.

Sistema	Tipo de interacción	Actividad de procesos	Definición de roles	Conocimiento de colaboración	Comunicación entre participantes	Beneficios para todos
Alliance	Síncrono /Asíncrono	Escritura	Editor, lectores, usuarios nulos	Trabajo síncrono	Ninguno	Autores
CES	Síncrono	Escritura	Autor	Trabajo sincronizado	Conferencia en tiempo real	Autores
D3E	Asíncrono	Revisión	Revisores	Comentarios	Comentarios entre autores	Revisores
Group works	Asíncrono	Revisión de trabajo coordinado	revisores	Anotaciones	Anotaciones	Revisores y autores
Grove	Síncrono	Escritura	Autores, modificación de privilegios	Trabajo sincronizado	Ninguno	Autores
Intant Update	Síncrono /Asíncrono	Simple escritura	Autores, Lectores	Cambio de las listas de usuarios	Ninguno	Autores
Quilt	Asíncrono	Escritura y Revisión	Autores, revisores y lectores	Comentarios y e-mail	Contribuciones e-mail	Autores y revisores
PREP	Asíncrono	Escritura y revisión	Definición de funciones de autores	Comentarios	Comentarios	Autores y revisores
SASSE	Síncrono /Asíncrono	Escritura	Autores	Trabajo sincronizado	Teleapuntadores	Autores
SEPIA	Síncrono /Asíncrono	Creación, planeación, Argumentación, Estructuración	Autores	Trabajo sincronizado, barras de estado de los participantes	Audio teleapuntadores	Autores

0.3 PLANTEAMIENTO DEL PROBLEMA

Desde siempre se ha pensado que el trabajo en grupo debería de dar mejores resultados que el trabajo individual, porque la colaboración puede facilitar el desarrollo de una tarea, aproximar a los mejores resultados y agilizar el tiempo de culminación de la misma. Pero no siempre sucede así.

En este trabajo de investigación, se aborda el problema de la falta de coordinación cuando se es integrante de un grupo de trabajo, en el cual todos operan con los mismos objetivos en mente, pero la colaboración y los resultados no son los esperados o los resultados muchas veces son prolongados más de lo planeado.

Existen muchas razones por las cuales el trabajo colaborativo no siempre tiene los resultados esperados. Una causa importante es que las personas no están acostumbradas al trabajo donde tienen que interactuar constantemente con un gran número de individuos, otra causa es que la mayoría de veces que se divide el trabajo no se hace de la manera correcta, ya que no se realiza un estudio previo de las capacidades y conocimientos de los que integran el equipo de trabajo, lo que provoca que se extienda la culminación del mismo o que los resultados no sean los mejores.

La raíz del problema, como se mencionó al inicio, es la falta de coordinación y comunicación por parte de los integrantes, pero más que eso es por la falta de herramientas de software que puedan promover y facilitar la comunicación entre éstos. Estas herramientas tienen que satisfacer las necesidades básicas como son:

- Debe facilitar la comunicación entre cada uno de los integrantes y de manera simultánea sin importar la ubicación física de estos.
- Debe promover la participación oportuna de los integrantes debido a la diferencia de horarios de trabajo.
- Debe dar la facilidad de poder interactuar con recursos que componen el proyecto, de forma colaborativa y de forma consistente.
- Debe de poder soportar un gran número de participantes, sin que esto afecte la comunicación.

El enfoque que se propone en este proyecto para resolver el problema planteado, consiste en proporcionar una herramienta de software que pueda promover y facilitar el trabajo cooperativo entre los integrantes de un grupo de trabajo. Esa herramienta debe de cubrir las necesidades anteriormente planteadas. También tiene como objetivo el poder compartir un recurso en el que todos los integrantes estén interesados, este será un archivo de edición, al cual todos tendrán acceso y podrá ser utilizado de una forma coordinada.

La herramienta está pensada para interactuar con una red inalámbrica en un ambiente ubicuo en el que se puede acceder a un recurso dependiendo de la ubicación de quien lo

solicita. Esta herramienta tendrá la interfaz de cualquier mensajero instantáneo (IM) el cual es familiar a un gran número de usuarios del *Internet*.

Podrá ser accedida:

- A través de una estación de trabajo, localizada en un lugar fijo y que forma parte de la red inalámbrica.
- A través de un dispositivo móvil (como puede ser un Asistente Personal Digital (PDA)) al ingresar éste a la red inalámbrica.

La interfaz proporcionará información de los colaboradores conectados en ese instante de tiempo, y dará la posibilidad de entablar una comunicación con quien se deseé.

No solo los participantes formarán parte de la interfaz, sino también los recursos que componen el proyecto a los cuales los participantes tienen permisos de acceder, por lo que cualquier recurso formará parte de la interfaz como un contacto más (en este caso el archivo de edición), al que todos pueden acceder coordinadamente.

Al momento en que cualquier integrante acceda al recurso, se le informará de las prioridades que tiene sobre el recurso, también se le proporcionará información si hay algún integrante utilizando el recurso, así como su identificación.

- En caso de que un usuario no encontrara a alguien más utilizando el recurso, él tendría la prioridad más alta y contaría con los permisos de edición sobre el documento y los cambios serían observados por cualquier integrante que acceda a ese recurso en un instante después.
- En el caso de que hubiera algún otro integrante editando el recurso, entonces se tendría que comparar prioridades sobre el archivo, aunque todos estarían observando los cambios realizados por quien tiene la prioridad más alta. De cualquier forma, todos podrían exponer sus comentarios sobre lo observado enviando algún mensaje instantáneo a la persona que está editando haciéndole alguna aclaración.

Se utilizará como caso de prueba un sistema para el proceso de revisión de un documento de tesis, donde normalmente, primero se envía al primer revisor del jurado, después de que lo revisa y hace las correcciones que le parecen necesarias las manda de regreso al autor para que haga las correcciones propuestas, después de hacer las correcciones este proceso se repite con cada uno de los revisores de la tesis, lo cual se convierte en un trabajo ineficiente por que los cambios propuestos por el primero que revisó podrían no coincidir con los propuestos por el último, esto pondría en duda al autor sobre qué cambios hacer, la participación de los integrantes no sería justa por que la revisión debería hacerse en forma conjunta para que todos concuerden en los cambios, a la vez esto puede traer como consecuencia que la tesis se alargue más de lo planeado.

Por lo anterior, el poder proporcionar una herramienta colaborativa para solucionar satisfactoriamente los problemas de comunicación y coordinación entre un grupo de revisores, nos permitirá obtener un mejor desempeño en el proceso de revisión de una tesis o un artículo. Y por consecuencia se podrían obtener más y mejores resultados en un menor tiempo.

Desde el punto de vista tecnológico, la presente tesis integrará tecnologías relacionadas con el desarrollo de sistemas *Peer to Peer* como JXTA. JXTA es un conjunto de protocolos abiertos para construir sistemas *Peer to Peer*. Es decir, permite a los integrantes de una red de computadoras y de dispositivos móviles comunicarse y colaborar conjuntamente.

0.3.1 Escenario

Cuando un nuevo elemento se une a la red P2P del sistema, ya sea desde un dispositivo móvil o una computadora de escritorio, se desplegará una interfaz parecida a un mensajero instantáneo (como lo indica la figura 0.1a) donde se le informa de las personas que están involucradas con el documento. En la figura 0.1b se muestra que en la lista de usuarios también se expone el estatus relacionado al documento de cada uno de los usuarios, por ejemplo Emilio y Rolando se encuentran editando el documento mientras Jesús está observando la edición y Anabel está en línea pero no está interactuando con el documento.

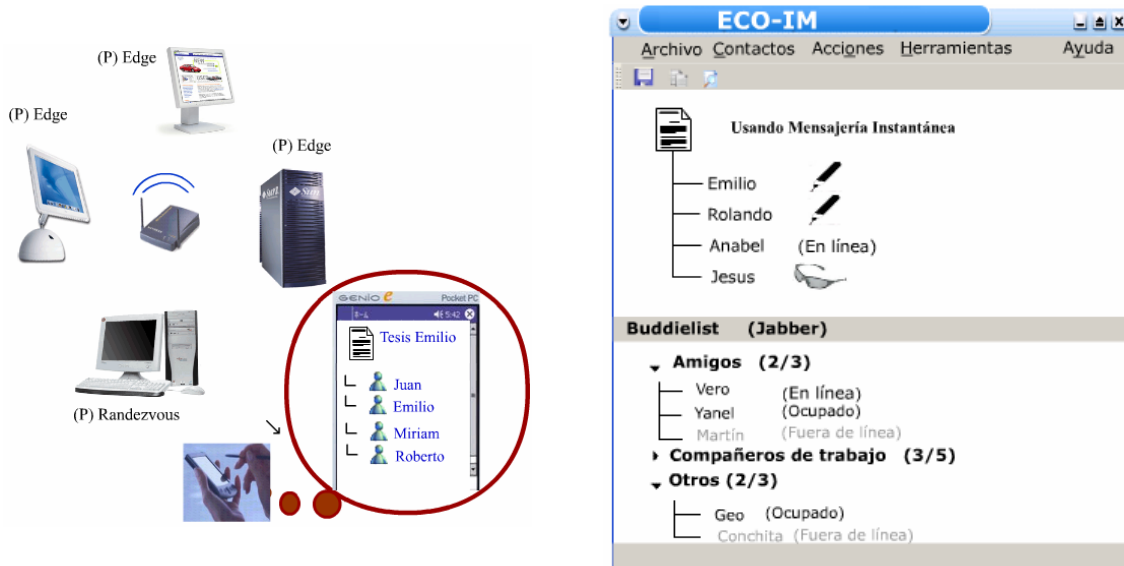


Figura 0.1a. Se despliega una interfaz de mensajería instantánea

Figura 0.1b. La lista de usuario también representa el estatus sobre el documento

En la figura 0.1b, también se muestra en la parte de abajo una interfaz de un mensajero comercial como lo es Jabber, con Jabber se puede estar en contacto con los usuarios de otros mensajeros instantáneos comerciales. Es decir, lo que se quiere en el sistema es presentar una interfaz de mensajería instantánea comercial y una interfaz para la colaboración sobre un documento, no queremos restringirle al usuario lo que ya conoce, al contrario, el objetivo es agregar la colaboración a una interfaz que ya conoce y maneja como lo es la mensajería instantánea.

Desde el momento en que el documento se pone a disposición de los revisores, éste puede ser editado dependiendo las prioridades que cada usuario tenga sobre el documento, la figura 0.2a muestra la interfaz de un usuario que tiene la prioridad más alta sobre el documento, por ejemplo el propietario del documento, él puede decidir qué permisos asigna a los revisores. En la figura 0.2b se muestra la interfaz de un revisor, en ella se puede ver una parte de permisos con la que puede solicitar un permiso sobre el documento, esto será solicitado al usuario que tenga la más alta prioridad.

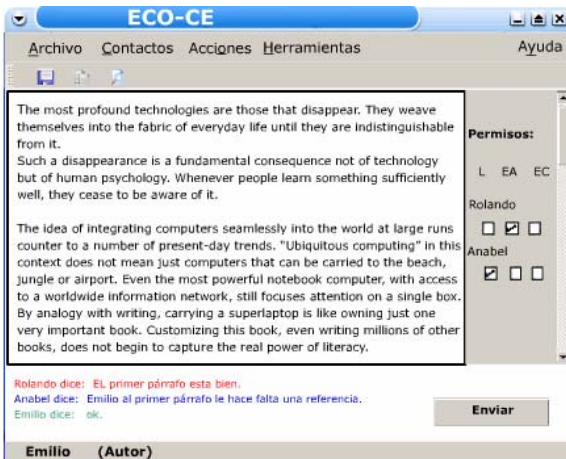


Figura 0.2a. La interfaz del autor del documento, en la parte derecha un cuadro donde asigna permisos y en la parte de abajo una ventana de chat.

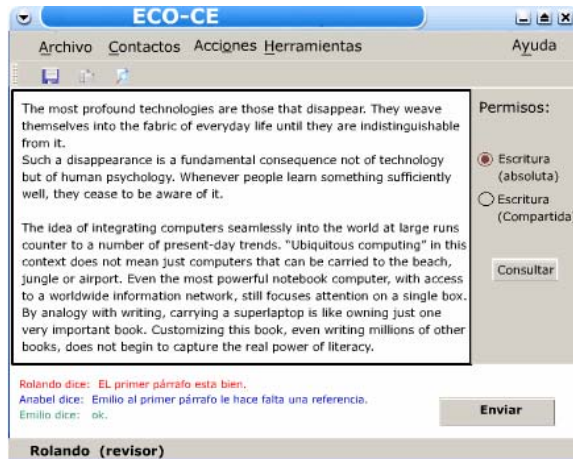


Figura 0.2b. La interfaz de un revisor del documento, en la parte derecha un cuadro donde solicita permisos y en la parte de abajo una ventana para enviar mensajes cortos.

En las figuras 0.2a y 0.2b también se exhibe, en la parte de abajo, una ventana de chat donde las personas pueden estar continuamente enviándose mensajes cortos de los comentarios y aportaciones sobre el documento, por ejemplo, Rolando que es un revisor puede hacerle sugerencias al autor que es Emilio sobre la redacción de su documento.

Después de un tiempo, un nuevo integrante también relacionado con el documento puede entrar al sistema como se muestra en la figura 0.3. Victor (otro revisor) entra al sistema, en su lista se dará cuenta que Rolando y Emilio ya se encuentran interactuando con el documento, entonces él notará que es una buena oportunidad para que él se una a la revisión del documento, entonces solicitará permiso para participar con el documento, las personas que se encuentran editando el documento aprobarán la participación de Victor.

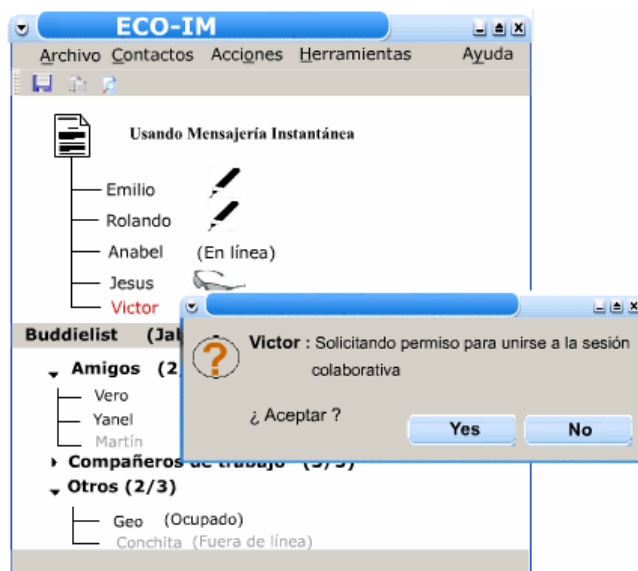


Figura 0.3. La interfaz cuando un usuario solicita ingresar a la colaboración sobre un documento

Por todo esto la mensajería instantánea puede ayudar a la colaboración oportunistas sobre un documento, porque en todo momento se observa la presencia de los integrantes y sus acciones sobre el documento.

0.4 JUSTIFICACIÓN

Es de gran importancia la actividad de realizar una herramienta de software que facilite y mejore la comunicación entre un grupo de personas que trabajan para los mismos fines, de tal forma que exista una mejor coordinación entre ellos. En este documento se pretende retomar el enfoque de Trabajo Colaborativo Asistido por Computadora (CSCW), el cual estudia el desempeño de una herramienta de software diseñada para auxiliar y facilitar el trabajo de grupo. Así mismo, se integrarán las tecnologías de los sistemas *peer to peer* y la Computación ubicua, las cuales pueden ser de gran utilidad en el intercambio y procesamiento de información en un grupo de trabajo.

Se hace importante introducir el concepto de trabajo colaborativo asistido por dispositivos móviles debido a la proliferación de los diferentes dispositivos móviles, la gran aceptación que tienen hoy en día y la movilidad que ofrecen a los usuarios, ya que no siempre se puede estar trabajando desde una estación de trabajo fija, debido a las necesidades que tiene las personas de desplazar su centro de trabajo de un lugar a otro. Luego entonces, con la participación de los dispositivos móviles el trabajo puede ser realizado desde cualquier ubicación y en cualquier instante de tiempo.

Es importante la contribución con el desarrollo de sistemas colaborativos, el proponer el diseño de un entorno de trabajo en el que se facilite la constante comunicación entre los miembros y así mejorar la coordinación cuando se trata de interactuar con un recurso que es de interés para todos los integrantes.

En un entorno de edición y revisión de documentos, donde la participación es oportunista (también llamada informal), el proceso de colaboración coordinada sobre archivos de edición podría realizarse de una manera más óptima y dar mejores resultados. Por ejemplo, imaginando el problema de la revisión de un documento de tesis en un entorno colaborativo, donde el documento está disponible para cualquiera de los miembros del jurado, cualquier integrante podría iniciar la revisión en cualquier momento, y la participación oportunista haría posible que algún otro miembro iniciase el mismo proceso de revisión. Este último se daría cuenta que alguien más se encuentra realizando la revisión, por lo tanto estaría observando los cambios propuestos e inclusive podría hacer algunos comentarios de manera instantánea al primer revisor, o también podría conjuntar a otros autores para que iniciasen una revisión en el mismo momento. Por lo tanto, el proceso de revisión se hace colaborativo y de una manera coordinada. Ya que dos o más miembros del jurado de la tesis podrían participar de manera simultánea con el proceso de la revisión de una tesis e inclusive hasta el mismo autor podría estar interactuado con los miembros de su jurado y hacer los ajustes y correcciones en el instante en que se lo proponen, y estos cambios podrían ser observados inmediatamente por todos los demás participantes, lo cual haría que el proceso fuera menos tedioso, promovería la participación de los integrantes del jurado en una forma conjunta y la revisión sería realizada en un menor tiempo.

El poder compartir un archivo de edición, hará posible una mejor y mayor participación de los integrantes del equipo de trabajo, ya que todos pueden dar sus puntos de vista que daría

como resultado un mejor producto final. Este archivo de edición puede ser cualquier documento que pueda ser editado Artículos, Reportes Técnicos, Tesis, etcétera.

Como se mencionó anteriormente, para este trabajo de investigación se incorporan los servicios *peer to peer* de los protocolos JXTA lo cual, dado las características es de gran valor para los objetivos y metas esperadas, ya que esta tecnología permitirá la interconectividad de recursos móviles con estaciones de trabajo fijas sobre una red, con lo cual lograremos la comunicación entre distintos usuarios y la participación colaborativa de los mismos.

También, el tratar de formar un entorno ubicuo con el uso de los dispositivos móviles, hará posible que la parte de trabajo distribuido sea mejor realizada por la movilidad que se puede conseguir de este entorno.

Por todo lo anterior este proyecto es de gran utilidad, porque introduce una serie de nuevas tecnologías con el único objetivo de mejorar el proceso de comunicación en un entorno colaborativo entre un grupo de trabajo, permitiéndonos compartir recursos de forma coordinada.

0.5 OBJETIVOS

0.5.1 Objetivo general

El objetivo principal es diseñar e implantar un sistema que soporte la colaboración oportunista durante la edición y revisión de documentos, y que pueda ser ejecutado tanto en dispositivos móviles como en computadoras de escritorio sin importar que se utilicen redes inalámbricas o fijas para llevar a cabo la comunicación.

0.5.2 Objetivos Específicos.

- Diseñar un ambiente ubicuo que facilite la comunicación coordinada de documentos.
- Utilizar el paradigma de mensajería instantánea para proporcionar una interfaz de colaboración oportunista mediante la cual los usuarios puedan enterarse de posibilidades de colaboración.
- Utilizar arquitecturas P2P para mejorar propiedades como la escalabilidad y aumentar la disponibilidad de los servicios.
- Diseñar e implementar aplicaciones que puedan ser ejecutadas en dispositivos móviles que utilicen redes inalámbricas.
- Diseñar e implementar protocolos de consistencia de réplicas que tomen en cuenta las restricciones de los dispositivos móviles.
- Implementar protocolos que reduzcan el número de mensajes necesarios para llevar a cabo la comunicación con el fin optimizar el uso de la batería en dispositivos móviles.
- Desarrollar un conjunto de pruebas tanto objetivas como subjetivas que demuestren la utilidad y funcionamiento del sistema.

0.6 Resumen

Del presente capítulo podemos resumir lo siguiente. En la Introducción abordamos los temas de sistemas distribuidos para trabajo colaborativo, es decir, sistemas pensados para asistir a un grupo de personas con un interés común. Mencionamos cuales eran las ventajas de la utilización de éste tipo de sistemas debido a que el presente trabajo se refiere al desarrollo de un sistema para asistir a un grupo de autores que editan un documento en conjunto. Dimos una introducción de lo que son las redes *peer to peer* debido a que nuestro sistema está basado en una comunicación *peer to peer*. Donde los autores del documento son *peers* dentro de la red *peer to peer* que se encuentran y se comunican para editar el recurso que en éste caso es el documento de edición compartido.

Por otra parte, en el apartado de antecedentes proporcionamos una lista de los sistemas que más se aproximan a las características de nuestro sistema, la lista se refiere a sistemas colaborativos sobre documentos compartidos. Cada uno de estos sistemas ha servido de base para la realización del presente sistema de edición colaborativa.

Por último, abordamos una breve visión de lo que es computación ubicua debido a que estamos pensando en la realización de un sistema ubicuo de edición colaborativa, donde podamos aprovechar cada uno de los recursos disponibles dentro de un ambiente ubicuo como son los dispositivos móviles, los cuales consideramos en este sistema como un recurso más para la colaboración debido a su gran aceptación y su accesibilidad.

Como el título de este trabajo menciona, queremos desarrollar un sistema ubicuo de colaboración, por lo tanto, en el siguiente capítulo se refiere al estado del arte de computación ubicua y se considerarán algunos sistemas ubicuos que puedan servir de referencia para el nuestro.

0.7 REFERENCIAS

- [0.1] Ellis C.A., Gibbs S J., Rein G.L. *Groupware Some Issues and Experiences*, Communications of the ACM, Vol. 34 not. 1, 1991, pp. 38-58.
- [0.2] Rodrigo Rodrigues, Barbara Liskov, Liuba Shriru. *The Design of a Robust Peer-to-Peer System*. Proceedings of the 10th ACM SIGOPS European Workshop, Saint Emilion, France, September 2002.
- [0.3] Saul Greenberg, "*Computer Supported Cooperative Work and Groupware: an Introduction to the Special Issues*", International Journal of Man-Machine Studies, v.34, 1991, p. 133-141.
- [0.4] Khoshafian, Setrag y Buckiewicz, Marek: "Introduction to Groupware, Workflow and Workgroup Computing", John Wiley & Sons, Inc.1995.
- [0.5] M.Weiser. *The computer of the 21st century*. Scientific American, pages 94-100, September 1991.
- [0.6] Favela, J., & Ruiz, D. Collaborative Authoring and Reviewing over the Internet. *WebNet Journal: Internet Technologies, Applications & Issues* 3(3), 26-34 (2001).
- [0.7] A. Prakash, H.S. Shim, and J.H. Lee. Data management Issues and Trade-Offs in CSCW Systems. IEEE Transactions on Knowledge and Data Engineering, 11(1) (1999).
- [0.8] Dominique Decouchant, Jesús Favela, Ana María Martínez Enríquez: PIÑAS: A Middleware for Web Distributed Cooperative Authoring. SAINT 2001: 187-194.
- [0.9] Decouchant A. M. Martínez et E. Martínez, "AllianceWeb: Cooperative Authoring on the Groupware, IEEE Computer Society, Cancún, México, 15-18 September 1999"
- [0.10] Robert S. Fish, Robert E. Kraut, Mary –D.P. Leland and Michel Cohen, "Quilt – A Collaborative Tool for Cooperative Writing", Proc. COIS'88 Office Information System (Palo Alto, Ca, March 1988).
- [0.11] Clarence Ellis, Simon Gibbs, "Desing and use of a Group Editor", in Cockton (ed.), "Engineering for human-Computer Interaction", North-Holland, 1990.
- [0.12] Dourish, P. and Bellotti, V., "Awareness and Coordination in Shared Workspaces," Proc. ACM CSCW Conf., 1992, 107-114.

- [0.13] Favela, J., C. Navarro, and M. Rodríguez “Supporting Opportunistic Interactions with People and Services in Pervasive Computing Environments”, Supporting Spontaneous Interaction in Ubiquitous Computing Settings a workshop at Ubiquitous Computing International Conference (UbiComp 2002), October 1-3, Gotengorg, Suecia.

Computación UBICUA

“La tecnología más trascendente es aquella que es desapercibida”

Mark Weiser.



En la siguiente sección se presenta el estado del arte de Computación Ubicua (*ubiquitous computing*), se exhiben algunos ejemplos de sistemas ubicuos con características similares a las presentadas en esta tesis donde se describen las ventajas y desventajas que nos podemos encontrar al desarrollar una herramienta ubicua.

1.1 ESTADO DEL ARTE DE COMPUTACIÓN UBICUA

La computación ubicua es un área nueva de investigación, la idea surge en 1988 para describir el futuro de la computación, cuando se concibe que la computadora pasará a ser desapercibida a las personas. Tal afirmación fue hecha por *Mark Weiser* a quien se le considera el padre de la computación ubicua.

Lo que se refiere a “desapercibido” no solo es una consecuencia del avance de la tecnología sino también se refiere a la psicología humana. Es decir, involucra el hecho en cómo las personas reciben información, cómo la interpretan y cómo actúan ante ella. Por ejemplo, cuando una persona se encuentra manejando por la autopista y ve una señal de tránsito, la persona asimila la información sin hacer un análisis muy consciente del proceso de lectura, porque ya ha visto muchas veces la misma señal.

La idea es que en el momento en que las cosas pasen a ser desapercibidas para el ser humano, es cuando podremos dejar de pensar en el objeto mismo y tendremos la libertad de poder pensar en cómo utilizarlas y con esto poder hacer surgir nuevas ideas. Sin embargo, la interacción que se tiene con la computadora actualmente no es la mejor para el cómputo ubicuo, porque en vez de que la herramienta nos facilite enfocarnos más a la tarea que a la herramienta, sucede todo lo contrario [1.1].

Comparando la computación ubicua con los agentes inteligentes, en donde la idea es programar a la computadora para que pueda tomar algún comportamiento parecido al de un humano, esto toma un poco de tiempo y atención por parte de las personas, porque a una computadora se le tiene que estar comunicando, se le tiene que dar comandos para una determinada tarea y debe existir una constante y estrecha relación, esto hace que la computadora se vuelva el centro de atención, de lo cual la computación ubicua se quiere deshacer.

Podemos tomar la realidad virtual como otra comparación, en la realidad virtual la intención es hacer un mundo dentro de la computadora lo más parecido al mundo real, donde a las personas se les permita explorar un mundo virtual que de otra forma no sería posible hacerlo. Con ella, podemos explorar planetas, bosques inaccesibles, las células, una computadora internamente; pero solo vemos lo que deseamos ver sin considerar lo que realmente compone al mundo. La realidad virtual se enfoca a un aparato enorme sobre el que se simula al mundo, más que en mejorar la invisibilidad del mundo que realmente existe. En [1.2] se le llama “virtualidad incorporada” para el proceso. La virtualidad de las computadoras se refiere a las diferentes formas en que los datos pueden ser alterados, procesados y analizados acercándolas a un mundo físico.

La computación ubicua se enfoca a dos metas cruciales en todo sistema de computadoras como son la *localización* y la *escala*. En cómputo ubico se cree que se tendrían mejores resultados de trabajo si las computadoras conocieran el entorno que les rodea. Es decir, que los dispositivos que transmiten y despliegan información en un sistema ubicuo deben de conocer dónde se encuentran localizados, si éstos conocieran el lugar que ocupan

físicamente y los dispositivos que le rodean, éstos podrían adaptar su comportamiento al medio y así poder interactuar con su alrededor sin la necesidad de ayuda de la Inteligencia Artificial.

1.1.2 Computación Ubicua – Computación Móvil

En computación ubicua existe también una estrecha relación con el área de computación móvil porque es posible interactuar en un entorno ubicuo desde algún dispositivo móvil. Pero trabajar con dispositivos no es una tarea fácil, porque se pueden presentar problemas para trasladar la información hacia los dispositivos personales, como por ejemplo: El Ancho de Banda, el cual es limitado en estos dispositivos, y el contenido de la *web* incluye contenido que requiere adaptarse a ciertas capacidades cuando los dispositivos acceden a ella, entonces se ven las limitantes de estos dispositivos. Por ejemplo, la información puede ser contenido multimedia, entonces el dispositivo no podría soportar tan fácil este tipo de información. Por lo tanto, existen algunos desafíos en los que se tiene que trabajar.

El desafío es aún más grande si se contempla también que los dispositivos personales son cada día de tamaños más reducidos, lo cual trae como consecuencia que las áreas donde se despliega la información sea en varios casos muy pequeña, el introducir pantallas pequeñas hace que la resolución en píxeles sea también más limitada para presentar la información. Mientras que el contenido *Web* codifica diferentes modos tales como JPG, WAP (formato de *Wireless Application Protocol*). Y más aún, estos dispositivos móviles trabajan con diferentes lenguajes para las transacciones como pueden ser HTML, HDL (*Handheld Device Markup Language*), y WML (*Wireless Markup Language*) [1.3]. Por lo tanto, en un ambiente ubicuo con dispositivos móviles se tiene que buscar la forma de cómo se les presentará el contenido, de manera que no existiese algún tipo de pérdida mientras se envía o por lo menos tratar de modificar los datos para tratar de resolver los problemas por alguna deficiencia o alguna mala interpretación de la información.

En [1.3] se propone un sistema basado en contenido. En éste se propone que el sistema se encargue de verificar cuál será la mejor versión que se entregará al usuario basada en el mismo contenido del documento que se esté solicitando. También verificará la forma en que será presentada. Éste debe tomar en consideración las capacidades del dispositivo, así como considerar el conocimiento del contexto [1.4] enfocándose a las preferencias del usuario.

El proceso inicia cuando un usuario con su dispositivo móvil lanza una solicitud al sistema, a partir de un intermediario servidor proxy. Entonces el sistema identifica al usuario e introduce la información de contexto a un denominado motor de decisiones que se encuentra en el servidor. Después el motor de decisiones ejecuta un algoritmo que busca la versión óptima que es interpretada con el dispositivo móvil del usuario presente y con las características de la red. La versión óptima significa cubrir las necesidades deseadas como color de fondo, factor de escala, y formato de presentación (ver figura 1.1). El motor de decisión envía los resultados a otro servidor, el cual genera la versión de contenido deseada. El servidor proxy entonces envía el contenido adaptado al recurso solicitante.

El sistema asimila el contexto del usuario en cuatro dominios de calidad: color, tiempo de descarga, modalidad y segmentos. El usuario puede especificar el tiempo de descarga máximo que podría tolerar para la entrega del documento. El dominio de modalidad permite al usuario especificar cómo se preservará el contenido original o se cambiará con respecto a la transferencia. El dominio de escala tiene cuatro valores correspondientes al formato de salida que ya se mencionaron WML, HTML, mapa de bits y PDF.

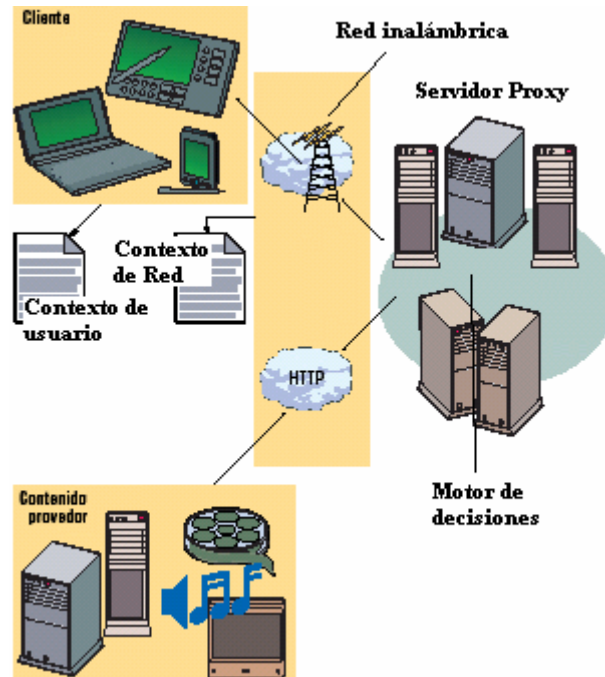


Figura 1.1. Arquitectura de un sistema de adaptación del servicio por contenido

Para verificar las capacidades del dispositivo acerca del contexto, el sistema toma en consideración el tamaño de la pantalla, la capacidad en píxeles y tamaño de buffer de memoria.

Para el contexto en la red, se consideraron el ancho de banda y el tiempo de recorrido de algunos canales de comunicación populares como por ejemplo: Acceso múltiple por división de código (CDMA), Servicio General de Paquetes de Radio (GPRS), Datos de Paquetes Digitales Celulares (CDPD).

Con respecto a la máxima tolerancia de tiempo de descarga, si el usuario especifica un máximo de tolerancia, el sistema negociará la versión del contenido automáticamente para reunir estas especificaciones, en este caso el contenido podría requerir segmentación. En la Figura 1.2 se muestra una especificación del usuario con respecto al máximo tiempo de tolerancia al tiempo de descarga decrementado de izquierda a derecha.

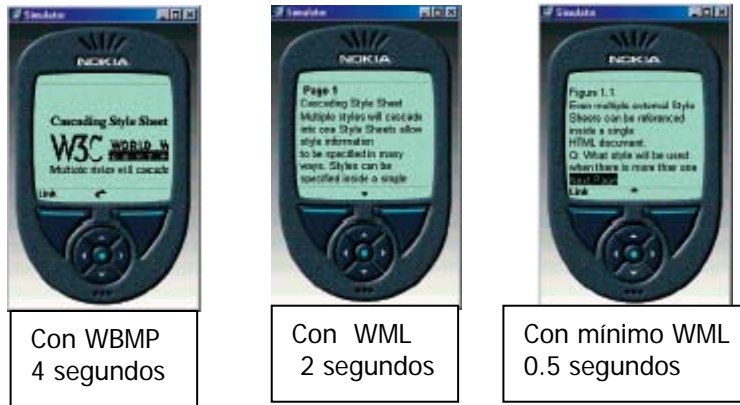


Figura 1.2. Tiempo que tardaría en descargarse

Considerando el factor de color contra el tiempo de descarga, la Figura 1.3 muestra el efecto de negociación del algoritmo intentando hacer un balanceo entre los factores de color y de tiempo de descarga. Decrementando los pesos sobre la percepción del color del usuario, el sistema responde ajustando los valores a 256, 16 y 2 colores para el documento PDF.

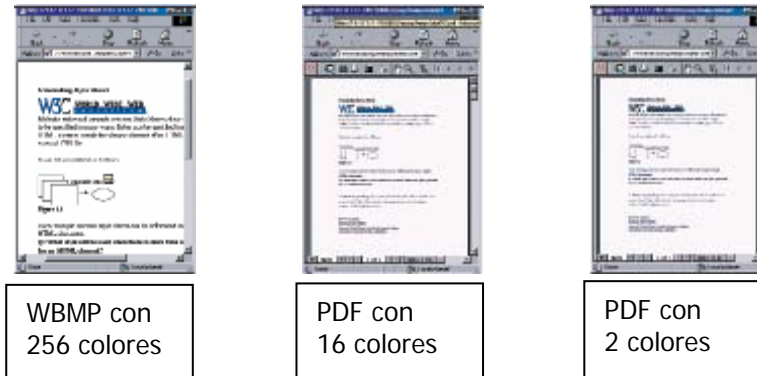


Figura 1.3. Ajuste del color al tiempo de descarga

Con respecto a la red, este sistema intenta adaptarse a diferentes características de la red, variando el ancho de banda mientras obtiene otros factores constantes. En la Figura 1.4 se cambia de modalidades para ajustar el ancho de banda y para minimizar el tiempo de descarga.

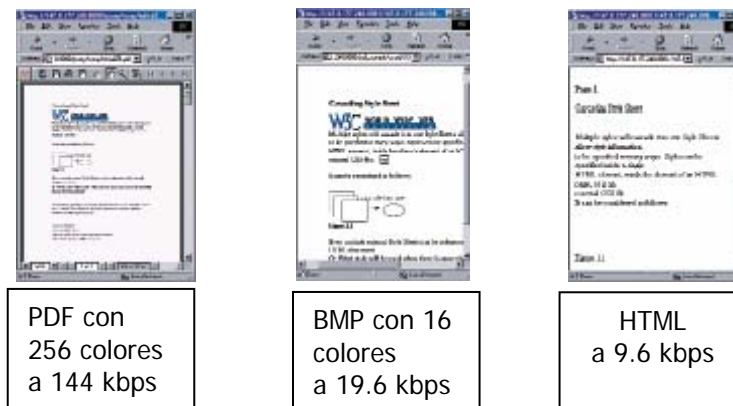


Figura 1.4. Ajuste del ancho de banda minimizando el tiempo de descarga.

Y en cuanto a las capacidades del recurso, ajustaron el tamaño del buffer de memoria para ver si el sistema automáticamente regresaba la versión de contenido óptima.

1.1.3 Computación de bolsillo

A su vez, con la aparición de la computación móvil con dispositivos de bolsillo se ha apostado que esto mejorará más la interacción de las personas con una computadora, ya que se tiene la oportunidad de llevar una computadora contigo a todas partes, esto se puede ver como una oportunidad para trabajar en cualquier lugar sin la necesidad de llevar la oficina contigo, solo tu herramienta de trabajo. Esto aumentará la interacción social porque podrías estar comunicado con cualquier persona en cualquier lugar, y aún más, esto podría aumentar el trabajo colaborativo entre un conjunto de personas en movimiento.

En la actualidad se está trabajando en cómo podría incrementarse el potencial social con esta tecnología. En [1.5] se plantean algunas preguntas interesantes a considerar:

¿Qué tiene la computación de bolsillo para hacer una sociedad potencial?

¿Cuáles son los requerimientos y características del sistema que realizarán ese potencial?

¿Cómo podemos sistemáticamente y efectivamente construir esos sistemas?

En [1.5] se presenta una arquitectura de la tecnología de bolsillo para la interrelación social.

Las comunidades de computación de bolsillo denotan las redes sociales que podrían surgir al unir a un conjunto de personas con esa tecnología en su vida diaria, se hace una comparación con las comunidades que se forman con el Internet pero ahí le denominan una comunidad en línea. La diferencia es que los encuentros en una comunidad con tecnología de bolsillo son a partir de las computadoras móviles y a través de una WPAM (*Wireless Personal Area Network*), esto es lo que hace surgir nuevas formas de interacción espontánea entre personas.

Este sistema introduce un enfoque de trabajo a partir de lo que denominan espacios de sociedades aumentadas, generando esferas digitales que involucran conjuntos de usuarios y computadoras de bolsillo, formando un espacio social donde las personas interactúan (ver figura 1.5). En [1.6] se sugieren cuatro distintos niveles de comunicación interpersonal como son: Intimo, personal, social y público.

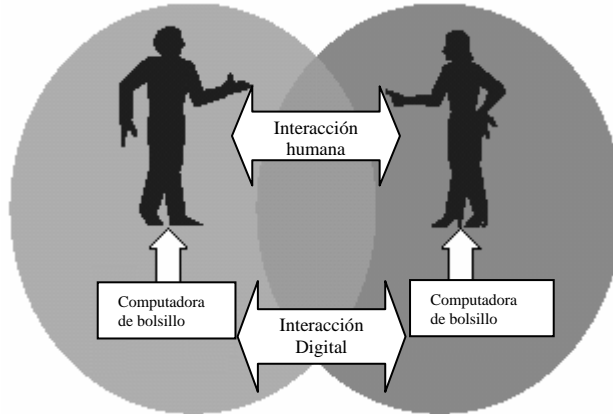


Figura 1.5. Muestra una interacción entre personas a través de su computadora de bolsillo

El *broadcast* de información de la computadora de bolsillo cubre un denominado espacio social digital determinado por el papel de la red WPAN, cuando dos computadoras se encuentran en un espacio digital social, entre ellas se asimilarían [1.4]. Basándose en el espacio social aumentado, se define una comunidad de tecnología de bolsillo como una red social que los participantes pueden formar a partir de sus computadoras de bolsillo. Las comunidades se pueden formar a partir de encuentros oportunos.

También en [1.5] se introduce un sistema llamado GENIE el cual hace una combinación a partir de las habilidades de los grupos disponibles en una WPAN, un usuario GENIE define un conjunto de preguntas que se almacenan en su computadora de bolsillo, cuando se encuentra con otro usuario los dispositivos intercambian las preguntas. Es decir, se pregunta al usuario nuevo si conoce la respuesta a la pregunta. El sistema solo transmite con la indicación de la persona. Para el problema de identificación el dispositivo también puede intercambiar la información personal de cada usuario.

En [1.5] también se introduce otro sistema donde la interacción es implícita, es decir, sin el conocimiento del usuario pero si con su consentimiento. Se denomina “el guía de música colaborativo PIRATE”, éste permite compartir algunas recomendaciones de música, debido al encuentro oportuno esto hace que la interacción sea en un periodo corto mientras dura el encuentro. A los usuarios se les proporciona conocimiento de alguna comunidad con un tipo de música favorita y se les permite conocer con quiénes pueden compartir ese tipo de música. A diferencia de GENIE los miembros de la comunidad del sistema PIRATE no necesitan establecer una interacción cara a cara, los intercambios suceden cuando dos o más se acercan físicamente hasta el alcance de las computadoras de bolsillo.

También en [1.5] se propone otra comunidad denominada WALID (*Wearabled Augmented Task-List Interchange Device*) en este tipo de comunidad existe un poco la idea de que tú puedes ayudar a alguien porque algún día necesitarías un favor de esa persona. En esta comunidad, dos usuarios se encuentran de manera espontánea entonces intercambian tareas del mundo real a través de sus computadoras de bolsillo. WALID usa un software que tiene la funcionalidad de un agente que se encarga de localizar los miembros de la comunidad más cercana y negociar el intercambio de favores, el agente mantiene una lista de tareas y conocimiento de la ubicación. Cuando dos o más personas se encuentran, los agentes

producen una negociación, el agente evalúa el tamaño del favor sacando comparaciones, para ver si le conviene hacer los favores, para esto utiliza lo que se conoce como teoría de juegos para asegurar que los resultados de la negociación sean convenientes para las dos partes.

Aunado a esto, la visión de estos sistemas es hacer comunidades con tecnología de bolsillo tales como:

- Comunidades computacionales: los miembros comparten recursos computacionales como puede ser el ancho de banda, velocidad de procesador, etc.
- Comunidades de ayuda: los miembros se comprometen a asistir a algún otro que solicite ayuda.
- Comunidades negociantes: En una comunidad los miembros buscan colectivamente vender sus productos.
- Comunidades mercantiles: Los miembros comparten mercancías sin la necesidad de dinero.
- Comunidades de conocimiento: Los miembros colectivamente acumulan información y tratan de interpretarla en forma conjunta.
- Comunidades políticas: Los miembros crean nuevas formas de democracia instantánea y activismo local.

Parecido a las comunidades en línea como el Internet, las comunidades con tecnología de bolsillo consisten de *WEB's* interrelacionadas con un enfoque a las interacciones sociales. Las diferencias, sin embargo, son de tres formas.

- **Contexto social:** Las comunidades con tecnología de bolsillo se forman a partir de encuentros directos más que encuentros indirectos, los intercambios deben depender de una corta distancia limitada por las redes *WPAN*. La forma de interacción entre los participantes es más directa por que cada uno puede ver a la otra persona sin perder detalles, como por ejemplo, la ropa que está usando, el estado de ánimo, los gestos durante la comunicación, etc.
- **Uso de contexto:** Las personas utilizan diferentes computadoras de bolsillo con distinto contexto, es decir, de su trabajo, de la oficina, de su casa, por lo tanto requieren de una escasa atención hacia el dispositivo móvil, por que los usuarios podrían estar realizando alguna otra tarea, este es uno de los objetivos principales en cómputo ubicuo [1.2].
- **Contexto técnico:** Como las computadoras de bolsillo, dentro de una red *WPAN*, con frecuencia tienen disponible poco ancho de banda y están limitados en cuanto a capacidad de procesamiento, entre otras desventajas, las aplicaciones tienen que

tratar con estas situaciones de hardware, por lo tanto se deben que adaptar al dispositivo [1.3].

En una comunidad con tecnología de bolsillo cada uno de los participantes, en el momento de ingresar, tiene las mismas oportunidades, dependiendo de cómo los participantes actúen y cuándo resuelvan sus conflictos de intereses es cuando una comunidad se dice estable. En una comunidad con dispositivos de bolsillo se identifican tres factores importantes que determinan el comportamiento de los participantes:

- **Identidad:** En este tipo de comunidades donde las personas interactúan cara a cara, cada uno de los participantes tiene información suficiente para determinar el estado de la persona con quien está interactuando, es decir, puede conocer el estado de ánimo de la persona, los ademanes que hace en la conversación, la ropa que está usando, los gestos que efectúa, etc, mientras que en una comunidad en línea, esto no es tan fácil de identificar las personas pueden aparentar un comportamiento que no tienen.
- **Privacidad:** En este tipo de sistemas los participantes tienen el control de lo que se mostrará a los demás, y también se pueden seleccionar los participantes, es decir, se le otorgará permisos diferentes a cada usuario.
- **Veracidad:** Dado que en una comunidad con tecnología de bolsillo podemos conocer mucho de la otra persona, como sus gestos y ademanes, eso nos puede proporcionar métricas para ver si la persona está diciendo o no la verdad, y ver si se le puede tener suficiente confianza.

Para una comunidad de colaboración es de gran utilidad el contar con dispositivos móviles de bolsillo, redes *ad hoc* inalámbricas, mecanismos para encuentros espontáneos, y lo que se le denomina tecnología de contexto, como pueden ser los dispositivos que permiten conocer el contexto de algún elemento de la red. Por tal motivo, se necesitan nuevas aplicaciones de software que interactúen con estos dispositivos, se necesita una infraestructura de software distribuido para que las personas puedan interactuar con los dispositivos, la idea es formar diferentes comunidades generales a partir de algunas particulares como lo muestra la figura 1.6. Esto da como resultado una comunidad distribuida de gran escala, formando comunidades en cualquier lugar y en cualquier momento. Por lo tanto, los sistemas tienen que hacer algunas tareas de administración de comunidades, como por ejemplo, descubrir nuevas comunidades, identificar sus dispositivos así como verificar las identidades de los integrantes de la comunidad.

A continuación se describe un sistema de mensajería instantánea con capacidades para conocer el contexto de los participantes.

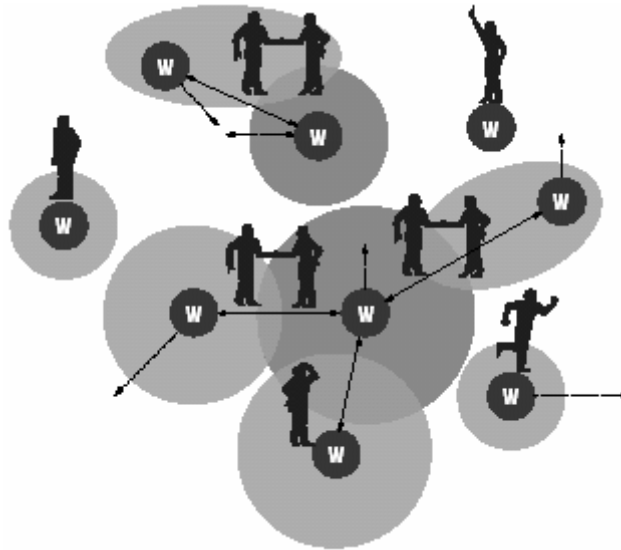


Figura 1.6. Formando comunidades generales a partir de particulares

1.1.4 ConChat: A Context-Aware Chat Program

En cómputo ubicuo una característica que siempre se ha buscado, es conocer el contexto de cada uno de los elementos que participan en un entorno ubicuo, desde las características de los dispositivos que interactúan, las condiciones de la red, así como el comportamiento de las personas participantes, ya que es de gran importancia para este tipo de entornos [1.4].

Mientras tanto, en un programa de *chat* común donde las personas interactúan a partir de una aplicación que les permite intercambiar mensajes entre los participantes, solo interrogando se podría conocer cómo es su lugar físico donde se encuentran, cómo es su estado de ánimo, y qué es lo que está haciendo, además de estar entablando una conversación. Por lo tanto se tendría que confiar en las respuestas de los participantes porque es lo que se tiene más al alcance.

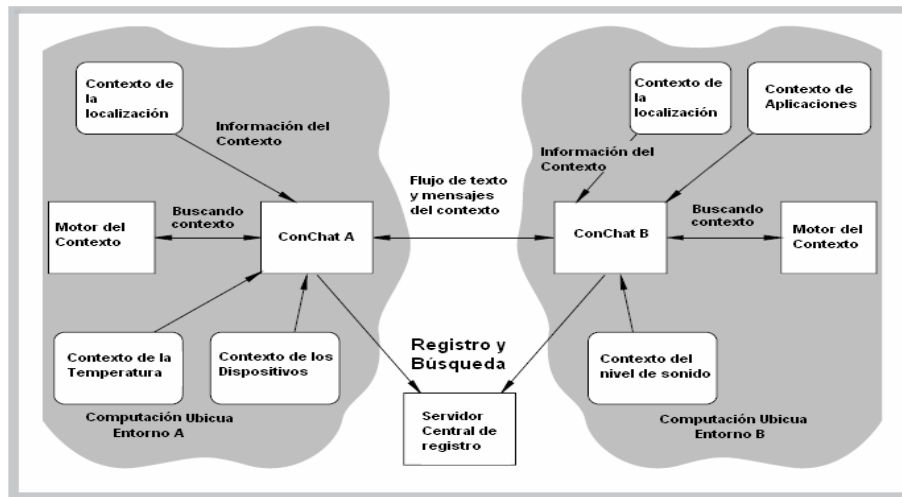


Figura 1.7 Sistema ConChat

Pero qué pasaría si se establecieran programas de *chat* en un entorno ubicuo donde se pudiera conocer el contexto de cada uno de los participantes, sin tener que hacer un interrogatorio. En [1.7] se presenta un sistema con esas características llamado ConChat. En ConChat introducen lo que se llama información contextual en la comunicación electrónica, con esto se podría conocer la información contextual de la otra persona, es decir, lo que está pasando en su entorno, como se muestra en la figura 1.7.

Para esto cada uno de las partes requiere un modelo común de información contextual. El sistema está basado en cálculos de predicados de primer orden y álgebra *booleana*, las operaciones básicas involucradas son la conjunción, la disyunción y cuantificadores sobre contexto. Con ello se construyen expresiones que contienen el contexto.

La estructura básica del sistema se basa en una estructura de predicados en el idioma inglés, una sentencia es de la forma <subject><verb><object> y el sistema define la estructura como:

Context (<ContextType> , <Subject> , <Related> , <Object>)

ContextType, se refiere al tipo de contexto que el predicado describe, *Subject* se refiere a la persona, lugar ó cosa de la que se refiere el contexto, *Object* es un valor asociado al *subject* y *Related* es alguna acción o adjetivo relacionado al *subject* y pueden existir operaciones relacionadas al *object* usando operadores booleanos (or \vee , and \wedge , negación \neg).

1.4.1 Ejemplos:

Context (<Localización><Emilio><Entrando><Cubículo 67>)

Context (<Temperatura><Laboratorio de Distribuido><frío><10gC>)

Este sistema está pensado en información de contexto simple, pero es independiente del lenguaje de programación, sistema operativo o del *middleware*.

1.1.4.2 Operaciones

Con este tipo de modelo es posible hacer operaciones *booleanas* las cuales expresan situaciones de la vida diaria, algunas operaciones *booleanas* podrían ser:

Context (<Localización> <Emilio> <Entrando> <Cubículo 67>) \wedge Context (<Actividad> <Reunión> <en> <Cubículo 67>)

Context (<Localización> <Emilio> <Entrando> <Cubículo 67>) \vee Context (<Localización> <Emilio> <Entrando> <Cafetería>)

\neg Context (<Localización> <Emilio> <Entrando> <Cubículo 80>)

Algunas veces se necesita información del lugar físico en que nos encontramos, como por ejemplo, si en el lugar donde estamos existe alguna otra persona más, en esta solicitud de

información de contexto podemos utilizar los denominados cuantificadores e existencialidad:

$\exists X$ = Existe al menos un valor de X

$\forall X$ = Para todos los valores de X

1.1.4.3 Ejemplos:

$\exists_s X$ *Context*(t,x,r,o) es verdad si *Context*(t,x,r,o) es verdad para alguna x perteneciente al conjunto S

Para expresar “Emilio está en alguna localización”, sería:

$\exists_{\text{localización}} Y$ *Context*(Localización,Emilio,en,Y)

Para expresar que todas las personas están en el laboratorio de distribuido

$\forall_{\text{personas}} Y$ (Localización,Y,en,Laboratorio de Distribuido)

ConChat trabaja como un *chat* normal donde se comparte conocimiento acerca del contexto, en éste se asume que se encuentra en un entorno de computación ubicua. Entre un elemento A y un elemento B, ellos se encuentran separados a una distancia, entonces los dos ejecutan la aplicación ConChat para establecer una conversación. El sistema ConChat usa un servidor central que permite a las personas conocerse entre ellas, en éste se registran y también pueden encontrar a las personas que estén registradas. En el servidor donde se registran hay direcciones IP que pertenecen a las máquinas donde se localizan las aplicaciones, y la forma de comunicarse es a partir de un *socket*² que es por donde se intercambian los mensajes.

Como ya se mencionó antes el objetivo de ConChat es compartir información contextual entre los integrantes por lo tanto en éste sistema se puede compartir:

- Número de personas
- Localización de los participantes
- La identidad de las otras personas en el cuarto
- Temperatura del cuarto, iluminación y información del sonido en el cuarto.
- La existencia de otras aplicaciones y dispositivos en el cuarto
- Actividad del cuarto (de junta, de lectura, de presentaciones)

En el momento que se ejecuta ConChat lo primero que se hace es llamar al proveedor de contexto para actualizar la información de la persona. Este sistema tiene varias reglas para hacer cálculos de predicados de primer orden. A continuación se presentan algunos ejemplos de los cálculos que puede hacer.

² Designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada.

$Context (Num_personas, Lab_distribuido, \geq, 3) \wedge Context (Aplicaciones, PowerPoint, en, ejecución) \Rightarrow Context (Actividad, Lab_distribuido, en, Presentación)$

Si en el laboratorio distribuido se encuentran más de 3 personas y existen aplicaciones *power point* en ejecución esto implica que hay una presentación en el laboratorio distribuido.

$Context (Num_personas, Lab_distribuido, \geq, 3) \wedge \neg Context (Aplicaciones, PowerPoint, en, ejecución) \Rightarrow Context (Actividad, Lab_distribuido, en, junta)$

Si en el laboratorio distribuido se encuentran más de 3 personas y no existen aplicaciones *power point* en ejecución esto implica que hay una junta en el laboratorio distribuido.

ConChat mantiene una vista de su espacio, que representa la información contextual en forma de conjuntos de expresiones y esta información la hace disponible hacia los otros participantes.

Por ejemplo, cuando el usuario Emilio desea conocer el contexto de alguna otra persona digamos su localización, entonces Emilio envía la consulta, podría ser de la siguiente forma $Context(Localización, PersonaX, en, Lab_Distribuido)$. Y como ConChat tiene la información de quiénes están en el laboratorio de Distribuido, se evalúa la consulta basándose en esta información, es decir, se evalúa la información intentando unificar la consulta con el conjunto de sentencias verdaderas que tiene, y una vez evaluada envía el resultado de la evaluación a Emilio. También Emilio podría estar interesado en que le sea notificado cuando alguna persona entra al Laboratorio de Distribuido, entonces para esto se asume que alguna persona está en el Laboratorio de Distribuido y es a ella a la que se le va hacer la consulta y se envía la siguiente expresión: $\exists_{Personas} Y Contexto (Localización, Y, Entrando, Lab-Distribuido)$ y se evalúa de nuevo la consulta y se regresa el resultado a Emilio.

1.1.4.4 Privacidad

En cuanto a la privacidad, ésta está controlada porque ConChat permite a los usuarios especificar su contexto por medio de una tabla, de forma que cada usuario tenga el control de acceso a los demás, es decir, a los demás usuarios se le asignan el tipo de información al que tienen acceso. Esta tabla incluso puede modificarse de manera dinámica. Aún así sigue siendo un sistema para estaciones de trabajo estáticas donde la localización está bien definida.

1.1.5 Administrando Datos en servicios de información dependiente de la localización

Un problema al trabajar con sistemas ubicuos en el que actualmente se está investigando es manejar la movilidad. Es decir, cómo acceder a la información cuando se trabaja con dispositivos móviles; el usuario desearía poder acceder a la información aún cuando se encontrara en constante movimiento.

En [1.8] se trabaja en este tipo de problema al tratar de acceder a la información dependiendo de la localización del dispositivo en un entorno de computación ubicua, en especial de la tecnología celular. Para esto los [LDIS] se encuentra actualmente con tres desafíos como son: **Restricciones del entorno móvil, espacio de datos y Movilidad del usuario.**

- *Restricciones del dispositivo móvil.* Dado que en la computación móvil se tiene algunos problemas por las capacidades de los dispositivos como son: un ancho de banda bajo, problemas de estabilidad con la red porque no hay una conexión segura como en una red alambrada y los recursos locales son también limitados [1.9, 1.3]. Eso podría causar que no se pudiera acceder de manera apropiada a la información, o que existieran problemas de pérdida de la información debido a la inestabilidad de los dispositivos móviles.
- *Espacio de Datos.* Dependiendo del espacio físico donde realice una solicitud de información es como se ha de manejar la consulta. Es importante porque esto puede verse reflejado en la respuesta a la petición del usuario.
- *Movilidad del Usuario.* Como en computación móvil, al usuario se le provee con dispositivos que le permiten estar accediendo a información desde cualquier lugar que se encuentre, y como estos cambian constantemente de posición, se debe pensar en buscar la forma de poder contestar satisfactoriamente a la demanda.

En [1.9] establecen que la localización en un LDSI se debe especificar implícita o explícitamente antes que un cliente pueda acceder a la información, y establecer un modelo de localización depende de la técnica de identificación de localización del propio sistema. Entonces proponen dos modelos: El modelo Geométrico y el Modelo Simbólico.

Modelo Geométrico. El sistema especifica una localización en una coordenada n -dimensional ($n = 2$ ó 3), por ejemplo un conjunto de coordenadas definiendo un área limitada. Entonces el sistema puede procesar formas geométricas para una representación concisa (por ejemplo, un área en forma de círculo puede ser representada por el centro y un radio). Este sistema tiene la característica de ser compatible con cualquier plataforma, por lo tanto, el grado de heterogeneidad es alto. Sin embargo, en términos de volumen de datos envuelve la necesidad de mapear la representación geométrica a un nivel semántico adecuado a las aplicaciones.

Modelo simbólico. En este modelo las entidades del mundo real definen espacios de localización, éstas pueden ser por ejemplo calles, ciudades, o incluso células inalámbricas y son identificadas únicamente por un sistema de asignación de nombres en forma jerárquica. Tiene más ventajas que el modelo geométrico porque éste no se basa en coordenadas, sino que con sus entidades puede ser de más ayuda y a su vez con esto, un nivel semántico estable para aplicaciones LDIS. Por lo tanto la localización simbólica es más fácil de manejar. Aunque se dice que el modelo apropiado va a depender de la aplicación.

1.1.5.1 Tipos de consultas

En un sistema así se pueden encontrar con diferentes tipos de consultas de las cuales se tiene que proceder de manera adecuada a la consulta.

1.1.5.2 Consulta local contra no local:

Las consultas locales. Son limitadas a la localización presente del usuario.

La consulta no local. Es limitada a la localización de otros usuarios más que del usuario presente. Porque son los que le resolverán la consulta.

En una consulta local el sistema debe resolver la consulta basándose en el contexto del usuario local, y para un usuario no local debe primero identificar el contexto de quien evaluará la consulta.

1.1.5.3 Consulta simple. Las consultas simples son aquellas cuando las solicitudes son de acuerdo a las condiciones de cómo la información se encuentra disponible sin tener que hacer más trabajo. Por ejemplo, cuando se solicita el reporte de asistencias del personal del CIC-IPN esa información ya se encuentra organizada de la forma que se está solicitando, el sistema no tiene que hacer ningún trabajo.

1.1.5.4 Consulta general. Esta envuelve condiciones complejas acerca de cómo los datos están organizados. Pueden participar muchos factores que se tendrían que considerar para poder cumplir con la petición, por ejemplo las consultas pueden ser divididas en restricciones espaciales, digamos una consulta que exprese: "En qué salón está un grupo de Maestría del CIC y que tenga más de 10 alumnos", por lo menos un predicado tiene restricciones espaciales. Entonces, para esto se tendrían que buscar la forma de poder hacer operaciones espaciales comunes entre los predicados, así, se requieren nuevas técnicas de procesamiento de consultas [1.7].

Para satisfacer una consulta se debe pensar en el aspecto del ámbito de la consulta; en un LDIS el resultado de la consulta depende de la localización donde se hizo la consulta, entonces no hay que olvidar que la respuesta debe de estar de acuerdo al ámbito del solicitante, por ejemplo en un sistema de ayuda geográfica, en donde la consulta sea: ¿Dónde encuentro un hotel? Entonces, la respuesta tiene que ser de acuerdo al ámbito donde se realizó la consulta (donde se encuentra la persona), el sistema no puede proporcionar información que sea inconveniente al solicitante.

En [1.8] la forma de la consulta tiene la siguiente estructura (consulta, resultado)

En una consulta simple donde por ejemplo se pide un grupo de Sistemas Distribuidos de la Maestría del CIC (Grupo_SistemasDistribuidos,A)

En una consulta general donde, por ejemplo, el salón de un grupo de la Maestría del CIC con más de 10 alumnos (Salon-grupo-mayora-10alumnos,B)

Donde A y B respectivamente contendrán el resultado de la consulta.

En el modelo de localización geométrico se toma la forma de un polígono en un espacio de 2 dimensiones. En el modelo de localización simbólica representa un ámbito válido como un conjunto de zonas lógicas identificadas.

Las propiedades espaciales de los datos dependientes de la localización y la movilidad del usuario son los problemas más importantes que los que se tiene que trabajar en un ambiente ubicuo.

Por los ejemplos anteriores vemos que el trabajo en Computación Ubicua es todavía grande, entonces, con computación móvil se tiene el problema que las aplicaciones se tiene que adaptar al dispositivo porque de eso va a depender cómo ellos tendrán que procesar la información, aunque eso no nos limita para poder formar grupos de interacción con este tipo de tecnología y así realizar trabajo colaborativo móvil a partir de interacción espontánea. De lo que sí ya se puede hablar es de aplicaciones ubicuas que trabajan con la percepción del contexto de los elementos del sistema lo cual es un importante y gran avance en esta área de investigación. Ahora hay que saber interpretar y manipular la información, porque es un gran problema el tratar con información donde tanto los usuarios y herramientas (elementos de un sistema ubicuo) se encuentran en constante movimiento en el sistema.

Entonces se tiene que seguir trabajado para tratar de resolver los problemas de la computación ubicua y para hacer la tecnología más estable en la que todos podamos colaborar.

1.2 Resumen

Del presente capítulo podemos resumir lo siguiente. Se abordó una introducción de lo que es computación ubicua, haciendo algunas comparaciones con agentes inteligentes y realidad virtual para su mejor comprensión. Un punto principal que mencionamos es que computación ubicua se enfoca a 2 metas cruciales en una red de computadoras las cuales son localización y escala. También describimos algunos sistemas móviles colaborativos donde el sistema se adapta a las características físicas del dispositivo móvil y cómo se puede lograr una colaboración oportunista a lo mismo que en nuestro sistema. Abordamos sistemas como ConChat donde el principal objetivo fue hacer un sistema ubicuo que proporcionara información contextual de cada uno de los elementos involucrados en el sistema el cual es uno de los objetivos principales en computación ubicua.

Como mencionamos en la introducción, nuestro objetivo es hacer un sistema colaborativo ubicuo por lo tanto, de los sistemas mencionados en éste capítulo se consideraron las características más importantes que pudieran ser de utilidad para el desarrollo del nuestro.

En el siguiente capítulo abordaremos un concepto importante para nuestro sistema, como lo es colaboración espontánea o también llamada colaboración oportunista donde se puede alcanzar una participación en grupo sin un acuerdo previo. Así también, abordaremos las características más importantes de mensajería instantánea y el por qué de su utilización en nuestro sistema.

1.3 REFERENCIAS

- [1.1] Mark Weiser. "*The world is not a desktop*". Interactions; January 1994; pp. 7-8.
- [1.2] M. Weiser. *The computer for the 21st century*. Scientific American, 19-25, Sep 1991.
- [1.3] W.Y. Lum and F.C.M. Lau, "*A Context-Aware Decision Engine for Content Adaptation*", *IEEE Pervasive Computing, Vol. 1, No. 3*, July-September 2002.
- [1.4] Ranganathan, R. Campbell, A. Ravi and A. Mahajan, ConChat: A Context-Aware Chat Program. *IEEE Pervasive Computing*, pp. 52-58, July-Sept 2002, Context-Aware Computing *Gregory D. Abowd, Maria Ebling, Hans-Werner Gellersen, Guerney Hunt, Hui Lei*
- [1.5] Gerd Kortuem, Zary Segall. Wearable Communities: Augmenting Social Networks with Wearable Computers. *IEEE Pervasive Computing Magazine*, Vol. 2, No 1, January-March 2003, p71-78.
- [1.6] E.T. Hall. *The Silent Language*, Anchor, Garden City, N.Y., 1973.
- [1.7] Ranganathan et al. ConChat: A Context-AwareChat Program. *IEEE Pervasive Computing*, July-September 2002, 51-57.
- [1.8] D. L. Lee, W.-C. Lee, J. Xu, and B. Zheng. "Data Management in Location-Dependent Information Services." *IEEE Pervasive Computing*, 1(3): 65-72, July-September 2002.
- [1.9] P. Bellavista, A. Corradi, C. Stefanelli, "*The Ubiquitous Provisioning of Internet Services to Portable Devices*", *IEEE Pervasive Computing, IEEE Computer Society Press*, Vol. 1, No. 3, pp. 81-87, July-Sep. 2002.

- Colaboración Oportunista
- Mensajería Instantánea (IM) para colaboración
- Casos de Uso



En la siguiente sección se aborda el concepto de colaboración oportunista; la importancia de este tema radica en el hecho de que es el objetivo principal de nuestra herramienta. Es decir, damos la posibilidad de crear nuevos espacios de colaboración espontánea en el trabajo de edición colaborativa entre las personas relacionadas a un documento. Para esto, presentamos a la mensajería instantánea como una forma de fomentar la colaboración oportunista al implementar un espacio potencial de colaboración. De igual forma presentamos los casos de uso de nuestro sistema de colaboración.

2.1 COLABORACION OPORTUNISTA

En el área de trabajo colaborativo asistido por computadora (CSCW) se han buscado nuevas herramientas que promuevan y faciliten la interacción entre un grupo de trabajo, así como nuevas formas de trabajo donde se explote lo más posible la labor en equipo. De ahí que se ha encontrado una nueva manera de trabajo colaborativo que algunos autores llaman colaboración por encuentros casuales, colaboración oportunista [2.1], colaboración informal [2.2], interacción espontánea [2.3], donde se puede interactuar sin un previo acuerdo.

Cuando un grupo de trabajo con intereses en común participan en un mismo proyecto, pero debido a la diferencia de horarios o de las distintas tareas que desempeñan en el equipo, no pudieran estar sincronizados en el tiempo o el lugar, entonces podrían surgir problemas de participación y comunicación. Pero con la ayuda de las redes WPAN (*Wireless Personal Area Network*) donde se forman grupos de trabajo delimitados por el alcance de las WPAN y sistemas, por ejemplo, de mensajería instantánea los cuales proporcionan la percepción del estatus de los participantes, se pueden formar grupos espontáneos de colaboración porque en cualquier momento y en algún determinado lugar se puede encontrar a algún participante conectado a la red y así poder interactuar, a esto es a lo que se le denomina trabajo colaborativo oportunista o colaboración espontánea, porque se pueden formar grupos de trabajo a partir de algún encuentro casual, es decir, sin un previo aviso.

En [2.3] definen a la interacción espontánea como la acción que toma lugar donde el humano y/o participantes de algún proyecto con ayuda de herramientas computacionales participan por una coincidencia temporal de algún lugar e interoperan para lograr un producto. En [2.3] plantean que la colaboración existe cuando al menos dos personas se reúnen para negociar ó colaboran para desempeñar una tarea.

Para entender los problemas que se pueden encontrar al trabajar con aplicaciones para usuarios móviles colaborando en forma oportunista, describiremos algunas características de los sistemas AIDA [2.1], Hocman [2.5,2.6] y SoundPryer [2.7], donde el común denominador de los tres prototipos es el concepto de actividad móvil dependiente del lugar físico de los participantes.

AIDA es un sistema que soporta trabajo con colaboración oportunista para dispositivos móviles, como ya se ha mencionado en el capítulo 0 en los antecedentes, es un sistema que utiliza mensajería instantánea para conocer la presencia de los participantes y mostrarla en una interfaz para PDA's (Asistente Personal Digital). AIDA también comparte recursos y muestra el estatus del recurso en la interfaz de mensajería instantánea. El recurso en AIDA es un documento de edición que está a disposición de un grupo de revisores y autores. La información en el dispositivo móvil se sincroniza con el servidor cada vez que el recurso y el cliente se conectan a un punto de acceso.

AIDA ofrece oportunidades de encuentro causal en una comunidad de autores y revisores. Por ejemplo, cuando un autor nota que su documento ha sido abierto por alguno de los

revisores gracias a la interfaz de Mensajería Instantánea, él podría encontrar esto como una oportunidad para enviar un mensaje a la persona que tiene activo el documento, o de igual forma él puede entrar a editar colaborativamente el documento, resolver algunas dudas del revisor ó más aún podría iniciar una sesión de revisión colaborativa sincronizada con los demás revisores que se encontraran en línea.

El sistema Hocman es una aplicación *peer to peer HTTP* donde se permite la compartición de documentos de texto, archivos de audio e imágenes sobre una red *ad hoc* por medio de un dispositivo móvil.

Este proyecto es de importancia porque describe cómo los elementos pueden estar interactuando al estar en movimiento con los dispositivos móviles, describe además las herramientas que ellos utilizan para coordinar sus actividades desde una perspectiva de (CSCW). Tiene dos modos de trabajo:

- Un modo donde descubre nuevos participantes (*peers*) a través de un navegador (*browser*) es decir, detecta qué nuevos *peers* se unieron a la red *ad hoc*.
- Un modo de exploración por vecindad, cada uno de los usuarios proporciona la lista de los vecinos que tiene a su alrededor.

Por otra parte, el sistema SoundPryer es un sistema de compartición de música a través de un sistema *peer to peer*. No es muy difícil imaginar un sistema donde se comparte música en forma de archivos, es decir, que se comparten los archivos de una computadora sobre la red por que esto ya existe y es muy común, el problema es cuando se está interesado en compartir y escuchar música con dispositivos móviles donde el usuario se encuentra en constante movimiento.

SoundPryer se forma con la compartición de archivos MP3 entre diferentes nodos de una *had hoc*. Dado que se está hablando de dispositivos móviles, la aplicación se encuentra con algunos problemas tales como: Un efecto de congestión de tráfico que se puede formar hacia el dispositivo, la distribución de la carga del trabajo debe ser considerada para que se adapte a cambios, entre otros. Cuando se comparten archivos es importante conocer el origen de los archivos, por lo tanto, se despliega una interfaz que tiene las propiedades del usuario que comparte, para encontrar la música se le proporciona una base de datos que contiene una bitácora de: el título, el artista, el tiempo y el álbum.

Cada uno de los dos anteriores sistemas se encuentran con problemas en la movilidad en la red *ad hoc* para soportar la colaboración. A continuación se enumeran una serie de problemas estudiados:

- Actividad simultánea
- movilidad
- Compartición

2.1.1 Actividades simultáneas

La disponibilidad eleva la complejidad debido a que el usuario se encuentra ocupado en varias actividades al mismo tiempo. Por lo tanto, la idea primordial es tratar de hacer que esos sistemas no distraigan al usuario de las actividades principales.

2.1.2 Movilidad

Cuando el usuario se mueve de un lugar a otro eso puede causar varios problemas de cambio continuo de posición y son escasas las situaciones en que se encuentra en un lugar estático, por lo tanto, en estos sistemas se trata de tomar en consideración los casos extremos donde existan condiciones de extrema movilidad.

2.1.3 ¿Compartición de recursos?

En estos sistemas que se han diseñado para asistir en la colaboración en tráfico, la sincronización de la información compartida es importante.

Por todo esto la colaboración espontánea se puede ver como una oportunidad de poder trabajar con algún compañero de equipo por algún encuentro no planeado. Como se mencionó en los ejemplos anteriores, con ayuda de la tecnología de computadoras móviles que son de tamaños pequeños y cómodos que podemos llevar a cualquier lado, es posible crear colaboración espontánea a partir de encuentros casuales, Aunque no es una necesidad porque también podemos crear colaboración asíncrona por medio de mensajería instantánea donde en todo momento se puede conocer el estatus de los participantes.

2.2 Mensajería Instantánea (MI) para Colaboración

Cuando se habla de colaboración en un grupo de trabajo, donde cada participante puede hacer alguna aportación hacia el equipo, se hace importante el poder contar con los espacios de trabajo informales de comunicación para realizar de forma más efectiva la colaboración. Aunado a esto, en los últimos años los sistemas de mensajería instantánea (MI) han sido ampliamente adoptados entre las personas debido a la facilidad que otorgan para estar en comunicación con una o varias personas al mismo tiempo, independientemente del lugar físico que ocupen.

Los sistemas de Mensajería Instantánea están basados en una comunicación sincronizada a través del Internet [2.8] [2.9], estos pueden asistir a la comunicación uno a uno ó comunicación uno a varios donde a una persona se le puede invitar a integrarse a una comunicación. Para la comunicación utilizan una interfaz en forma de ventana y un cuadro de diálogo por donde los participantes teclean los mensajes que son enviados hacia la otra persona, se parece a un sistema de comunicación por teléfono que se basa en un modelo de llamadas en forma dinámica [2.8], los mensajeros instantáneos soportan lo que se le llama un grupo *Chat*, donde cada participante puede entrar a un salón de *Chat* donde se pueden encontrar a varias personas con quienes se puede establecer una plática.

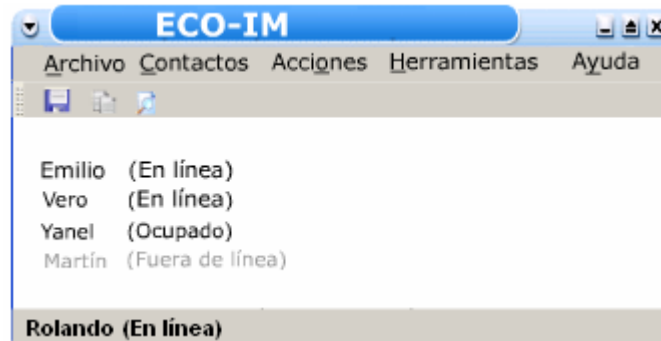


Figura 2.1 Interfaz de un mensajero instantáneo

Las personas disponibles con las que se puede interactuar se representan en forma de lista, la cual puede ser organizada dependiendo del ámbito y la relación que existe con las personas, puede ser organizada, por ejemplo: Por amigos, compañeros de trabajo ó familiares con las que en dado momento se puede establecer una interacción, esto es posible porque la lista despliega el estado de las personas que la integran, en ella podemos ver quiénes están disponibles y quiénes están indispuestos, como se muestra en la figura 2.1. Por ejemplo, AOL [2.10] da la posibilidad de que el usuario asigne su estado el cual verán sus contactos, esto está relacionado con resolver problemas de privacidad, en [2.11] se puede poner un estado de: Ocupado, al teléfono, regreso enseguida, en línea, en la hora de comida o simplemente estado inactivo. Con esto los demás comprenderán que en esos momentos no es posible interactuar. Los sistemas más populares son por ejemplo AOL, Yahoo messenger, MSN messenger, ICQ y Jabber [2.12]. Es preciso mencionar que estos sistemas además de servir como un medio para enviar mensajes instantáneos a las personas

disponibles, también es posible enviar mensajes en forma de correo electrónico a las personas que no se encuentran en ese momento en línea, con esto se busca que de cualquier forma se pueda estar en contacto en cualquier momento.

Algunos otros sistemas inclusive proporcionan algunas características adicionales, como por ejemplo en [2.13] el mensajero instantáneo proporciona información acerca del contexto de su espacio físico de la persona con la que se está interactuado, digamos, su localización, el número y la identidad de los participantes del mismo espacio físico, temperatura del lugar, iluminación e información del sonido en el cuarto y se está trabajando para compartir los gestos de las personas. Por esto, además de permitir establecer una interacción con la persona también proporciona suficiente información de su espacio físico.

También se ha mencionado que los espacios de colaboración potencial se caracterizan por que permiten a los usuarios descubrir, identificar o crear oportunidades para iniciar una interacción que permita realizar trabajo colaborativo entre un equipo de trabajo [2.14].

En nuestro sistema la mensajería instantánea es la manera en la que formaremos un espacio potencial de colaboración, porque la lista de contactos que se forme será la lista de posibles personas con las que se pueda colaborar y el estado de los contactos será la manera de saber el momento en que se puede colaborar. A partir del espacio potencial, las personas involucradas en el documento tendrán la oportunidad de formar el espacio real de edición, debido a que sabrán en todo momento qué personas están en línea dispuestas a editar el documento o qué personas ya se encuentran trabajando sobre el documento, porque el sistema tomará en cuenta los estados normales de los mensajeros instantáneos comerciales y también un estado que indique si la persona se encuentra editando o leyendo el documento.

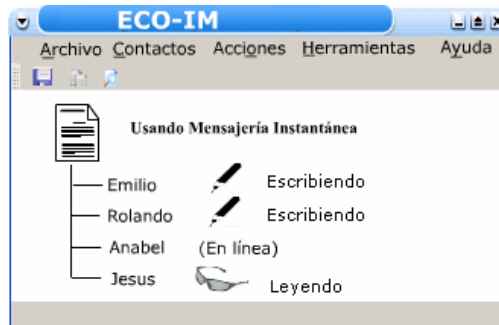


Figura 2.2 El espacio potencial de colaboración

Por ejemplo, la figura 2.2 muestra la interfaz del mensajero instantáneo. En la interfaz se exhibe el documento que está a disposición de los revisores, abajo las personas involucradas en el documento (revisores y autores), a su lado derecho el estado que tienen y el estado sobre el documento. Esta interfaz sería nuestro espacio potencial de colaboración porque aquí se identifican las oportunidades de colaboración o se establecen la interacción inicial que formará el trabajo de colaboración real [2.14].

Capítulo 2 Mensajería instantánea para colaboración

Como se puede ver, en el trabajo colaborativo de edición donde se tiene que estar en constante comunicación, una buena opción sería los sistemas de mensajería instantánea para formar los espacios de colaboración potencial que darán lugar a los espacios reales de trabajo.

2.3 Casos de Uso

2.3.1 Caso de uso de un ambiente ubicuo de colaboración.

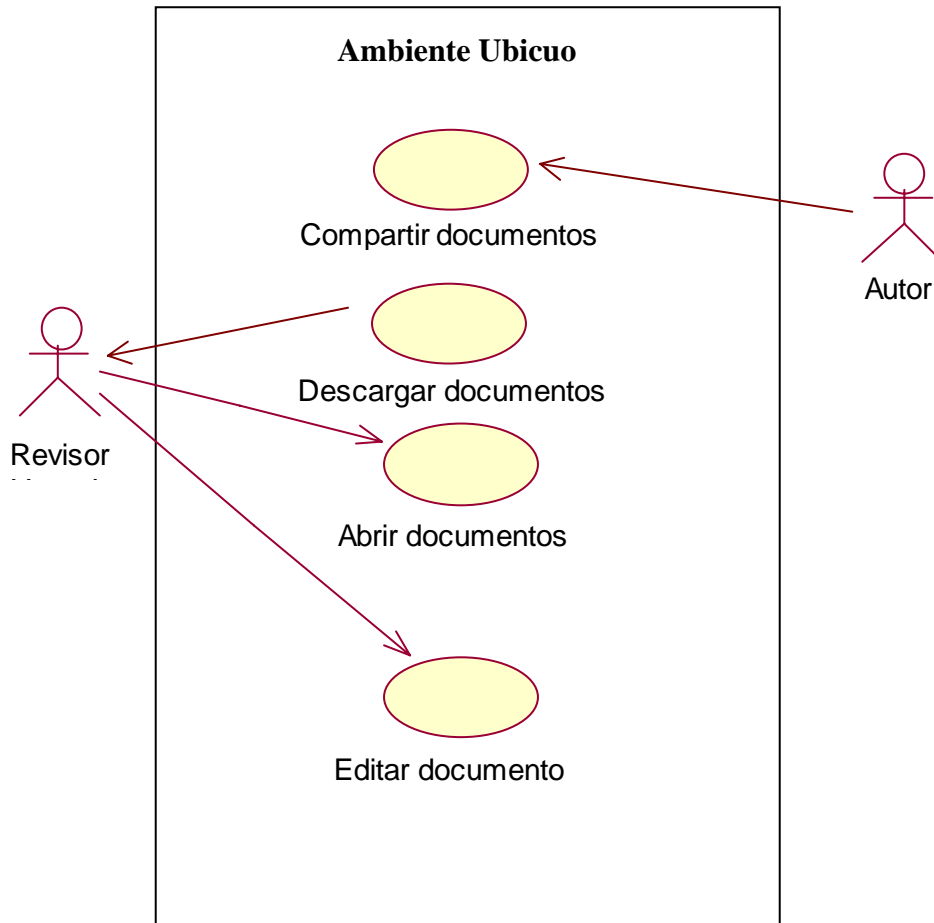


Figura 2.3 Caso de uso de la edición de un revisor en un ambiente ubicuo

En la figura 2.3 podemos encontrar el caso de uso de la edición de un documento por parte de un revisor donde un autor comparte un documento al revisor para que éste pueda editarlo y así participar en un ambiente ubicuo de edición, primeramente encontramos la operación de compartir, donde un autor comparte un documento para que pueda ser editado colaborativamente por quienes crea conveniente. A su vez, un revisor entra al ambiente ubicuo y se descarga el documento para que pueda ser editado, después de que el revisor tiene posesión del documento puede abrir el documento para iniciar su participación editando el documento.

2.3.2 Caso de uso de colaboración oportunistista entre Revisor-Autor.

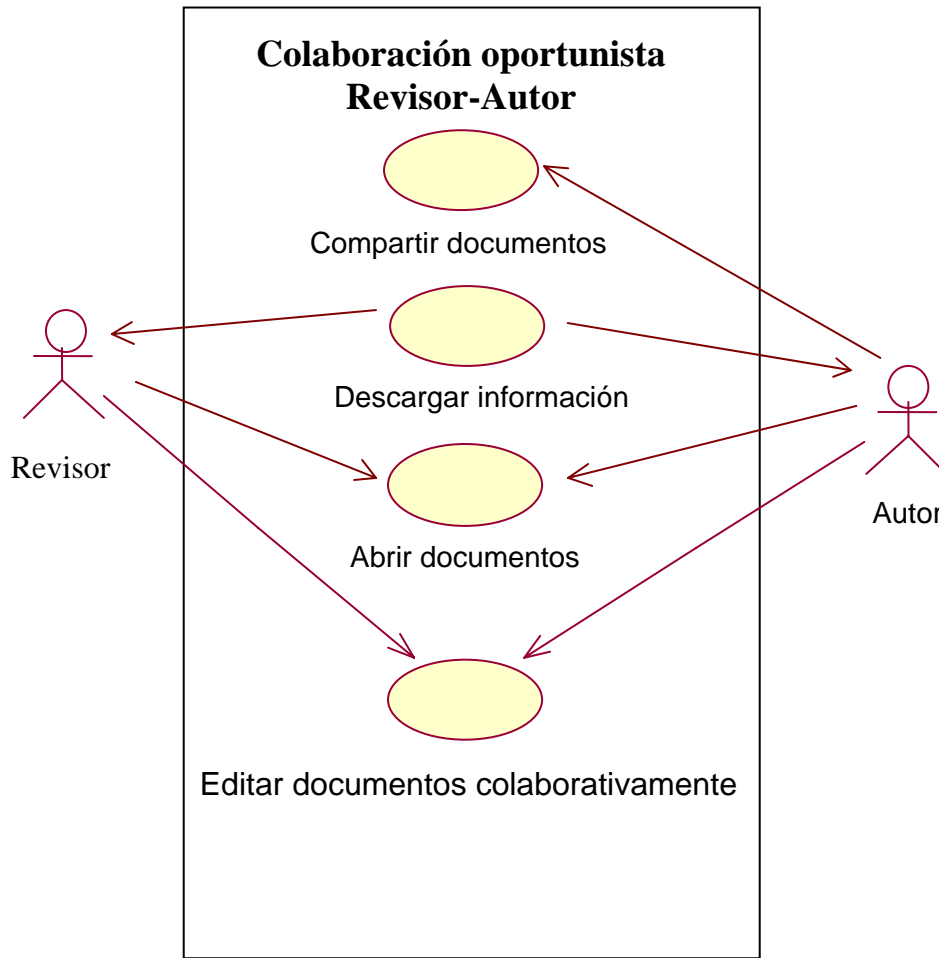


Figura 2.4 Caso de uso de colaboración entre un autor y un revisor

En la figura 2.4 podemos encontrar el caso de uso de una colaboración oportunistista entre un Autor y un revisor. Primeramente el autor del documento comparte un documento para que pueda ser editado colaborativamente por algún revisor. Después el documento es puesto a disposición al revisor y al propio autor, de este modo cualesquiera de los dos puede iniciar su participación sobre el documento, es decir, cualesquiera de los dos participantes puede abrir el documento y editarlo. Este proceso es colaborativo debido a que los dos participantes estarán siempre informados de cualquier acción sobre el documento. Es decir, el Autor conocerá cuándo el revisor abre el documento, lo edita y cuáles son los cambios que realiza, de igual forma el revisor.

2.3.3 Caso de uso de colaboración oportunistamente entre Revisores.

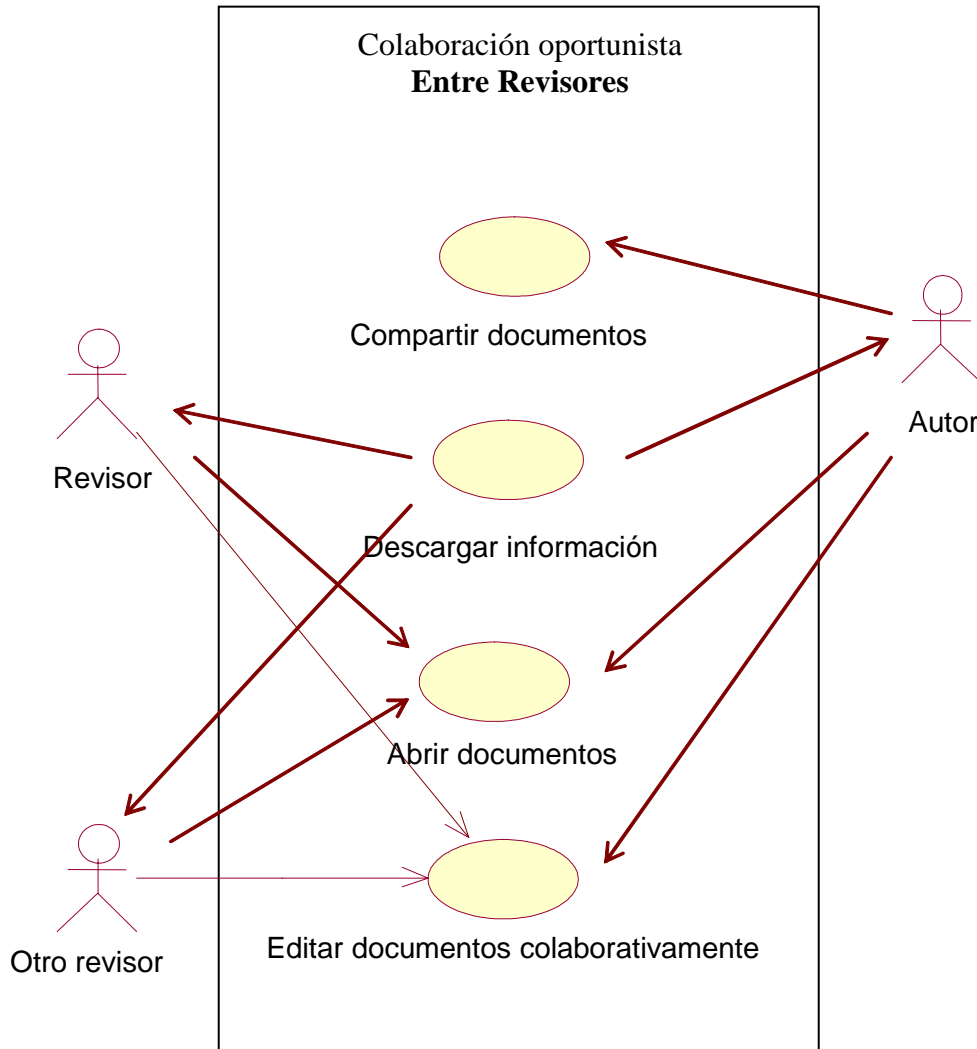


Figura 2.5 Caso de uso de colaboración oportunistamente entre el autor y más de un revisor

En la figura 2.5 podemos encontrar el caso de uso donde se presenta la participación de un autor y varios revisores dando lugar a la colaboración oportunistamente. En este caso el autor comparte un documento, después éste documento es puesto a disposición de los revisores para su edición, cualesquiera de los revisores puede abrir el documento para iniciar su participación, inclusive todos los revisores pueden abrir el documento pero sólo uno podrá editarlo a su vez. Inclusive, si llega un nuevo revisor el documento se pone a su disposición también, entonces su participación se vuelve oportunistamente por su colaboración con los demás revisores.

2.4 Resumen

De este capítulo podemos resumir lo siguiente. Se abordó una introducción de colaboración oportunista, es decir, un espacio donde la colaboración es informal, donde no existe un previo acuerdo para dar lugar a la colaboración, mencionamos algunos ejemplos de sistemas donde la idea es una participación espontánea, los cuales comparten diferente tipo de información con los participantes que se van uniendo al sistema. Se presentaron también, algunos de los problemas que pudiera encontrarse en este tipo de sistemas, como por ejemplo la movilidad y la distracción al usuario por la participación inesperada de varios usuarios. Otro concepto que manejamos en el presente trabajo es mensajería instantánea donde la idea principal es estar en comunicación constante con grupo de personas a través de un sistema, los participantes pueden conocer su estatus y mandar mensajes cortos entre ellos, mencionamos algunos de los sistemas de mensajería instantánea más populares que existen. Para nuestro sistema el objetivo de usar mensajería instantánea es para dar lugar a un espacio potencial de colaboración entre el documento y los participantes y así llegar a un espacio de colaboración real. Al final describimos diferentes casos que se pueden presentar en nuestro sistema.

En el siguiente capítulo abordaremos lo que son las redes *peer to peer* ya que son la base de las comunicaciones en nuestro sistema de colaboración. Hablaremos de los protocolos JXTA para formar redes *peer to peer* en computadoras de escritorio y JXME para llevar las redes *peer to peer* a dispositivos móviles.

2.5 REFERENCIAS

- [2.1] Favela, J., C. Navarro, and M. Rodríguez “Supporting Opportunistic Interactions with People and Services in Pervasive Computing Environments”, Supporting Spontaneous Interaction in Ubiquitous Computing Settings a workshop at Ubiquitous Computing International Conference (UbiComp 2002), October 1-3, Gotengorg, Suecia.
- [2.2] Max van Kleek. Supporting Informal Computer-Human Collaboration: The K22 Kiosk Platform. In Student Oxygen Workshop (SOW'02). Cambridge, MA, 2002.
- [2.3] Esbjörnsson, Mattias & Östergren, Mattias *Issues of Spontaneous Collaboration and Mobility*. Presented at the workshop Supporting Spontaneous Interaction in Ubiquitous Computing Settings, at UBICOMP'02.
- [2.4] Gerd Kortuem, Zary Segall. Wearable Communities: *Augmenting Social Networks with Wearable Computers*. *IEEE Pervasive Computing Magazine*, Vol. 2, No 1, January-March 2003, p71-78.
- [2.5] Esbjörnsson, M., Juhlin, O. and Östergren, M. (2002). Making Motor Bikers Come Together - Fast Moving Users and Mobile Ad Hoc Networks. In Proceedings of IRIS'25.
- [2.6] Esbjörnsson, M., Juhlin, O. and Östergren, M. (2002). How to Evaluate Prototypes Supporting Opportunistic Meetings. Presented at the Workshop on Mobile Ad Hoc Collaboration, CHI'02.
- [2.7] Axelsson, F. & Östergren, M. (2002). SoundPryer: Joint Music Listening on the Road. In Extended Abstracts of UBICOMP'02.1
- [2.8] Grinter, R. & Palen, L. (2002). *Instant Messaging in Teen Life*, Proc. CSCW '02, New Orleans, LA.
- [2.9] Nardi, B., Whittaker, S. and E. Bradner. "Interaction and Outeraction: Instant Messaging in Action." Proceedings of CSCW 2000. (2000), ACM Press, 79-88.
- [2.10] AOL Instant Messenger Home Page <http://www.aim.com/>
- [2.11] MSN Instant Messenger Home Page <http://messenger.msn.com/>
- [2.12] Jabber Software Foundation, <http://www.jabber.org/>
- [2.13] Ranganathan, R. Campbell, A. Ravi and A. Mahajan, ConChat: A Context-Aware Chat Program. *IEEE Pervasive Computing*, pp. 52-58, July-Sept 2002,

Context-Aware Computing Gregory D. Abowd, Maria Ebling, Hans-Werner Gellersen, Guerney Hunt, Hui Lei.

- [2.14] Morán, A., et-al, “On the design of potential collaboration spaces”. International Journal of Computer Applications in Technology, Vol. 19, Nos. ¾, 2004, pp. 184-194.

- Redes *Peer to Peer*
- El proyecto como red virtual
- JXME



En la siguiente sección se presentan los protocolos JXTA para desarrollar redes *Peer to Peer*. En esta sección se explican los servicios principales de búsqueda y comunicación de JXTA para hacer posible la colaboración dentro de una red. Se presenta también a JXME la cual es una API para lograr la posibilidad de que los dispositivos móviles puedan participar dentro de una red *Peer to Peer* y así poder colaborar como un recurso más.

3.1 REDES PEER TO PEER (P2P)

Como se mencionó en la introducción en el capítulo 0, este trabajo comprende la implementación de un sistema *Peer to Peer* (P2P), donde los elementos de la red pueden colaborar sin poseer algún tipo de control centralizado o un tipo de organización jerárquico.

Así mismo, la tecnología JXTA es una programación en red que está diseñada para resolver un número de problemas en la computación distribuida moderna, especialmente en el área ampliamente referenciada como lo es la computación *peer to peer*, o redes *peer to peer*, o simplemente P2P.

En este sistema implementamos los protocolos JXTA, porque son un conjunto generalizado y abierto de protocolos que permite a cualquier recurso sobre la red comunicarse y colaborar como un peer, desde computadoras de escritorio hasta PDA's. Además, son independientes del lenguaje de programación [3.1].

El proyecto de JXTA fue originalmente concebido por *Sun Microsystem Inc.* Actualmente esta tecnología ha sido empleada en una gran variedad de aplicaciones, de las cuales podemos mencionar algunas, como por ejemplo, sistemas de administración de información, sistemas de archivos compartidos, aplicaciones colaborativas, etc. El proyecto JXTA estandariza un conjunto común de protocolos para construir redes virtuales P2P. Así mismo, los protocolos definen los requerimientos mínimos para formar y unirse a redes P2P, así como la forma en que se comunicarán los elementos involucrados en la red virtual P2P.

3.1.1 El proyecto JXTA como red virtual.

Los protocolos JXTA crean una red virtual encima de la infraestructura de la red física y la red virtual formada permite a los participantes intercambiar mensajes con cualquier otro peer independientemente de su localización en la red, esto posibilita a los peers móviles desplazarse libremente en la red. Los mensajes intercambiados son transparentemente ruteados, atravesando *firewalls* o *NAT's*, y usan diferentes protocolos, como pueden ser TCP/IP o http, para alcanzar al peer destinatario, como lo muestra la figura 3.1. El proyecto de red virtual JXTA estandariza la manera en la cual los peer se descubren, se auto organizan, descubren recursos en la red y la forma en que se comunican entre ellos. La implementación del proyecto JXTA se construye encima de 5 abstracciones de red virtual. Primero, un *modelo de direccionamiento lógico* de los peers que se expande sobre la red. Segundo, los grupos de *peers* (*peergroups*) que permite a los peers dinámicamente auto organizarse en un dominio virtual protegido. Tercero, los *anuncios* para que los peer publiquen sus recursos (*peer*, *peergroup*, servicios). Cuarto, un mecanismo de enlace llamado *resolver* para desempeñar todas las operaciones de enlace requerido en los sistemas distribuidos. Finalmente, los *pipes* los cuales son los canales de comunicación que permite a los peers comunicarse. En las siguientes secciones se abarcarán cada una de los 5 puntos.

3.1.2 Los JXTA ID's

El modelo de direccionamiento de JXTA está basado en un modelo de direccionamiento lógico uniforme e independientemente de la localización. Cualquier recurso en la red (*peer*, *pipe*, *peergroup*, etc.) se le asigna un único JXTA ID. Los JXTA ID's son objetos abstractos que permiten una representación de un ID (IPv6, MAC) para coexistir en la misma red JXTA. La referencia de implementación es usar UUID's (*Universal Unique Identifier*) aleatorios de 128 bits permitiendo a cada peer generar su propio ID. De este modo, un peer en la red solamente puede ser identificado por su *peer* ID permitiéndole al peer ser direccionado independientemente de su dirección física. Así mismo, un peer puede soportar múltiples interfaces de red (*Ethernet*, *Wi-Fi*, etc.) y será direccionado como un único peer independientemente de la interfaz usada. La abstracción de *Peer ID* permite a los peers encapsular no solo protocolos de transporte físicos, sino también protocolos de transporte lógico como HTTP o TLS (*Transport Layer Security*). En nuestro sistema cada uno de los colaboradores es un peer dentro de la red JXTA con un identificador único que lo represente en el ambiente de colaboración y se le pueda hacer llegar el documento de edición.

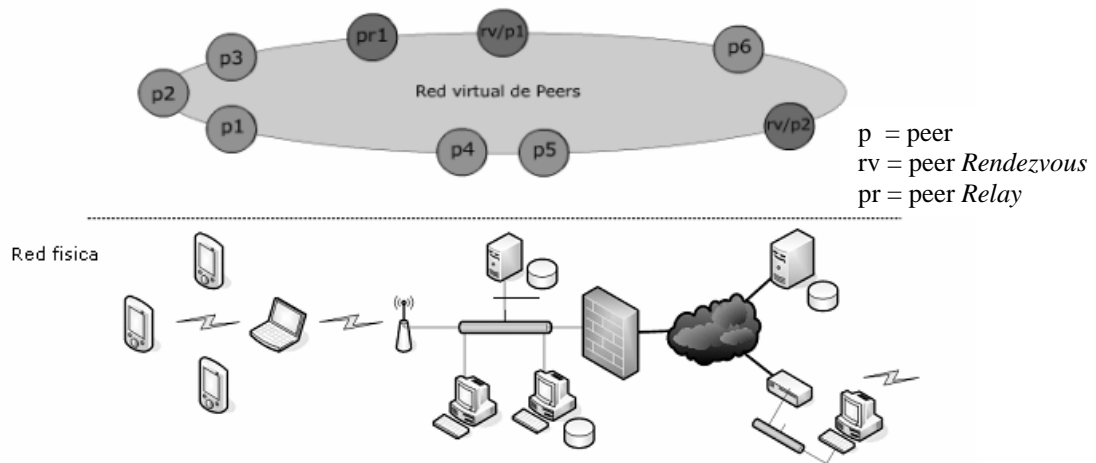


Figura 3.1 La red virtual del proyecto JXTA

3.1.3 Los anuncios en JXTA

Todos los recursos en el proyecto JXTA, son representados por anuncios (*Advertisements*). Los anuncios son descriptores de un recurso representados como documentos XML. El proyecto JXTA estandariza los anuncios de los recursos principales como son: los *peers*, los *peergroups*, los *pipes*, los servicios, etc. Aunque los desarrolladores pueden crear sus propios anuncios, la figura 3.2 muestra un ejemplo de un *anuncio de un peergroup* (jxta:PGA); (GID) describe un *peergroup* por su ID; un módulo de especificación (MSID) que apunta a un anuncio que describe los servicios disponibles para este grupo, (Name) el nombre del *peergroup*, y una descripción del grupo (Desc). Los *peers* publican e

intercambian anuncios para descubrir y encontrar recursos en la red. Todos los anuncios son publicados con un tiempo de vida que especifica la duración de los anuncios sobre la red.

```
<?xml version="1.0"?>
<!DOCTYPE jxta:PGA>
<jxta:PGA xmlns:jxta="http://jxta.org">
  <GID> urn:jxta:jxta-NetGroup</GID>
  <MSID>urn:jxta:uuid-DEADBEEFDEAFBABAFEEDBABE00000010206</MSID>
  <Name>NetPeerGroup</Name>
  <Desc>NetPeerGroup by default</Desc>
</jxta:PGA>
```

Figura 3.2. Ejemplo de un Anuncio de un *Peergroup*

3.1.4 Los peer Rendezvous

El proyecto de red JXTA usa un mecanismo de enlace al recurso llamado *resolver* para desempeñar todas las operaciones de resolución encontradas en los sistemas distribuidos tradicionales, desde resolver el nombre de un peer en una dirección IP (DNS), enlazar el puerto de un *socket*, localizar un servicio por medio del directorio de servicios (LDAP), o buscar por el contenido en un sistema de archivos distribuidos (NFS). En los protocolos del proyecto JXTA, todas las operaciones de resolución son unificadas bajo un simple descubrimiento de uno o más anuncios. No se especifica cómo es la búsqueda de anuncios, pero proporcionan una arquitectura de protocolo de enlace genérico con unas políticas por omisión que pueden ser sobre escritas.

La red JXTA proporciona unas políticas basadas en los *Peers Rendezvous*. Los *Rendezvous* son *peers* que tienen agregado a su *cache* índices de anuncios (es decir, apuntadores a los *peers* más básicos como son los *peers edge* de los que ha guardado sus anuncios). Los *peers edge* sólo representan a un recurso dentro de la red. Los *Rendezvous* son *peers* cuya localización es bien definida y éstos pueden tener un mejor conocimiento de la distribución de los demás peers en la topología de la red [3.2][3.3][3.4].

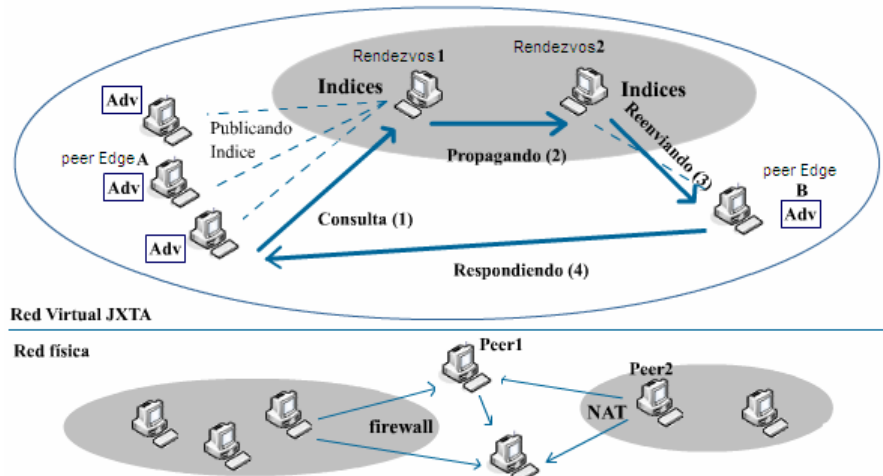


Figura 3.3. Los super *peers Rendezvous* del proyecto JXTA

Los *peer Edge* mantienen una relación especial con su *peer Rendezvous*. Cualquier *peer* puede llegar a ser un *Rendezvous* si cumple con las credenciales correctas. En los *PeerGroups* seguros se puede restringir qué *peer* actuará como *Rendezvous*. Los *Rendezvous* solo mantienen el índice del anuncio publicado por el *peer edge* sin mantener el anuncio completo, de esta forma reducen el problema de obtener anuncios fuera de tiempo. Los *peers edge* usan el servicio de Índice Distribuido de Recurso Compartido (SRDI) para indexar su anuncio sobre el *Rendezvous*. Los índices pueden ser puestos sincronizadamente cuando un nuevo anuncio es publicado o en forma asíncrona por el demonio SRDI que se corre en intervalos fijos. En la figura 3.3 por ejemplo, ambos el *peer A* y el *peer B* están publicando sus anuncios a sus respectivos *Rendezvous* Rdv1 y Rdv2. Cuando el *peer A* hace una consulta de un anuncio almacenado en el *peer B*, la consulta es enviada al *Rendezvous* del *peer A* Rdv1 (1), el Rdv1 busca en su *cache* si tiene el índice de ese anuncio. Si no lo encuentra, propaga la consulta al próximo *Rendezvous* Rdv2 (2), Rdv2 encontrará el índice en su *cache* y reenviará la consulta al *peer B* (3). Cuando el *peer B* recibe la consulta enviará el anuncio al *peer A* (4).

3.1.5 Redes *Rendezvous* de baja consistencia

En una red fluctuante, es difícil mantener una vista consistente de los *peers*, y este problema viene a ser mayor si el número de *peers* se incrementa [3.2]. En una red empresarial P2P donde las computadoras son estáticas se puede ver una alta estabilidad, pero en una red móvil donde los participantes son del tipo de PDA's o teléfonos celulares no es tan estable. JXTA introduce una Tabla *Hash* Distribuida común (DHT) como en [3.2], [3.3], o [3.5] que asumen una infraestructura estable de *peers* donde se puede mantener una vista distribuida consistente.

3.1.6 La vista de los *peers* del *Rendezvous*

Cada *Rendezvous* mantiene su propia vista de *peers* (Una lista ordenada de *peers* que conoce en su *peergroup* por su ID). Los *peers* pueden tener inconsistencia en su vista RPV (vista de *peers rendezvous*) por eso, usan un algoritmo para que puedan resolver el problema. Los *Rendezvous* periódicamente seleccionan un número aleatorio de *Rendezvous* de su vista local, y envían una lista aleatoria de los *peers* que conocen. Cada *peergroup* tiene la habilidad de definir su propio conjunto de *peers Rendezvous* a los que les va a enviar la vista, el envío puede acelerar una convergencia en la RPV.

3.1.7 Conexión de los *peers edge* a los *Rendezvous*

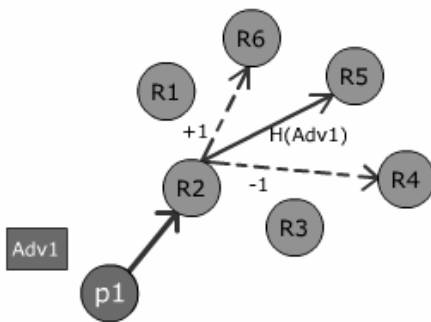
Para que un *peer Edge* se pueda conectar a un *Peer Rendezvous*, el *peer edge* primero verifica en su *cache* local por algún anuncio de un *Rendezvous*. Los *peer edge* ordenan los *Rendezvous* candidatos en una *batch* de cinco e intenta conectarse con los cinco hasta que una conexión sea establecida. Los *peer edge* solo mantiene una conexión. Si ninguna conexión es establecida en un periodo de 30 segundos, el *peer edge* intentará hacer nuevamente una búsqueda en IP *multicast*. Existirán *Rendezvous* que estén escuchando entonces ellos contestarán a la solicitud del *peer edge*, pero si en un periodo de 5 minutos

ningún *Rendezvous* ha sido alcanzado, el *peer edge* intentará convertirse en un *Rendezvous* (si tiene la capacidad).

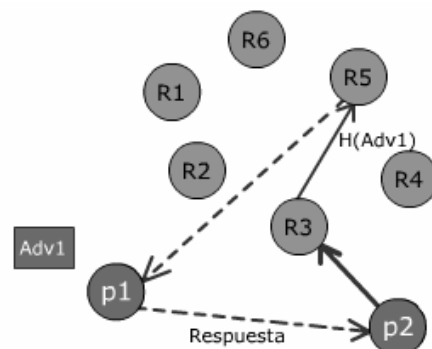
3.1.8 Propagación en los *Rendezvous*

En la figura 3.4.1a se muestra cómo un nuevo anuncio publicado es indexado sobre la RPV del *Rendezvous*. El P1 publica un nuevo anuncio sobre su *Rendezvous* R2 por medio del servicio RDSI antes mencionado. R2 usa la función (H(adv1)) de una tabla *hash* distribuida DHT para mapear el índice a un *Rendezvous* en su RPV local. La RPV de R2 contiene los *Rendezvous* R1 al R6. Por poner un ejemplo, suponga que la función DHT nos regresa R5. Entonces R2 asignará el índice a R5 para aumentar la probabilidad de recuperar el índice en una vecindad de R5, el índice también es replicado en los vecinos de R5 (+1 y -1 en la lista RPV ordenada). En este ejemplo el índice se replica sobre R4 y R6. Es importante notar que la forma de la RPV no tiene que ver con la forma de la red física. R5 y R6 pueden estar en lados opuestos del hemisferio. Replicar el índice alrededor de R5 significa que se está aumentando la región lógica en la RPV donde el índice pueda ser localizado. Vamos a suponer que el *peer edge* P2 está buscando el anuncio adv1 (figura 3.4.2a). El P2 intentará resolver su consulta con su *Rendezvous* R3. El servicio SRDI sobre R3 realizará la función (H(adv1)) usando la vista local de R3. Si la RPV sobre R2 y R3 son la misma, la función DHT retornará el mismo *Rendezvous* R5. R3 reenviará la petición a R5 que a su vez la reenviará a P1 para así responder a P2.

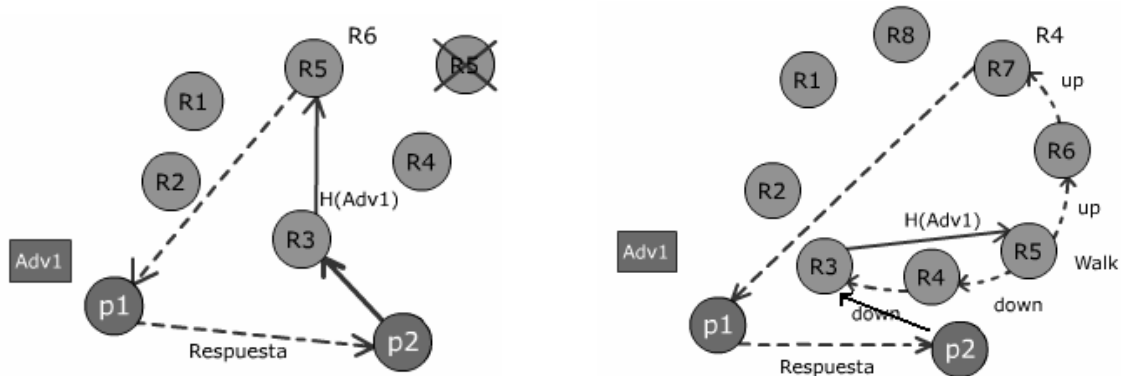
Ahora suponga que se cancela a R5, R3 actualiza su RPV para reflejar el hecho de que R5 desapareció de su RVP (figura 3.4.2b). De esto, R3 tendrá una nueva RPV, R6 ahora tomará el lugar de R5 de nuestro anterior ejemplo. Para resolver la petición de P2 nuevamente se realizará la función DHT que regresará el nuevo R5, dado que también se publicó el índice sobre R6 como parte de la estrategia de replicación. Este ejemplo demuestra que aún con una RPV inconsistente la DHT puede ser utilizada con éxito.



1a) Publicando Adv1



2a) Buscando Adv1 (RPV consistente)



2b) Buscando Adv1 (RPV inconsistente) 2c) Buscando Adv1 (encaminamiento de Rango limitado)

Figura 3.4. Una DHT consistente

Existen escenarios más caóticos de la RPV donde los cambios son masivos (ver figura 4.2.c). La RPV de R3 está ahora integrada de 8 *rendezvous* (R1-R8) correspondientes al R4 original. Cuando R3 recibe la consulta de P2 realizará la función DHT que regrese el índice a R5, pero como la RPV ha cambiado drásticamente el índice no será encontrado sobre R5. En este caso, se usa un mecanismo alternativo para encaminar el *peer rendezvous*. Se usa *un encaminamiento de rango limitado*, después que se alcanzó a R5 el encaminamiento será en ambas direcciones hacia arriba por R6 y hacia abajo por R4. El objetivo es buscar el *rendezvous* que pueda tener el índice buscado, este encaminamiento tiene las ventajas de incrementar la probabilidad de encontrar el índice en la región por el esquema de replicación en los vecinos.

3.1.9 Los *peers Relay*

El proyecto JXTA es una *red ad hoc*, en donde los mensajes ruteados no son deterministas, donde las rutas pueden ser unidireccionales y cambian rápidamente. Esto puede ser un problema cuando trabajemos con *peers* móviles (colaboradores móviles) dado que tienen la posibilidad de unirse y dejar la red en cualquier momento. El proyecto JXTA introduce la noción de *Peers Relay* para auxiliar a los *peers* que no tengan una conexión física directa (que estén bajo NAT o firewalls). Cualquier *peer* puede llegar a ser un *peer Relay* asumiendo que tiene las capacidades (ancho de banda, conectividad directa). Los *peer Relay* proporciona la habilidad para encolar mensajes hacia *peers* inalcanzables o temporalmente indisponibles. Por ejemplo en la figura 3.5, un *peer A* quiere enviar un mensaje al *peer B* pero como el *peer B* está bajo una NAT el *peer A* no puede enviar mensajes directamente al *peer B* (es decir, el *peer B* no está al alcance de A). Entonces el *peer B* usa un *peer relay D* que lo hace alcanzable. De este modo el *peer D* facilita el ruteo de mensajes. Una parte del anuncio de los *peers* contiene un conjunto de *peer relay* preferidos para ayudar al ruteo. Aunque los *peer* siempre intentarán una conexión directa antes de usar un *relay*.

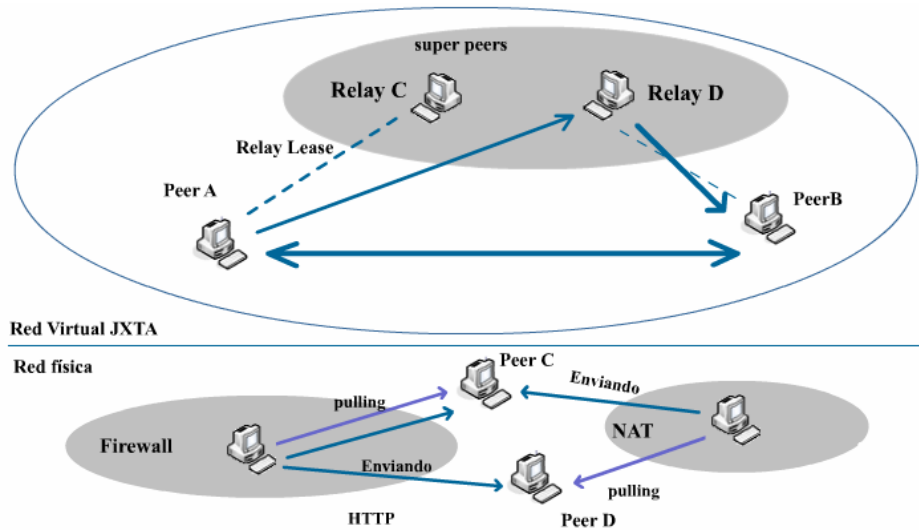


Figura 3.5. Los super *peers* Relay del proyecto JXTA

El ruteo de los mensajes vía un *relay* es hecho transparentemente y dinámicamente por la red virtual JXTA. Incluso los *peers* pueden enviar mensajes a través de cualquier *relay* disponible, no solo por su *relay* preferido. Igual que los *Rendezvous* los *relays* pueden mantener una vista entre ellos mismos para así obtener una lista de los relays disponibles.

JXTA usa un ruteo adaptativo. Las rutas son construidas inicialmente por el que envía, esto es hecho para descentralizar el manejo de la ruta y se evita el utilizar tablas de ruteo centralizadas. Como cualquier otro recurso, las rutas son representadas por los *anuncios de ruteo*. Los *peers edge* pueden cachear anuncios de rutas para descubrir un camino hacia un destinatario. Un anuncio de ruteo describe la forma de cómo alcanzar un *peer* en una secuencia ordenada de saltos. Cada salto es definido por el ID del peer, los saltos son normalmente *peers relay*.

Los mensajes JXTA contienen información de ruteo como uno de sus elementos. En cada salto la información es actualizada. Cuando un *peer* recibe un mensaje, éste puede usar la información de ruteo del mensaje para contestar al que hizo la petición.

Por lo tanto cuando el *peer* colaborador sea un peer móvil, primeramente se tiene que buscar una conexión con un *peer relay* para que los mensajes del *peer* móvil y hacia el peer móvil puedan ser ruteados por *el peer relay*.

3.1.10 Los mensajes en JXTA

Los mensajes son la unidad básica de intercambio de información entre los *peers*, los *peers* interactúan enviando y recibiendo mensajes. Los mensajes pueden ser XML o Binario, ambos formatos pueden ser usados dependiendo de la situación. Un mensaje es una secuencia ordenada de nombres y contenido denominado elementos.

A partir de los mensajes JXTA es como el documento de edición será transportado por la red JXTA así también, las actualizaciones del documento y aportaciones hechas sobre el documento de edición.

3.1.11 Los PeerGroups

Como se mencionó anteriormente, los *peers* se auto organizan en *peergroups* (grupo de *peers*). Un *peergroup* representa un conjunto dinámico de *peers* que tienen un conjunto común de intereses y adicionalmente tienen un conjunto común de políticas (membresía, intercambio de contenido). Cada uno de los *peergroups* es identificado únicamente por su ID. En el proyecto JXTA no se estipula cuándo, dónde o por qué los *peergroups* son creados, solamente describe como un *peergroup* es creado, publicado, y descubierto. Los *peergroups* pueden ser creados dinámicamente por los usuarios, por los desarrolladores o por los administradores de red para separar el ámbito de interacción entre los *peers*. En la figura 3.6, se muestra un *peergroup* B que está abarcando múltiples dominios de la red física, el *peergroup* C muestra a un *peergroup* que abarca exactamente lo que compone el dominio de un NAT, como es posible observar, un *peer* puede pertenecer al mismo tiempo a múltiples *peergroups*.

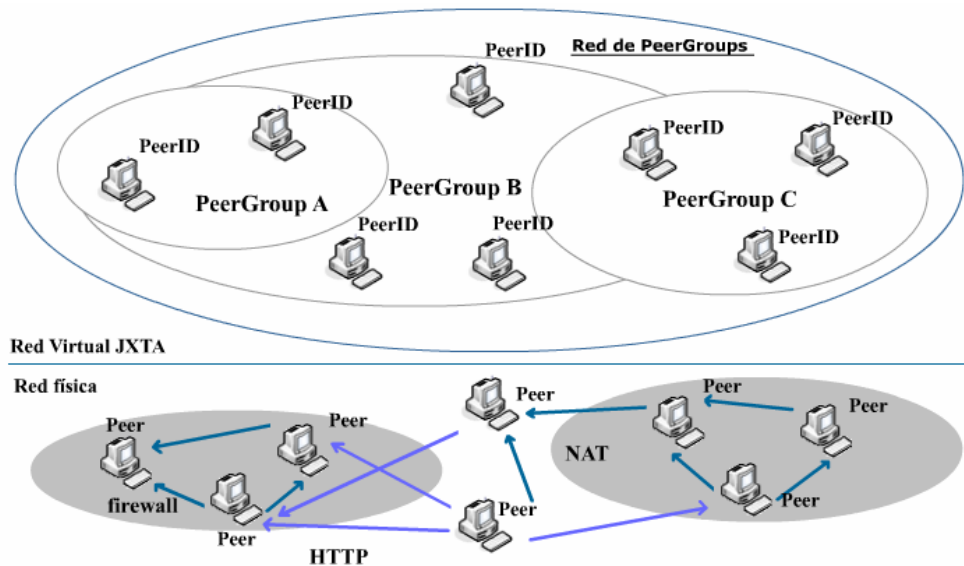


Figura 3.6. Los PeerGroups en el Proyecto JXTA

En JXTA se recomienda crear *PeerGroups* por tres razones:

- Cuando se quiere crear dominios seguros para el intercambio seguro de información, los *peergroups* forman una región lógica que limita a los miembros del *peergroup*. El *peergroup* subdivide la red en regiones seguras con respecto a la red física.
- Para crear entornos comunes. Los *peergroups* se forman basándose en el interés mutuo de los *peers*. Los *peers* con el mismo interés tienden a unirse, eso sirve para

dividir la red en regiones abstractas restringiendo la propagación de anuncios de búsqueda y de descubrimiento.

- Para crear un entorno fácil de monitorear. Los *peergroups* permiten el monitoreo de los *peers* (por ejemplo, el tráfico).

De los *PeerGroups* se desea aprovechar esta característica para que el documento de edición sea quien delimite los grupos de colaboración con los autores involucrados en el trabajo de edición, y a su vez cada grupo de colaboración forme un *PeerGroup* colaborativo.

3.1.12 Los Pipes

Los *pipes* son canales virtuales de comunicación usados para enviar y recibir mensajes entre los servicios y las aplicaciones. Los *pipes* proporcionan una abstracción virtual sobre los *peers* para proporcionar la ilusión de entradas y salidas que no están físicamente limitadas a la localización del peer. Los *pipes* pueden conectarse a uno o más *peer* receptores. En JXTA puede haber *pipes de entrada* y *pipes de salida*, como lo muestra la figura 3.7.

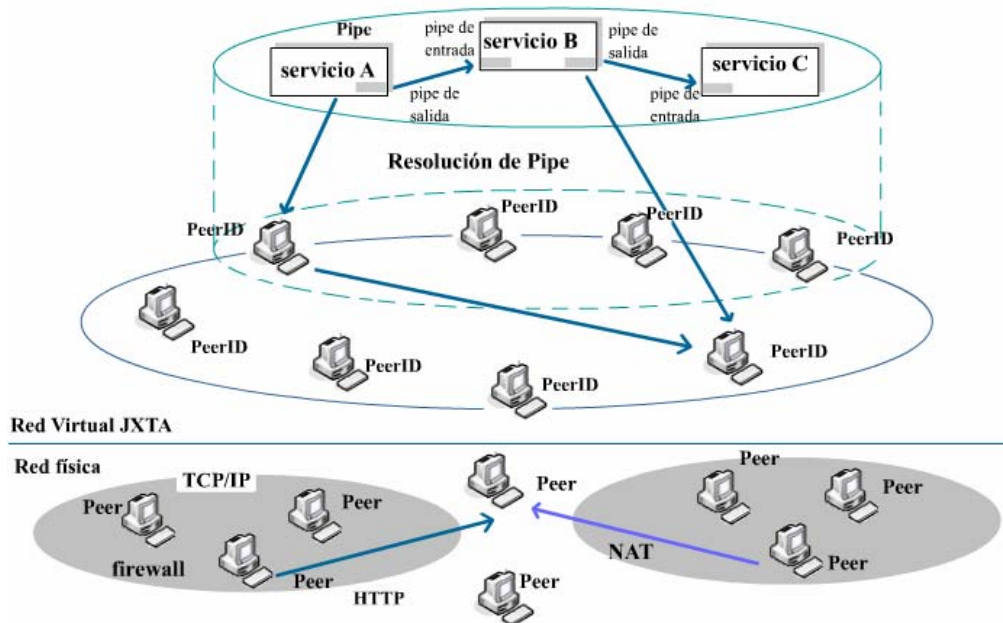


Figura 3.7. Los Pipes en JXTA

Los *pipes* son publicados y descubiertos usando un anuncio de *pipe* y son identificados únicamente por su *pipeID*. El proceso de enlazado de un *pipe* consiste en buscar y conectar dos o más *pipes*. Cuando un mensaje es enviado a un *pipe*, el mensaje es transmitido por el

pipe de salida local a la entrada de un *pipe* que esté escuchando. Por lo regular los *pipes* ofrecen dos modos de comunicación:

- Un *pipe de conexión punto a punto*, que son dos *pipes* conectados uno al otro con un canal unidireccional y asíncrono, un *pipe* de entrada que recibe el mensaje de un *pipe* de salida. No existe alguna operación de retransmisión o reconocimiento de que llegó el mensaje. El mensaje puede contener un anuncio de *pipe* que puede ser usado para abrir un nuevo *pipe* para responder al *pipe* que le envió el mensaje.
- Un *pipe de propagación* que conecta un *pipe* de salida a múltiples *pipes* de entrada. El mensaje de propagación es enviado a todos los *pipes* de entrada que estén escuchando en el contexto del *peer group* presente.

Los servicios de *pipe* bidireccionales, confiables y *pipes* seguros han sido implementados en la cima de los servicios principales de los *pipes*. Los anuncios de *pipes* pueden contener esquemas de datos tipados XML que describen el conjunto básico de mensajes que son enviados o recibidos a través del *pipe*.

3.1.13 Seguridad en el Proyecto JXTA

Debido a que los modelos de confianza basados en certificados para proporcionar infraestructura de llave pública son extremadamente difíciles de manejar, el proyecto JXTA proporciona un modelo de confianza a nivel de entrada [3.6] que no tiene costo y puede fácilmente ser soportado por los modelos de confianza de Internet existentes. Este modelo es apropiado, por ejemplo, en salones de *Chat*, en la compartición de contenido y en las transacciones seguras financieras. El modelo de confianza en JXTA permite a los peers ser su propia autoridad certificante, proporciona potentes algoritmos de cifrado para los datos en transmisión sobre la red virtual y para los datos almacenados remotamente. JXTA ha implementado un transporte virtual basado en TLS para proporcionar comunicaciones seguras entre los peers. La suite de cifrado por defecto es RSA1024 con 3DES y SHA-1.

3.2 JXME

El principal propósito del proyecto JXTA para J2ME es proporcionar compatibilidad de los servicios de JXTA para dispositivos móviles tales como: los teléfonos celulares y los asistentes personales digitales (PDA's), usando JXTA para J2ME cualquier dispositivo MIDP (*Mobile Information Device Profile*) podría participar en una red P2P.

3.2.1 Principales Objetivos JXME

- Ser interoperables con cualquier elemento de la red JXTA.
- Ser simple y fácil para los desarrolladores.
- No necesitar grandes recursos para funcionar sobre teléfonos celulares y PDA's

J2ME incluye unos perfiles especiales diseñados para asistir a dispositivos pequeños de recursos limitados [7]. La CLDC (Configuración de Dispositivos de Conexión Limitada) y la MIDP (Perfil para Dispositivos de Información Móvil), son las clases principales y API's especializadas para trabajar con entornos restringidos sobre dispositivos con conexión inalámbrica. MIDP y CLDC nos ofrece herramientas necesarias para desarrollar aplicaciones inalámbricas como nuestra aplicación de trabajo colaborativo donde los componentes (Computadoras de escritorio, teléfonos celulares o PDA's) puedan operar con Java.

CLDC. Es adecuado para dispositivos con 16/32 bits con microprocesadores RISC/CISC y al menos de 160 KB de memoria total.

MIDP. Es un conjunto de API's de Java que está orientada a dispositivos móviles, la especificación MIDP- 1.0 se enfoca a la interfaz del usuario, el almacenamiento de la información y las comunicaciones de red.

3.2.2 Proyecto JXTA para J2ME

La comunidad JXTA de Java comprende que MIDP tiene algunas restricciones, por ejemplo, librerías limitadas, limitaciones para el parseo de documentos XML, solo soporta comunicación HTTP y la seguridad está limitada, de ahí que se vea reflejado en las capacidades de JXTA con J2ME.

3.2.3 JXTA Relays

Dadas las limitaciones de los dispositivos móviles, los *peers* J2ME solo pueden actuar como *edge peers*. Es decir, no pueden ofrecer servicios a otros miembros de un *PeerGroup*. Los *peers* inalámbricos solo pueden ser ventanas en una red para los usuarios [1], por lo tanto, las tareas más complicadas como la búsqueda de recursos debe ser hecha por *peers* estables dentro de la red JXTA llamados *Peers Relays* (Ver figura 3.8).

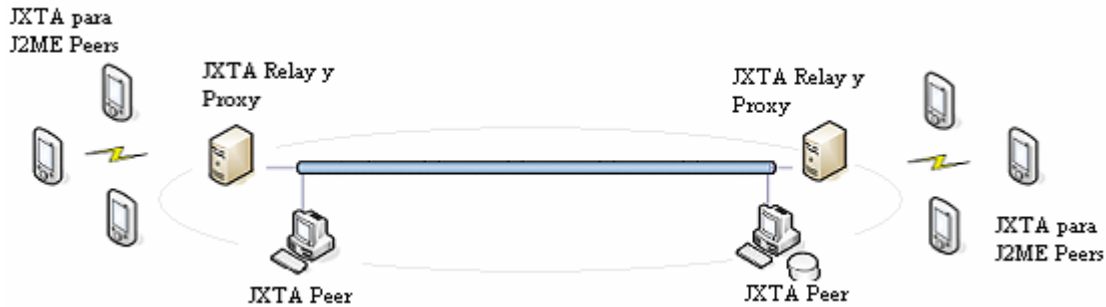


Figura 3.8. Dispositivos inalámbricos que son miembros de una red JXTA conectados a través de *peerRelay*, y pueden comunicarse independientemente de los protocolos de red.

Los *JXTA relays* realizan las tareas importantes para los peers inalámbricos y así mantener su estatus como miembro de un *PeerGroup*.

De las tareas más importantes podemos mencionar:

3.2.3.1 Proporcionar interoperabilidad con los protocolos JXTA.

- Proporcionar a los *peers* J2ME tareas de usuario, de grupo y de descubrimiento de *peers*. Así como crear *pipes* y grupos dentro de la red JXTA.
- Filtrado de Tráfico.
- Ajuste para la optimización de anuncios.

3.2.3.2 Los *Relays* no comprometen la visión P2P.

A diferencia del modelo cliente-servidor, los *peers* inalámbricos JXTA no requieren establecer y mantener una relación estática con un *relay* designado. Dos *peers* J2ME pueden ser conectados a diferentes *relays* y aun así descubrirse y comunicarse. De hecho un *peer* inalámbrico puede dinámicamente cambiar su *relay* o tener múltiples *relays*.

Los *relays* JXTA filtran todas las peticiones de entrada de la red y responden a las consultas hacia los *peers* inalámbricos permitiendo así reducir el volumen del tráfico con los protocolos JXME.

El modelo *peer to peer* es especialmente apropiado a las características de los dispositivos inalámbricos. Su movilidad implícita requiere un método de descubrimiento de recursos sobre la red. Así mismo, sus recursos limitados significan una mayor confianza de los servicios sobre la red para crear aplicaciones de confianza.

3.3 Resumen

En este capítulo abordamos una introducción de lo que son las redes *peer to peer*. Mencionamos principalmente los protocolos JXTA para formar las redes *peer to peer*, es decir, una red virtual donde los elementos se auto organizan y se comunican por medio de anuncios para formar los *peer groups*. Nuestro sistema retoma el concepto de *peer group* para formar los grupos de edición colaborativa sobre los documentos. Hablamos de la importancia de contar con los *peers rendezvous* para mantener la consistencia de los *peer* dentro de la red *peer to peer* y de cómo se comunican éste tipo de *peers* para encontrar los anuncios de los *peer edge*, esto es de gran utilidad para nosotros para mantener la consistencia de los anuncios sobre el documento y de los mensajes entre los colaboradores del sistema. Mencionamos otro tipo de *peer* llamado *relay* cuya principal función es hacer transparente las comunicaciones dentro de la red virtual entre diferentes NAT's. Como lo hemos mencionado en los objetivos, nuestra intención es hacer un sistema colaborativo también en dispositivos móviles, para esto utilizamos los protocolos JXTA para Java *Microedition* (JXME) y nos dimos cuenta de la importancia de los *peer relay* para cualquier dispositivo móvil que quiere ingresar a la red virtual.

Dada las características principales de los paradigmas y herramientas que vamos a utilizar para el desarrollo de nuestro sistema de edición colaborativa, en el siguiente capítulo abordaremos la arquitectura de nuestra herramienta y cómo todas estas tecnologías interactúan para lograr nuestro propósito.

3.4 REFERENCIAS

- [3.1]. Traversat, B., et-al. Project JXTA 2.0 Super-Peer Virtual Network. <http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf>
- [3.2]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, *A Scalable Content Addressable Network*, ACM SIGCOM, 2001.
- [3.3]. F. Dabek, E. Brunskill, M.F. Kaashoek, D. Karger, R. Morris, I. Stoica, and H. Balakrishnan, *Building Peer-to-Peer Systems with Chord, a Distributed Lookup Service*, 2001
- [3.4]. T. Berners-Lee, J. Hendler, and O. Lassila, *The Semantic Web*, <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>
- [3.5]. B. Zhao and al., *Brocade: Landmark Routing on Overlay Networks*, in Proceeding of the First International Workshop on Peer-to-Peer Systems, Mar. 2002.
- [3.6]. B. Traversat and al, *The Project JXTA Virtual Network*, May 2001, www.jxta.org/docs/JXTAprotocols.pdf,
- [3.7]. JXTA for J2ME – Extending the Reach of Wireless With JXTA Technology, Mar 2002.

Arquitectura de ECO



En este capítulo corresponde hablar de la arquitectura de la herramienta colaborativa ECO, hablaremos de cuales son los requerimientos para formar sistemas de computación ubicuos colaborativos, también se explicará cada uno de los componentes involucrados en el sistema, así como la forma en que se relacionan e interactúan para lograr la edición colaborativa. Hablaremos de los protocolos de comunicación JXTA para redes *Peer to Peer* que son la base de las comunicación en nuestra herramienta.

4.1 ARQUITECTURA DE ECO

En este capítulo presentamos el diseño de la arquitectura del proyecto ECO, donde se explica cada uno de los elementos participantes del sistema de colaboración ECO. Así como la forma de implementación de éstos.

En muchos de los documentos creados y elaborados en colaboración [4.1]. Las personas envueltas en la sesión de revisión, al mismo tiempo tienen otras varias actividades, Así, es difícil llevar a cabo una sesión real de colaboración donde los autores intercambian sus ideas acerca de las principales partes del documento, en lugar de solo leer los cambios hechos o propuestos por algún coautor. Esta situación podría traer problemas de una mala sincronización en el trabajo de edición colaborativa, donde las partes críticas del documento (los cuales necesitan el acuerdo y las contribuciones de los demás autores) son cambiadas varias veces hasta que el documento satisface a todos los autores involucrados en la sesión. Esta situación puede aumentar significativamente el tiempo necesario para completar el documento.

Como un paso para dar solución al problema, proponemos usar la mensajería instantánea para mantener a los otros usuarios informados cuando alguien está presente trabajando con un documento compartido. En esta forma, los autores estarán aptos para llevar a cabo una sesión colaborativa de edición, particularmente cuando se empieza a editar una parte crítica del documento. Este esquema tiene la posibilidad de discutir cambios críticos del documento mientras ellos están en el proceso de elaboración.

Esto es similar a los propuestos en [4.2] donde presentan el diseño de espacios potenciales de colaboración, argumentan que es importante proporcionar asistencia no solo para actividades colaborativas dentro de los espacios de colaboración, sino también para aquellos que pudieran ocurrir fuera de él y que podrían proporcionar una asistencia importante para colaboración, introducen dos conceptos importantes: *Espacios de colaboración reales* y *Espacios de colaboración potenciales*. Un espacio de colaboración actual es donde las actividades de colaboración toman lugar y un espacio de colaboración potencial representa un espacio adicional donde la posibilidad de colaboración es idéntica y de la cual una interacción inicial es establecida. Nosotros implementamos el editor de colaboración ECO que podría ser clasificado en un espacio de colaboración actual y el Mensajero Instantáneo ECO que puede ser clasificado en un espacio de colaboración potencial porque satisface el conjunto de requerimientos identificados en [4.2]:

- En ECO, un usuario tiene conocimiento de una colaboración potencial, sin tener que dejar el modo de trabajo actual. En este caso usamos mensajería instantánea donde a pesar de no ser muy intrusivo, permite a los usuarios conocer el estado de los demás usuarios con respecto a los documentos compartidos relevantes.
- Indicar a otros usuarios, directamente o indirectamente, el interés o disponibilidad (o la carencia de ella) para obtener colaboración. En particular, nuestro sistema indica si cualquier otro usuario está presente trabajando sobre un documento compartido.

■ Ser capaz de lanzar un mensaje de interacción inicial para negociar la posibilidad de colaboración. En nuestro caso, cuando un usuario identifica que otro usuario está editando un documento (por ejemplo un documento el cual él ha estado trabajado), solo necesita dar doble clic en el icono del documento para preguntar por una sesión de edición colaborativa. En este caso, como se vio en el capítulo dos de colaboración oportunista, las personas envueltas en esta interacción son capaces de negociar si entran o no en una sesión colaborativa.

■ Ser capaces o ignorar cualquier aviso de colaboración potencial, incluyendo peticiones directas por otros usuarios. Como es en el caso de Mensajería Instantánea, un usuario puede ignorar una solicitud de colaboración. Esto es importante para nosotros porque una de nuestras principales ideas fue diseñar una herramienta lo menos intrusiva posible.

Entre las ventajas de espacios de colaboración potencial podemos mencionar: los espacios de colaboración potencial proporcionan soporte para identificar la posibilidad de colaborar y establecer una interacción inicial para colaborar, mientras se trabaja en forma individual. Proporcionan mecanismos para transiciones parecidas entre modos de trabajo individual y colaborativo.

En otro de los casos, para maximizar la posibilidad de que un autor interesado en un documento en particular llegara a percibir los cambios hechos sobre el documento, implementamos acceso a nuestro sistema desde una gran variedad de recursos, desde teléfonos celulares hasta computadoras de escritorio. En la misma forma, se desarrollan un conjunto de servicios distribuidos para soportar escenarios que puedan ser totalmente implementados por dispositivos móviles. Los ejemplos de esos servicios son: el servicio de percepción distribuida de ECO (ECO-DA) y el servicio de repositorios de archivos de ECO (ECO-FR).

Los archivos compartidos en ECO pueden ser accedidos de una manera ubicua por medio de diferentes tipos de recursos tales como teléfonos celulares o PDA's. Sin embargo, los teléfonos celulares y PDA's no tienen suficiente memoria para almacenar archivos grandes y su acceso a la red es comúnmente discontinuo y ello podría cambiar su identificador de red cada vez que cambie de una NAT a otra o cuando la PDA se una a diferente red *ad hoc*. Todas esas características imponen restricciones a la infraestructura distribuida así como al comportamiento del sistema. Para hacer frente a esas restricciones usamos los *pipes* Jxta [4.3] como principal método para enviar y recibir todo tipo de información en ECO. Seleccionamos los Jxta *Pipes* porque ellos proporcionan la ilusión de una red virtual a la que no se une a una localización de transporte específica. Los *pipes* son dinámicamente enlazados a un *peer endpoint* en tiempo de ejecución. Usando la abstracción de *pipe* nuestras aplicaciones y nuestros servicios pueden moverse transparentemente de un *peer* a otro enmascarando la movilidad de un recurso o la falla de un servicio. Más aún, la infraestructura *peer to peer* provee al sistema con útiles propiedades tales como tolerancia a fallas, la posibilidad de usar infraestructura de llave pública para implementar transacciones seguras, y escalabilidad.

En ECO los archivos compartidos son manejados por el repositorio de archivos *peer to peer* llamado ECO-FR. En ECO-FR, los archivos compartidos son descritos por un documento XML que es también usado para anunciar la presencia del documento en el sistema. En esos anuncios, el ECO-FR almacena el nombre del documento así como alguna otra meta-información tal como el nombre del propietario, una lista de la prioridad de los autores, la fecha de creación, la última fecha de modificación, palabras clave, si pertenece a un proyecto, etc. Esos datos están con la intención de ser usados en búsquedas más complejas. Para hacer frente a la movilidad y a las desconexiones, todas las operaciones de ECO-FR son idempotentes y las comunicaciones están basadas en *Jxta pipes*.

4.1.1 Requerimientos para sistemas de computación ubicuos para colaboración.

Una herramienta ubicua colaborativa debe permitir a los usuarios interactuar con el entorno (artefactos y humanos) para llevar a cabo el trabajo colaborativo. Para hacer eso, identificamos siete características importantes que debe reunir la implementación de una herramienta colaborativa ubicua: escalabilidad, acceso independiente a los recursos, asistencia para movilidad y desconectividad, percepción del contexto, mejoramiento de los mecanismos de seguridad, protocolos abiertos y tolerancia a fallas.

La escalabilidad es importante porque un crecimiento excesivo en el tiempo de respuesta podría limitar la utilización de un sistema colaborativo. Este incremento en el tiempo de respuesta podría ser causado por un crecimiento considerable en el número de usuarios y recursos compartidos administrados por un sistema colaborativo. En otro caso, los escenarios ubicuos descritos por *Marc Weiser* comúnmente incluyen cientos de computadores [4.4] interactuando por medios inalámbricos y redes fijas. Entonces, las arquitecturas deben estar diseñadas para permitir la inclusión dinámica de software y componentes de hardware en respuesta a un incremento en el número de usuarios y recursos compartidos.

De hecho, una herramienta colaborativa para ser ubicua no se debe atar al usuario a una computadora fija ni tener acceso a Internet como un prerequisite para el funcionamiento apropiado. El sistema debe permitir a los usuarios moverse físicamente mientras se usa y debe tener soporte para una gran variedad de dispositivos tales como PDA's, teléfonos celulares o computadoras portátiles. En otro de los casos, una herramienta colaborativa debe ser capaz de encontrar y interactuar dinámicamente con otros componentes ubicuos [4.5] y dispositivos colaborativos tales como displays públicos o proyectores. Por lo tanto, la infraestructura distribuida así como los principales servicios que implementa la aplicación deben soportar la utilización de diferentes tipos de redes (fijas, inalámbricas, *ad hoc*), manejar las posibles desconexiones, e implementar protocolos estándares que permitan el descubrimiento dinámico de los componentes ubicuos.

La *Percepción del Contexto* es una característica principal de una herramienta colaborativa ubicua. Una herramienta ubicua debe conocer el estado actual del entorno de los usuarios y presentar una vista resumida de los aspectos más relevantes. Más específicamente, debe hacer notar al usuario situaciones de interés tales como colaboraciones potenciales y el

estado localización física, lo que estoy usando, etc.) de los dispositivos compartidos. Por ejemplo, conocer qué documento compartido se está empezando a editar usando *display* o proyector, eso podría ser una buena evidencia de colaboración potencial.

Son necesarias las comunicaciones seguras, cuando los documentos compartidos son confidenciales y cuando se necesitan implementar mecanismos para asegurar que un autor no pudiera denegar una modificación hecha por él a un documento particular. Las violaciones de privacidad podrían también limitar la utilización de una herramienta colaborativa.

Finalmente, como en cualquier sistema distribuido, los cuellos de botella y las fallas de un solo punto deben ser reducidos. Así que es importante diseñar arquitecturas para soportar fallas en alguno de sus componentes sin afectar el sistema en su funcionamiento total.

4.1.2 Aspectos de Implementación

Como se mencionó en los capítulos anteriores, se propone usar los servicios *Peer to Peer* de los protocolos JXTA y JXTA para Java *microedition* [4.6], como se muestra en la figura 4.1, la principal ventaja de las redes *Peer to Peer* es la infraestructura de red virtual encima de *Internet* permitiendo a los *peers*, comunicarse, descubrirse y organizarse independientemente de la topología de la red física, atravesando *firewalls* o NAT's [4.3]. En ECO los servicios ECO-FR y ECO-DA así como ECO-IM y ECO-CE son todos implementados como *peers* en JXTA.

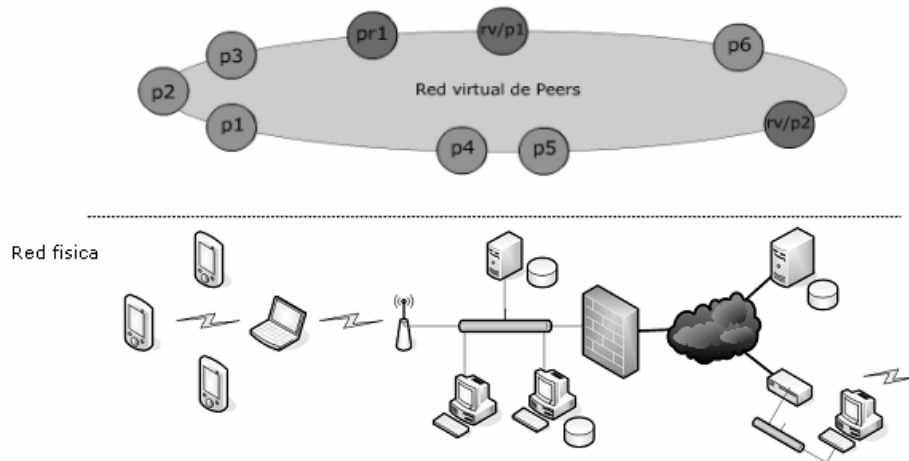


Figura 4.1 La red virtual del proyecto JXTA

4.1.2.1 ECO-Repositorio de archivos (FR)

El servicio de repositorios de archivos en ECO está compuesto por un conjunto variable de *peers* los cuales comúnmente son localizados en computadoras fijas (dadas sus características). Esos *peers* publican sus anuncios de sus servicios en una tabla *hash* distribuida implementada por un tipo especial de peer conocido como *peer rendezvous*. La cantidad de información que el anuncio contiene es una descripción del recurso o del

servicio que anuncian (en este caso el de un repositorio de archivos) y un identificador Jxta necesario para localizar el servicio. Cuando un nuevo ECO-FR es creado, como se muestra en el diagrama de secuencia de la figura 4.2, tiene primero que publicar su anuncio. Para interactuar con un *peer* repositorio de archivos, los *peers* ECO-IM y ECO-CE consultan a su *rendezvous* que ellos conocen para encontrar los anuncios de los ECO-FR's. Para eso, ECO-IM (cliente 1) envía una consulta a un *peer rendezvous* (RV1) buscando por un ECO-RF. RV1 verifica si tiene algún índice de ese anuncio, si no encuentra el índice, entonces propaga la consulta al próximo *rendezvous* (RV2). Cuando la consulta llega a RV2, RV2 encuentra un índice del anuncio y reenvía la consulta al *peer* ECO-FR1. Cuando ECO-FR1 recibe la consulta envía el anuncio al *peer* Cliente1. Cuando el cliente 1 recibe el anuncio, entonces se encuentra en condiciones de invocar el servicio implementado por ECO-FR1.

Para crear un nuevo documento compartido, el Cliente1 envía un documento XML al *peer* ECO-FR1. ECO-FR1 usará este documento como el anuncio del nuevo documento compartido. Así, ECO-FR1 pondrá el anuncio del archivo en una tabla *hash* distribuida para leer o escribir al documento creado, un cliente debe encontrar el anuncio del documento sobre la tabla *hash* distribuida usando un proceso similar al anteriormente descrito.

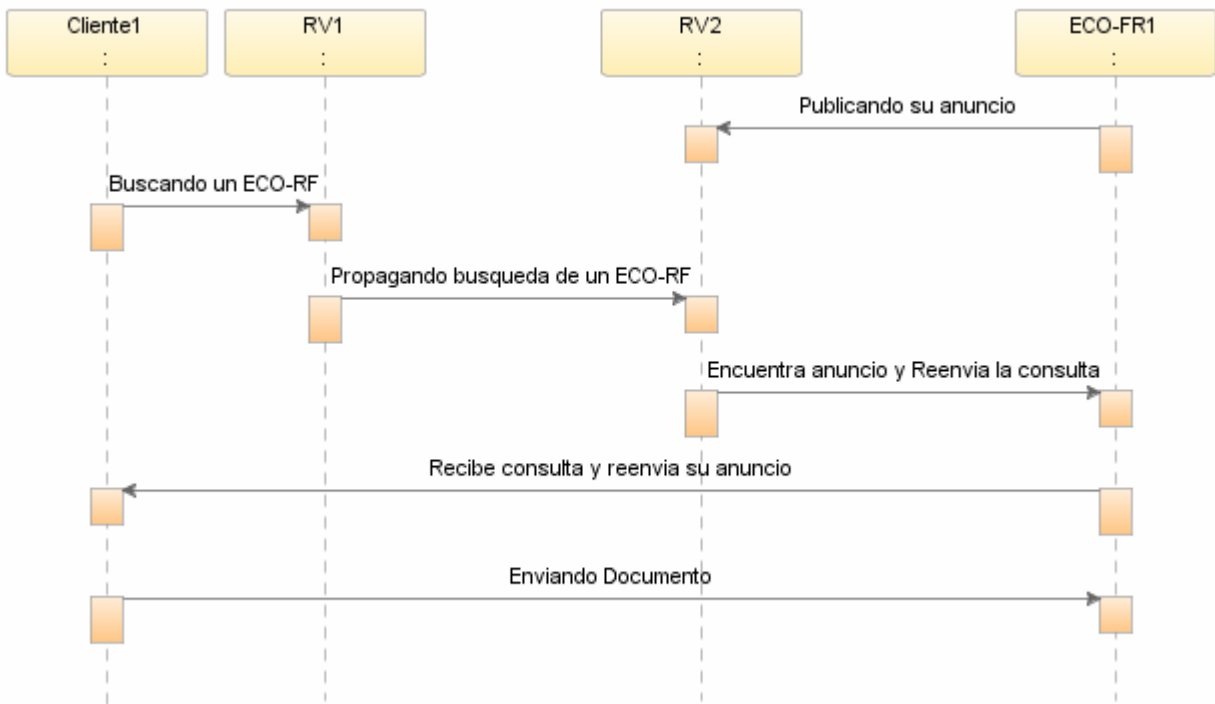


Figura 4.2. Diagrama de secuencia de la creación de un ECO-FR, la búsqueda de un ECO-FR para la publicación de un documento compartido

ECO-FR define tres simples operaciones sobre el documento compartido: Lectura Idempotente, Escritura Idempotente, Actualización Idempotente, donde no importa realizar dos o más operaciones consecutivas (Lectura, Escritura o Actualización) sobre el documento con parámetros idénticos, por que el ECO-FR siempre devolvería el mismo resultado (el mismo documento). Este tipo de operaciones puede facilitar la tolerancia a

fallas, porque el ECO-FR podría reenviar el documento más de una vez sin peligro de que sea duplicado incorrectamente, por lo tanto, facilitarían la implementación de las operaciones de edición colaborativas.

4.1.2.2 ECO - Servicio de percepción Distribuida (DA)

Es necesario mencionar que en el sistema ECO, estamos interesados en el desempeño de actividades sobre el documento compartido. Dejamos la percepción de presencia a servicios tradicionales como *Jabber* o *MS Messenger*. El servicio de percepción distribuida es implementado por el ECO-FR. Cuando un peer (cliente1) crea un archivo compartido, el ECO-FR1 busca por los anuncios de los coautores registrados en el documento XML usado para crear el documento compartido. Los anuncios de los autores contienen la información necesaria para contactar al ECO-IM que está actualmente usando un cliente en particular. Si un usuario no está actualmente conectado, el ECO-FR no encontrará su anuncio ya que todos los anuncios son publicados con un tiempo de vida antes de que expiren. Así, ECO-FR recolecta los anuncios de todos los coautores conectados y usando un *pipe* de propagación los mantiene informados (Diagrama de secuencias de la figura 4.3) de las actividades inicialmente hechas sobre el archivo compartido.

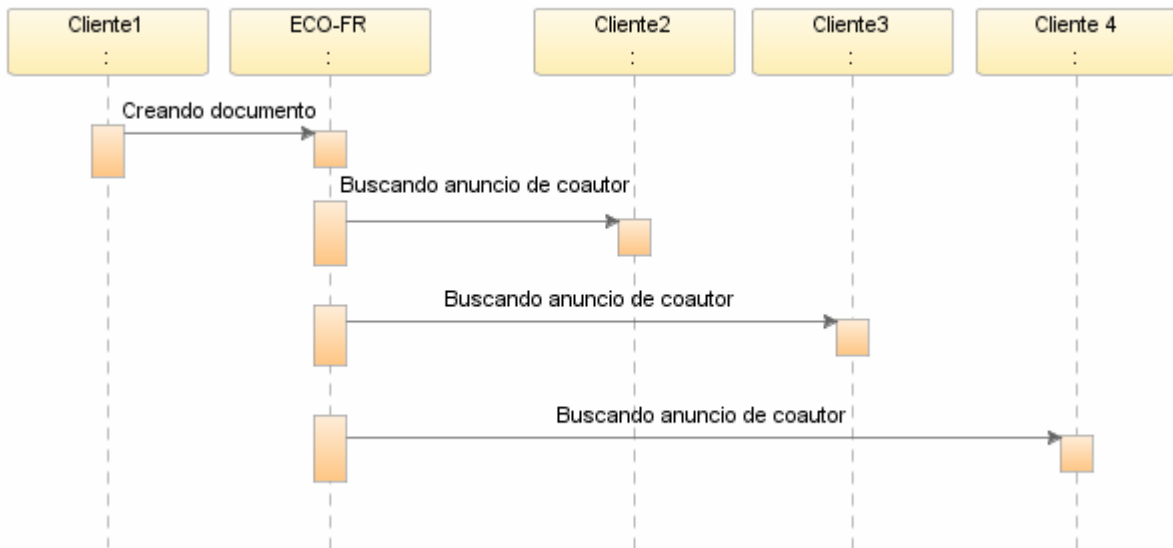


Figura 4.3 Diagrama de secuencias de la búsqueda de los coautores para mantenerlos informados del documento.

En esta situación, el Cliente2 podría estar interesado en participar en una sesión de edición colaborativa y envía una solicitud a ECO-FR1 (Diagrama de secuencia de la figura 4.4). Cuando ECO-FR1 recibe la solicitud, ECO-FR1 la propaga al cliente1 quien decide aceptar la solicitud o no. Si la solicitud es permitida ECO-FR1 lo notifica al cliente2 (Diagrama de secuencia de la figura 4.5) y la sesión colaborativa inicia.

Siguiendo este esquema, un ECO-IM podría estar recibiendo anuncios de información de n diferentes ECO-FR, si el usuario es coautor de n documentos compartidos localizados sobre n distintos ECO-FR's.

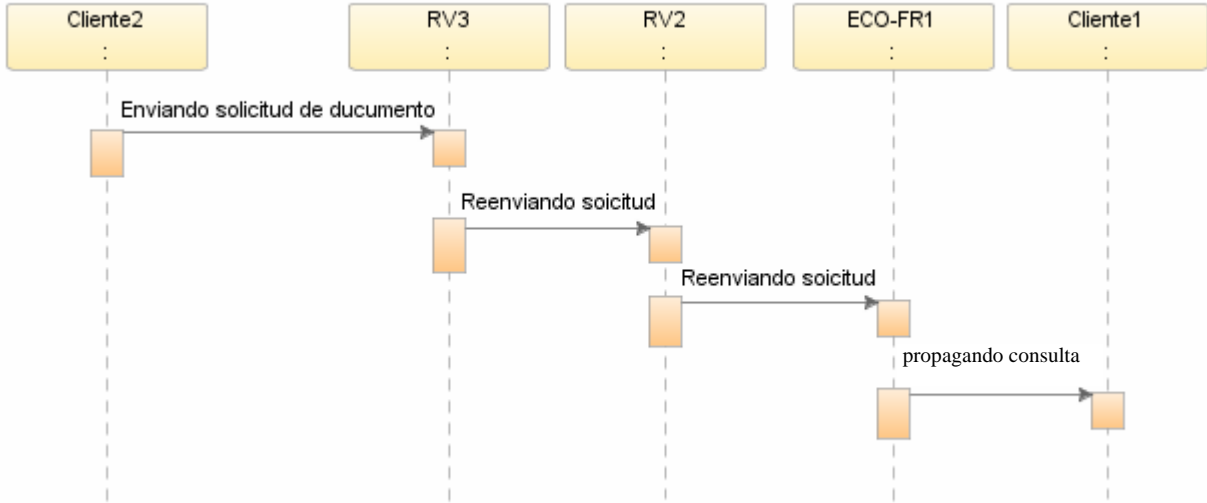


Figura 4.4. Diagrama de secuencias de la solicitud del documento para participar en una sesión colaborativa.

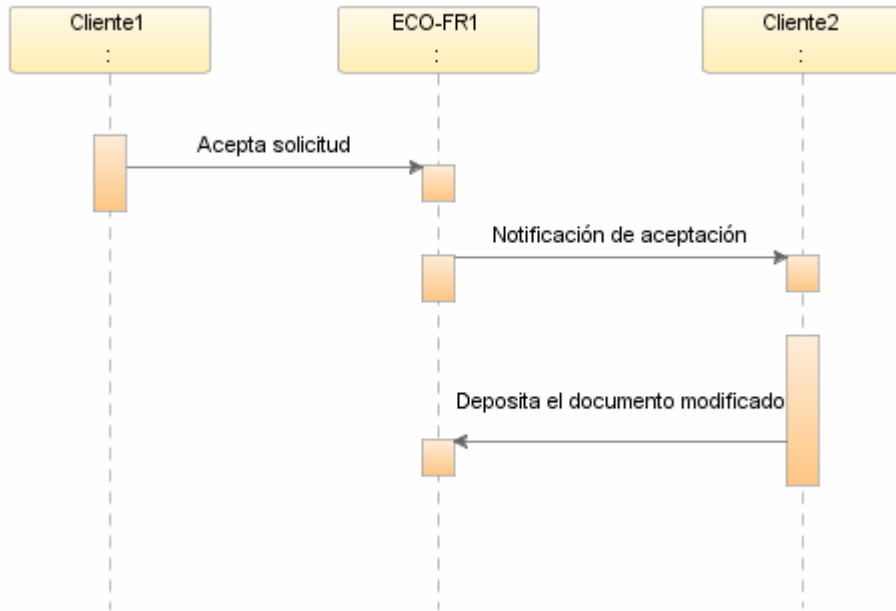


Figura 4.5. Diagrama de secuencias de la aceptación de la participación del Cliente2.

Los servicios de ECO-FR y ECO-DA son escalables porque en cualquier tiempo, más ECO-FR's pueden ser añadidos al sistema sin interferir con sus presentes operaciones. Los *peers* ECO-FR usan *pipes* como su principal método para intercambiar información con los demás *peers*, como se mencionó cuando se habló de las redes *peer* en el capítulo 3, los *pipes* proporcionan una ilusión de entradas virtuales y mensajes los cuales son transportados transparentemente e independientemente. Así un *peer* ECO-FR o un ECO-DA son capaces de comunicarse con cualquier otro *peer* sin importar si ambos están ejecutándose sobre una computadora de bolsillo (PDA) o una computadora de escritorio.

4.1.2.3 ECO Herramienta colaborativa

El sistema ECO está compuesto de dos herramientas colaborativas: el Mensajero Instantáneo (ECO-IM) y el Editor Colaborativo (ECO-CE). El principal propósito de ambos es penetrar naturalmente en el estilo de vida de los usuarios y ayudarles a llevar acabo sus actividades de manera más eficiente. Para hacer eso, ambas herramientas fueron diseñadas para modelar de la mejor manera posible la forma en la cual una comunidad de autores interactúen en una sesión tradicional colaborativa entre autores, donde dos o más personas interactúen a través de una computadora de trabajo sobre un documento común, pero presentando mejores facilidades para lograr una sesión colaborativa más productiva.



Figura 4.3. Interfaz principal de ECO (ECO-IM). a) ECO-IM corriendo sobre PDA. b) ECO-IM corriendo sobre una computadora de escritorio. En ECO el estado de un usuario está relacionado al documento compartido, es decir, si un usuario está editando o leyendo o no está en línea.

La primera de esas facilidades es la implementación de un entorno de trabajo ubicuo permitiendo a los usuarios estar concientes de la edición presente o sesión de edición colaborativa no obstante si el usuario está accediendo al sistema por medio de una computadora fija o móvil. Como se mencionó, se eligió dar percepción al usuario de “las actividades de edición sobre documentos relevantes” más que solo la percepción del usuario o la presencia del documento o disponibilidad, porque el conocer que un usuario o conjunto de usuarios están editando un documento es una evidencia mejor de un espacio colaborativo potencial. Esta información se muestra en una herramienta parecida a un mensajero instantáneo porque la mensajería instantánea ha demostrado sus capacidades

para penetrar en la vida diaria de sus usuarios y son ya una herramienta de colaboración comúnmente empleada. Como puede verse en la figura 4.3, la principal interfaz de ECO (ECO-IM) es similar a aquellos usados en mensajería instantánea comercial donde los usuarios ordenan sus contactos en diferentes grupos y esos sistemas muestran su estado presente (en línea, sin conexión, ocupado, etc.) de sus contactos. En ECO cada grupo corresponde a un documento compartido en particular y los miembros de ese grupo son los autores y revisores de ese documento en particular.

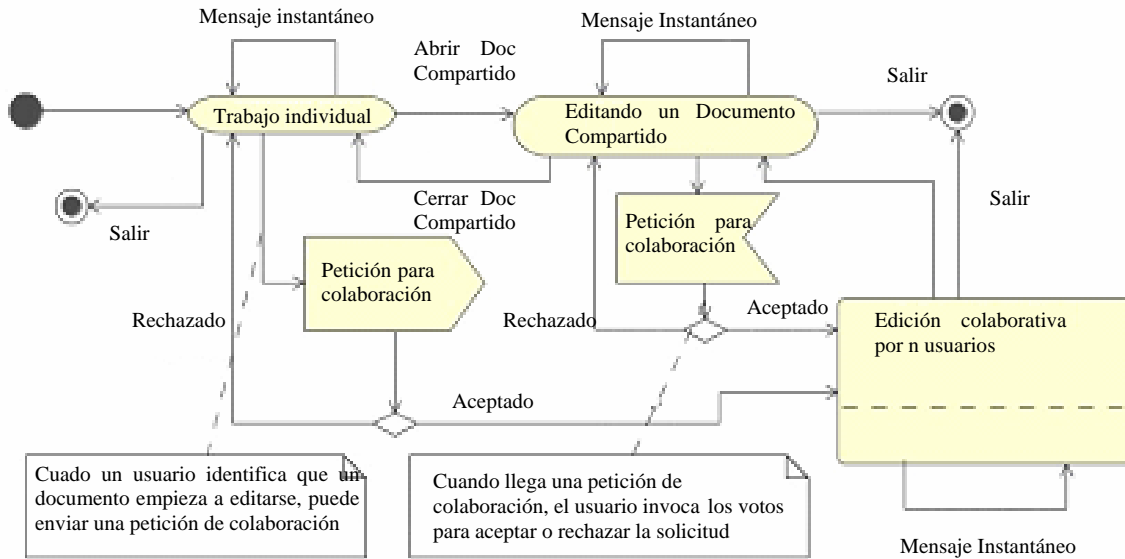


Figura 4.4. Diagrama de transiciones de actividades entre un trabajo individual y colaborativo en ECO

En la figura 4.4, se muestra un diagrama de actividades resumido del sistema ECO, en el cual se muestran cómo son llevadas a cabo las transiciones entre el trabajo individual y el trabajo colaborativo:

- Cuando un usuario está haciendo un trabajo individual (cualquier actividad que no esté relacionada con un documento compartido en ECO) él puede estar consciente de los otros usuarios de manera discreta. Así, puede identificar una colaboración potencial y decidir solicitar unirse a una sesión de autores colaborativa.
- Si la solicitud es permitida por los presentes participantes de la sesión colaborativa, el ECO-CE del usuario que hizo la solicitud será abierto.
- Para regresar a modo de trabajo individual, el autor solo necesita cerrar el ECO-CE.

En las siguientes secciones explicaremos con más detalle el comportamiento de las herramientas colaborativas de ECO.

4.1.2.4 El Espacio de Colaboración Potencial (ECO-IM)

ECO-IM es nuestro espacio de colaboración potencial y también el encargado del conjunto de actividades como son:

La Creación y eliminación de documentos compartidos. Cuando un documento compartido es creado, ECO-IM ayuda a los propietarios del documento a producir un documento XML con la siguiente información: nombre del documento compartido, propietario, fecha de creación, lista de prioridades de los autores, un conjunto de palabras clave, nombre del proyecto al cual el documento pertenece, y si se necesita mejorar la seguridad en certificados X509.V3. Esta información puede ser modificada en cualquier tiempo por el propietario del documento.

Una vez que el nuevo documento es creado, ECO-IM les pregunta a los usuarios propuestos si ellos desean participar como coautores del nuevo documento creado. Esos usuarios pueden aceptar o rechazar la invitación.

La creación del nuevo documento compartido es reflejada por la creación del nuevo grupo en ECO-IM. El grupo será integrado por aquellos usuarios quienes aceptaron la invitación por defecto. El nombre del grupo es el mismo que el nombre del documento compartido. Cuando un documento es eliminado del ECO-FR el grupo correspondiente es también eliminado de los ECO-IM's de los autores relacionados al documento. Solo el propietario del documento compartido tiene permiso para borrarlo del ECO-FR.

El Manejo de las transiciones de un trabajo individual a trabajo colaborativo. Cuando un usuario identifica una colaboración potencial, solo tiene que hacer doble clic en el icono que representa el documento para enviar una solicitud a participar en una sesión entre autores colaborativa. Como se ve en la figura 4.5, cuando una petición para colaboración llega, ECO-IM despliega una ventana preguntado a los participantes presentes de la sesión colaborativa si ellos están de acuerdo con la inclusión del nuevo participante. El ECO-IM colecta los votos para hacer la decisión final, ECO-IM utiliza las prioridades dadas por el usuario (cuando el documento fue creado) de hecho dan más peso los votos de los usuarios presentes con más prioridad. Como una alternativa, el presente usuario con más alta prioridad puede decidir solo si el nuevo coautor es aceptado en la sesión de edición colaborativa.

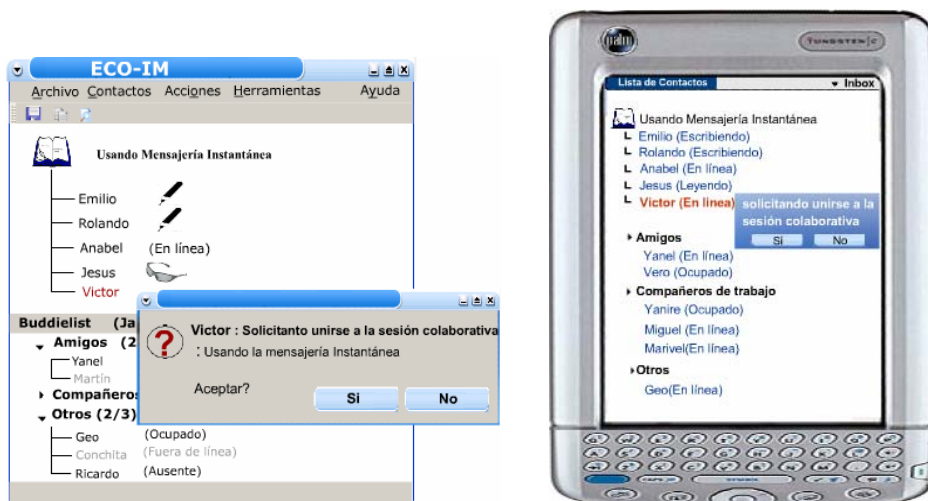


Figura 4.5. Implementación de una sesión de votos en ECO-IM para decidir si un nuevo miembro es admitido en la presente sesión colaborativa entre autores.

4.1.2.5 El Espacio de Colaboración Real (ECO-CE)

El espacio de colaboración real es implementado por el editor colaborativo ECO (ECO-CE). Como mencionamos anteriormente, se define un esquema similar al usado en una sesión entre autores no distribuido tradicional. En ECO-CE, las sesiones de edición colaborativas son dirigidas por los autores participantes actuales con más alta prioridad. En cualquier tiempo, el autor con más alta prioridad decide quiénes y cuándo pueden modificar el documento. Como se puede ver en la figura 4.6b, la interfaz de este autor es mejorada con unos botones de opciones que pueden ser usados para administrar las capacidades de acceso de los otros usuarios.

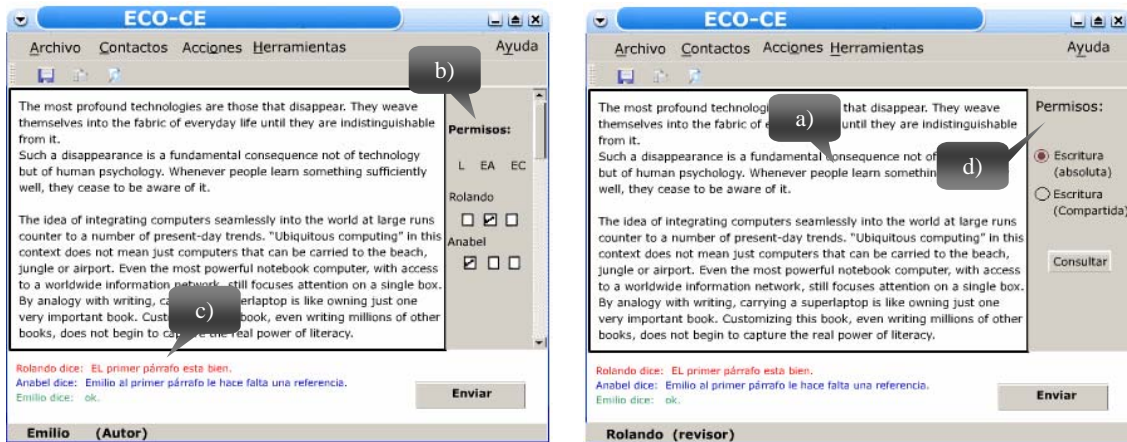


Figura 4.6. ECO-CE que implementa el espacio de colaboración real, (a) sección de edición, (b) sección del panel de administración en la interfaz del usuario con más alta prioridad, (c) sección de chat, (d) la interfaz usada por un usuario particular para preguntar por un permiso en particular sobre el documento.

Si el autor con más alta prioridad deja la presente sesión, el próximo autor con la más alta prioridad tomará el control de la sesión colaborativa. En la misma forma, si un colaborador con más alta prioridad llega, él o ella tomarán el control de la sesión colaborativa. Cuando un autor no tenga la más alta prioridad el panel mostrado en la figura 4.6b es ocultado y se despliega un panel para un autor regular (figura 4.6d). Un participante regular puede solicitar en cualquier momento por un derecho en particular sobre el documento compartido. Para hacer eso, el colaborador solo necesita activar un botón de opción etiquetado con el permiso que él está solicitando. Esto es mostrado en la figura 4.6d. Las solicitudes de permisos son desplegadas en el cuadro de mensajes de la sección de Chat (figura 4.6c) del presente coordinador. Este mensaje será desplegado periódicamente hasta que la solicitud sea contestada. Finalmente, en cualquier momento, los participantes de una sesión colaborativa pueden intercambiar ideas o comentarios por medio de una herramienta parecida al Chat que es desplegada en la interfaz. Esto se puede ver en la figura 4.6c.

Varios autores [4.7-4.14], han identificado la necesidad de herramientas de edición colaborativas y varios proyectos han sido desarrollados para asistir en las secciones de edición colaborativa. La colaboración entre autores requiere asistir en la coordinación entre las actividades entre autores, así como la flexibilidad para soportar los cambios dinámicos necesarios entre el grupo de edición. La coordinación de las actividades entre autores incluye la necesidad de estrategias de escritura para comunicar los cambios entre los coautores, así como proporcionar a los colaboradores información relacionada al estado presente del proceso en el grupo de edición. Los requerimientos de flexibilidad incluyen la necesidad de métodos simples para asistir a transiciones parecidas entre las estrategias de edición (secuencial, recíproca y paralela), entre las actividades (de edición, investigación, plan inicial, escritura), entre formas de comunicación síncronas y asíncronas y de espacios compartidos y privados [4.11]. Otro aspecto principal de CSCW es que la colaboración en la edición de un documento no necesariamente significa que los usuarios sean notificados de todas las actualizaciones o informados de todos los documentos compartidos. Entonces, reducimos la carga cognoscitiva de los usuarios por una mejor organización en las actividades de coordinación, filtrando información innecesaria y entregando la información importante podemos incrementar grandemente la eficacia de los sistemas de CSCW [4.15].

4.2 Resumen

En este capítulo abordamos la explicación de nuestra arquitectura, primeramente vimos la importancia de crear espacios de colaboración potencial para un grupo de personas involucradas en la edición de un documento, es decir, proporcionar un espacio donde los coautores puedan interactuar y puedan encontrar el documento, y a partir de ello dar la posibilidad de alcanzar un espacio de colaboración real. Para implementar una herramienta ubicua colaborativa como nuestro sistema, identificamos siete características importantes como son: la escalabilidad, acceso independiente a los recursos, movilidad, percepción del contexto, seguridad, protocolos abiertos y tolerancia a fallas, explicamos la importancia de cada uno de ellos y cómo los implementamos en el sistema. Para la implantación de nuestro sistema ECO, involucramos dos servicios importantes como lo son el servicio de repositorio de archivos y el servicio de percepción distribuida y con diagramas de secuencia vimos los pasos que hay que seguir para llegar a un espacio real donde se lleva a cabo la edición del documento, el servicio de repositorio de archivos es el encargado de administrar el documento y hacerlo llegar a las personas involucradas y el servicio de percepción distribuida es el encargado de conocer la presencia de los autores y coautores dentro del sistema. También hablamos de las herramientas colaborativas por las que está compuesto ECO, como son el mensajero instantáneo que da lugar al espacio potencial y el editor colaborativo que conlleva al espacio real de la edición del documento.

Después de haber explicado la arquitectura de nuestro sistema, de los servicios que utiliza y de las herramientas que se implementan en ECO, a continuación es necesario mencionar cómo es el diseño de las clases que forman nuestro sistema. Así también, creemos importante presentar las pruebas del funcionamiento de la aplicación describiendo un caso donde se lleva a cabo la revisión entre el autor y el revisor detallando cada uno de los pasos del proceso.

4.3 REFERENCIAS

- [4.1]. Baecker, R.M., D. Nastos, R. Posner, and K. L. Mawby. (1993). "The User Centered Interactive Design of Collaborative Writing Software". Proceedings of the ACM Conference on Human Factors in Computing Systems (INTERCHI'93), Amsterdam, The Netherlands, 24-29 April. 399-405 p.
- [4.2]. Morán, A., et-al, "On the design of potential collaboration spaces". International Journal of Computer Applications in Technology, Vol. 19, Nos. ¾, 2004, pp. 184-194.
- [4.3]. Traversat, B., et-al. Project JXTA 2.0 Super-Peer Virtual Network. <http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf>
- [4.4]. Weiser, M. The Computer for the 21st Century. Scientific American, Vol. 265, No. 3, September 1991.
- [4.5]. Kindberg, T. and Fox, A. System Software for Ubiquitous Computing. IEEE Pervasive Computing, Vol. 1, No. 1, January-March 2002.
- [4.6]. JXTA for J2ME. <http://jxme.jxta.org>
- [4.7]. Favela, J., & Ruiz, D. (2001). Collaborative Authoring and Reviewing over the Internet. WebNet Journal: Internet Technologies, Applications & Issues 3(3), 26-34.
- [4.8]. Robert S. Fish, Robert E. Kraut, Mary –D.P. Leland and Michel Cohen, "Quilt – A Collaborative Tool for Cooperative Writing", Proc. COIS'88 Office Information System (Palo Alto, Ca, March 1988).
- [4.9]. Neuwirth, C., Kaufer, D., Chandhok, R. And Morris, J. (1990). Issues in the design of computer support for co-authoring and commenting. Proceedings of the third conference on CSCW'90. Baltimore, MD: ACM Press, pp. 183-195.
- [4.10]. J. Favela, C. Navarro, M. Rodriguez, "Extending Instant Messaging to Support Spontaneous Interactions in Ad-hoc Networks," in Proceedings of ACM 2002 Conference on Computer Supported Cooperative Work (CSCW '02), New Orleans, Louisiana, 2002.
- [4.11]. Moran, A.L., Favela, J., Martinez, A.M., and Decouchant, D.. Document presence notification services for collaborative writing. Proceedings CRIWG-2001, 7th International Workshop on Groupware, IEEE Computer Society, Darmstadt, Germany, pp. 12-1330.

- [4.12]. Dominique Decouchant, Jesús Favela, Ana María Martínez Enríquez, “PIÑAS: A Middleware for Web Distributed Cooperative Authoring”. SAINT 2001: 187-194 (piñas)
- [4.13]. Malone, T.W., Grant, K.R., Lai, K.Y., Rao, R., Rosenblitt, D., "Semi Structured Messages are Surprisingly Useful for Computer-Supported Coordination," ACM Transactions on Office Information Systems, Vol. 5, No. 2, Apr. 1987, 115-131. (Lens)
- [4.14]. Clarence A. Ellis, Simon J. Gibbs, and Gail L. Rein. Design and use of a group editor. Proceedings of IFIP WG2.7 Working Conference on Engineering for Human Computer Communication, August 1989, pages 13--28, 1990.
- [4.15]. Prakash, A., Shim, H.S., and Lee J.H., Issues and Trade-offs in CSCW Systems, IEEE Transactions on Data and Knowledge Engineering , Jan.-Feb. 1999, Vol. 11, Issue 1, pp. 213-227.

- Diseño
- Implantación y pruebas



En la siguiente sección exponemos el diseño de ECO a partir de los diagramas de las clases principales. Así también, presentamos la implantación y pruebas de nuestra aplicación, indicando una serie de pantallas donde se demuestra el funcionamiento de la aplicación, tomamos un ejemplo práctico de colaboración entre un autor y un revisor.

5.1 DISEÑO

A continuación se presenta el diseño del sistema en lo referente a creación de la red P2P, así como a la edición y colaboración sobre el documento a través de diagramas de clases, haciendo referencia a las clases principales del sistema.

Primeramente se menciona la forma en que se construye la red JXTA, es decir, todo lo necesario para que un usuario se convierta en un peer colaborador dentro de la red, para esto, es necesario que el peer cuente con un ID que le servirá como identificador dentro de la red *peer to peer* y para identificarse dentro de la comunidad de colaboradores. El peer abre puertos para recibir los mensajes dentro de la red por medio de los *pipes* de comunicación. Cada uno de los participantes se une a un grupo de trabajo (*peer group*).

Después del proceso de conexión se continúa con la descripción de la forma en que los peer se encuentran y se envían mensajes. Como se mencionó en el capítulo de la arquitectura, ECO involucra un servicio de percepción distribuida donde cada uno de los participantes está consciente de la disponibilidad de un documento, así como de la disponibilidad de un colaborador, es necesario preguntar constantemente por el estatus de los demás colaboradores dentro de la comunidad peer, para esto se forman diferentes tipos de mensajes que pueden ser de utilidad para conocer el estatus de los participantes, así como de las acciones hechas al documento.

Dos clases importantes son la clase *Buddy* y la clase *ProfileAdvertisement*, la primera administra todo lo referente a la identificación del peer colaborador y la segunda crea un anuncio con la información del peer y lo distribuye en la red para que sea encontrado y sea tratado como uno más dentro del grupo de trabajo.

Finalmente, hacemos referencia a la clase *GuiFile* la cual conlleva todo lo referente al documento (nombre, peers involucrados, prioridades, etc) y la clase *ChatGuiEdition*, que es utilizada al momento de hacer la edición colaborativa sobre el documento.

El diagrama de la figura 5.1 nos muestra la definición de la clase *jxtaConnection*, la cual nos permite hacer la conexión a la red JXTA. En la misma figura se puede observar que para hacer un objeto de tipo conexión usamos la clase *jxtaConnection* donde implementamos los métodos *setPeerGroup()* y *getPeerGroup()* para unirse al *peerGroup*. Se implementan también los métodos *setPipeAdvertisement()* y *getPipeAdvertisement()* para que los peers dentro de la red conozcan los pipes para comunicarse.

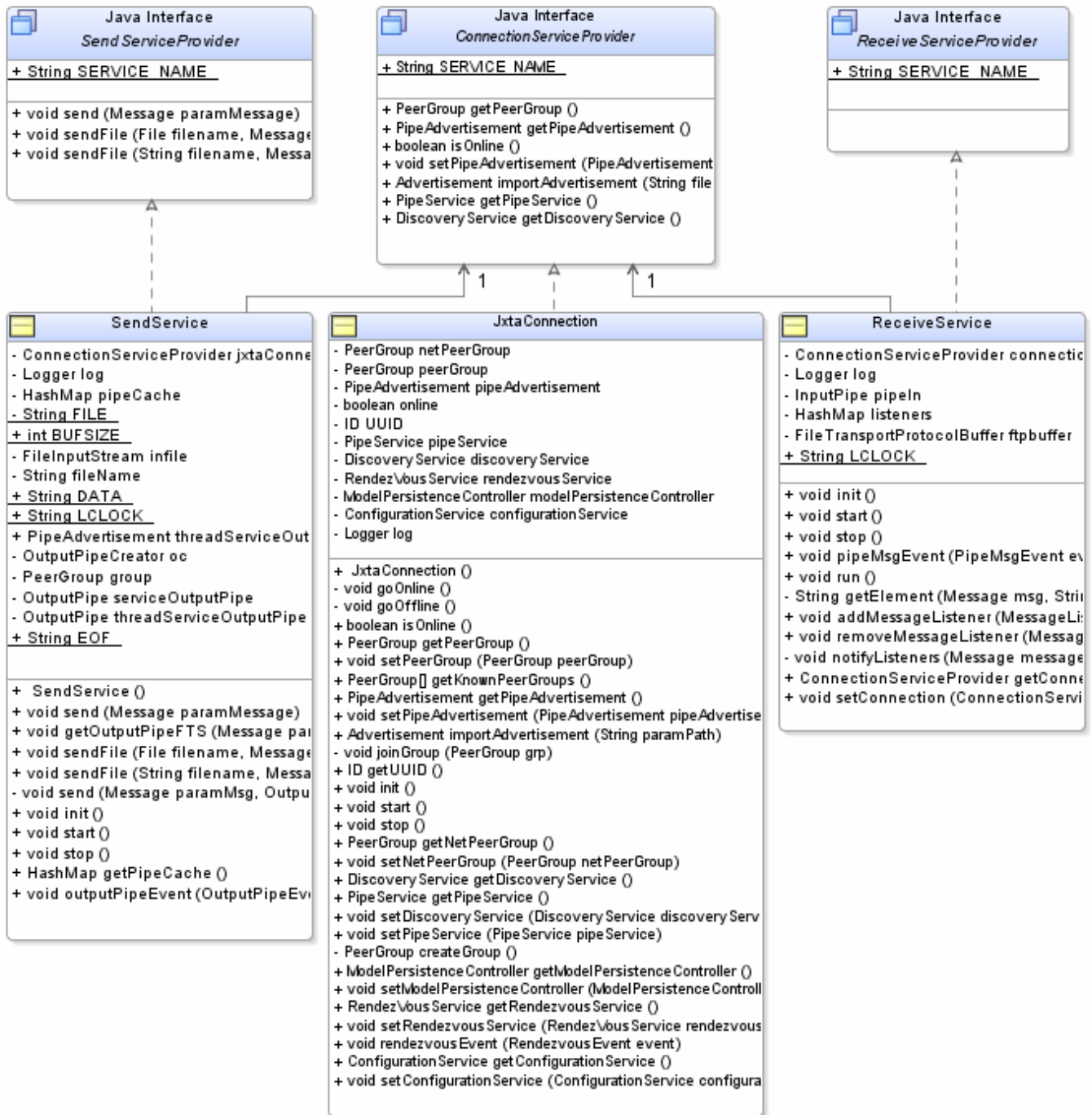


Figura 5.1. Diagrama de clase de la conexión a la red JXTA

En el diagrama de clases de la figura 5.1, también podemos encontrar las clases *SendService* y *ReceiveService* de las cuales podemos crear objetos con la función de enviar mensajes sobre la red P2P, cada vez que queremos que los otros peers conozcan de nuestra presencia, necesitamos mandar un mensaje con nuestro estatus. Cuando deseamos enviar el documento se necesita primeramente crear un *pipe* de salida con *getOutputPipeFTS()* con una variable de tipo *Message* y un parámetro, el cual contiene el anuncio del peer al cual queremos mandarle el documento de tal forma que con su anuncio formamos nuestro pipe de salida. Para enviar el documento utilizamos el método *sendFile(filename,message)* con el nombre del archivo como parámetro, así como con la referencia al pipe de salida. Por su parte los objetos de tipo *ReceiveService* son utilizados por los peers al momento que reciben mensajes a través de su pipe de entrada (*pipeMsgEvent(PipeMsgEvent)*).

En la figura 2 podemos encontrarnos con el diagrama de las clases que participan para formar los mensajes de comunicación. Cuando se extiende de la clase *Message* heredamos los métodos de identificación como son los métodos de: *getDate()* para conocer la fecha del mensaje, *getType()* el tipo del documento, *getSender()* el *peer* quien envía el mensaje, *getReceiver()* el *peer* hacia quien va dirigido el mensaje. Por ejemplo, cuando un repositorio recibe un mensaje relacionado a un documento (*FileMessage*) puede obtener información suficiente para su administración, como por ejemplo, la lista de usuarios relacionados al documento *resultUsr*, la lista de prioridades sobre el documento *resultPri*, El nombre del archivo del mensaje *FileName* así también, el contenido del documento *contentFromXML(String)*.

De la clase *SearchMessage*, podemos obtener suficiente información para buscar un *peer* dentro de nuestra comunidad de *peers*.

De la clase *StatusPingMessage* podemos crear un objeto para mandar un mensaje XML para dar a conocer nuestra presencia, para esto heredamos de la clase *StatusMessage* para conocer primero nuestro estatus. Para estar en condiciones de poder contestar a los mensajes de presencia creamos un objeto de tipo *StatusPongMessage*.

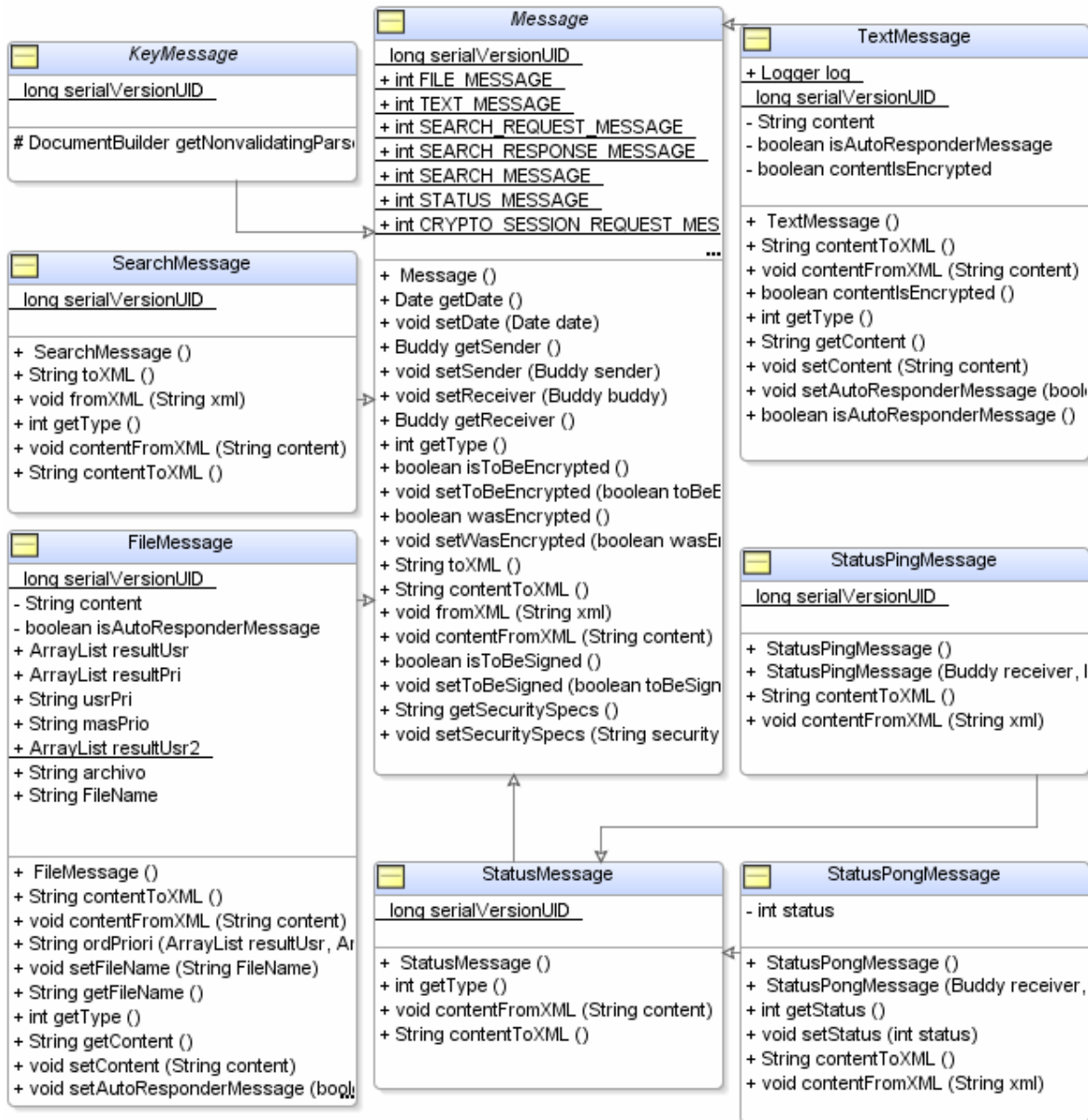


Figura 5.2. Diagrama de clases de los mensajes para la comunicación en la red JXTA.



Figura 5.3. Pantalla para la identificación del nuevo peer.

En la figura 5.4 se muestra la definición de la clase Buddy, esta clase contiene la definición de un peer dentro de la red JXTA, en ella se observa que para que un peer pueda ser un participante más sobre el documento, el participante tiene que introducir cierta información para que los otros puedan identificarlo. La información necesaria se exhibe en la figura 5.3, como lo es el nombre (*name*), un nombre identificador en la red (*nickName*) y un *email*, estos son los datos que se deben llenar cuando un nuevo peer se une a la red por primera vez y automáticamente se le crea un identificador uuid único.

Para guardar los datos de identificación utilizamos la clase *Buddy*, la clase *Buddy* además es la que contiene el estatus actual del peer dentro de la red *getStatus()* para que el peer se pueda dar a conocer con los demás peers, crear un anuncio de Pipe para que los demás puedan encontrarlo y así le puedan mandar mensajes de identificación. Para esto utiliza el *setPipeAdvertisement()*.

```

class Buddy
{
    long serialVersionUID
    + int WRITING
    + int READING
    + int ONLINE
    + int ABSENT
    + int BUSY
    + int OFFLINE
    - String nick
    - String name
    - String email
    - String uuid
    - int status
    - AdvertisementHolder pipeAdvertiser
    - TextMessageBuffer textMessageBuffer
    - long profileTimestamp
    - Vector buddyListenerList

    + Buddy ()
    + Vector getBuddyListenerList ()
    + String getNick ()
    + void setNick (String nick)
    + String getName ()
    + void setName (String name)
    + String getEmail ()
    + void setEmail (String email)
    + String getUuid ()
    + void setUuid (String uuid)
    + int getStatus ()
    + void setStatus (int status)
    + PipeAdvertisement getPipeAdvertisement ()
    + void setPipeAdvertisement (PipeAdvertisement pipeAdvertisement)
    + TextMessageBuffer getTextMessageBuffer ()
    + void setTextMessageBuffer (TextMessageBuffer textMessageBuffer)
    + boolean isOffline ()
    + long getProfileTimestamp ()
    + void setProfileTimestamp (long profileTimestamp)
    + void addBuddyListener (BuddyListener buddyListener)
    + void removeBuddyListener (BuddyListener buddyListener)
    # void emitStatusChanged (int oldStatus)
    # void emitProfileChanged ()
}
    
```

Figura 5.4. Definición de la clase Buddy.

Para crear un anuncio de *pipe* utiliza la clase *ProfileAdvertisement* donde guarda la información necesaria para que los demás *peers* puedan encontrarlo; la información de identificación es la misma que el usuario capturó la primera vez que entró al sistema (ver figura 5.3), como lo es el nombre, el *nickname*, el *email*, así como el ID de identificación del peer.

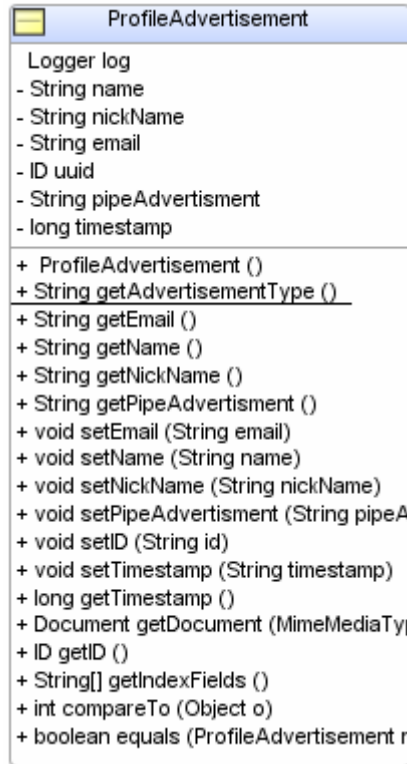


Figura 5.5. Definición de la clase ProfileAdvertisement.

En la figura 5.5, también podemos encontrar un *setTimestamp()* el cual contiene el tiempo que se desea que el anuncio esté publicado en la red. Aunque los anuncios dentro de las redes tienen un *Timestamp* asignado, en esta aplicación se consideró modificar el *Timestamp* de los anuncios de mensajes de identificación.

Cuando se crea un nuevo documento el propietario del mismo tiene que llenar cierta información (ver figura 5.6) para que un Repositorio lo pueda administrar, en la figura 5.7 se puede ver la definición de la clase *GuiFile* que obtiene toda la información de administración sobre el documento.

El propietario obtiene el nombre del documento *getFileName()*, la lista de peers a los que va dirigido el documento *saveUsrList()*, la lista de prioridades de los peers sobre el documento *saveUsrPrio()*, si existen algunas palabras reservadas que los peers puedan usar para encontrar un documento en particular sobre la red *saveWordRes()*. Después de que se llena la información de administración del documento se busca el identificador de un repositorio *getUuidRep()*, cuando se encuentra el *uuid* del repositorio se le manda un archivo XML con la información de administración recaudada *sendFileMessage()* y por ultimo se envía el documento físico al repositorio *sendFile()*.



Figura 5.6. Información necesaria para la administración de un nuevo documento

Cuando a un peer le llega el documento puede manipularlo dependiendo de la prioridad que le fue asignada por el propietario, para esto crea un objeto de la clase ChatGuiEdition() de la figura 5.8 con los métodos de *openFile()*, *readTheFile()* y *closeFile()*, dependiendo de las modificaciones que pretenda hacer.

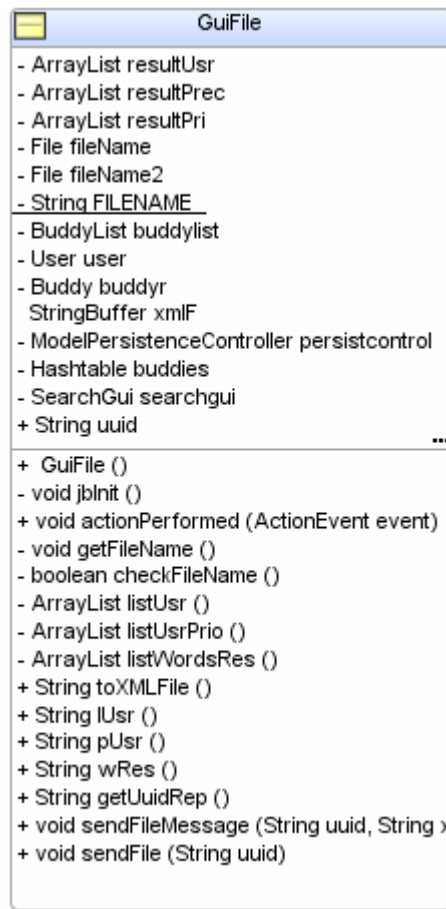


Figura 5.7. Definición de la clase GuiFile.

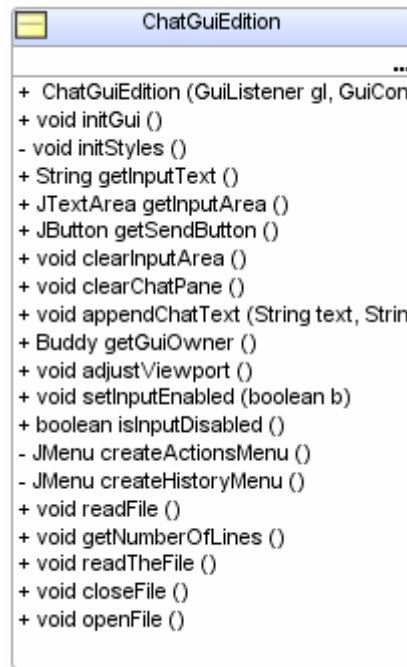


Figura 5.8. Definición de la clase ChatGuiEdition.

Como se mencionó al principio del capítulo, en esta sección solo se consideraron las clases más importantes. Como se puede ver ECO implementa las clases principales y necesarias para formar un sistema bajo una arquitectura puramente P2P, dado que cada uno de los colaboradores dentro del grupo de trabajo es un peer dentro de la red *Peer to Peer*. Así también, los repositorios de documentos son un peer que se comunican con los otros peers para elaborar el documento colaborativamente.

Así también se consideró mencionar en este capítulo los métodos más importantes que nos permiten llevar a cabo la colaboración sobre el documento entre los colaboradores, principalmente desde que se crea el documento hasta la edición colaborativa.

5.2 IMPLANTACION Y PRUEBAS

En esta sección se explicará la implementación de ECO, una herramienta de edición colaborativa, así como el funcionamiento de cada una de las partes de la herramienta. Hablaremos primeramente de la forma en que se inicia un documento y cómo es que llega el documento de edición a los colaboradores participantes. Se explicará cómo es la administración del documento así como también la forma en que los autores y revisores participan en la herramienta.

Primeramente cualquier autor puede iniciar editando un documento, y ponerlo a disposición de los revisores que considere que tengan un interés sobre el documento. Aunque también se puede dar el caso en que solo se cree el documento en blanco y así se mande a un repositorio (ECO-R) para que todos los involucrados en el documento desarrollen colaborativamente el documento.

La figura 5.9 muestra la interfaz principal (ECO-IM) de los colaboradores la cual es muy parecida a la de cualquier interfaz de mensajería instantánea, aunque en este caso no aparecen una lista de contactos esto es porque la lista de contactos se forma en base a un archivo compartido y la lista de contactos son los colaboradores del documento.

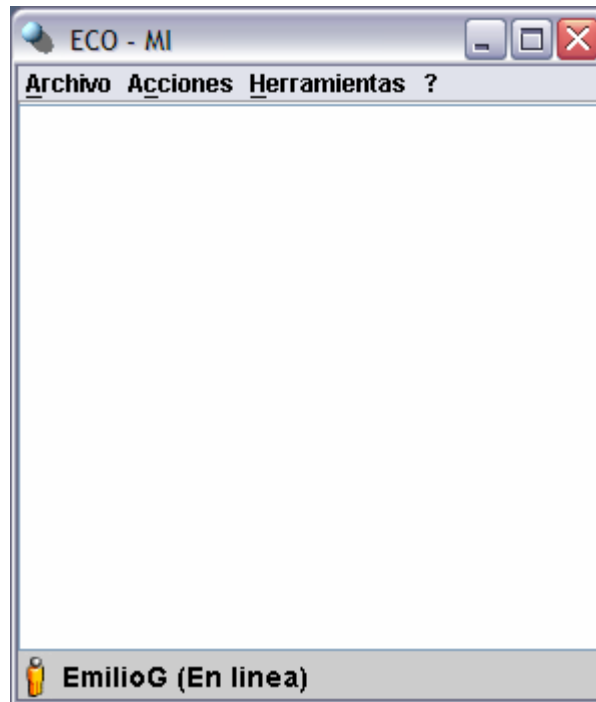


Figura 5.9. Interfaz de colaboración entre los miembros de la red P2P

5.2.1 Creando un documento de colaboración

Cualquier colaborador puede iniciar un documento, éste puede ser un documento en blanco o un documento que ya contiene algunas líneas. Cuando creamos un documento en ECO un cuadro de diálogo se abre, donde podemos elegir el documento que se desea compartir. Para abrir un documento se da un clic sobre el botón “ABRIR ARCHIVO” de la figura 5.10.



Figura 5.10. Interfaz para crear un documento de colaboración

Cuando damos clic “Abrir archivo” se abre un cuadro de diálogo (ver figura 5.11) donde podemos elegir el documento editado o nuevo. Aquí se podrá elegir cualquier documento que se encuentre en la máquina del autor. Hasta el momento solo la herramienta está pensada para trabajar con archivos planos.

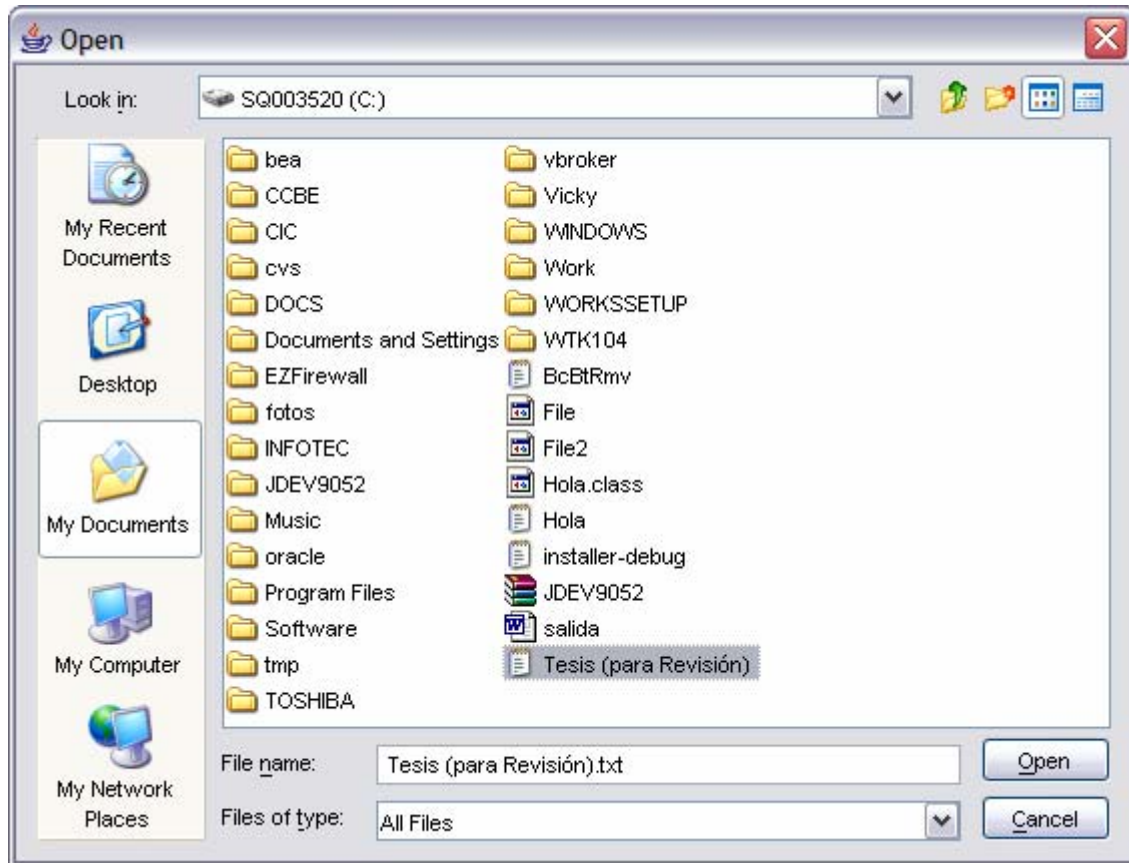


Figura 5.11. Pantalla para elegir un documento para la colaboración

Una vez que sabemos cuál es el archivo que queremos compartir, agregamos los *nicknames* de las personas con las que queremos compartir nuestro documento así como también la prioridad que tienen sobre el documento, toda esta información se guarda en un archivo XML de administración. Se puede agregar también palabras clave que puedan ser de utilidad en la administración de los documentos.

Podemos agregar también un certificado de seguridad en la administración del documento o en la etapa de edición.

5.2.2 Enviar el documento al repositorio de Archivos

Una vez que tenemos todo lo necesario para la administración damos clic sobre el botón ENVIAR, e internamente se buscará un repositorio donde el documento pueda residir. El repositorio de archivos se obtiene de la red P2P de los *peers* que tienen la responsabilidad de ser repositorios de documentos. Una vez que se encuentra el repositorio se le manda el documento. Cuando llega el documento al repositorio, el repositorio abre el archivo de administración XML para buscar la información de administración del documento (es decir, quién es el autor de documento, a quiénes va dirigido el documento y cuáles son las prioridades de cada uno de los involucrados en el documento). Cuando encuentra la información de administración el repositorio busca en la red a cada uno de los *peers* de la

lista de personas a quienes va dirigido el documento. Cuando encuentra un peer que representa a la persona involucrada se envía una copia del documento.

5.2.3 Abrir el documento en WS

Cuando le llega el documento a un representante del grupo de colaboradores, en su interfaz le aparece un icono que representa al documento con el nombre del documento y debajo del documento una lista con todos los involucrados en el documento; al lado de cada uno de los colaboradores aparece su estatus dentro del grupo de colaboración. Este estatus puede ser:

5.2.3.1 En línea

Indica que el colaborador está activo y en cualquier momento puede iniciar una edición colaborativa. Así cualquier colaborador al ver a un compañero en línea tiene la oportunidad de enviar un mensaje para que inicien una actividad de edición colaborativa sobre el documento debido a que se presenta una oportunidad potencial de colaboración.

5.2.3.2 Inactivo

Indica que en esos momentos no se encuentra el colaborador, pero tenemos presente su estatus y podemos saber cuándo entra al grupo de colaboración y así poder considerarlo en una sección de colaboración.

5.2.3.3 Editando

Su estatus indica que es la persona que está haciendo cambios sobre el documento ya que cuenta con la aprobación del grupo y tiene el control total sobre el documento. Esta persona puede hacer los cambios que considere necesarios aunque los demás pueden aconsejarlo por medio de una pantalla de *chat* que se encuentra debajo del documento. Solo la persona que edita puede ver los cambios que va haciendo al documento. Para que los cambios realizados puedan ser vistos por los demás colaboradores tiene que apretar el botón ENVIAR DOCUMENTO, así los cambios sobre el documento son enviados al repositorio y el repositorio lo redirige a los demás participantes dentro de la red P2 para mantenerlos actualizados.

5.2.3.4 En Lectura

Indica que el usuario solo está revisando los cambios realizados por la persona que está editando, por medio de una pantalla que le muestra el avance del documento. Ésta no tiene ningún control sobre el documento, su pantalla se actualiza cada vez que existe una modificación sobre el documento.

Como se mencionó en los objetivos del capítulo 0, la colaboración también se podrá hacer desde dispositivos móviles, para esto se envía el documento hacia un dispositivo móvil que pertenezca a la red de colaboradores. Esto se logra gracias a que el repositorio envía el

documento al colaborador móvil en cuanto se da cuenta de que acaba de integrarse a la red P2P.

Cuando el documento llega al dispositivo móvil, el documento es abierto automáticamente para que el colaborador tenga un seguimiento real de las modificaciones hechas en el documento, por lo tanto, los cambios del documento se van reflejando en la pantalla del dispositivo móvil.

5.2.4 Las pruebas realizadas

Tomamos dos colaboradores, una persona que inicia el documento y otra que participa colaborativamente con la edición del documento. Inicialmente un autor crea un documento con algunas líneas de texto, después lo manda hacia un repositorio para que le pueda llegar al otro colaborador y pueda abrir el documento y ver los cambios sobre el documento. EmilioG (Emilio Gutiérrez) es quien representa al autor y a partir de su interfaz de chat inicia el documento. Para iniciar el documento primeramente carga un archivo (ver figura 5.12), después Emilio asigna las personas que participarán en el documento, así también, asigna la prioridad sobre el documento, en este caso asignamos a RolandoM (Rolando Menchaca) como el otro colaborador (La persona quien revisará el documento). A Rolando se le asignará la prioridad número 2 y a Emilio la prioridad 1.



Figura 5.12. Pantalla de propiedades sobre el documento.

Antes de que se mande el archivo aparece un cuadro de diálogo (ver figura 5.13) donde se presenta un resumen de las propiedades que tecleó el autor del archivo, presentando cada uno de los participantes en el documento en este caso a EmilioG con la prioridad 1. Esto quiere decir que él tendrá el control total sobre el documento, en segundo lugar aparece

RolandoM quien representa un revisor del documento dado que tiene la prioridad 2. Esto quiere decir que inicialmente solo podrá llevar un seguimiento de lectura sobre el documento, es decir, solamente podrá ver los cambios que realiza EmilioG .

En el mismo cuadro de diálogo aparece el botón OK, dar clic sobre el botón OK significa aceptar los datos y entonces se envía el documento con el archivo XML que internamente se creó con toda la información necesaria para la administración del documento por parte del repositorio



Figura 5.13. Pantalla de confirmación de los datos de administración.

Luego Emilio manda el documento al repositorio, en nuestro ejemplo encontramos un repositorio llamado REPOSITORIO (como lo muestra la figura 5.14) que fue el peer repositorio que se encontró en la red P2P. Después el repositorio envía el documento a Rolando y Emilio que son las personas a quienes va dirigido. Entonces en la pantalla de ECO_IM de Emilio y Rolando aparece el documento para que cualquiera pueda abrirlo en cualquier momento.

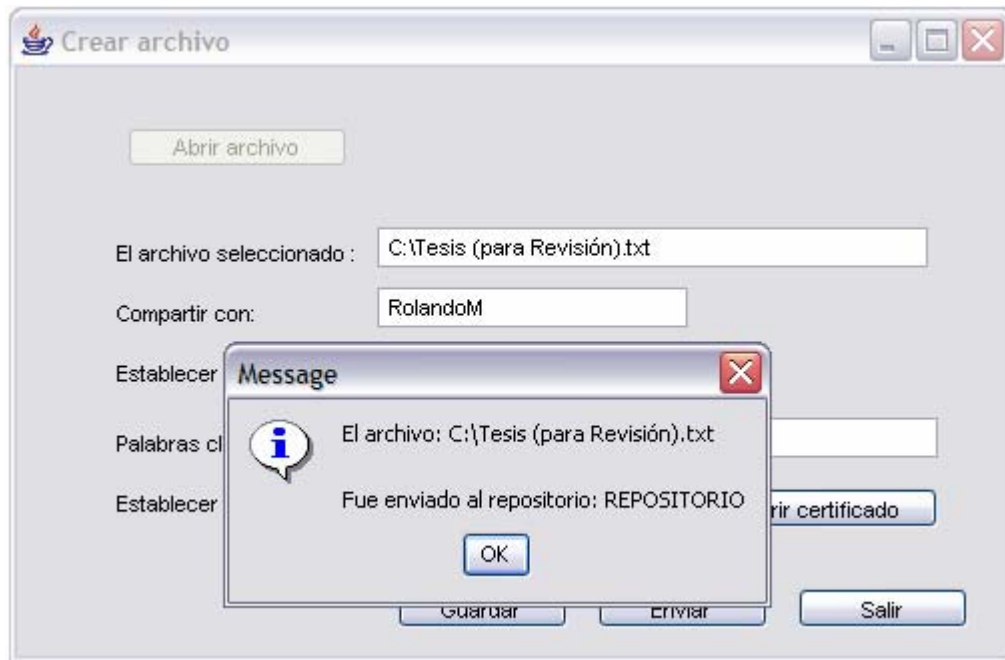


Figura 5.14. Cuadro de diálogo donde indica que encontró un Repositorio para el archivo.

Después de que llega el documento al repositorio éste consulta el archivo XML de propiedades para ver a quiénes va dirigido, entonces los busca y reenvía el documento a cada uno de los colaboradores.

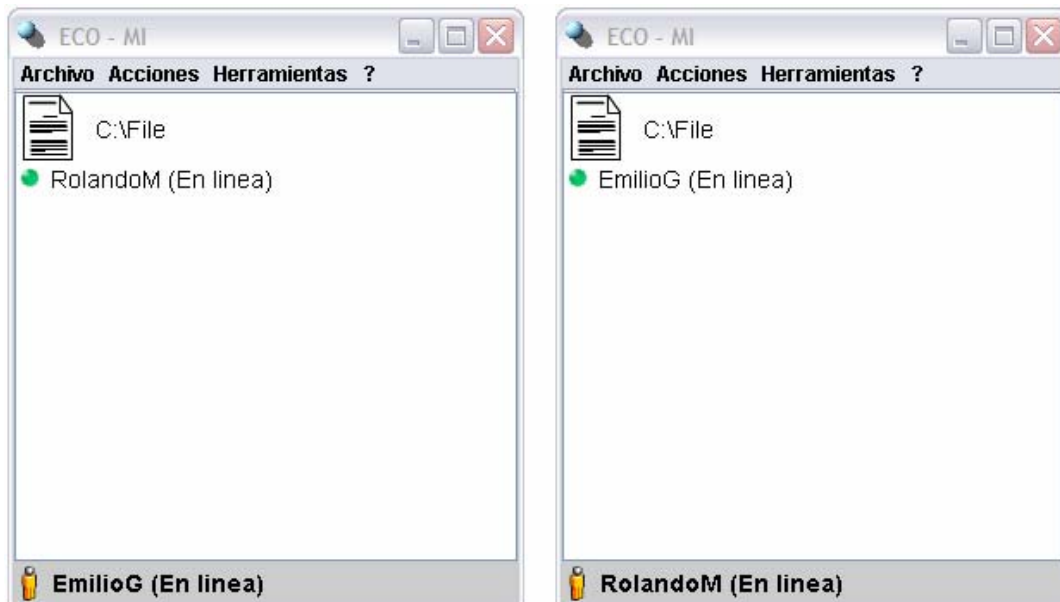


Figura 5.15. Pantalla de los colaboradores después de que les llega el documento.

Cuando llega el documento a los dos participantes aparece un icono que representa al documento así como el nombre del documento y debajo del documento aparece la lista de contactos, en este caso en la pantalla de EmilioG aparece el nombre del revisor del

documento RolandoM; a su vez en RolandoM aparece el nombre del autor del documento EmilioG, como lo muestra la figura 5.15.

Entonces EmilioG abre el documento, en este caso tiene la prioridad más alta y entonces EmilioG tiene la posibilidad de abrir y editar el documento. Para abrir el documento solo tiene que dar doble clic al icono del documento; cuando realiza esta operación le aparece la pantalla de la figura 5.16, en esta pantalla primeramente aparece un cuadro de edición donde puede hacer cualquiera de las siguientes operaciones:

- Agregar nuevas líneas al documento, en cualquier parte del documento.
- Modificar alguna palabra, modificar párrafo del documento.
- Borrar en cualquier parte del documento.

Cuando se hace alguna modificación sobre el archivo, el cambio realizado no se ve reflejado instantáneamente en la pantalla de los demás colaboradores, sino hasta que EmilioG le da clic en el botón “Enviar documento”, esto hace que el archivo modificado sea enviado al Repositorio y éste los reenvíe a los nuevos participantes.

En esta misma pantalla, en la parte derecha, aparecen unos *checkbox* donde el usuario puede decidir pasar el control a otro colaborador. Para saber quiénes son los otros colaboradores, en la parte de en medio aparece un *combobox* con la lista de colaboradores que pueden participar en el trabajo de edición colaborativa, entonces para asignar un permiso primeramente selecciona al colaborador y después decide qué permiso asignar.

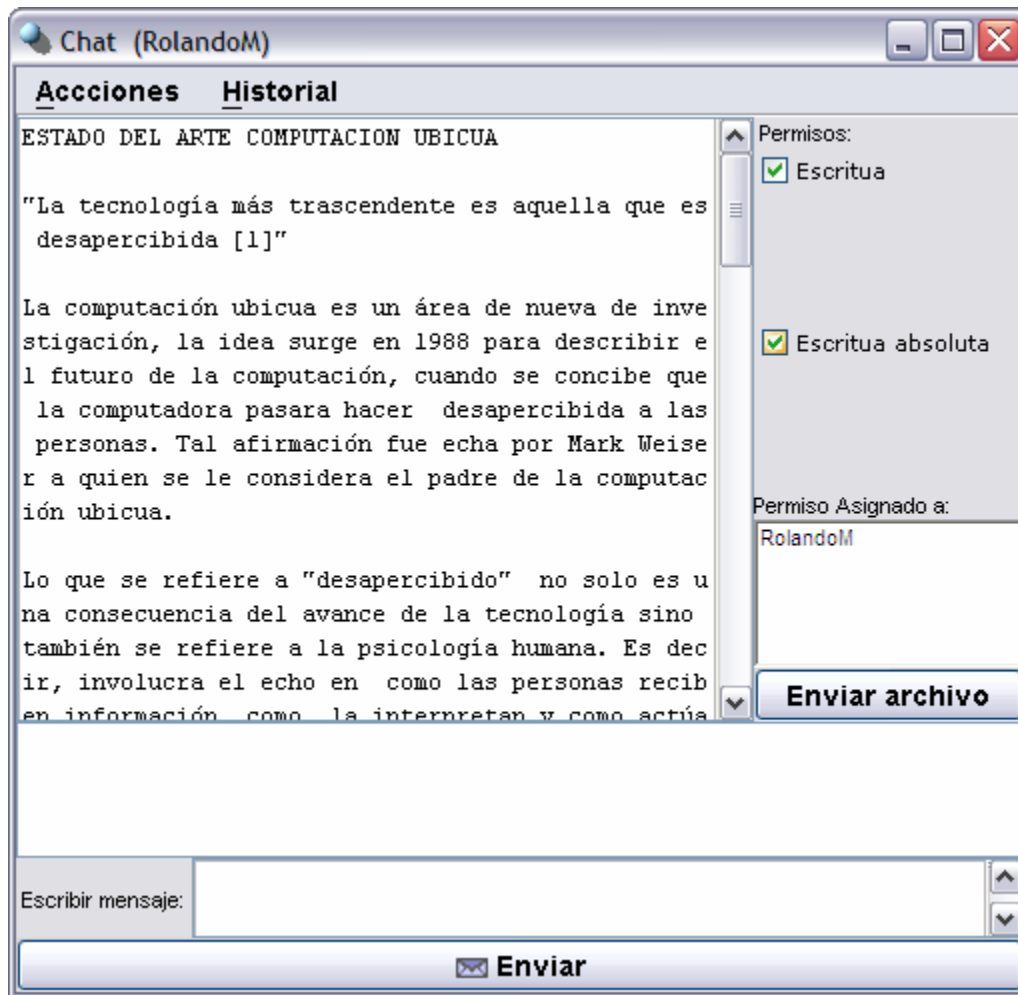


Figura 5.16. Pantalla de edición, de asignación de permisos y de envío de mensajes instantáneos para un colaborador con la máxima prioridad.

En la misma pantalla de la figura 5.16 EmilioG puede enviar mensajes por medio de una interfaz de chat que se encuentra en la parte inferior y estar en contacto con los colaboradores debido a que el mensaje es enviado a todos los participantes al momento de dar clic sobre el botón Enviar.

En cambio RolandoM cuando abre el documento debido a la prioridad asignada solo puede leer el documento y ver los cambios que está agregando EmilioG. A RolandoM le aparece la pantalla de la figura 5.17, con una pantalla de edición donde lee los cambios realizados al documento, unos botones de opción donde se le notifica los privilegios que tiene sobre el documento y en la parte de abajo cuenta con una interfaz de *chat* donde puede mandar mensajes, así como solicitar el control para hacer algunas observaciones y modificaciones sobre el documento.

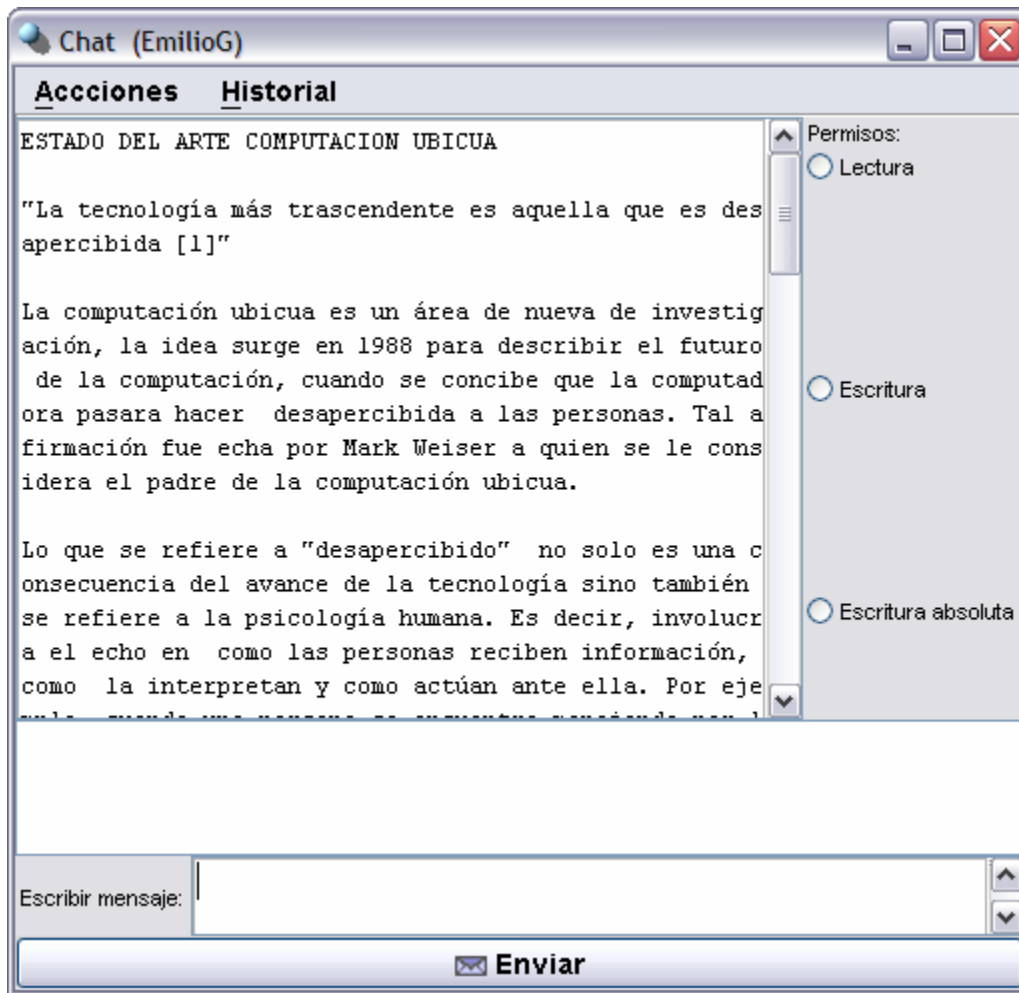


Figura 5.17. Pantalla de edición, de asignación de permisos y de envío de mensajes instantáneos para un colaborador con una prioridad menor.

Luego, si RolandoM quiere hacer algunas modificaciones sobre el documento entonces solicita el control del documento enviando un mensaje por el cuadro de diálogo. En este caso EmilioG decide dar el control del archivo a Rolando. Esto hace que la ventana de edición cambie para los dos ya que ahora Rolando tiene la prioridad más alta sobre el documento y puede hacer cualquier cambio sobre el documento (Ver figura 5.18) y Emilio solo puede ver los cambios y esperar que le regrese el control del documento.

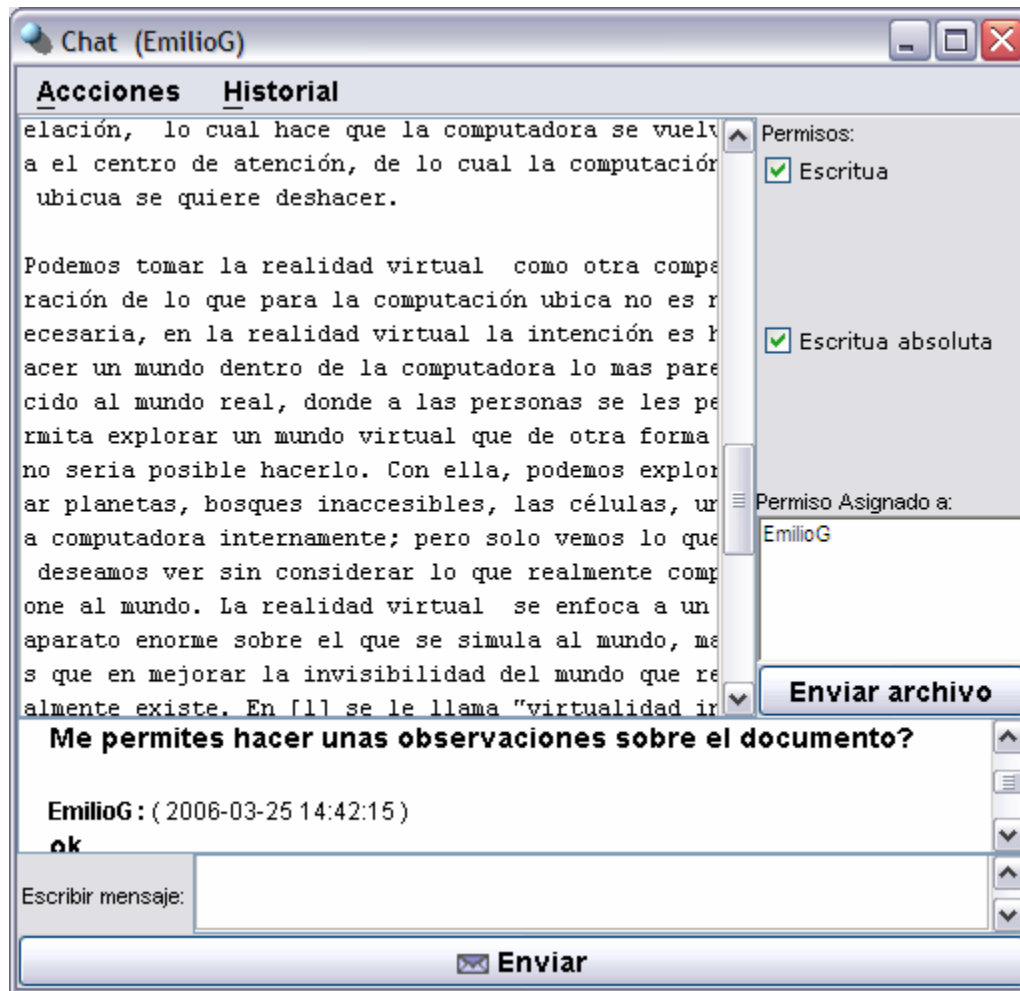


Figura 5.18. Pantalla donde un colaborador solicita hacer algunas observaciones sobre el documento.

Cuando Rolando termina de hacer las modificaciones pertinentes sobre el documento tendrá que repetir el mismo proceso para devolver el control a Emilio.

5.2.5 Trabajando con dispositivos móviles.

5.2.5.1 Encontrar a un peer móvil

Para encontrar a un peer móvil, el dispositivo tendrá que mandar constantemente un archivo XML donde representa su estatus en la red de colaboradores, desde que se une a una red P2P hasta que la abandona. Este archivo sirve para que los otros colaboradores sepan cuándo contar con este colaborador móvil.

5.2.5.2 Enviar el documento a un peer móvil.

Cuando el *nickname* de un usuario móvil aparece en la interfaz de algún usuario, esto indica que se encuentra unido a la red colaborativa y listo para participar en el documento de

edición. Luego entonces, el repositorio, quien también lo encuentra listo, envía una copia del documento al usuario móvil. Cuando el documento llega completamente al dispositivo móvil, será desplegado en la pantalla del dispositivo como en la figura 9.19.



Figura 5.19. Pantalla de un dispositivo móvil donde aparece el documento para que pueda ser leído.

Como mencionamos en capítulos anteriores, un usuario del dispositivo móvil solo puede llevar el seguimiento del documento de forma visual debido a que no le es permitido hacer modificaciones. Esto es por que en un dispositivo móvil sería muy complicado hacer modificaciones de escritura en el documento como lo sería, por ejemplo, en un teléfono celular.

5.3 Resumen

En este capítulo se demostró el funcionamiento de ECO como una herramienta de edición colaborativa, tomando en cuenta que se cubrieron todos los aspectos planteados en los objetivos particulares y generales; se presenta un ejemplo real de colaboración entre un autor y un revisor de un documento, así como de la participación de un usuario móvil.

Cabe mencionar que en la herramienta se encuentran los protocolos y servicios de comunicación P2P los cuales son la base de las comunicaciones entre los participantes del documento. Empleando una arquitectura puramente P2P.

Hasta aquí hemos visto el diseño y las pruebas de nuestro sistema ECO, con esto hemos demostrado el funcionamiento de nuestra herramienta de colaboración tanto en computadoras de escritorio y dispositivos móviles. Lo último que nos queda por explicar son las conclusiones y las extensiones a nuestro sistema, es decir, los trabajos a futuro, todo esto lo abarcamos en el siguiente capítulo de nuestro documento.

- Conclusiones y trabajos futuros



En la siguiente sección presentamos las conclusiones y resultados obtenidos a partir del presente trabajo de tesis. Así mismo, presentamos algunas propuestas de trabajos futuros que podrían ser llevados a cabo a partir del presente estudio.

6.1 CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo de tesis se desarrolló una herramienta colaborativa cumpliendo el principal objetivo de una comunicación coordinada entre los participantes y el documento editado. Como lo mencionamos en los objetivos, el sistema desarrollado utiliza una arquitectura P2P para mejorar la escalabilidad y disponibilidad de los servicios y para que dispositivos móviles puedan también ejecutar nuestra aplicación tomando en cuenta sus restricciones físicas.

Para la presente tesis, proponemos una extensión de la mensajería instantánea y del paradigma de la percepción de la presencia para dar lugar a la implementación de un espacio de colaboración potencial donde los participantes de un grupo de colaboración tienen la misma oportunidad de iniciar una sesión de colaboración. El espacio de colaboración potencial permite a los participantes conocer cuándo llevar a cabo una sesión de edición colaborativa y oportunamente unirse a la sesión. Para facilitar el uso y la aceptación, ECO implementa un sencillo y útil esquema de colaboración que asemeja a la colaboración que puede existir entre un grupo de autores cuando se encuentran en una misma computadora, donde los participantes expresan sus ideas sobre el documento y todos los demás están conscientes de las actualizaciones.

Se desarrolló un cliente ECO para una gran variedad de dispositivos tales como computadoras de bolsillo y computadoras de escritorio en redes estáticas y redes inalámbricas. Así también se implementó un editor colaborativo especializado ECO-CE de tal forma que podría ser extendido para asistir a una gran variedad trabajos que puedan ser resueltos colaborativamente como pueden ser diagramas UML, diagramas electrónicos, desarrollo de programas, imágenes, presentación, etc. Cada uno de los recursos mencionados puede ser encontrado dinámicamente sobre una red de computadoras usando tablas *Hash* distribuidas implementadas por los protocolos JXTA. La intención fue integrar dispositivos que pudieran ser útiles en una sesión de trabajo colaborativo. Por ejemplo, asistir en la colaboración por medio de dispositivos geográficamente cercanos tales como ver el documento formado colaborativamente en el *display* o proyector más cercano a los participantes dentro de un área de trabajo.

Desde el punto de vista de la implementación, en nuestra arquitectura identificamos siete características importantes que debe reunir una herramienta ubicua colaborativa:

6.1.1 Escalabilidad

Nuestro sistema es escalable porque puede crecer el número de colaboradores sin que esto afecte al funcionamiento del sistema, pueden participar cualquier número de computadoras tanto en una red fija como en una red inalámbrica. La administración del recurso compartido no se ve afectado por el número de usuarios que participan en el sistema.

6.1.2 Acceso independientemente del dispositivo

Los usuarios pueden participar en nuestro sistema desde una computadora de escritorio hasta cualquier dispositivo móvil. Con cualquiera de estos dispositivos el usuario puede conocer el estatus del documento en todo el ciclo de edición de un documento. Así también estar en todo momento en un espacio potencial donde en cualquier momento pueda participar en el documento con los demás usuarios.

6.1.3 Asistencia a la movilidad y a la desconectividad

Nuestro sistema está pensado para trabajar en una red fija así como en una red inalámbrica, en esta última cualquier usuario puede entrar y salir en cualquier momento, por esto la administración del documento la ponemos en manos de un repositorio que está representado por un elemento de la red fija, esto nos permite poder atender al usuario móvil porque conocemos el momento en que está disponible y cuando está ausente de la red. Si sabemos que está dentro de nuestra red colaborativa podemos involucrarlo con el documento como cualquier colaborador más.

6.1.4 Percepción del contexto

En todo el ciclo de edición del documento los participantes están conscientes del avance del documento, así como de la presencia y ausencia de los demás colegas participantes. Uno de los principales objetivos de la presente tesis era poder dar la posibilidad de un espacio de colaboración potencial a un grupo de autores que realizan un trabajo de edición colaborativa permitiéndoles conocer el estatus y a partir de ello saber cuándo se está en la posibilidad de participar conjuntamente, pasando de un espacio de colaboración potencial a un espacio de colaboración real.

6.1.5 Mecanismos de seguridad

En nuestro sistema existe la posibilidad de que los mensajes que se envían entre los participantes puedan ser *encriptados*, abriendo la posibilidad de que en trabajos futuros el documento y las modificaciones puedan ser enviados de una manera más segura con algún sistema de encriptación cuando el documento dada sus características así lo requiera.

6.1.6 Protocolos abiertos

Nuestro sistema utiliza los protocolos JXTA los cuales son protocolos abiertos de SUN *microsystems* para crear redes *Peer to Peer* P2P, donde dadas las características de la red no importa cuándo un nodo de la red falle porque cualquier otro peer puede ocupar su lugar.

6.1.7 Tolerancia a fallas

En ECO el nodo más crítico podría ser el nodo quien representa el repositorio debido a que el repositorio contiene realmente el documento y los demás solo una copia, pero en una red *Peer to Peer* no se presenta el problema ya que el número de repositorios a quien se envía el documento para su administración puede ser más de uno o también un nodo de colaboración podría llegar hacer un repositorio que administre los documentos.

6.1.2 Limitaciones

Es necesario mencionar que en el sistema ECO dos o más colaboradores no podrían modificar concurrentemente el documento, debido al sistema de prioridades establecido al crear el documento, solo uno a la vez podrá editar el documento siempre y cuando los colaboradores de más prioridad así lo decidan. A su vez, un dispositivo móvil tal como un teléfono celular no podría operar si previamente no encuentra a un nodo *relay*.

Para cubrir con todos esos requerimientos, en la presente tesis se desarrolló un servicio de repositorio de archivos y un servicio de percepción del contexto distribuido basado en una arquitectura P2P. La arquitectura P2P nos permite implementar escenarios con redes *ad hoc* debido a que sus servicios principales pueden ser utilizados por dispositivos móviles interconectados. De esta manera los usuarios de ECO no necesariamente necesitan acceso a Internet para participar en una sección de colaboración real. Así también JXTA tiene implementado medios de transporte virtual basados en TLS para la comunicación entre peers, por ejemplo, la suite de cifrado de *jxta* es RSA 1024 con 3DES y SHA-1. También desarrollamos un cliente de mensajes instantáneos y un editor colaborativo como un peer y pueden ser integrados con los servicios distribuidos de las redes P2P y obtener todas las ventajas de su diseño.

GLOSARIO



3DES	Data Encryption Standard (DES) es un algoritmo de cifrado, es decir, un método para cifrar información, escogido como FIPS en los Estados Unidos en 1976, y cuyo uso se ha propagado ampliamente por todo el mundo. El algoritmo fue controvertido al principio, con algunos elementos de diseño clasificados, una longitud de clave relativamente corta, y las continuas sospechas sobre la existencia de alguna puerta trasera para la <i>National Security Agency</i> (NSA). Posteriormente DES fue sometido a un intenso análisis académico y motivó el concepto moderno del cifrado por bloques y su criptoanálisis.
AIDA	AIDA: IM&PA OF PEOPLE AND SHARED RESOURCES es un cliente para mensajería instantánea y para determinar la presencia de los participantes para dispositivos móviles.
ALLIANCE	AllianceWeb es un proyecto donde el objetivo es que los entornos <i>web</i> asistan al trabajo de edición colaborativo entre los autores de un documento.
AOL	America Online Instant Messenger (AIM). Es un programa comercial para mensajería instantánea.
BEARSHARE	BearShare es un cliente gratuito <i>Gnutella</i> para intercambio de archivos, para Windows.
BROWSER	Un navegador web, hojeador o browser es una aplicación software que permite al usuario recuperar y visualizar documentos de <i>hipertexto</i> , comúnmente descritos en HTML, desde servidores <i>web</i> de todo el mundo a través de Internet. Esta red de documentos es denominada <i>World Wide Web</i> (WWW). Los navegadores actuales permiten mostrar o ejecutar: gráficos, secuencias de video, sonido, animaciones y programas diversos, además del texto y los hipervínculos o enlaces.
CDMA	Code Division Multiple Access (multiplexación por división de código). Es un término genérico que define una interfaz de aire inalámbrica basada en la tecnología de espectro extendido (<i>spread spectrum</i>). Para telefonía celular, CDMA es una técnica de acceso múltiple especificada por la TIA como IS-95.

CDPD	Cellular Digital Packet Data. Usa canales libres en terminales AMPS/TDMA para transmitir datos. Requiere un canal libre de 30 KHz, con una velocidad de 19200 bps y una tasa de transferencia neta de 9600 bps, si todos los canales de voz de la red están ocupados, los datos son almacenados hasta que alguno de ellos quede libre.
CHAT	Es un sistema donde los usuarios pueden acceder para escribir mensajes y llevar a cabo una conversación en línea vía una red de computadoras.
CLDC	Connected Limited Device Configuration (Configuración de dispositivos de conexión limitada). Define las bases para la programación de interfaces de dispositivos limitados en recursos como los teléfonos celulares o PDA's
CLIENTE-SERVIDOR	La arquitectura cliente-servidor llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificarlas.
COARSY	Collaborative Editing and Review System (Sistema de revisión y de edición colaborativa). El sistema usa una arquitectura cliente servidor para el trabajo de edición colaborativa.
COMPUTING PERVASING	Computación penetrante. Permiten a los usuarios realizar sus transacciones, personales o profesionales, empleando una nueva clase de dispositivos inteligentes y portables. Estos nuevos dispositivos inteligentes cuentan con microprocesadores que permiten a los usuarios conectarse a la red y conseguir un acceso directo, simple y seguro para recabar información y servicios
CONCHAT	Context-Aware Chat Program. Es un programa de <i>chat</i> con conocimiento del contexto, es decir proporciona información contextual entre los usuarios.

CSCW	Computer Support for Collaborative Work (Trabajo colaborativo asistido por computadora) Término genérico que combina el trabajo en grupo con las tecnologías de redes de computadoras disponibles, el hardware y Software asociados y sus técnicas.
DHT	Tabla <i>Hash</i> Distribuida. Una tabla hash o mapa hash es una estructura de datos que asocia <i>llaves</i> o <i>claves</i> con <i>valores</i> . La operación principal que soporta de manera eficiente es la <i>búsqueda</i> : permite el acceso a los elementos almacenados a partir de una clave generada.
DNS	Domain Name System (Sistema de nombres de dominios) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque, como base de datos, el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.
EMAIL	Correo electrónico , o en inglés e-mail , es un servicio de red para permitir a los usuarios enviar y recibir mensajes mediante sistemas de comunicación electrónicos (normalmente por Internet). Esto lo hace muy útil comparado con el correo ordinario, pues es más barato y rápido. Junto con los mensajes también pueden ser enviados ficheros como paquetes adjuntos.
ETHERNET	Es el nombre de una tecnología de redes de computadoras de área local (LANs) basada en tramas de datos. El nombre viene del concepto físico de <i>ether</i> . <i>Ethernet</i> define las características de cableado y señalización de nivel físico y los formatos de trama del nivel de enlace de datos del modelo OSI. <i>Ethernet</i> se refiere a las redes de área local y dispositivos bajo el estándar IEEE 802.3 que define el protocolo CSMA/CD, aunque actualmente se llama <i>Ethernet</i> a todas las redes cableadas que usen el formato de trama descrito más abajo, aunque no tenga CSMA/CD como método de acceso al medio.

FIREWALLS	Un cortafuegos , es un elemento de hardware o software utilizado en una red de computadoras para prevenir algunos tipos de comunicaciones prohibidos según las políticas de red que se hayan definido en función de las necesidades de la organización responsable de la red.
GENIE	Es un sistema para el intercambio de mensajes en una red de dispositivos móviles.
GNUTELLA	Es un proyecto de software distribuido para crear un protocolo de red de distribución de archivos entre pares, sin un servidor central.
GPRS	General Packet Radio Service o GPRS es una tecnología digital de telefonía móvil. Es considerada la generación 2.5, entre la segunda generación (GSM) y la tercera (UMTS). Proporciona altas velocidades de transferencia de datos (especialmente útil para conectar a Internet) y se utiliza en las redes GSM. GPRS es sólo una modificación de la forma de transmitir datos en una red GSM, pasando de la conmutación de circuitos en GSM (donde el circuito está permanentemente reservado mientras dure la comunicación aunque no se envíe información en un momento dado) a la conmutación de paquetes.
GROUPWARE	Programa Informático Colaborativo o Groupware se refiere a los programas informáticos que integran el trabajo en un solo proyecto con muchos usuarios concurrentes que se encuentran en diversas estaciones de trabajo, conectadas a través de una red (Internet o Intranet).
GROVE	Group Outline Viewing Editor . Es un editor de texto en tiempo real cuyo objetivo fue permitir a investigadores que se encontraban geográficamente dispersos y bajo presión de tiempo, redactar un documento en forma conjunta.
HDML	Handheld Devices Markup Language (Lenguaje de Marcado para Dispositivos Handheld). Lenguaje que permite que los dispositivos Handheld puedan tener acceso a la Web. Actualmente ha evolucionado hacia WML (Wireless Markup Language).

HOCMAN	Es una Sistema <i>peer to peer HTTP</i> para asistir a la colaboración móvil, está pensado para su utilización en dispositivos de mano y redes inalámbricas.
HTML	Acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas <i>web</i> . Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos, y también de los más fáciles de aprender.
ICQ	Es un servicio de mensajería instantánea y el primero de su tipo en ser ampliamente utilizado en Internet, mediante el cual es posible chatear y enviar mensajes instantáneos a otros usuarios conectados a la red de ICQ. También permite el envío de archivos, videoconferencias y charlas de voz.
INTELIGENCIA ARTIFICIAL	La inteligencia artificial (abreviado IA, en inglés AI) es definida como la inteligencia exhibida por una entidad artificial. Generalmente se asume que dicha entidad o sistema sea una computadora. Forma una rama vital en las ciencias de la computación, la cual lidia con el comportamiento inteligente, el aprendizaje y la adaptación en las máquinas.
INTERNET	Es una red mundial de computadoras interconectadas con un conjunto de protocolos, el más destacado, el TCP/IP. Aparece por primera vez en 1960. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que Internet, independientemente de su extensión o de que sea pública o privada.
IPV6	IPv6 es la versión 6 del Protocolo de Internet (Internet Protocol), un estándar del nivel de red encargado de dirigir y encaminar los paquetes a través de una red.
JABBER	Jabber es un protocolo libre gestionado por Jabber Software Foundation basado en el estándar XML para mensajería instantánea.

JPG	<p><i>Joint Photographic Experts Group</i> es un algoritmo diseñado para comprimir imágenes con 24 bits de profundidad o en escala de grises. JPEG es también el formato de fichero que utiliza este algoritmo para comprimir imágenes. El formato de archivos JPEG se abrevia frecuentemente JPG debido a que algunos sistemas operativos sólo aceptan tres letras de extensión.</p>
JXME	<p>Es una API traductora para conectar dispositivos J2ME a redes JXTA, teniendo presente las limitaciones técnicas de los dispositivos móviles frente a los de escritorio.</p>
JXTA	<p><i>Juxtapose</i>. Conjunto de protocolos abiertos P2P que permite que cualquier dispositivo (incluyendo dispositivos móviles) conectado a la red pueda comunicarse y colaborar.</p>
LDAP	<p><i>Lightweight Directory Access Protocol</i>. Es un protocolo de red que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP puede considerarse una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas.</p>
LDIS	<p><i>Location-Dependent Information Services</i> (Servicios de Información Dependientes de la Localización). Son servicios para la comunicación entre dispositivos móviles.</p>
MAC	<p>En redes de computadoras la dirección MAC (<i>Media Access Control address</i>) es un identificador hexadecimal de 48 bits que se corresponde de forma única con una tarjeta o interfaz de red. Es individual, cada dispositivo tiene su propia dirección MAC determinada y configurada por el IEEE (los primeros 24 bits) y el fabricante (los 24 bits restantes).</p>
MENSAJERÍA INSTANTÁNEA	<p>Los mensajeros instantáneos son un conjunto de programas que utilizan el protocolo TCP IP que sirven para enviar y recibir mensajes instantáneos con otros usuarios conectados a Internet u otras redes, además saber cuándo están disponibles para hablar.</p>
MIDDLEWARE	<p>Es un software de conectividad que permite ofrecer un conjunto de servicios que hacen posible el</p>

funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

MIDP

MIDP es el acrónimo de Perfil para dispositivos de información móvil (*Mobile Information Device Profile*) y nos proporciona un perfil que se apoya en CLDC y que nos va a proporcionar los paquetes y clases necesarios para el desarrollo de nuestras aplicaciones. MIDP está orientado principalmente a teléfonos móviles, aunque existe una implementación para PalmOS (versión 3.5 y superiores) y PocketPC, por lo que es también utilizable en casi cualquier PDA.

MIDDLEWARE

El **Middleware** es un software de conectividad que permite ofrecer un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida que se sitúa entre la capa de aplicaciones y las capas inferiores (sistema operativo y red).

MP3

MPEG-1 Audio Layer 3, más conocido como **MP3**, es un formato de audio digital comprimido con pérdida desarrollado por el *Moving Picture Experts Group* (MPEG) para formar parte de la versión 1 (y posteriormente ampliado en la versión 2) del formato de video MPEG. Su nombre es el acrónimo de MPEG-1 Audio Layer 3.

MSN

MSN o **Microsoft Network** es un ISP (Proveedor de Servicios a Internet) y un portal desarrollado por la empresa Microsoft. MSN distribuye servicios como cuentas de correo en Hotmail, posibilidades de mensajería instantánea con el protocolo *.NET Messenger Service*, un motor de búsquedas y un servicio para la creación de Blogs.

NAPSTER

Napster fue un servicio de distribución de archivos de música (en formato MP3) y pionero de las redes P2P de intercambio creado por *Shawn Fanning*. Su popularidad comenzó durante el año 2000. Su tecnología permitía a los aficionados a la música compartir sus colecciones de MP3 fácilmente con otros usuarios.

NAT	<p>NAT (<i>Network Address Translation</i> - Traducción de Dirección de Red) es un estándar creado por la <i>Internet Engineering Task Force</i> (IETF) el cual utiliza una o más direcciones IP para conectar varios computadores a otra red (normalmente a Internet), los cuales tienen una dirección IP completamente distinta (normalmente una IP no válida de Internet definida por el RFC 1918). Por lo tanto, se puede utilizar para dar salida a redes públicas a computadores que se encuentran con direccionamiento privado o para proteger máquinas públicas.</p>
NFS	<p>El <i>Network File System</i> (<i>Sistema de archivos de red</i>), o NFS, es un sistema de archivos distribuido para un entorno de red de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales. Originalmente desarrollado por <i>Sun Microsystems</i> en 1984.</p>
NICKNAME	<p>Un alias es un nombre que puede acompañar o reemplazar el nombre de una persona, (por ejemplo, Pepe es un alias de José). Puede ser humorístico, derogatorio, agudo, etc. Puede aplicarse genéricamente a un nombre, o ser particular de una persona.</p>
PDA	<p><i>Personal Digital Assistant</i>, (Ayudante personal digital) es una computadora de mano originalmente diseñado como agenda electrónica. Hoy en día se puede usar como una computadora doméstica.</p>
PDF	<p><i>Portable Document Format</i>, (Formato de Documento Portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa <i>Adobe Systems</i>.</p>
PEER TO PEER	<p>Una red informática entre iguales (<i>peer-to-peer</i> -que se traduciría de par a par ó de punto a punto) se refiere a una red que no tiene clientes y servidores fijos, sino una serie de nodos que se comportan simultáneamente como clientes y como servidores de los demás nodos de la red. Este modelo de red contrasta con el modelo cliente-servidor la cual se rige de una arquitectura monolítica donde no hay distribución de tareas entre sí, solo una simple comunicación entre un usuario y una terminal en donde el cliente y el servidor no pueden cambiar de roles.</p>

PIÑAS	A Middleware for Web Distributed Cooperative Authoring. Un sistema de edición colaborativa en un ambiente de <i>web</i> .
PIRATE	Sistema para la compartición de archivos de música entre dispositivos móviles.
QUILT	Quilt. Desarrollado por Bellcore es un sistema para edición colaborativa y acciones abstractas sobre un documento, basado en el control de roles.
RDSI	Red Digital de Servicios Integrados (RDSI o ISDN) como: una red que procede por evolución de la Red Digital Integrada (RDI) y que facilita conexiones digitales extremo a extremo para proporcionar una amplia gama de servicios, tanto de voz como de otros tipos, y a la que los usuarios acceden a través de un conjunto de interfaces normalizados.
REDES AD HOC	Las redes ad hoc, también conocidas como MANETs, son redes inalámbricas que no requieren ningún tipo de infraestructura fija ni administración centralizada. Las computadoras que las constituyen deben proporcionar también servicios de encaminamiento, retransmitiendo paquetes entre aquellos ordenadores que no tienen conexión inalámbrica directa.
RENDEZVOUS	Es una primitiva de sincronización asimétrica que permite a dos procesos concurrentes, el cliente y el servidor, intercambiar datos de forma coordinada.
RISC	Complex Instruction Set Computer (Computadora de Conjunto de Instrucciones Complejas). Esto corresponde a procesadores que son capaces de ejecutar un gran número de instrucciones pre-definidas en lenguaje de máquina (del orden del centenar).
CISC	Reduced Instruction Set Computer (Computadora de Conjunto de Instrucciones Reducido). Esto corresponde a procesadores que son capaces de ejecutar un número reducido de instrucciones.

RSA1024	El sistema criptográfico con clave pública RSA se basa en que todo usuario de dicho sistema hace pública una clave de cifrado y oculta una clave de descifrado. Una llave es un número de gran tamaño, que una persona puede conceptualizar como un mensaje digital, como un archivo binario o como una cadena de bits o bytes.
SERVIDOR PROXY	El término proxy hace referencia a un programa o dispositivo que realiza una acción en representación de otro. La finalidad más habitual es la del servidor proxy , que sirve para permitir el acceso a Internet a todos los equipos de una organización cuando sólo se puede disponer de un único equipo conectado, esto es, una única dirección IP.
SHA-1	La familia SHA (<i>Secure Hash Algorithm</i> , Algoritmo de Hash Seguro) es un sistema de funciones <i>hash</i> criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el <i>National Institute of Standards and Technology</i> (NIST). El primer miembro de la familia, fue publicado en 1993 es oficialmente llamado SHA . Sin embargo, hoy día, no oficialmente se le llama SHA-0 para evitar confusiones con sus sucesores. Dos años más tarde el primer sucesor de SHA fue publicado con el nombre de SHA-1 .
SHREDIT	Shared Editor (Editor compartido). Es un editor de texto colaborativo usado, en primera instancia para estudios de comportamiento de grupo
SOCKET	Designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada.
SOUNDPYER	Un software wireless del tipo <i>peer to peer</i> para "asociaciones musicales móviles" en coches. Cada conductor se convierte en una estación de radio móvil, transmitiendo su <i>stream</i> de música digital a otros coches que se encuentren en un área wi-fi determinada.

SRDI	<i>Shared Resource Distributed Index</i> (Servicio de índices distribuidos de los recursos compartidos). Es un servicios para indexar los anuncios de los recursos dentro de la red <i>peer to peer</i> en los <i>rendezvous peers</i> .
SUN MICROSYSTEM INC	<i>Sun Microsystems</i> es una empresa informática de Silicon Valley, fabricante de semiconductores y software. Fue constituida en 1982 por el alemán Andreas von Bechtolsheim y los norteamericanos Vinod Koshla, Bill Joy y Scott McNealy. Las siglas SUN se derivan de Stanford University Network, proyecto que se había creado para interconectar en red las bibliotecas de Stanford.
TCP/IP	La familia de protocolos de Internet es un conjunto de protocolos de red que implementa la pila de protocolos en la que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.
TLS	TLS Transport Layer Security (Seguridad para Capa de Transporte) es una versión estandarizada por el IETF del protocolo SSL que pretende abarcar toda la capa de transporte de la pila OSI. <i>Upper Atmospheric Research Collaboratory</i> . Prototipo para la investigación colaborativa atmosférica.
UARC UBIQUITOUS COMPUTING	<i>Ubiquitous Computing</i> es un término caracterizado por Mark Weiser, cuya meta es el incremento en el uso de sistemas de cómputo a través del ambiente físico, haciéndolos disponibles y a la vez invisibles al usuario.
UML	Lenguaje Unificado de Modelado (UML , por sus siglas en inglés, <i>Unified Modeling Language</i>) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (<i>Object Management Group</i>).
WALID	<i>Wearabled Augmested Task-List Interchange Device</i> . Comunidad para el intercambio de mensajes entre dispositivos móviles.

WAP	Wireless Application Protocol o WAP (protocolo de aplicaciones inalámbricas) es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, por ejemplo, Acceso a servicios de Internet desde un teléfono móvil.
WI-FI	Es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11.
WML	El Wireless Markup Language es un lenguaje cuyo origen es el XML. Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA) dotados de tecnología WAP. Es una versión reducida del lenguaje HTML que facilita la conexión a Internet de dichos dispositivos y que además permite la visualización de páginas <i>web</i> en dispositivos inalámbricos que incluyan la tecnología WAP.
WORLD WIDE WEB	La World Wide Web , <i>la Web</i> o WWW , es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador web para extraer elementos de información (llamados "documentos" o "páginas <i>web</i> ") de los servidores <i>web</i> (o "sitios") y mostrarlos en la pantalla del usuario.
WPAN	Wireless Personal Area Networks (Red Inalámbrica de Área Personal) Una PAN es una solución de redes, que realza nuestro espacio personal ya sea de trabajo o privado, por una red con una variedad de dispositivos personales, las cuales pueden usarse como vestimenta personal, a su vez estos dispositivos proveen capacidades de comunicación dentro del espacio personal, así como fuera de ella.
X509.V3	X.509 es un estándar ITU-T para infraestructura de claves pública (public key infrastructure o PKI). X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de ruta de certificación.

- XML** *eXtensible Markup Language* (lenguaje de marcas extensible) es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML).
- YAHOO** **Yahoo Inc.** es una empresa global de medios. Posee un portal de *Internet*, un directorio *web* y una serie de servicios, incluido el popular correo electrónico Yahoo.

APÉNDICE A



Manual de Instalación



MANUAL DE INTALACION Y DE CONFIGURACION

Requerimientos mínimos

Para poder ejecutar ECO en una computadora de escritorio es necesario contar con lo siguiente:

- Procesador Pentium 4 ó superior.
- 4 Gb de Disco duro (mínimo).
- 1208 Mb de memoria RAM.
- 64 Mb de Tarjeta de video.

Para poder configurar y ejecutar ECO es necesario contar con los siguientes programas y librerías, así también se proporciona algunas *URL's* donde el usuario podría descargar los programas.

Instalación Máquina Virtual de Java

Para poder ejecutar ECO, teniendo en cuenta que lo único que se desea es poder ejecutar la aplicación Java, tan sólo es necesario descargar el *Java Run Time Environment* (JRE) la cual podrá encontrar en la dirección *web*: <http://java.sun.com/j2se/1.4/download.html>.

Ejecute el fichero de instalación descargado y siga las instrucciones de pantalla y tenga en cuenta que nos recomendable elegir un directorio de instalación que contenga espacios o caracteres especiales como acentos en el nombre.

Java Micro Edition

Máquina virtual de Java (JVM), un programa encargado de interpretar código intermedio (*bytecode*) de los programas *Java* precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje *Java*. De esta forma, la JVM proporciona al programa *Java* independencia de la plataforma con respecto al hardware y al sistema operativo. En J2ME tenemos una serie de máquinas virtuales *Java* con diferentes requisitos, cada una para diferentes tipos de pequeños dispositivos.

- KVM (*Kilobyte Virtual Machine*) es la JVM más pequeña desarrollada por Sun, especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria. El verificador de clases estándar de *Java* es demasiado grande para la KVM, por lo que se introduce un algoritmo de verificación de clases en dos pasos (preverificación durante el proceso de desarrollo y verificación por el propio dispositivo).
- CVM (*Compact Virtual Machine*), que soporta las mismas características que la JVM de J2SE.

Las **configuraciones** son conjuntos de clases básicas orientadas a conformar el corazón de las implementaciones para dispositivos de características específicas. Existen dos configuraciones definidas en J2ME:

- CLDC (*Connected Limited Device Configuration*), orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad gráfica, cómputo y memoria, como por ejemplo teléfonos móviles o PDAs. Se definen sobre una máquina virtual KVM.
- CDC (*Connected Limited Configuration*), orientada a dispositivos con más recursos, como por ejemplos decodificadores de televisión digital, televisores con Internet, electrodomésticos o sistemas de navegación en automóviles. Su JVM de referencia es CVM.

Los **perfiles** son bibliotecas Java de clases específicas orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos.

Los perfiles se construyen sobre configuraciones:

- *Foundation Profile*, *Personal Profile* y *RMI Profile* se construyen sobre una configuración CDC.
- *PDA Profile* y *Mobile Information Device Profile* (MIDP) se construyen sobre CLDC. MIDP está orientado para dispositivos como teléfonos móviles, buscapersonas (*paggers*) o PDAs de gama baja con conectividad.

Para desarrollar aplicaciones en J2ME existen dos alternativas que se describirán para *Windows XP* y para *Linux*.

- Instalar las APIs CLDC 1.0 y MIDP 2.0.
- Instalar un entorno desarrollo.

Las dos alternativas requieren la instalación previa del *J2SE Development Kit* (JDK), versión 1.4.2 o posteriores. Puede descargarse de <http://java.sun.com/j2se/1.5.0/download.jsp>.

- *Windows*: jdk-1_5_0_06-windows-i586-p.exe (59.86 MB)
- *Linux*: jdk-1_5_0_06-linux-i586.bin (46.71 MB). No olvide darle permiso de ejecución mediante `chmod u+x <Archivo>`.

Instalación de las APIs

1. Entrar en <http://www.sun.com/software/communitysource/j2me/midp/download20.xml>, pulsar sobre *Download*, rellenar los datos de un formulario para obtener una licencia gratuita de uso y seleccionar el fichero:

- *Windows*: midp-2_0-src-windows-i686.zip (7.81 MB)

- *Linux*: midp-2_0-src-linux-i686.zip (7.71 MB)

2. Descomprimir el fichero utilizando *Winzip* en *Windows* o *unzip* en *Linux*. El directorio *docs* contendrá la documentación de J2ME-MIDP 2.0.

3. Añadir a la variable de entorno *path* los directorios <ruta JDK/bin> y <ruta MIDP/bin>.

<ruta JDK/bin> y <ruta MIDP/bin> representan la ruta del subdirectorio bin de JDK y MIDP, respectivamente.

Para acceder a las variables de entorno en *Windows XP*:

- Botón secundario del ratón en: *Mi PC, Propiedades*.
- Seleccionar la pestaña: *Opciones avanzadas*.
- Pulsar sobre el botón *Variables de entorno*. En este momento, aparece una ventana mostrando las variables de entorno, similar al de la figura A1.



Figura A1. Variables de entorno en *Windows XP*.

- Para modificar un valor, seleccionar la variable, pulsar sobre el botón *Modificar* en la lista que aparece en la mitad inferior de la ventana y añadir el nuevo valor separándolo por un punto y coma de los anteriores.
- Para crear una nueva variable, pulsar sobre el botón *Nueva*.

4. Añadir una nueva variable de entorno *MIDP_HOME*, con la ruta de instalación de MIDP.

Plataforma JXTA

Para poder trabajar con JXTA se necesitan una serie de librerías. Estas librerías se pueden obtener de www.jxta.org las aplicaciones de prueba MyJXTA o JXTA Shell, ambas soportadas por *Microsoft Windows*, *Linux*, *Apple Macintosh* y *Solaris*.

Cuando se desarrolla un programa basado en el API de *Java* y se quiere arrancar la aplicación creada, aparece una serie de ventanas para configurar el entorno JXTA.

El programa de configuración JXTA muestra 4 paneles.

Básico. Se debe introducir un nombre de *peer* en la opción *PeerName* de la figura A2. Este nombre será usado en los anuncios que se difundirán por la red (no tiene por qué ser único).

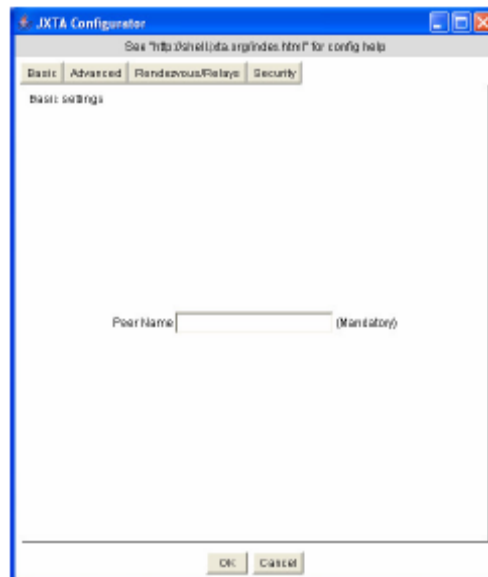


Figura A2. Opciones básicas.

Avanzado. Permite configurar información de la red donde se encuentra el *peer* (Ver figura A3). Soporta TCP/IP y HTTP. Por defecto viene configurado TCP/IP y HTTP. Se debe activar las propiedades TCP si hay otros *peers* en la red local que se desean encontrar mediante la versión *broadcast* del PDP. Aparece también una lista de todas las redes que el mecanismo conoce (sólo se puede seleccionar una donde escuchar mensajes).

Si están corriendo múltiples instancias de la plataforma JXTA en un ordenador se debe cambiar el puerto (por defecto 9701). Cada instancia debe tener su puerto.

Se activan las propiedades HTTP si se quiere conectar a los *rendezvous*. También se activan si hay un cortafuegos o NAT de por medio.

Si se está detrás de un cortafuegos se debe activar la opción de *proxy*. Si no se está conectado a *Internet*, las opciones HTTP deben estar desactivadas.

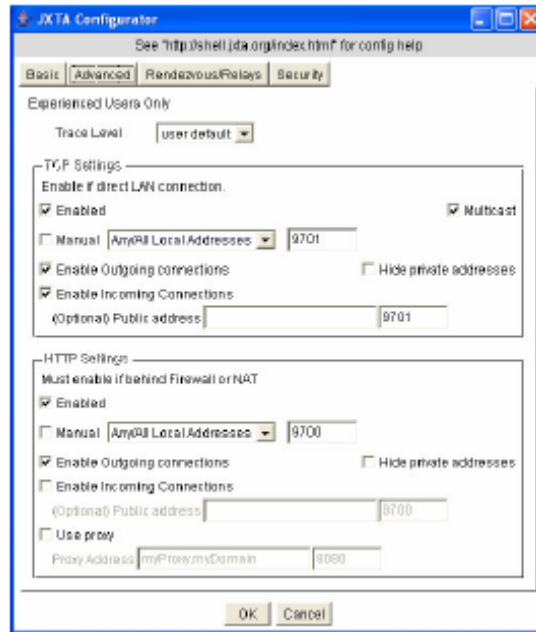


Figura A3. Opciones Avanzadas.

Rendezvous /relays. Permite especificar que un *peer* será usado como *rendezvous* o *relay* (ver figura A4). Por defecto se puede descargar una lista de *peers relay/rendezvous*.

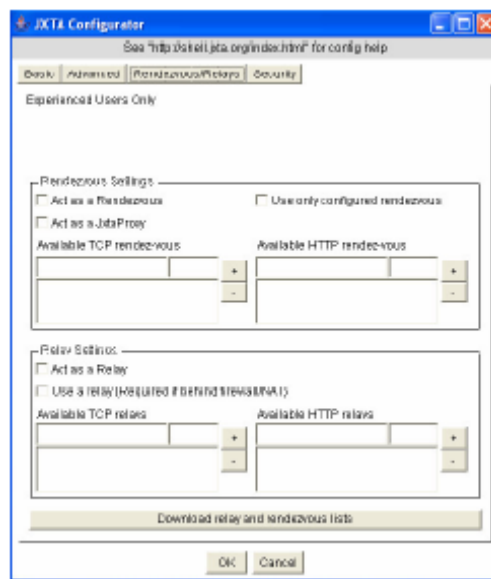


Figura A4. Opciones Relay/Rendezvous.

Seguridad. Solicita un nombre y *password* para poder acceder a una determinada configuración de *peer* (ver figura A5).

Los anuncios descubiertos (caché local) y propiedades de la configuración son almacenados en una carpeta denominada *jxta*.

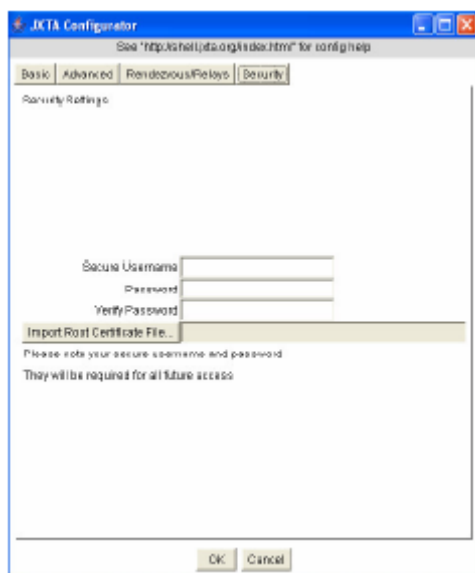


Figura A5. Opciones de Seguridad.

Plataforma JXME

Para poder ejecutar una aplicación de JXTA para J2ME deberá contar con cualquiera de los siguientes sistemas operativos:

- *Linux (kernel 2.4 o superior)*
- *Solaris 8*
- *Microsoft Windows 98 Segunda Edición, Microsoft Windows NT, Microsoft Windows 2000 o Microsoft Windows XP.*
- *Mac OS X*

También, necesitará del la siguiente lista de librerías y programas.

- *J2ME Connected Device Configuration (CDC)*
- *Java™ 2 SDK, Edición estándar, v 1.3.1 o superior.*
- *Ant.*

Una vez, instalada la lista de software, tendrá que modificar las siguientes variables de entorno.

JAVA_HOME	Direccionar al JDK <i>home</i>
ANT_HOME	Direccionar al JDK <i>home</i>
CDC_HOME	Direccionar a las librerías de J2ME CDC (<i>btclasses.zip</i> , and <i>foundation.jar</i>)

Tabla A1. Variables de entorno modificadas para JXME.

Para Ejecutar una aplicación en un dispositivo móvil es nuestro dispositivo cuenta con los siguientes requerimientos:

Kilobyte Virtual Machine

- Versión mínima de la JVM
 - Tamaño reducido (60Kb)
 - Procesador de 16-bits a 8 MHz
 - Memoria total de 160 Kb
 - 128 Kb de no-volátil (KVM y librerías CLDC)
 - 32 Kb de volátil para entorno de ejecución

Hardware MIDP

- Memoria necesaria
 - 256Kb de no-volátil para librerías MIDP
 - 8 Kb de no-volátil de datos persistentes
 - 64 Kb de volátil para el entorno de ejecución
- Pantalla
 - Mínimo de 96x54 píxels
 - Profundidad de 1 bit

Instalación de ECO

Ciente ECO

Una vez instalado y configurado nuestro ambiente de trabajo y las variables de entorno el primer paso para instalar los clientes de ECO en una computadora de escritorio es obtener del archivo “ecoim.zip” que contiene un empaquetado de los archivos fuente y de las clases de ejecución del cliente de ECO, el proceso de descompresión de “ecoim.zip” dará como resultado una serie de archivos y carpetas como en la figura A6.1

Repositorio ECO

Así también para instalar el repositorio de ECO, es necesario obtener el archivo “ecoim.zip” que contiene un empaquetado de los archivos fuente y de las clases de ejecución del repositorio de ECO. El proceso de descompresión de “ecoim.zip” dará como resultado una serie de archivos y carpetas como en la figura A6.2

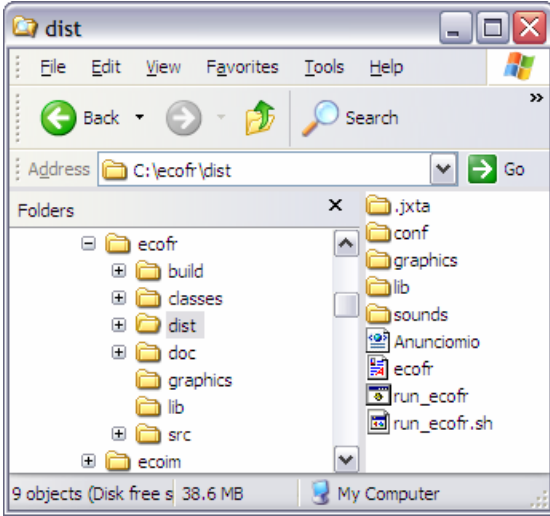


Figura A6.2 Carpetas generadas para la ejecución del cliente de ECO

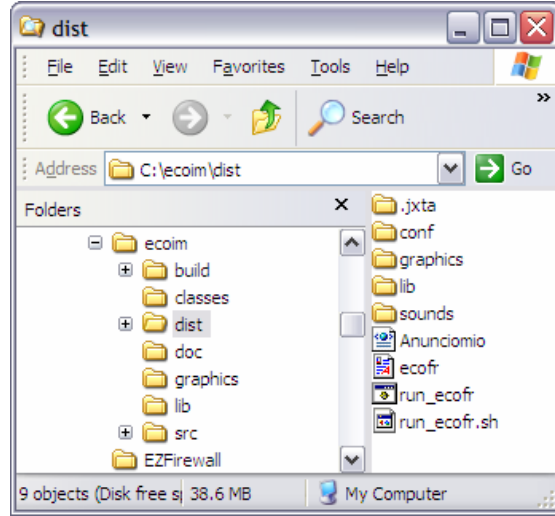


Figura A6.2 Carpetas generadas para la ejecución del repositorio de ECO

Ejecución de ECO

Cliente ECO

Para ejecutar un cliente de ECO en una computadora de escritorio es necesario acceder a la carpeta denotada “dist”, la cual contendrá los archivos llamados “run_ecofr.bat” y “run_ecoim.sh”, para ejecutar el cliente ECO en *windows* es necesario dar “enter o doble clic” sobre el archivo “run_ecoim.bat”. Para ejecutar el cliente ECO en *Linux* es necesario ejecutar `./run_ecoim.sh`. En la tabla A2 se muestra el contenido del documento “run_ecoim.bat” y “run_ecoim.sh”.

<i>run_ecoim.bat</i>	<pre>@echo off REM IF exist ".jxta" del /S /F /Q .jxta* REM IF exist ".jxtaim" del /S /F /Q .jxtaim* start javaw -Djxtaim.home=. -Djxtaim.rdv.enabled=false -jar ecoim.jar Exit</pre>
<i>run_ecoim.sh</i>	<pre>#!/bin/sh #echo test java -Djxtaim.home=. -jar ecoim.jar</pre>

Tabla A2. Contenido del archivo *run_ecoim.bat* y *run_ecoim.sh*.

Repositorio ECO

Para ejecutar un repositorio de ECO en una computadora de escritorio es necesario acceder a la carpeta llamada “dist”, la cual contendrá los archivos denominados “run_ecofr.bat” y “run_ecofr.sh”, para ejecutar el cliente ECO en *windows* es necesario dar “enter o doble clic” sobre el archivo “run_ecofr.bat”. Para ejecutar el cliente ECO en *Linux* es necesario ejecutar `./run_ecofr.sh`. En la tabla A3 se muestra el contenido del documento “run_ecofr.bat” y “run_ecofr.sh”.

<code>run_ecofr.bat</code>	<pre>@echo off REM IF exist ".jxta" del /S /F /Q .jxta* REM IF exist ".jxtaim" del /S /F /Q .jxtaim* start javaw -Djxtaim.home=. -Djxtaim.rdv.enabled=false -jar ecofr.jar Exit</pre>
<code>run_ecofr.sh</code>	<pre>#!/bin/sh #echo test java -Djxtaim.home=. -jar ecofr.jar</pre>

Tabla A3. Contenido del archivo `run_ecofr.bat` y `run_ecofr.sh`.

Cliente ECO móvil

Para Ejecutar el cliente ECO móvil es necesario haber instalado el API de ant y configurado en entorno de variables. Luego entonces, de la carpeta de “ecomovil” de la figura A7.a podemos encontrar un archivo llamado “build.xml” el cual es un archivo de construcción de nuestros ejecutables, inclusive nos presentará un emulador para que podamos ver nuestra aplicación sobre un teléfono celular o una PDA mostrándonos cuál sería la salida en un dispositivo móvil real. Para ejecutar el archivo “build.xml” necesitamos primeramente abrir una consola de Windows y teclear el comando “ant build.xml” como en la figura A7.

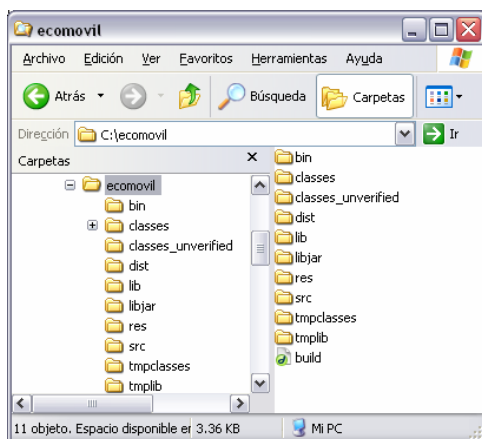


Figura A7a. Ejecución del archivo build.xml.

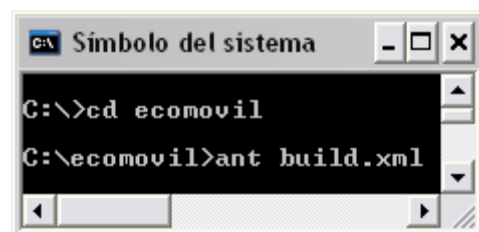


Figura A7b. Ejecución del archivo build.xml.

En la figura A8, podemos encontrar la salida de nuestra aplicación en un emulador de un teléfono celular, esto es, después de haber ejecutado el archivo “build.xml”.

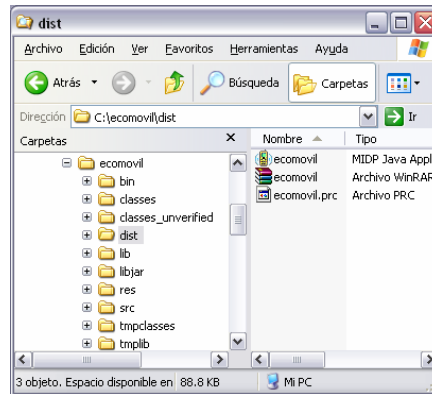


Figura A8. Archivos de salida después de ejecutar “build.xml”.

Después de ejecutar el archivo “build.xml” en la subcarpeta “bin” de la carpeta “ecomovil” se generan los archivos “ecomovil.jar” y “ecomovil.jad” como se observa en la figura A8. Luego entonces, para poder ejecutar la aplicación en un dispositivo móvil real, es necesario copiar el archivo “ecomovil.jar” al dispositivo móvil y después ejecutarlo desde ahí (ver figura A9).



Figura A9. ECO móvil en un emulador de un teléfono celular.