

REPOSITORIO DISTRIBUIDO PARA REDES DE  
TRABAJO COOPERATIVO

TESIS

MAESTRO EN CIENCIAS DE TECNOLOGIA  
INFORMATICA  
ESPECIALIDAD EN INGENIERIA DE SOFTWARE



INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY

CAMPUS MONTERREY

POR  
SUSANA EGUILA FRANCO

DICIEMBRE DE 1999

**REPOSITORIO DISTRIBUIDO PARA REDES DE  
TRABAJO COOPERATIVO**

**TESIS**

**MAESTRO EN CIENCIAS DE TECNOLOGIA  
INFORMATICA  
ESPECIALIDAD EN INGENIERIA DE SOFTWARE**



**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY**

**CAMPUS MONTERREY**

**POR  
SUSANA EGUIA FRANCO**

**DICIEMBRE DE 1999**

**REPOSITORIO DISTRIBUIDO PARA REDES DE TRABAJO  
COOPERATIVO**

**TESIS**

**MAESTRO EN CIENCIAS DE TECNOLOGÍA INFORMÁTICA  
ESPECIALIDAD EN INGENIERÍA DE SOFTWARE**



**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES  
DE MONTERREY**

**Campus Monterrey**

**POR**

**SUSANA EGUÍA FRANCO**

**DICIEMBRE DE 1999**

# Instituto Tecnológico y de Estudios Superiores de Monterrey

## Campus Monterrey

### Programa de Graduados de la División de Computación, Información y Comunicaciones

Los miembros del comité de tesis recomendamos que la presente tesis de Susana Eguía Franco sea aceptada como requisito parcial para obtener el grado académico de Maestro en Ciencias de Tecnología Informática, con especialidad en:

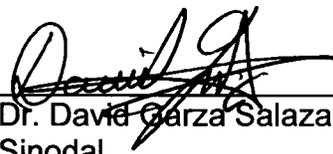
#### Ingeniería de Software

#### Comité de Tesis:



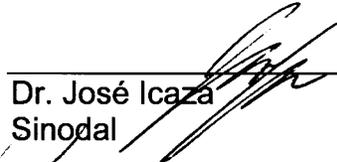
---

Dr. Ramón Brena Pinero  
Asesor



---

Dr. David Garza Salazar  
Sinodal



---

Dr. José Icaza  
Sinodal



---

Dr. Carlos Scheel Mayenberger  
Director del Programa de Graduados de la División de Computación, Información y Comunicaciones

Diciembre de 1999

**Repositorio distribuido para redes de trabajo  
cooperativo**

**Por**

**Susana Eguía Franco**

**TESIS**

Presentada a la División de Graduados en Computación, Información y  
Comunicaciones como requisito parcial para obtener el grado  
académico de Maestro en Ciencias de Tecnología Informática

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Monterrey**

**Diciembre de 1999**

# Indice

<b>Capítulo 1.- Marco de referencia</b> .....	<b>1</b>
1.1 Antecedentes .....	1
1.2 Definición del problema .....	2
1.3 OBJETIVOS .....	3
1.3.1. Objetivo general .....	3
1.3.2. Objetivos específicos .....	3
1.4 Justificación.....	4
<b>Capítulo 2.- Trabajo colaborativo apoyado por computadora (CSCW)</b> .....	<b>5</b>
2.1 Introducción a CSCW .....	5
2.2 Espacios compartidos .....	6
2.3 Conceptos .....	8
2.4 Trabajo colaborativo en grupos .....	10
2.5 CSCW en el contexto de investigación y el trabajo creativo .....	11
2.6 Casos de estudio .....	13
2.6.1. BSCW- Basic Support for Cooperative Work .....	14
2.6.2. Lotus Notes .....	15
<b>Capítulo 3.- Consideraciones de sistemas distribuidos</b> .....	<b>18</b>
3.1 Características de los sistemas distribuidos .....	18
3.2 Consideraciones para elegir un modelo de sistemas distribuidos .....	19
3.3 Alternativas de comunicación entre procesos .....	20
3.4 Problemática de la coordinación distribuida .....	20
3.5 Sistemas distribuidos en el contexto de sistemas cooperativos.....	21

<b>Capítulo 4.- Especificaciones del repositorio .....</b>	<b>22</b>
4.1 Características deseables del repositorio .....	22
4.2 Repositorio como medio de cooperación .....	22
4.3 Especificación funcional.....	23
4.3.1. Servicios al usuario .....	23
4.3.2. Servicios administrativos.....	27
4.4 Modelo del sistema propuesto .....	29
 <b>Capítulo 5.- Diseño propuesto .....</b>	 <b>31</b>
5.1 Decisiones de diseño .....	31
5.1.1. Arquitectura distribuida .....	31
5.1.2. Tolerancia a fallas .....	33
5.1.3. Control de acceso .....	34
5.1.4. Manejo de versiones .....	36
5.1.5. Implementación del prototipo .....	36
5.2 Especificación arquitectónica y diseño general .....	38
5.2.1. Servicio al usuario .....	38
5.2.2. Constitución interna del repositorio local .....	39
5.2.3. Interfaces con mensajería y otros sitios .....	42
5.3 Interfaces con RICA.....	46
5.3.1. Notificación de eventos .....	47
5.3.2. Implementación.....	47
5.4 Interfaces con otros medios digitales .....	48
5.4.1. Interfaz con sistema Phronesis .....	50
5.5 Diseño detallado .....	53
 <b>Capítulo 6.- Manejo de versiones .....</b>	 <b>54</b>
6.1 Alternativas para el manejo de versiones .....	54
6.2 Manejo de versiones concurrentes.....	54
6.2.1. Lectura virtual.....	56
6.3 Manejo de versiones local .....	57
6.4 Manejo de versiones distribuidas .....	58

<b>Capítulo 7.- Evaluación comparativa de la herramienta .....</b>	<b>60</b>
7.1 Objetivos y métodos .....	60
7.2 Comparación y evaluación .....	60
7.2.1. Comparación.....	60
7.2.2. Evaluación.....	64
<b>Capítulo 8.- Conclusiones y trabajos futuros .....</b>	<b>66</b>
8.1 Conclusiones .....	66
8.2 Limitaciones y trabajos futuros.....	67
<b>Anexo 1.- Documentación del análisis .....</b>	<b>669</b>
<b>Anexo 2.- Diseño detallado .....</b>	<b>66</b>
<b>Anexo 3.- Prototipo.....</b>	<b>66</b>
<b>Referencias bibliográficas .....</b>	<b>66</b>
<b>Vita .....</b>	<b>66</b>

## Indice de figuras

Figura 2.1. Matriz de tiempo/espacio en CSCW .....	5
Figura 2.2. Marco de trabajo cooperativo .....	8
Figura 2.3. Fases del proceso creativo .....	13
Figura 4.1. Modelo del sistema propuesto .....	30
Figura 5.1. Diseño arquitectónico, visión topológica .....	33
Figura 5.2. Diagrama de componentes de JatLite para el RouterServer (ANS)....	37
Figura 5.3. Visión arquitectónica jerárquica ....	38
Figura 5.4. Interfaces con aplicación del usuario.....	39
Figura 5.5. Constitución interna de un repositorio local .....	41
Figura 5.6. Interfaces con otros sitios y mensajería.....	42
Figura 5.7. Clasificación de mensajes .....	43
Figura 5.8. Modelo general y específico de interfaces con medios digitales externos .....	49
Figura 6.1. Mecanismo de lectura virtual .....	57
Figura 6.2. Proceso para actualización de versiones de manera local .....	57

## Indice de tablas

Tabla 4.1. Esquema de privilegios por tipo de usuario .....	25
Tabla 4.2. Requerimientos para la notificación de eventos.....	25
Tabla 5.1. Formato de un mensaje KQML estándar .....	44
Tabla 5.2. Performativas más comunes en KQML .....	44
Tabla 5.3. Eventos de notificación y candidatos de los mismos .....	47
Tabla 5.4. Metadatos a exportar a Phronesis .....	52
Tabla 6.1. Eventualidades posibles en el manejo de versiones .....	59

## Introducción

En 1998, dentro del Centro de Inteligencia Artificial del ITESM, se dio inicio al proyecto RICA (Redes Informáticas de Conocimiento con Agentes, antes conocido como CORREA), cuyo propósito es la creación de un conjunto de herramientas de apoyo a la Administración del Conocimiento, enfocados hacia recursos de educación e investigación, con el objetivo de propiciar la integración de profesores e investigadores internos y externos al ITESM, aplicando principalmente Sistemas Basados en Agentes [BRE 99].

El tipo de herramientas que actualmente se desarrollan para RICA pueden identificarse como software de apoyo al trabajo colaborativo enfocadas hacia investigadores y educadores, las cuales se implementan en una red que brindará servicios de información tales como manejo de información, servicios de alerta y monitoreo, clasificación de textos, etc., buscando establecer contactos en base a líneas de interés para favorecer la integración y colaboración.

En este contexto, se observó la necesidad de incluir un repositorio que permita compartir documentos electrónicos de trabajo entre grupos colaborativos dispersos geográficamente, y como una fuente para adquirir nuevos conocimientos, establecer contactos, someter a revisión y evaluación una idea preliminar y permitir la evolución de las mismas hasta un estado que finalmente permita su difusión a la comunidad en general. El trabajo realizado en esta tesis comprende el diseño general y detallado de tal herramienta, para un ambiente distribuido y con las características que le permitan apoyar efectivamente el trabajo colaborativo en grupo. Durante la elaboración de la misma se implementó un prototipo, del cual se implementó únicamente la versión local pero con preparaciones para un ambiente distribuido, y el cual no implementa algunos de los servicios tales como administración de comentarios a documentos, evaluación a los mismos, y los servicios de notificación y administrativos, pero esto puede ser implementado en un futuro con base en lo aquí descrito. Cabe aclarar que no se han contemplado para esta tesis los conceptos de Sistemas Multiagentes, pero se establece la base para futuras implementaciones o adaptaciones al mismo que permitan la experimentación en ese tipo de sistemas.

# Capítulo 1.- Marco de referencia

## 1.1 Antecedentes

Para cualquier persona que esté al tanto de los avances tecnológicos y científicos que se suscitaron durante este siglo, es indudable que gracias a ellos hemos podido obtener los altos niveles de desarrollo con que contamos hoy. Sin embargo, al estudiar el papel que ha desempeñado nuestro país en este desarrollo, se ha observado que muchos de estos trabajos se han llevado a cabo por grupos atomizados, y que existe una carencia de esfuerzos concentrados. Esto, aunado a la carencia de políticas de desarrollo tecnológico ligadas a problemas y oportunidades de nuestro país, nos ha llevado, de acuerdo al Dr. José Albarrán, miembro de la Academia Mexicana de Ingeniería [GYR 95], a que *“a los esfuerzos realizados les ha faltado continuidad y persistencia a largo plazo para que generen productos de gran valor”*, a pesar de que existe en México un amplio grupo de ingenieros y científicos dedicados a la investigación científica y muchos de ellos han sido reconocidos internacionalmente.

A partir de observaciones como ésta, se visualiza la necesidad de fomentar la cooperación e integración de los esfuerzos de investigación; sin embargo, ésta no es una tarea fácil si consideramos que la complejidad de comunicación e interacción humana, que se incrementa a mayor número de participantes pues se incrementan los canales de comunicación entre cada uno de ellos y la necesidad de comprender las ideas expresadas a través de diferentes lenguajes y semántica. Cuando esta dinámica se da entre personas en distintas ubicaciones geográficas o pertenecen a distintas instituciones, se enfrenta además con posibles retardos en la comunicación, problemas de sincronización e incluso burocracia, que ponen barreras para el intercambio de ideas y la toma de decisiones.

El proyecto RICA (Redes Inteligentes de Conocimiento basado en Agentes) fue iniciado por un grupo de investigadores del ITESM a principios de 1998, financiado parcialmente por el CONACYT, con el propósito de brindar servicios específicos útiles a la comunidad de investigadores, los cuales lleven a un mejor aprovechamiento de recursos humanos y materiales a través de la colaboración e integración de los mismos [BRE 98].

De manera general, los servicios que incluiría RICA son los siguientes:

- Ofrecer una interfaz única a una colección de servicios de información modulares (agentes) que propicien la integración y colaboración.
- Localización de documentos electrónicos para una determinada tarea.

- Contactar a dos personas relacionadas con base a sus intereses específicos o que deseen hacer un intercambio de ideas.
- Manejo de la información para hacerla más accesible a quien la necesite, ya sea a través de adaptación, conversión entre formatos o traducción.
- Direccionamiento de mensajes a grupos de investigadores, según sus áreas de interés.
- Monitoreo de grupos de noticias, enviando notificaciones a los miembros que pudieran estar interesados.
- Monitoreo de bases de datos sobre reportes de investigación y otros recursos, notificando los cambios que pudieran ocurrir en ellos.
- Localización (en la forma de una dirección electrónica) de investigadores bajo una cierta área de interés.

## **1.2 Definición del problema**

Ya sea por compartir el interés en un área de estudio en particular, o por trabajar de manera conjunta en un determinado proyecto, los investigadores o educadores que utilicen las herramientas suministradas por RICA requerirán de un medio que les permita compartir información de trabajo o discusión, hacer aportaciones sobre algún determinado tema y exponer sus ideas ante otros para someterlas a evaluación o refinar algún concepto, y al mismo tiempo conocer las líneas de interés de otros participantes y presumiblemente unirse a un proyecto o establecer algún contacto.

Además de los medios tradicionales como el correo electrónico, servicios de noticias y de conversación en línea, mucha de esta comunicación puede llevarse a cabo a través de documentos de naturaleza básicamente temporal, que se crean y comparten de manera asíncrona, como por ejemplo: propuestas, reportes en proceso de elaboración, documentos de trabajo, etc. Un repositorio electrónico puede proveer una plataforma de colaboración a este respecto, de manera que mantenga y organice los materiales necesarios en un punto central y permita localizar la información deseada bajo criterios relevantes, y establezca mecanismos para compartir adecuadamente la información a diferentes niveles de audiencias.

Sin embargo, dado que se busca fomentar la integración entre personas que potencialmente se encuentran dispersas a lo largo diferentes regiones geográficas o pertenecen a instituciones independientes entre sí, surgen los siguientes problemas:

- Asignar justamente los recursos de cómputo para proveer dichos servicios.
- Anticipar el crecimiento o demanda en capacidad necesaria.
- Asignar la responsabilidad administrativa de mantener tal herramienta en correcto funcionamiento y responder por un servicio adecuado a todos los participantes, tomando en cuenta la complejidad de adquirir el conocimiento

necesario para la administración y transferirlo en el tiempo para asegurar la continuidad.

Como una solución se plantea el establecimiento de un repositorio electrónico implantado sobre una red de recursos distribuidos, la cual permita mayor flexibilidad y capacidad de crecimiento que un solo repositorio central, pero sin perder una visión centralizada hacia el usuario final, de manera que no se distraiga de su actividad principal con detalles de la arquitectura computacional.

Una funcionalidad importante a la que debe darse soporte es la del manejo de versiones o historial de los documentos compartidos, para dar soporte a la evolución y sucesivo refinamiento de ideas o propuestas expresadas a través de los mismos, mediante la cual puedan explorar, evaluar y repasar diferentes alternativas. Como resultado de esa interacción, se puede llegar a un documento como "producto final" como un resultado valioso de investigación el cual es deseable dar a conocer a la comunidad, por lo que es necesario permitir establecer enlaces con otros medios digitales como bibliotecas digitales con el fin de facilitar la divulgación de los esfuerzos de investigación y educación.

### **1.3 OBJETIVOS**

#### **1.3.1. Objetivo general**

Explorar la factibilidad y aplicabilidad de un repositorio de documentos electrónicos virtual a partir de recursos distribuidos y las cualidades que tal herramienta deba presentar para ser una opción efectiva en el apoyo al trabajo colaborativo apoyado por computadora, en el contexto de la comunidad de investigadores que interactúen bajo la plataforma de trabajo creada en el proyecto RICA.

#### **1.3.2. Objetivos específicos**

- Diseñar y validar una solución que permita reunir un conjunto de documentos electrónicos dispersos en un solo repositorio virtual de manera transparente a sus usuarios y las plataformas específicas en que éstos operen.
- Estudiar el grado en que dicha herramienta puede o no apoyar el trabajo colaborativo en el contexto del proyecto RICA.
- Diseñar e implementar un prototipo del repositorio de contribuciones.
- Diseñar e implementar la interfaz con medios digitales externos con fines de divulgación, específicamente con la biblioteca digital Phronesis, desarrollada en el ITESM.
- Comparar las alternativas de herramientas similares existentes con la solución propuesta.
- Integración con servicios de RICA para proveer una plataforma más completa de colaboración.

#### **1.4 Justificación**

En la actualidad ya existen varias herramientas de software cuyo propósito es apoyar el trabajo colaborativo, sin embargo, en muchos casos se tiene un enfoque especializado en un solo aspecto de la comunicación. El repositorio que se plantea, por estar dentro del contexto del proyecto RICA, complementará el conjunto de servicios ya existentes para crear una plataforma de colaboración más integral y dar soporte a la gran variedad de interacciones que puedan surgir entre sus participantes.

Por otra parte, son pocas las herramientas que pretenden alcanzar a personas que no forman grupos de trabajo bien definidos dentro de una estructura organizacional o que no comparten la misma infraestructura computacional. Es deseable la existencia de una herramienta como la que se propone pues se busca un enfoque más democrático de cooperación donde se minimice la problemática de asignar la responsabilidad de control y mantenimiento a una determinada persona o institución, y al mismo tiempo dar flexibilidad suficiente para crecer y adaptarse a los cambios.

La importancia de los esfuerzos hacia un entorno con mayor integración son evidentes si examinamos el incremento en los últimos años de proyectos de sistemas de soporte al trabajo colaborativo (CSCW), como por ejemplo, el sistema BSCW (Basic Support for Collaborative Work) desarrollado en Alemania, [1,2] y el auge de herramientas comerciales con estos propósitos, como Lotus Notes, por mencionar uno de ellos. Herramientas como BSCW se basan fuertemente en el uso de espacios compartidos, sin embargo, esto se realiza a través de un repositorio centralizado.

El problema que aquí se presenta surge inevitablemente en cualquier ambiente de colaboración distribuido, lo cual implica que la investigación de una solución podrá servir de referencia en otras áreas o aplicaciones similares. El enfoque que se da va dirigido especialmente al apoyo de instituciones educativas y de investigación, pero pudiera aplicarse a cualquier grupo de organizaciones que colaboren mediante recursos compartidos.

## Capítulo 2.- Trabajo colaborativo apoyado por computadora (CSCW)

### 2.1 Introducción a CSCW

Así como se ha encontrado utilidad de las herramientas computacionales para resolver problemas de diversas naturalezas y complejidad, se observa cada vez más la importancia que éstas pueden tener para impactar favorablemente el trabajo colaborativo, en un entorno cada vez más abierto, a través de redes y el enorme crecimiento de Internet en los últimos años. El término de “trabajo colaborativo asistido por computadora”, (Computer Supported Collaborative Work-CSCW), es utilizado para englobar un área multidisciplinaria de investigación dedicada a estudiar y entender la forma en que la tecnología puede aplicarse como soporte a la colaboración en equipos. Dicha tecnología abarca tanto hardware como herramientas y técnicas de software. El beneficio que se espera es reducir la complejidad implícita en la coordinación de actividades cooperativas y facilitar la interacción humana [SCH 97].

Dentro del marco del CSCW, podemos encontrar diferentes esquemas de colaboración: síncronos o asíncronos, recursos compartidos o dispersos, en el mismo lugar o de manera distribuida, de manera directa o indirecta. [SCH 97] Es posible encontrar herramientas de soporte bajo cualquiera de estas condiciones, ya sea a través de boletines electrónico, videoconferencia, kioscos informativos, correo electrónico, uso de aplicaciones compartidas, herramientas para votación, etc. Los esfuerzos se han concentrado principalmente en las siguientes áreas:

- “*Groupware*”, un conjunto herramientas dedicadas a grupos colaborativos que trabajan en un proyecto con tareas específicas y un calendario establecido [4].
- Automatización de oficinas
- Comunicación en línea
- Espacios compartidos

	Mismo lugar	Distinto lugar
mismo tiempo	- Conversación cara a cara - herramientas votación - juntas con apoyos electrónicos	- Teléfono - Videoconferencia - Aplicaciones compartidas
Tiempo diferente	- Notas - Pizarras - Secciones de noticias - Kioskos	- Cartas - Correo electrónico - Fax

Figura 2.1. Matriz de tiempo/espacio en CSCW

Una de las preocupaciones más importantes en el área del Trabajo Colaborativo Apoyado por Computadora es el de comprender hasta que punto puede la tecnología apoyar o entorpecer las complejas relaciones que se dan entre los miembros de un grupo, ya que modifica y crea nuevas "dinámicas sociales" entre ellos [BER 98]. Algunos de los argumentos contrarios son que en ocasiones la tecnología tiende a imponer formas rígidas, es posible perder la valiosa comunicación informal entre las personas y puede no adaptarse a los modelos mentales que las personas usan para trabajar y relacionarse con otros. La tecnología, más que forzar modelos de trabajo predefinidos y poco prácticos, debe adaptarse a la forma en que estos son creados por quienes lo llevan a cabo. Por ejemplo, un sistema que reduce el trabajo de unas áreas a la vez que incrementa el de otras tiene muchas posibilidades de no ser aceptado, así como aquellos donde los beneficios no son percibidos por quienes efectúan el trabajo y no alcanzan a ver los beneficios compartidos, o aquellos que interfieren con las interacciones informales de las personas.

Sin embargo, también se han podido observar efectos positivos [BER 98], pues es posible que el uso de estas herramientas cree nuevas oportunidades para estructurar las actividades, incremente la democracia y disminuya el control centralizado. Se ha observado que a menudo son exitosas cuando se integran de manera natural dentro del trabajo o son definidas en el contexto de su uso, cuando son controladas por quienes van a usarlas y no son demasiado elaboradas o complicadas, y cuando ayudan a redefinir (para mejorar) el trabajo en lugar de imponer un estilo o limitarlo. Por esto, muchos de los esfuerzos en el campo del CSCW se están redirigiendo hacia tecnologías que permitan la menor interferencia, como por ejemplo, tableros electrónicos, sistemas de correo electrónico mejorado, juntas virtuales, espacios compartidos, etc.

## **2.2 Espacios compartidos**

Por espacios compartidos se entiende un espacio pasivo que contiene todos aquellos elementos que son requeridos y manipulados por los miembros de un grupo, llamados "artefactos compartidos", que registran de manera persistente las actividades y son accesibles a quienes los necesitan. Estos permiten una forma de comunicación indirecta, y es adecuado solamente para grupos donde la cooperación pueda ser flexible y las dependencias entre los integrantes del equipo no están claramente definidas, por lo que la coordinación sólo es posible si se intercambia suficiente información para estar al tanto de las actividades de los otros [SCH 97].

Los artefactos compartidos no siempre permiten el ser utilizados o modificados por más de una persona a la vez, o pueden no estar disponibles por alguna causa cuando se necesitan, por lo que surgen consideraciones de concurrencia de uso y replicación. La concurrencia debe manejarse mediante un mecanismo de control

que asegure la consistencia y confiabilidad de la información. Mediante la replicación se facilita el control de concurrencia y se incrementa la disponibilidad de un recurso ante potenciales fallas, pero impone la responsabilidad de asegurar la consistencia entre las copias del mismo [SCH 97].

En ocasiones, tal vez se desee restringir a algunos miembros del grupo el acceso o modificación a dichos elementos, para que no interfieran en el trabajo de otros, por lo que también es necesario establecer políticas de acceso y asegurar que éstas se cumplan. Un ejemplo común de un artefacto compartido donde se presenta este problema es un documento que está siendo escrito de manera conjunta por un grupo de personas. Es posible encontrar soluciones a gran parte de estos problemas en técnicas y métodos utilizados en la teoría y práctica de bases de datos distribuidas, tales como bloqueos de acceso y transacciones. A este respecto, pueden observarse 2 diferentes posturas:

- Pesimista: aseguran la consistencia en todo momento con el fin de evitar conflictos, mediante bloqueos de acceso o restricciones fuertes.
- Optimista: asume que las personas, al trabajar con recursos compartidos, establecen protocolos sociales que les permiten definir el modo de control, les ayudan a estar al tanto de las actividades de los otros y a evitar y resolver conflictos. Un ejemplo de un protocolo social es cuando alguien avisa en voz alta a los compañeros que va a efectuar una modificación, pidiendo que los demás esperen hasta que finalice.

Debido a la naturaleza misma de las nuevas tendencias como computación móvil y los sistemas distribuidos, los esquemas pesimistas no son del todo adecuados, por lo que en años recientes se ha venido dando un mayor interés en enfoques optimistas y protocolos sociales, que son menos restrictivos y dan más libertad de ajustar el control de concurrencia al contexto en que se trabaja [SCH 97].

Este tipo de controles son aplicados a sistemas tales como editores colaborativos, donde los participantes escriben al mismo tiempo un documento. Existen también los sistemas de co-autoreo, que están pensados para periodos más largos de tiempo, por lo que son de naturaleza asíncrona. En ellos, los autores trabajan en un plan en conjunto, y más tarde se dividen una porción del trabajo entre ellos e intercambian ensayos y comentarios sobre las aportaciones de otros. En estos, los usuarios pueden asumir ciertos papeles tales como autor, comentarista, lector, etc., dependiendo de los cuales pueden asignarse responsabilidades y permisos de acceso, pero estos roles pueden cambiar para ciertas partes de un documento o en el transcurso del tiempo [DIX 98].

## 2.3 Conceptos

Cuando se habla de “cooperación” o “colaboración”, se trata de un conjunto de participantes que trabajan juntos para producir un producto o servicio. [SCH 97]. Aunque pueden no compartir las mismas metas, se ayudan y obtienen beneficios.

Por otra parte, se entiende por coordinación el acto de trabajar conjuntamente de manera armónica [AMER 81]. Para este tipo de tareas, la comunicación es esencial, pues a través de ella es posible establecer metas y objetivos, asegurar que se ha llegado al término de una tarea, que no se haga trabajo redundante, resolver situaciones conflictivas, etc.

La comunicación que se da entre los participantes busca llegar a un entendimiento entre las partes, y se da ya sea de manera directa, como la comunicación cara a cara, o de manera indirecta a través de artefactos compartidos, sobre los cuales los participantes ejercen control y a través de los mismos obtienen retroalimentación [DIX 98].

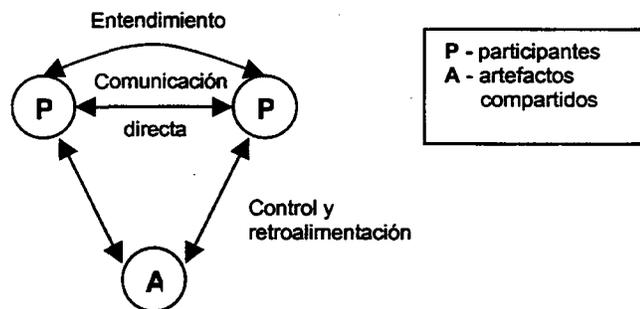


Figura 2.2. Marco de trabajo cooperativo

Tanto como para la comunicación directa como la indirecta, el mecanismo de coordinación no están claramente definidos, de manera que solo pueden ser apoyados si se intercambia suficiente información para llegar a dicho entendimiento.

Examinando más de cerca el concepto de coordinación pueden encontrarse las siguientes implicaciones [MAL 90]:

- Identificar y dividir las metas en acciones
- Seleccionar actores (grupos o individuos) y asignarles acciones
- Asignar recursos
- Compartir información

De éstas, la parte más compleja es administrar las interdependencias entre los actores, pues pueden tener metas conflictivas aunque con un mismo fin. Los tipos de interdependencias que se presentan pueden ser:

- Prerrequisitos, que requieren el ordenamiento de actividades o el movimiento de información.
- Recursos compartidos, relacionado con la asignación de recursos.
- Simultaneidad o necesidad de sincronización.

Como resultado de estas interdependencias, pueden surgir conflictos durante la coordinación, los cuales pueden resolverse por medios como:

- negociación
- intercambio de información
- intervención de una autoridad superior, o de un mediador
- estar al tanto de las actividades de los co-participantes

En la comunicación directa, la información se propaga activamente entre los miembros de la misma, mientras que, en la indirecta, es necesario un espacio común de trabajo en el cual puedan depositarse los artefactos compartidos. En este contexto, una modificación a uno de ellos es un evento de interés para otros miembros del grupo. Una manera de mantenerse al tanto de éstos eventos de interés es mediante notificaciones directas o indirectas, que pueden verse como un tipo de protocolo social, y la comunicación mediada por computadora. La importancia de tener un conocimiento de las actividades de otros es debido a que permite mantener un modelo actualizado del trabajo en progreso para mejorar la habilidad de tomar decisiones, siempre y cuando se trate solamente de información relevante y no se olviden consideraciones de privacidad y seguridad.

La comunicación a través de artefactos es importante, pero debe ser acompañada de comunicación directa, en especial en casos de conflicto, pues en ellos es más difícil tener la misma base o conocimiento del contexto, porque se debilitan las características de contemporaneidad, simultaneidad y secuencia que tiene la comunicación directa, lo cual puede verse en problemas como los siguientes [DIX 98]:

- Tomado de turnos – cuando se trata de 2 participantes, el problema es decidir cuando debe darse un cambio de turno, pero es complejo para más de 2 personas pues se agrega el problema de decidir a quien le corresponde el turno.
- Conocimiento del contexto – una conversación sólo es comprensible dentro de un contexto o dominio. El contexto externo es el tema a tratar, ambiente y circunstancias en que tiene lugar, y el interno, la dependencia con estados anteriores de la misma. La referencia que se toma para continuar una comunicación es importante, y en el caso de la participación asíncrona, es más

difícil de especificar o clarificar, o los objetos a los cuales se refiere pueden ya haber cambiado.

En estos u otros casos donde puede haber rupturas en la comunicación, se debe dar un proceso de corrección, lo cual se puede dar a través de canales alternos, redundancia, etc. En muchos casos, la automatización de la comunicación en medios electrónicos falla no porque ocurran estas rupturas, sino porque reducen la capacidad de recuperación al eliminar muchos de estos canales alternos [DIX 98].

## **2.4 Trabajo colaborativo en grupos**

En algunos casos, los grupos se forman de manera espontáneamente para responder a requerimientos de una situación. En otros casos, los grupos son de naturaleza algo más formal, formando comunidades, donde se entiende por el término “comunidad” a aquellos grupos de personas “con intereses comunes y valores compartidos que pueden residir en lugares geográficamente separados” [SCH 92].

Analizar el comportamiento de un grupo o comunidad es complejo pues es necesario tomar en cuenta toda la dinámica de las relaciones sociales que se dan entre sus miembros. Los roles y relaciones no son siempre estables, sino que pueden cambiar a través del tiempo. De la misma manera, puede cambiar la membresía a un grupo y la estructura del mismo.

Los sistemas de apoyo al trabajo colaborativo deben ser lo suficientemente flexibles como para soportar estos cambios, y permitir conocer la historia del trabajo en grupo, pues para un nuevo miembro, por ejemplo, puede ser difícil comprender el trabajo realizado sin revisar todas las contribuciones anteriores. De igual forma, un grupo puede a su vez dividirse en varios subgrupos de especialización.

Un modelo que puede ser usado para comprender el trabajo en grupo es el llamado “cognición distribuida” [DIX 98], el cual explica que el pensamiento humano no se da solamente de manera interna, sino como un proceso relacionado con el mundo exterior y con otras personas, y donde la comunicación no es solamente acerca de la transmisión de conocimiento entre personas, sino que además incluye la creación de un nuevo conocimiento grupal que no pertenece totalmente a un solo miembro y que puede estar representado de manera concreta en un artefacto compartido. En este modelo se considera que no es necesario conocer detalladamente el proceso cognitivo de cada individuo, lo cual es una tarea imposible, sino que es posible diseñar sistemas mediante el análisis de situaciones existentes en los grupos. Es importante notar que este análisis solo puede hacerse en el contexto en que ocurren dichas situaciones.

## **2.5 CSCW en el contexto de investigación y el trabajo creativo**

Existen 2 extremos en los cuales ubicarse el trabajo en grupo, como son las tareas altamente estructuradas, por seguir un procedimiento bien definido, y aquellas no estructuradas, donde las cosas no siempre se hacen de la misma manera y donde no existe un orden específico. Ejemplos de este tipo de tareas involucran aquellas que tienen que ver con el trabajo creativo. Se considera que en trabajos como el de investigación, el desarrollo de ideas y de un entendimiento entre los participantes constituye la tarea primaria [DIX 98], es decir, que el producto final es tan importante como el proceso que llevó a este.

El tipo de conversaciones que pueden tener lugar entre los miembros de un grupo como estos son [DIX 98]:

- Para llevar a cabo una acción, como aceptar, declinar, afirmar, etc.
- Para requerir aclaraciones
- Para explorar posibilidades futuras
- Para orientación y construcción de un conocimiento compartido

Durante la interacción se comparte mucha información informal, que puede estar en la forma de resultados de pruebas preliminares, reportes de juntas, anotaciones, etc. Para dar apoyo a una comunidad electrónicamente, es necesario preservar la mayor cantidad de conocimiento posible, tanto formal como informal [SCH 92]. Sin embargo, la información no estructurada o informal a menudo se pierde, ya que mucha de ésta se da en durante el proceso y no está visible en el producto final, siendo que puede ser vital porque ha sido la base para la toma de decisiones o porque permite conocer la evolución o razonamiento de lo hecho. A este respecto se ha dicho que *“si el conocimiento informal de una comunidad pudiera ser capturado y diseminado más ampliamente, la calidad y eficiencia de la investigación científica mejoraría. Esto es porque el colega invisible sería mayor y porque permitiría apertura a científicos de diversas disciplinas, que alentarían novedosas formas de investigación interdisciplinaria”* [SCH 92].

Existe una gran cantidad de dicha información por capturar, analizar, estructurar y reutilizar, por lo que es necesario el uso de tecnologías que apoyen este fin, pero en el diseño de las mismas se presenta el problema de estructurar y almacenar todo esto de manera adecuada. Entre las soluciones planteadas tradicionalmente se han mencionado bases de datos de textos o el uso de hipertexto. [BAE 93] Las herramientas que apoyan este tipo de actividades no deben seguir un flujo predeterminado y deben permitir al grupo la libertad de hacer lo que se considere necesario y adaptarse dinámicamente a cada situación, por lo que estar conscientes de las actividades de otros se convierte es un punto muy importante para la coordinación. Estas funciones pueden ser apoyadas por mensajes asíncronos, que no interrumpen el trabajo actual y permiten realizar las tareas con mayor autonomía y control.

Sin embargo, deben establecerse ciertos controles de seguridad y privacidad, pues en la comunidad científica pueden surgir problemas para otorgar el crédito cuando existe una fuerte competencia. Por ello, es importante que existan varios niveles de privacidad, ya sean: para uso personal, para el grupo de trabajo, para una comunidad con intereses afines o para el público en general.

Un grupo puede formarse para colaborar ya sea porque tiene un problema que resolver o porque en conjunto van a desarrollar algún tipo de solución o generar ideas. Se ha sugerido que el proceso creativo ocurre en un ciclo de 4 fases [SHN 97], que son:

1. La fase de documentación, recolección de información y conocimiento del dominio del problema o área de conocimiento: muchas soluciones creativas son basadas en el conocimiento de un dominio, el cual se combina de diferentes formas para llegar a una solución nueva. Las herramientas típicamente utilizadas en esta parte del proceso son bibliotecas digitales, servicios de búsqueda y visualización de la información.

El uso de la tecnología ha hecho que la posibilidad de localizar y acceder recursos de información sea un proceso más fácil cada vez, y en el proceso es posible darse cuenta que el problema en contextos de investigación no es encontrarla en cantidad suficiente, sino que, ante la gran cantidad de resultados, el poder discriminar entre ella efectivamente, en base a la calidad, contenido y organización del material. [BAE 93]

2. Fase creativa propiamente, mediante el uso de herramientas adecuadas que soporten tanto procesos exploratorios, combinación de ideas, prototipo rápido, lluvia de ideas basado en espacios, simulación, etc. Esta fase en particular se encuentra en fase de investigación pues no es muy claro como dichas herramientas pueden soportar directa o claramente el trabajo creativo. En este punto pueden mencionarse herramientas tales como ensambladores de documentos, herramientas de diseño arquitectónico o diseño, herramientas para la construcción de interfaces de usuario, modelos, simulaciones, "esqueletos" y registro del historial.
3. Refinamiento como un proceso social, mediante la consulta con miembros del equipo o expertos en el área, como un proceso social que lleve a confirmar la idea, sugerencias o argumentos contrarios. Aquí se pueden aplicar sistemas tales como grupos de noticias o listas de correo, o herramientas de votación a distancia.
4. Finalmente, la diseminación del trabajo, siendo ésta una parte central de la adquisición de conocimiento científico y de investigación. Todo este trabajo sería en vano si no se disemina o se hace público, sobre todo en el contexto de la investigación y educación. A través de esta última fase se completa un

círculo (o más bien una espiral) en la cual se están apoyando futuras investigaciones. Sin embargo, es adecuado escoger el medio adecuado de diseminación, buscando la audiencia adecuada. También, al considerar la naturaleza del trabajo de investigación, en ocasiones es deseable limitar la publicación de innovaciones para proteger la ventaja competitiva. Como soporte a esta etapa, encontramos correo electrónico, publicaciones, distribución selectiva, intranets, comunidades electrónicas, etc.

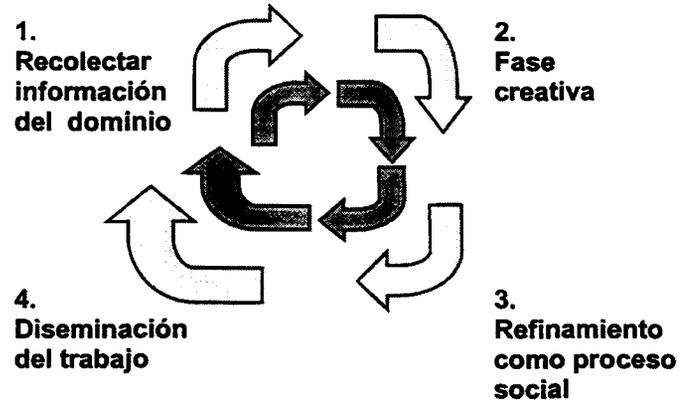


Figura 2.3. Fases del proceso creativo

En el caso de tareas poco estructuradas como las de investigación, las herramientas actuales no proveen mecanismos de control para permitir a los participantes establecer sus propios protocolos. Todos los usuarios son considerados iguales y pueden usar el sistema libremente. Sin embargo, muchos de las implementaciones actuales no llevan un registro de las actividades en el grupo.

La tecnología de apoyo debe permitir la coexistencia de diferentes modos de cooperación, reconocer la estructura u organización de grupos, permitir el trabajo dinámico y en formas inesperadas.

## **2.6 Casos de estudio**

Como casos de estudio, se escogieron sistemas como BSCW, desarrollado en Alemania, y Lotus Notes, porque se consideran buenos ejemplos de herramientas para trabajo colaborativo en el área que se quiere apoyar en esta tesis y por ser más conocidas o difundidas que otras similares. Sin embargo, existen otros ejemplos interesantes como el que se describe en [SCH 92].

### 2.6.1. BSCW- Basic Support for Cooperative Work

El proyecto BSCW o Basic Support for Cooperative Work (1995-1999) fue desarrollado en Alemania por el grupo GMD, con el objetivo de *“proveer soporte para compartir información independientemente de la plataforma para grupos de usuarios sobre World-Wide Web”* [GMD 97]. Actualmente se encuentra en su versión 3.3. BSCW es parte del proyecto de investigación CoopWWW, del cual es el núcleo. Actualmente se encuentra en su versión 3.3. En CoopWWW, el énfasis va dirigido hacia profesionales administrativos o técnicos que trabajan en diferentes organizaciones y ambientes tecnológicos.

El repositorio BSCW, se basa en el concepto de “espacios de trabajo”, donde cada usuario tiene uno propio en el cual pueden crearse directorios o carpetas, y crear una jerarquía de las mismas, cada una de las cuales se asocia a una discusión o tópico de trabajo en particular, a la cual pueden invitarse a participar a otros usuarios o grupos y en donde cada miembro invitado puede hacer sus aportaciones. Se trata de un proyecto muy completo y que aplica adecuadamente muchos de los conceptos importantes para el apoyo del trabajo en grupo a través de artefactos compartidos.

#### Principales características de BSCW

##### Colaboración

- Dirigido a grupos auto-organizados.
- Espacios compartidos sobre Internet, para la manipulación de artefactos de manera distribuida y asíncrona.
- Se manejan tipos de documentos tales como: documentos, ligas URL, grupos de discusión, etc.
- Se pueden iniciar varios temas o tópicos de discusión, en cada uno de los espacios de trabajo.
- Soporte a la publicación de documentos, dentro de BSCW, esto es, cambiando los atributos de los objetos para permitir accesos anónimos.
- Posibilidad de definir y convocar a juntas virtuales.
- Búsquedas personalizadas, permitiendo añadir campos de búsqueda de interés (ej. Fecha de creación, autor, tipo de documento, etc.).

##### Herramientas para el desarrollo de ideas y el refinamiento social

- Mantenimiento del historial de los objetos.
- No solo provee un medio de almacenamiento y recuperación, sino que además se puede compartir información informal como notas, comentarios, etc.
- Brinda la posibilidad de evaluar el trabajo de otros o juzgar la calidad del contenido de acuerdo a categorías predefinidas.
- Manejo de versiones de acuerdo al esquema RCS, que soporta desarrollar ramas diferentes de un mismo tema.

- Es posible evaluar cada versión y establecer estados para cada una de ellas, por ejemplo “estable”, o “experimental” o “liberada”.
- Establecimiento de seguros a los documentos para impedir modificaciones no deseadas.

Notificación y conciencia de actividades de otros:

- Opción de recibir vía correo electrónico, además de consultar en línea, los sucesos del área de trabajo, especificando los eventos de interés para el usuario.
- Se lleva un control de los eventos no leídos por el usuarios.
- Monitoreo de la presencia y actividad de otros usuarios y posibilidad de integrar comunicaciones síncronas con ellos.

Control de acceso:

- Administración de derechos de acceso, los cuales son heredables a través de la jerarquía de carpetas o directorios.
- Pueden establecerse grupos adicionales a los provistos por defecto (“Owners”, “Members”, “Others” y “anonymous”). Es posible “invitar” a otros grupos o usuarios a los espacios de trabajo y carpetas de otros.
- Establecimiento de seguros (bloqueo) a los documentos para impedir su modificación, mediante mecanismos de exclusión mutua (“check out”). Sin embargo, siempre es posible leer un documento, mediante una notificación de advertencia de que alguien más ha bloqueado el acceso.
- Se añaden miembros a un espacio de trabajo a través de invitación.
- Se especifican niveles básicos de acceso (añadir, modificar, eliminar, etc.) y se especifica que tipo de operaciones se pueden realizar con ello.

Distribución:

- El repositorio BSCW es accesible desde cualquier punto vía WWW.
- Es un esquema centralizado (1 servidor al que se conectan todos los clientes).

## 2.6.2. Lotus Notes

De acuerdo con Lotus, Lotus Notes es *“un administrador de información para grupos de trabajo, (...) un grupo de personas pueden compartir información a través de una computadora de red aún si esas personas se encuentran en diferentes partes del mundo”* [RAN 97].

El concepto de espacio compartido se ofrece a través de bases de datos, las cuales pueden mantener tanto información estructurada como no estructurada, además al soporte de uno o varios espacios de trabajo.

Cabe aclarar que el concepto de Lotus Notes es mucho más general que un espacio compartido de información, pues puede servir para un gran número de aplicaciones, incluyendo "workflow", automatización de oficinas o simplemente como un cliente de correo electrónico, además de que ofrece la posibilidad de adaptarse o extender las funcionalidades ofrecidas mediante programación a la medida, pero aquí el estudio se centrará en el contexto de soporte al trabajo en grupos de investigación y grupos poco estructurados.

De igual forma como se hizo en la referencia anterior, se listarán enseguida las características específicas en cada uno de los rubros de comparación de la herramienta propuesta.

### Principales características de Lotus Notes

#### Colaboración:

- Integrado con correo electrónico, agenda, lista de cosas por hacer, base de datos, etc.
- Permite dentro de sí mismo explorar Internet.
- Posible iniciar hilos de discusión
- Bases de datos diseñadas para información poco estructurada, pero también maneja la estructurada.
- Permite establecer ligas relacionales entre documentos que no residen en la misma área geográfica.
- Manejo de varios contextos a través de workspaces.
- Se da soporte también al trabajo en modo fuera de línea, a través el mecanismo de replicación.

#### Control de acceso

- Se definen dominios, a los cuales pertenecen los usuarios y dentro de los cuales se dispone de un "libro de direcciones" compartido.
- Todo se mantiene en un directorio o área de trabajo persona
- Soporte explícito a diferentes roles de trabajo (Manager, Designer, Editor, Author, Reader, Depositor, y No-Access).

#### Herramientas para el desarrollo de ideas y el refinamiento social

- A través de su capacidad de replicación, múltiples usuarios pueden actualizar información en bases de datos, de manera que los cambios más recientes son mezclados.
- El nivel de granularidad es muy grande, ya que pueden manejarse incluso a nivel campos de la base de datos.
- Pueden relacionarse lógicamente diferentes documentos entre sí (apuntadores a otros documentos), o enlaces hacia otras bases de datos o vistas diferentes.

- Para algunos tipos de documentos es posible establecer atributos de edición a diferentes secciones dentro de los mismos, de manera que pueden ocultarse o tener diferentes niveles de seguridad para ciertos usuarios.

Distribución:

- Mediante la definición de dominios, se permite la escalabilidad de la red y dar soporte a un mayor número de usuarios.
- La sincronización de ellos se lleva a cabo mediante la replicación de bases de datos
- El servidor de Lotus Notes, Lotus Domino, tiene capacidades de actuar como servidor de WEB, lo cual hace posible hacer aplicaciones cliente con independencia de plataforma.
- Es posible conocer la ubicación de los documentos y es necesaria la replicación explícita.

## **Capítulo 3.- Consideraciones de sistemas distribuidos**

En la herramienta propuesta en esta tesis, un repositorio de documentos electrónicos para apoyar el trabajo colaborativo de grupos de investigación dispersos, son deseables las siguientes características:

- Alta disponibilidad de los recursos.
- Acceso con buen desempeño.
- Mantener la integridad y el control.
- Tolerancia a fallas.
- Tomar en cuenta que los miembros pueden pertenecer a diferentes organizaciones y por lo tanto contar con diferentes plataformas de cómputo entre sí.

A través de una arquitectura distribuida se consigue un incremento de recursos virtuales, que son integrados rápidamente, y se solucionan problemas alrededor de la justicia para asignar recursos y responsabilidades administrativas.

Este capítulo no pretende ser un estudio detallado de sistemas distribuidos y sus conceptos, sino más bien ilustrar un panorama general de los aspectos a considerar en el diseño de la herramienta propuesta.

### **3.1 Características de los sistemas distribuidos**

Cuando se dice que un sistema es distribuido, se habla de un conjunto de computadoras autónomas enlazadas por medio de una red, que proveen un servicio integrado mediante un software diseñado para este propósito [COU 96]. A través del tiempo se han tenido diferentes paradigmas de cómputo, desde los sistemas monolíticos que funcionaban en mainframes hasta sistemas como cliente-servidor o la computación móvil en nuestros días. El cambio de paradigma hacia sistemas distribuidos se ha dado en busca de un mejor uso de los recursos, la integración de aplicaciones y nuevas oportunidades de negocio.

Entre las características (y beneficios) que podemos identificar en este tipo de sistemas encontramos [COU 96]:

- Posibilidad de compartir recursos, ya sea por conveniencia, por reducir costos, o para apoyar el trabajo de aquellos que trabajan cooperativamente. Esto es posible a través de administradores de recursos en cada computadora que implementan políticas y métodos para hacer los objetos compartidos accesibles desde otros puntos.

- Mayor apertura, que permiten construir sistemas con hardware y software heterogéneos sin problemas de integración, a través de mecanismos que facilitan la comunicación entre procesos.
- Permiten atender a muchos usuarios de manera simultánea y ejecutar varios procesos concurrentemente.
- Escalabilidad, a través de sistemas modulares y extensibles, que permitan al sistema crecer sin necesidad de cambiar la aplicación
- Tolerancia a fallas, a través de la redundancia en hardware y software, que incrementan la disponibilidad y la posibilidad de recuperarse de errores o fallas al menos parcialmente.
- Transparencia, que permite acceder servicios de manera uniforme sin necesidad de usar operaciones diferentes o estar al tanto de su ubicación.

Sin embargo, por la naturaleza misma de este tipo de aplicaciones es el hecho de no poder asegurar el 100% de la consistencia y sincronización, y que siempre se trata con un conocimiento parcial de los sucesos globales. Entre los requisitos de diseño necesitamos considerar mecanismos para localizar los recursos, cuestiones de desempeño y confiabilidad en relación a la carga y retardo de las comunicaciones en redes, mantener la consistencia a un costo razonable y establecer una arquitectura de software que permita balancear de manera óptima la carga de trabajo.

En particular, asegurar un buen desempeño puede implicar el uso de técnicas de "caching", donde se mantienen copias temporales de un recurso remoto de manera local para minimizar el impacto en la red de comunicaciones, pero esto implica tomar otras consideraciones para asegurar la consistencia, principalmente por el tiempo necesario para propagar actualizaciones sobre objetos compartidos.

### ***3.2 Consideraciones para elegir un modelo de sistemas distribuidos***

De entre los 2 modelos principales para los sistemas distribuidos, el modelo cliente-servidor y el modelo basado en objetos, se deben tomar en cuenta que factores los hacen más apropiados en el contexto de la aplicación que se piensa diseñar.

El modelo cliente-servidor es adecuado para muchos tipos de aplicaciones, como por ejemplo, sistemas de archivos, sistemas de noticias y correo electrónico, compartir impresoras, etc., donde los recursos y servicios compartidos son provistos por un servidor y solicitados por clientes. El proceso servidor actúa como un administrador centralizado de recursos, lo cual no es muy deseable, por ejemplo, en aplicaciones donde se necesita una cooperación directa entre los clientes o donde se necesite incrementar la disponibilidad de los recursos en prevención de posibles fallas. Además de esto, no siempre es eficiente manejar

los recursos comunes de forma centralizada en lugar de permitir que algunos de ellos residir de manera local.

En cuanto al modelo basado en objetos, cada recurso compartido es visto con un objeto con el cual se puede interactuar a través del intercambio de mensajes. Estos pueden residir en cualquier punto de la red y ser vistos de manera uniforme. Sin embargo, su implementación presenta las dificultades de localizar el objeto donde quiera que se encuentre.

### ***3.3 Alternativas de comunicación entre procesos***

Un proceso puede requerir servicios que le ofrecen otros procesos a través de solicitudes, ya sea de manera directa a través de invocaciones o llamadas a rutinas y procedimientos, o a través de objetos intermedios como archivos o memoria compartida. En ambientes distribuidos, la comunicación entre procesos tiene que darse por medio del intercambio de mensajes a través de la red.

Cuando se trata de llamadas o invocaciones a procedimientos, tenemos un enfoque "bloqueante", es decir, que quien hace la petición debe esperar hasta que ésta sea atendida para poder continuar con sus actividades, las cuales pueden o no depender de una respuesta del otro proceso. Sin embargo, si no es necesario tener dicha respuesta para continuar con las tareas, o pueden realizarse otras actividades en paralelo, el tiempo de espera es innecesario y para ello se han creado mecanismos no bloqueantes como los sistemas orientados a mensajes.

Los sistemas orientados a mensajes son más difíciles de implementar, pero permiten tener sistemas poco acoplados y ofrecen oportunidades interesantes tales como almacenar los mensajes de manera que se les de atención en una cola de espera o posponer la petición para cuando sea posible atenderla, permitir un cierto grado de independencia en el procesamiento local, etc.

### ***3.4 Problemática de la coordinación distribuida***

Existen 2 problemas principales alrededor de la coordinación en ambientes de redes, y estos son decidir con justicia el orden de los eventos que ocurren en máquinas independientes y sincronizar las actividades. Esto es especialmente importante en un sistema orientado a mensajes.

Para determinar el orden en que ocurren los eventos, el parámetro utilizado es el tiempo que los mensajes son generados. Sin embargo, sincronizar actividades de acuerdo al tiempo no es un problema trivial pues 2 computadoras independientes pueden tener sus relojes establecidos con horarios diferentes y solo pueden ser sincronizadas mediante comunicaciones en red, donde no

siempre son conocidos los tiempos de transmisión, por lo que solo pueden obtenerse resultados aproximados. Sin embargo, en ocasiones lo importante no es determinar la hora exacta en que se dió un suceso, sino determinar el orden lógico o relativo en que estos tuvieron lugar, lo cual puede estimarse gracias a relaciones de causalidad y la certeza del orden entre eventos que ve un mismo proceso. Para ambos tipos de sincronización existen algunos algoritmos propuestos, como los de Christian o de Berkeley para la relojes físicos, y el de Lamport para relojes lógicos [COU 96].

### **3.5 Sistemas distribuidos en el contexto de sistemas cooperativos**

La mayoría de los sistemas dentro del área de CSCW son aplicaciones distribuidas, ya que la colaboración que se da entre personas está relacionada con el tiempo en que interactúan (al mismo tiempo o de manera asíncrona) y la ubicación geográfica. El problema de la cooperación en sistemas distribuidos está en decidir el nivel de autonomía de sitios individuales en la red, donde a mayor autonomía, decrece el soporte a la cooperación, donde ambos son objetivos importantes de manera que a menudo se escogen enfoques mixtos [ROD 91].

El control a menudo se ha visto como un medio de proveer transparencia, que involucra los siguientes aspectos:

- Ubicación: el usuario no debe estar al tanto de la ubicación física de los recursos.
- Acceso: acceder y manipular los objetos de manera uniforme.
- Migración: los objetos pueden reubicarse o moverse sin que el usuario lo sepa.
- Concurrencia: los usuarios no tienen problemas de accesos concurrentes
- Replicación: el sistema mantiene automáticamente la consistencia de las copias de los objetos.
- Fallas: los problemas son ocultos a los usuarios.

La transparencia es lograda mediante mecanismo como la serialización de eventos y la creación de puntos de revisión seguros donde puede regresarse en caso de fallas. El problema actual es que las herramientas tradicionales han abarcado todos estos puntos, tomando todas las decisiones de control dentro del sistema [ROD 91], para lo cual, nuevamente la tendencia es hacia esquemas mixtos. Otro error común ha sido el enmascarar la presencia de otros usuarios.

## **Capítulo 4.- Especificaciones del repositorio**

### ***4.1 Características deseables del repositorio***

Con base en los conceptos del trabajo colaborativo apoyado por computadora revisados en el capítulo 2 y las consideraciones que surgen en torno de sistemas distribuidos del capítulo 3, se concretó la especificación para la herramienta propuesta, la cual busca mantener recursos de información compartidos como un medio para facilitar la colaboración, favoreciendo la comunicación a través de dichos recursos (documentos) y el proceso creativo y de generación de ideas, buscando:

- Dar suficiente flexibilidad para evitar burocracia en el manejo de los documentos compartidos y la administración de los mismos.
- Simplicidad en el manejo de la herramienta, al permitir una visión centralizada de recursos dispersos, de manera transparente al usuario, y sin complicarle las tareas administrativas de manera que pueda concentrarse en su actividad, por medio de servicios administrativos del sistema.
- Dar un orden, estructura y consistencia a los recursos compartidos, de manera que no sea simplemente un espacio donde colocar y obtener documentos, sino que además de facilidades para localización y el estudio de lo que allí se deposite.
- Incrementar la disponibilidad de los documentos para ser manipulados en el momento que se necesite.
- Control de acceso y seguridad sobre materiales sensitivos y dentro de grupos de trabajo establecidos.
- Fomentar la comunicación e intercambio de conocimiento entre personas con intereses afines.
- Proveer un medio efectivo de cooperación, con base en los conceptos del capítulo 2 y en relación a alternativas similares ya existentes.

### ***4.2 Repositorio como medio de cooperación***

Ya que se busca apoyar a investigadores y educadores cuya actividad principal en el repositorio será compartir y generar ideas, se proponen herramientas para apoyar cada una de las fases del proceso creativo propuestas por Shneiderman [SHN 97]:

1. Recolección de información y conocimiento del dominio:  
Mediante la obtención de la información, y la provisión de métodos de búsqueda y consulta en base a áreas de interés o académicas, u otros conceptos

relevantes tales como autores o grupos de trabajo. Esto es posible ya que en el proyecto RICA los participantes pueden inscribirse especificando sus áreas de interés de acuerdo a la taxonomía de ciencias computacionales de la ACM [BRE 98].

2. Fase creativa:

Favorecer la aportación, comparación y combinación de ideas a través del manejo de versiones, el historial de los documentos y herramientas de comparación de versiones.

3. Refinamiento social:

Las herramientas utilizadas en la fase 2 son también de utilidad en esta fase, complementadas con la posibilidad de hacer notas o comentarios externos a los documentos.

4. Divulgación:

A través de permitir exportar el "producto final" a otros medios digitales de divulgación, como por ejemplo bibliotecas digitales.

### **4.3 Especificación funcional**

En base a lo anterior, se presentan enseguida las especificaciones funcionales del repositorio. Se encuentran separadas en 2 secciones principales: un conjunto de servicios centrados en el usuario y visibles directamente a éste, y los servicios de administración y coordinación entre los servidores que pertenecen al repositorio. En este documento, el término "sitio" será empleado como sinónimo de la instalación física del servicio local de un repositorio.

#### 4.3.1. Servicios al usuario

##### Proceso de registro y acceso

Los usuarios harán uso de los servicios de registro de RICA, por medio de los cuales podrán acceder al repositorio. En dichos servicios se solicita información al usuario acerca de sus intereses académicos y sus preferencias acerca de cómo desea que se manejen los eventos de notificación. Cada usuario puede estar asociado a uno o varios grupos de interés.

La especificación de áreas de interés es global a todo el proyecto RICA, y está tomada actualmente de la taxonomía de las ciencias computacionales elaborada por la *Association for Computing Machinery* (ACM), la cual consiste en una clasificación arborescente en áreas, subáreas, subsubáreas, etc. Este método de clasificación permite contar con servicios de avisos y contacto entre personas

tomando en cuenta los diferentes grados de especialización sobre una misma área de interés. Cabe aclarar que el esquema de clasificación elegido actualmente no es el único aplicable, pues podría tomarse en un determinado caso otro estándar como el de la IEEE o algún otro específico de una rama del conocimiento específico. En el proyecto RICA se ha planteado la posibilidad futura de permitir que los usuarios creen dinámicamente las áreas, sus significados y relaciones entre estas [BRE 98].

Para hacer uso del repositorio, los usuarios pueden establecer una conexión al servicio a través de una página WEB la cual pueden acceder desde una computadora cliente. Se debe suministrar una contraseña, y pasar por el proceso de autenticación de usuarios definido en RICA. Solo pueden registrarse en un servidor, el cual será siempre el encargado de proveer el servicio de manera local, aunque le permitirá una visión virtual de todo el repositorio distribuido en el conjunto de sitios participantes.

#### Manejo de documentos:

Se permitirá a los miembros de repositorio llevar a cabo las siguientes operaciones:

- Creación (agregación) de documentos nuevos al repositorio, mediante la transmisión de un archivo en la máquina cliente al servidor del sitio ("upload"). Solo es posible crear documentos en el sitio en el que se tiene registro.
- Búsqueda de documentos en base a criterios considerados relevantes: autores, grupos definidos o áreas de interés a diferentes niveles de detalle. También se incluye el soporte a consultas en base a fechas de actualización, de manera que se pueda seguir el curso de eventos nuevos solamente.
- Acceso a documentos disponibles en el repositorio local o remotamente, de manera transparente, de acuerdo a sus privilegios de acceso y membresía a grupos definidos.
- Se permitirá realizar modificaciones a los archivos ya leídos o accedidos, lo cual implica solamente agregar un documento ya registrado, el cual se supone ha sufrido un cambio. Cada modificación implica registrar una versión nueva. (Ver control de versiones más adelante).
- Solicitar la promoción (exportar o divulgar) un artículo a bibliotecas digitales y medios digitales, de acuerdo a las preferencias del usuario. Dichos medios externos pueden requerir información adicional sobre el documento, lo cual implica solicitar metadatos dependiendo del contexto.

- Las modificaciones de archivos implican: copiarlo a la máquina del usuario, editarlo y subirlo. Cada modificación implica registrar una versión nueva. (Ver control de versiones más adelante).
- No se imponen restricciones de bloqueo o apartado de documentos para modificación. Un documento que está siendo modificado por un usuario podría ser bajado para modificación por otro usuario, para maximizar la disponibilidad del mismo y evitar la espera por periodos largos de tiempo por este recurso. El mecanismo de control es el manejo de versiones.
- Capacidad de asignar atributos y permisos a los documentos, sobre aquellos artefactos que le pertenecen al usuario o sobre los que tiene privilegios administrativos. El tipo de permisos asignados por omisión son:

TIPO DE USUARIO	LECTURA	MODIFICACIÓN	PROMOCIÓN	DESTRUCCIÓN
Creador	✓	✓	✓	✓
Del mismo grupo	✓	✓	*	
Todos	*			
Otros grupos	*	*		
Comparte 1 o más áreas de interés del documento	✓	*		

Tabla 4.1 Esquema de privilegios por tipo de usuario

Donde:

- "\*" representa que el usuario creador decide los permisos que asigna a este tipo de usuario(s), y no se asignan de forma automática.
- "TODOS" representa todo el universo de usuarios (locales y remotos), no es considerado un grupo.
- Los derechos de grupo y de "todos" pueden ser reducidos.
- Un grupo NO se compone sólo de usuarios locales.

Se considera que los grupos están completamente definidos, de manera que no hay necesidad de dar permisos sobre usuarios individuales.

- Para mantener la apreciación de los sucesos relevantes en el repositorio, se hará uso del mecanismo de notificación de eventos, como sigue:

EVENTO	SE NOTIFICA A
Alta de artículo	Grupo y área interés
Modificación de documento	Creador, grupo(s) y usuarios que lo están

	accesando
Promoción a biblioteca digital	Grupo(s) y área interés
Destrucción	Grupo(s)

Tabla 4.2 Requerimientos para la notificación de eventos

- Solo el autor de una versión puede deshacer ("rollback") a sus propios cambios sobre un documento. En caso de conflictos, sería necesario que las partes establecieran contacto y se pongan de acuerdo explícitamente.
- El borrado de documentos se hará mediante depuración automática y en base a la demanda, es decir, aquellos documentos que no han sido utilizados en un tiempo determinado. El sistema notificará que el documento será marcado para depuración por un tiempo antes de proceder con el borrado, dando tiempo al usuario de verificar esto. Si el usuario desea que permanezca más tiempo, deberá remover esa marca.
- En principio se puede introducir cualquier tipo de documento para almacenamiento y consulta, pero específicamente se prevee el soporte a operaciones especiales para tipos de documentos oficiales. Estas operaciones incluyen por ejemplo la posibilidad de comparación entre textos de versiones de un documento, o la provisión de conversión entre formatos, como una consideración para versiones futuras. Los formatos oficiales definidos inicialmente son: Archivos texto, HTML, PostScript, PDF, y RTF.

Propiedades de los documentos:

- Pueden estar asociados a una o varias áreas de interés, de acuerdo a la una taxonomía previamente definida en RICA.
- Cada documento tiene un dueño, que es quien originalmente lo agregó al repositorio, pero se consideran autores del mismo a todas aquellas personas que contribuyen creando versiones del mismo. Cada autor de versión es dueño de su propia versión.
- Un documento puede existir en uno o varios formatos. Una funcionalidad futura deseable es proveer la opción de conversión bajo demanda a otro formato, siempre y cuando se pueda desde y hacia formatos oficiales soportados.
- Es posible la existencia de varias instancias de documento para una misma versión (es decir, cuya referencia anterior es la misma). Al mismo tiempo, debe permitirse la visión de modificaciones a documentos y eventos sobre ellos en orden cronológico para permitir a los usuarios ponerse al día rápidamente de las actividades que se han llevado a cabo dentro del repositorio.

- Es posible deshacer una versión de algún documento ("rollback"). Un rollback implica regresar a una versión anterior del documento, acción que solo puede realizar el autor del último cambio. No se considera democrático el hacer rollback a varios niveles anteriores.

#### 4.3.2. Servicios administrativos

##### Servicios de administración local

- El respaldo de repositorio virtual busca maximizar el uso de recursos, de manera que se respalde solo lo más necesario, se incremente la disponibilidad de documentos altamente solicitados y se disminuyan costos de transmisión y sincronización. El mecanismo de respaldos está basado en la demanda (antigüedad de uso del archivo), el cual proveería la facilidad de guardar una copia de documentos en un área temporal para un número limitado de artículos, definido por el espacio asignado a este propósito, donde además mantener acceso rápido a ellos. Las áreas de almacenamiento temporal (buffers para lectura de acceso frecuente), serán depuradas con mayor frecuencia.
- Manejo de áreas temporales de almacenamiento para el mantenimiento de copias de alto uso y del uso reciente por los usuarios, las cuales sirvan a la vez como un mecanismo de respaldo en base a la demanda y se depuren con base en su antigüedad. Los documentos poco usados y con cierta antigüedad pueden ser depurados, dependiendo del espacio asignado por el administrador del sistema. Es necesario considerar los esquemas de almacenamiento del sitio, ya sea, que cada usuario tiene asignado un espacio fijo o si se dispone de un área común limitada a cierto espacio. En este caso, el enfoque será asignar a cada usuario un área en espacio en disco, que puede verse como "infinita" en el caso de no estar restringido a un tamaño, y proveer la verificación de no exceder el espacio físico disponible.
- Control de copias de documentos en áreas de respaldo -buffers- tales como conteo, ubicación, conocimiento de diversidad de formatos disponibles, sincronizar la actualización y/o información del documento maestro a las copias.
- Monitoreo de documentos "antiguos" a los que no se les ha hecho un cambio, para depurar ya sea automáticamente o preguntar al usuario si puede proceder a eliminar el archivo (mediante una notificación). No se deben depurar automáticamente documentos que contienen el último cambio o versión.

- Envío de documentos a bibliotecas digitales junto con sus metadatos correspondientes, una vez solicitado. Es posible recolectar automáticamente parte de esta información, dependiendo del caso específico.
- En un futuro, llevar a cabo el monitoreo automático de la consistencia del repositorio.

### Comunicación y coordinación entre sitios independientes

- Es necesario mantener una lista o registro de otros sitios o servidores que participan en el repositorio. Esto implica procesos de alta, baja, actualización de parámetros de los mismos, y registrar cuales se encuentran "activos". Para algunas o todas estas funciones se requiere de un administrador local, cuyas funciones serán definidas más claramente cuando se traten las decisiones de diseño, en el capítulo 5.
- Localizar y permitir acceder el archivo físico propiamente en el sitio donde reside verdaderamente, a partir de una lista de documentos disponibles virtualmente.
- Permitir la continuación del servicio aún en modo aislado del resto del repositorio. Cuando se restablecen las condiciones normales del servicio, el o los sitios desconectados deben resincronizarse por si se dieron eventos de interés global. También es importante mantener una cola de eventos sin recibir que pudieran ser entregados a un sitio que ha estado fuera de operación o comunicación por un determinado tiempo.

### Interfaces con proyecto RICA

- Interactuar con servicios de notificación de RICA, para dar aviso de la creación o modificación (creación de nuevas versiones) de documentos asociados a una persona y/o un área de interés.
- Notificación de "promoción" de documentos a bibliotecas digitales.
- Notificaciones de depuraciones programadas de documentos en desuso.

### Manejo de versiones:

El control de actualizaciones será basado en eventos en el tiempo: no se seguirá el esquema de bloqueo de documentos para modificaciones (muchas lecturas - 1 modificación). Se permitirá a todos los usuarios consultar el documento y subir cambios si así lo desean, generando nuevas versiones cada vez que esto ocurra, para evitar monopolios sobre documentos u "olvidos", y agilizar la dinámica de uso de archivos. Cuando un usuario inscriba un cambio, se registrará el tiempo en que esto ocurre. Si otro usuario tiene el mismo archivo con

fecha anterior (porque lo leyó antes de que ocurriera el cambio), recibirá una notificación de que el cambio ocurrió y deberá ser su responsabilidad revisar la última versión del documento.

El manejo de versiones en los documentos se hará mediante un esquema "distribuido". Todos los sitios que formen el repositorio mantendrán la misma lista con descriptores de documentos, y la información de donde residen el original y la última versión. Cada nueva versión se almacena localmente, pero se busca dar a conocer este hecho y notificar a todos los demás sitios participantes que él tiene la última versión. Cuando se requiera obtener un documento en una versión específica, cada sitio sabrá donde deberá buscar para solicitar una copia, la cual se almacena en un área temporal para agilizar peticiones posteriores del mismo documento, de manera que pueda ser usado localmente. En el caso de que se quisiera deshacer una versión, se debe notificar a los demás sitios para que reflejen estos cambios en su pila de versiones.

Una alternativa interesante para un futuro podría ser que se obtengan y almacenen solamente los cambios incrementales, esto es, guardar solo las diferencias entre archivos, para formatos soportados. Para subsecuentes actualizaciones, guardar solo un cierto número de cambios en pila con el fin de economizar espacio, y al llegar al límite permitido, tomaría el documento total y el cambio más antiguo para formar una nueva versión completa del documento.

En caso de manejar un esquema incremental, la base siempre es un documento completo para no buscar potencialmente  $n$  cambios en toda la red de  $n$  nodos.

#### ***4.4 Modelo del sistema propuesto***

Enseguida se presenta el modelo del sistema propuesto, de acuerdo a la notación OMT.

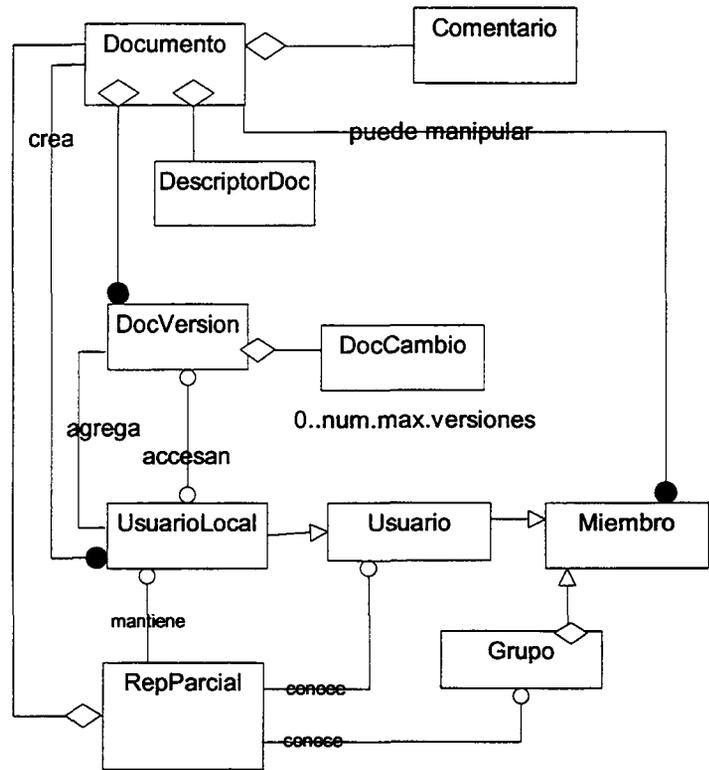
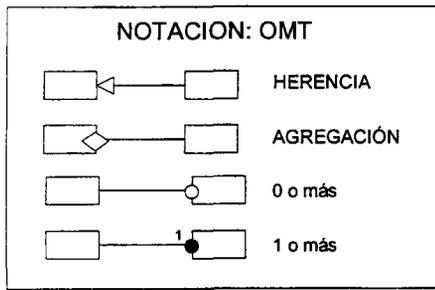


Figura 4.1. Modelo del sistema propuesto

## Capítulo 5.- Diseño propuesto

### 5.1 Decisiones de diseño

Con base en los conceptos revisados en el capítulos 2 y las consideraciones del capítulo 3, se tomaron las siguientes decisiones de diseño. Debe señalarse como una consideración importante el tratar en todo lo posible no afectar la operación normal de un sitio, ya que dichos servidores pertenecen a instituciones independientes. El servicio requiere al menos la especificación inicial (en tiempo de instalación) de parámetros tales como espacios asignados en disco, ubicaciones locales de los servicios, etc., con el fin de apegarse a las políticas locales de cada institución.

#### 5.1.1. Arquitectura distribuida

La estrategia general es que cada servidor denominado "sitio" mantiene y administra una serie de documentos creados por un conjunto de usuarios cliente registrados, los cuales pueden consultar los documentos disponibles y obtenerlos físicamente. Es necesario compartir la información del inventario de documentos con otros sitios, de manera que es preciso el intercambio de información relativa a los documentos y eventos de interés que ocurran sobre ellos.

Para una cierto número  $n$  de servidores, hay  $n-1$  canales de comunicación, donde, si  $n$  es pequeño, puede ser aceptable que todos ellos se envíen mensajes entre sí, pero si no, es necesario pensar en usar un esquema "federativo", es decir, subdividir en grupos y tener un nodo representante. Si este último no estuviera disponible en algún momento, algún otro nodo de su grupo, preparado para ello, pudiera asumir su rol. En principio, el prototipo consideraría un grupo pequeño de nodos, los cuales serían vistos como participantes de un solo grupo.

Este nodo representante, denominado "Mensajería", tendría el rol de especializarse en el manejo de las comunicaciones o mensajes entre sitios y actúe como intermediario entre todos ellos. La mensajería podría redistribuir mensajes a 1 o  $n$  sitios, y permitiría actuar como un servidor de nombres, es decir, mantener los datos necesarios para localizar un sitio, de manera que puedan contactarse directamente sin necesidad de guardar los detalles de ubicación de cada uno. Esto implica que deberá mantener una lista con los datos de otros sitios miembros, y realizar funciones de administración de registros, actualizaciones y bajas de los mismos. Esta misma mensajería podría actuar como ruteador entre diferentes grupos federativos de sitios y mantener los parámetros comunes.

El registro y baja de sitios participantes deberá ser realizado por la mensajería, es decir, los sitios se registran únicamente con ésta. Por este motivo, en la parte del sitio local, las labores del administrador se limitan a establecer la conexión con la mensajería, ajuste de parámetros locales y la operación y mantenimiento propio del sitio, y por parte de la mensajería, las peticiones de registro y comunicación con los otros sitios pueden ser manejadas de manera automática o manual, es decir, si el administrador da su aprobación a cada solicitud de registro, además de registrar el grupo al que pertenece la mensajería y relacionarlo con otros grupos .

Ya que los documentos que requieren los clientes pueden no estar siempre disponibles localmente, es responsabilidad del sitio que les da servicio iniciar una petición de transferencia hacia el sitio que lo contiene físicamente, con el fin de ofrecer el servicio. La transferencia física del documento deberá realizarse directamente, es decir, una vez que se recibe una petición de enviar un documento, se trata directamente con el sitio solicitante, con el fin de hacer más eficiente la transferencia.

El usuario posiblemente necesitará trabajar repetidamente con dicho documento y tal vez creará una o más versiones del mismo, por lo que, cuando un sitio recibe una transferencia en respuesta a una solicitud suya, el sitio se encargará de mantener una copia temporal. Con esto se pretende además minimizar la carga adicional en comunicaciones y ofrecer un mayor tiempo de respuesta al usuario.

Para esta tesis, se considera adecuado el uso de un modelo de sistema orientado a mensajes, que permita intercambiar información entre los sitios de manera poco acoplada y se soporten escenarios de posibles fallas o desconexiones temporales de la red, de manera que los eventos generados puedan ser almacenados en una cola de espera hasta que puedan ser procesados o distribuidos a otros sitios. La interacción directa del usuario con el sistema es adecuada en el modelo cliente-servidor.

El tiempo es un parámetro importante para conocer fechas de creación y modificación de documentos, para programar eventos, etc. Todos los servidores deberán "sincronizar su tiempo", es decir, no es posible sincronizar físicamente los relojes sin posiblemente interferir en la operación normal del sitio, pero sí es posible guardar las diferencias relativas de tiempo entre ellos. Un algoritmo adecuado para estos propósitos es el de Berkeley [COU 96], donde la mensajería puede actuar como árbitro para ajustar los tiempos de todos.

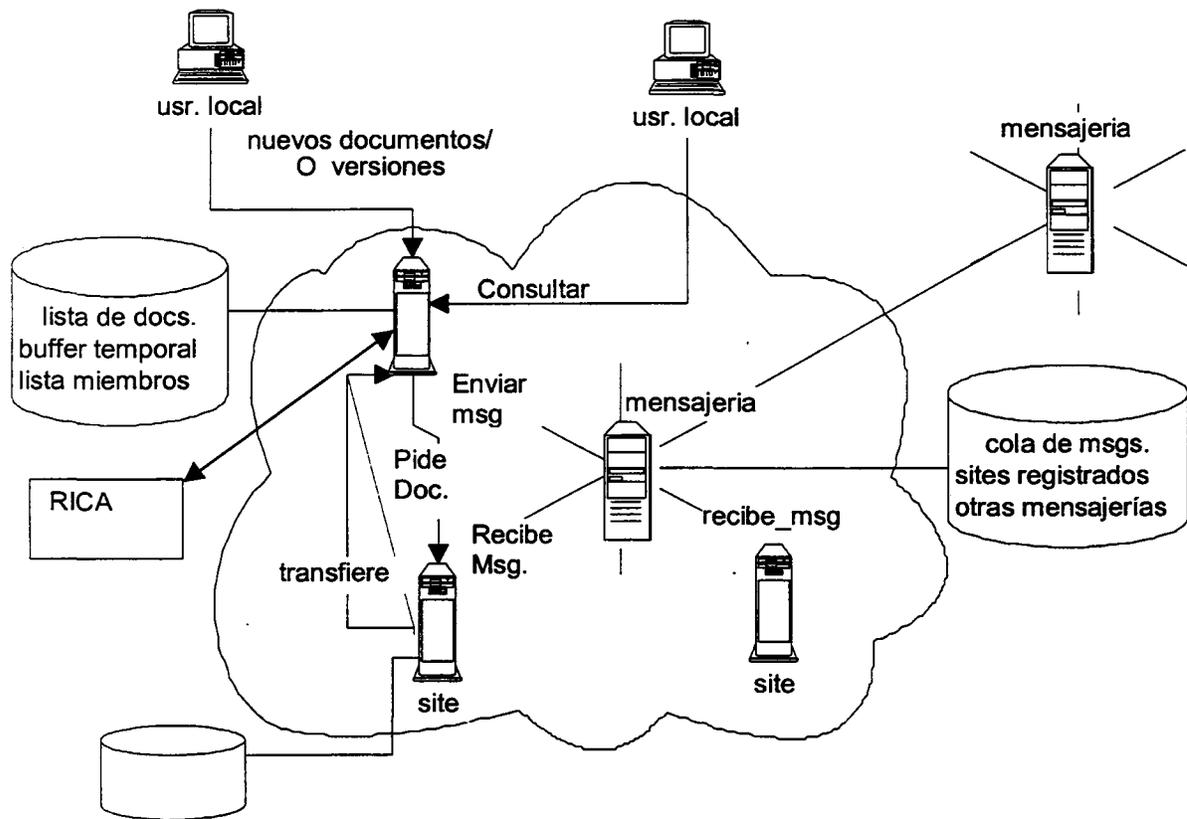


Figura 5.1 Diseño arquitectónico, visión topológica

En la figura 5.1 se muestra un doble enlace con el sistema RICA, el cual es un prerequisite para el funcionamiento del repositorio en el sitio, pues se integrará con el mismo a través de interfaces para solicitar información específica de usuarios y grupos, así como solicitar servicios de notificación de eventos de sucesos relevantes en el sistema.

### 5.1.2. Tolerancia a fallas

Como ya se mencionó anteriormente, se busca, mediante la distribución de recursos, tener un cierto grado de tolerancia a fallas al incrementar la disponibilidad al menos parcial del servicio, a través de la redundancia local de documentos bajo demanda, y la posibilidad de resincronizar el trabajo realizado cuando las condiciones normales se recuperan. Esto permitirá que los usuarios del sistema no tengan que esperar innecesariamente y que puedan hacer su trabajo en el momento que les parezca más adecuado.

El servicio en un sitio específico, que se recupera de una falla o de estar temporalmente fuera de comunicación (ej: se reinicia) tiene la responsabilidad de:

- notificar su "llegada"
- sincronizarse con respecto al tiempo
- solicitar información de los eventos que tuvieron lugar en su ausencia (adiciones, cambios en la lista de documentos, cambios en los documentos que contiene, etc.).

Esto se llevará a cabo mediante el sistema de mensajería, donde un nodo asumiría el papel de "central de mensajes" y se encargaría de enviar los mensajes a los nodos correspondientes y almacenarlos si no le es posible y reintentar posteriormente.

Al realizar una "transacción", se deberá guardar el tipo de operación y hora de inicio, manteniendo un estado "no terminado" hasta que se completa. Si por algún motivo se interrumpe la transacción, deberá ser posible:

1. reintentar
2. intentar con un nodo de respaldo (un área de buffer que tiene la versión solicitada, o si esto no es posible, notificar si se desea buscar una versión anterior)
3. si todo falla, preguntar al usuario si desea programarlo para más tarde.

Se asume que el nodo mensajería es confiable y robusto, pero en caso de que éste llegara a fallar, otro servidor de su grupo pudiera asumir temporalmente el rol (el primero que se dé cuenta del hecho y tenga la capacidad de desempeñar esa función, si existe alguno), y avisar a los demás. Cuando se reanude el servicio de la mensajería oficial, ésta debe notificar de su llegada y reconectar a todos los demás nodos. El nodo sucesor puede elegirse mediante algún algoritmo de elección [COU 96].

### 5.1.3. Control de acceso

La forma como se especifica el control de acceso en sistemas para el trabajo colaborativo tienden a ser muy restrictivas, pues no contemplan la posibilidad de un cambio de roles en los participantes y/o existe burocracia al momento de asignar permisos de acceso, pues en ocasiones existe una persona con derechos de administración, que no siempre está disponible [STI 98].

En este caso en particular se tomará un enfoque intermedio, mediante la posibilidad de delegar de tareas a 1 o varias personas, además del dueño actual de un documento, de manera que sea posible cambiar los roles en el tiempo y solicitar autorizaciones a más de una persona autorizada para maximizar la disponibilidad.

De manera más específica, existen 2 tipos de privilegios que se pueden tener sobre un documento:

- a) A nivel global (todo el documento)
- b) A nivel versión

Todos los permisos globales del documento son incrementales, es decir, existen 5 niveles de permisos, de menor a mayor, y cada nivel contiene a los anteriores.

0. Sin acceso- en este nivel, no se puede consultar el contenido del documento, pero si puede conocer la existencia del mismo. En este caso, el sujeto pudiera contactar a uno de los administradores del documento para solicitar permiso.

1. Lectura- implica capacidad de consulta del contenido documento y sus metadatos

2. Escritura- abarca la consulta y permite crear nuevas versiones.

3. Administración- puede dar y modificar permisos ya existentes, exceptuando el derecho de pertenencia sobre el documento y sus versiones. Esto incluye capacidad para tomar decisiones sobre el documento ante acciones automáticas del sistema tales como depuración. Los permisos de administración son en relación a características globales y no por versión.

4. Pertenencia – se da solo a los autores del documento original sobre todo el documento, y a cada autor de versión sobre sus modificaciones en particular. El derecho de pertenencia no puede ser quitado, a menos que el poseedor decida eliminar el documento o versión.

Los privilegios que tiene el dueño de una versión, le permiten de manera exclusiva el borrado de la misma ("rollback"). En este caso, el dueño del documento no puede deshacer los cambios efectuados por otras personas. El sistema puede tener permisos de borrado en base a tiempo en desuso, es decir, al efectuar la función de depuración, siempre y cuando se configure de esta forma.

Los beneficios de este esquema propuesto son los siguientes:

- Permite la delegación de funciones administrativas de documentos para mayor flexibilidad
- Promueve la democracia en la aportación de ideas y en caso de desacuerdo, promueve la comunicación de ideas y retroalimentación.

Los beneficios mencionados se hacen patentes en situaciones como las siguientes:

- Existen varios responsables de un proyecto, pero solo uno está registrado como dueño del documento, o si se da un cambio de funciones entre los participantes.

- Situaciones donde se necesita una rápida atención a la solicitud de permisos de lectura o escritura, pero el responsable no se encuentra disponible.
- Durante la generación de ideas, muchas veces se descartan algunas inicialmente, pero posteriormente pudieran ser retomadas.
- Puede presentarse el caso que necesite discutirse una idea, y por tratarse de grupos dispersos, y llegar a una decisión o conclusión por parte de un subconjunto del total de miembros de un grupo, eliminando una idea que tal vez pudiera ser reconsiderada por los miembros restantes.
- Prevención de antagonismos entre miembros de grupos.

Cabe aclarar que, como se menciona más adelante, cuando un usuario sea dado de baja del sistema RICA, debe ceder sus derechos de pertenencia sobre los documentos a usuarios existentes o depurarlos.

#### 5.1.4. Manejo de versiones

Ya que ésta es una parte central de este trabajo, se describe de manera detallada en el capítulo 6. Por ahora, baste decir que la estrategia a utilizar tendrá una perspectiva optimista (no bloqueante) para permitir un libre flujo de aportaciones y dar soporte a posibles fallas. Cada modificación a un documento se añade como una versión independiente del mismo, las cuales pueden ser observadas en 2 dimensiones: la referencia o versión de lectura anterior del autor y el orden cronológico de los eventos. El mecanismo propuesto de control de versiones se apoya de manera importante en la notificación de eventos, lecturas virtuales (falsas lecturas) y herramientas de comparación entre versiones, mediante los cuales es posible asumir que un usuario está al tanto de las actividades y propuestas de sus colaboradores.

Cabe aclarar que, en relación a los protocolos optimistas, existen varios de ellos, desde aquellos que son completamente automatizados y donde el manejador de base de datos se encarga de completar o deshacer las transacciones, hasta aquellos que son implementados de manera "social", como aquí se menciona.

#### 5.1.5. Implementación del prototipo

Una de las características deseables más importantes de este sistema es la independencia de plataforma, ya que los sitios o nodos participantes no están limitados a una sola organización o institución, por lo que la comunicación entre sitios se realizará sobre la base de TCP/IP, generalmente aceptado para la mayoría de los ambientes computacionales. Siguiendo la misma filosofía, el servicio al usuario se realizará vía Internet a través de páginas HTML en plataforma WEB, lo cual permite además una integración más fácil con otros

servicios de RICA. De igual manera, las transferencias de documentos se llevarán a cabo mediante los estándares HTTP, FTP o mediante sockets.

El manejo de las comunicaciones entre los diferentes sitios y la mensajería será implementado bajo el esquema que plantea JatLite [HEE 98], un framework escrito en el lenguaje Java y basado en TCP/IP, que provee una plantilla para la comunicación entre “agentes de software” o módulos independientes de software, con soporte al intercambio de mensajes basados en el estándar KQML [FIN 93].

Esta herramienta fue seleccionada por varias razones:

- Proporciona la infraestructura básica de comunicaciones requerida, mediante un servicio especial llamado “Agent Message Router” (AMR), que provee la funcionalidad básica de un servidor de nombres, ruteo de mensajes, comunicación asíncrona, y manejo de colas de mensajes en disco.
- Soporte a comunicación asíncrona y reserva de mensajes.
- Independencia de plataforma, por estar implementado en el lenguaje Java, y posibilidad de entablar comunicación con otras tecnologías a través de CORBA, RMI o simplemente por medio el uso de mensajes KQML, que son un estándar independiente de la plataforma o lenguaje de desarrollo.
- EL API de JatLite maneja diferentes niveles de abstracción y aplicables a diferentes necesidades, dependiendo del tipo de servicios que se requieran. Por ejemplo, existen una capa especializada en el manejo de comunicaciones vía HTTP o FTP, y más arriba de ella se encuentra la capa especializada en el manejo de mensajes KQML.

### The RouterServer Component Diagram

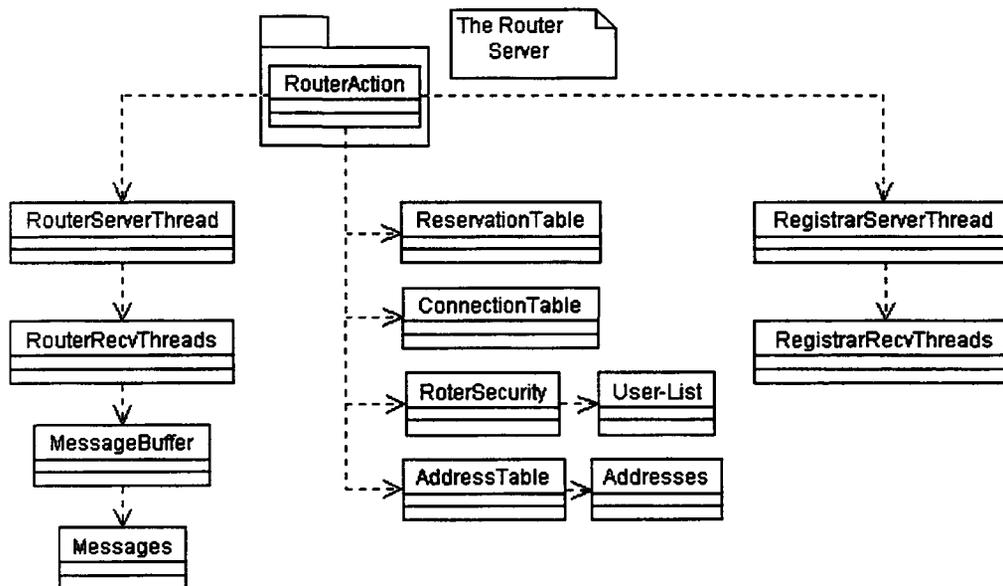


Figura 5.2 Diagrama de componentes de JatLite para el RouterServer (ANS)

## 5.2 Especificación arquitectónica y diseño general

En la sección 5.1.1, se describió la arquitectura general de la aplicación propuesta y el funcionamiento de la misma. Todo este proceso debe facilitar el trabajo del usuario, por lo que debe realizarse de manera transparente. Con el fin de enfatizar este objetivo, en la figura 5.3 se presenta otra perspectiva de la arquitectura desde un punto de vista jerárquico, el cual permite aislar los detalles de cada capa convenientemente.

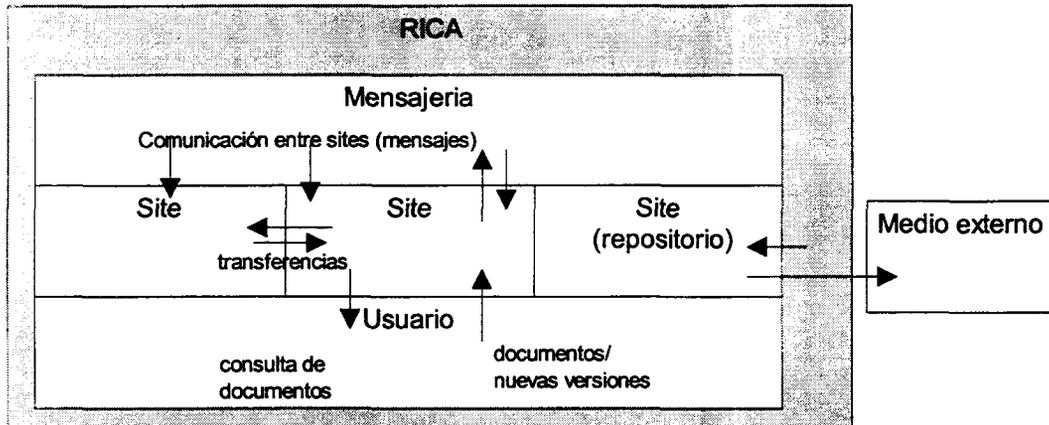


Figura 5.3. Visión arquitectónica jerárquica

Enseguida se detallará la especificación arquitectónica para las siguientes áreas de la aplicación:

- Aplicación o servicio al usuario
- Servicio de repositorio en el sitio (constitución interna)
- Interfaces con mensajería y otros sitios-repositorios
- Interfaces con sistema RICA
- Interfaces con otros medios digitales (ej. Bibliotecas digitales)

### 5.2.1. Servicio al usuario

Un usuario será atendido por un solo sitio, y deberá ser un usuario registrado del sistema RICA residente en el mismo. Después de pasar por un proceso de autenticación, podrá interactuar con el servicio del repositorio vía un navegador de páginas web. La notificación de eventos es responsabilidad de RICA, de manera que no habrá interacción con el usuario en este sentido. Las peticiones y servicios del usuario, que serán denominados "AppUsuario", serán atendidas a través de conexiones a un proceso en el lado del servidor, llamado "ServicioUsuario".

En cuanto a la inscripción de documentos, el usuario deberá iniciar una transferencia ("upload") iniciando un protocolo específico ftp o http, a través de la misma interfaz provista por la página del navegador. El archivo será recibido en el sitio por un módulo especializado en estas tareas (ftpSiteServer), por lo cual también será encargado de proveer las requisiciones que haga el cliente en comunicación con su contraparte en la proceso del usuario. Los detalles del funcionamiento interno de ftpSiteServer y ServicioUsuario se explicarán en la siguiente sección.

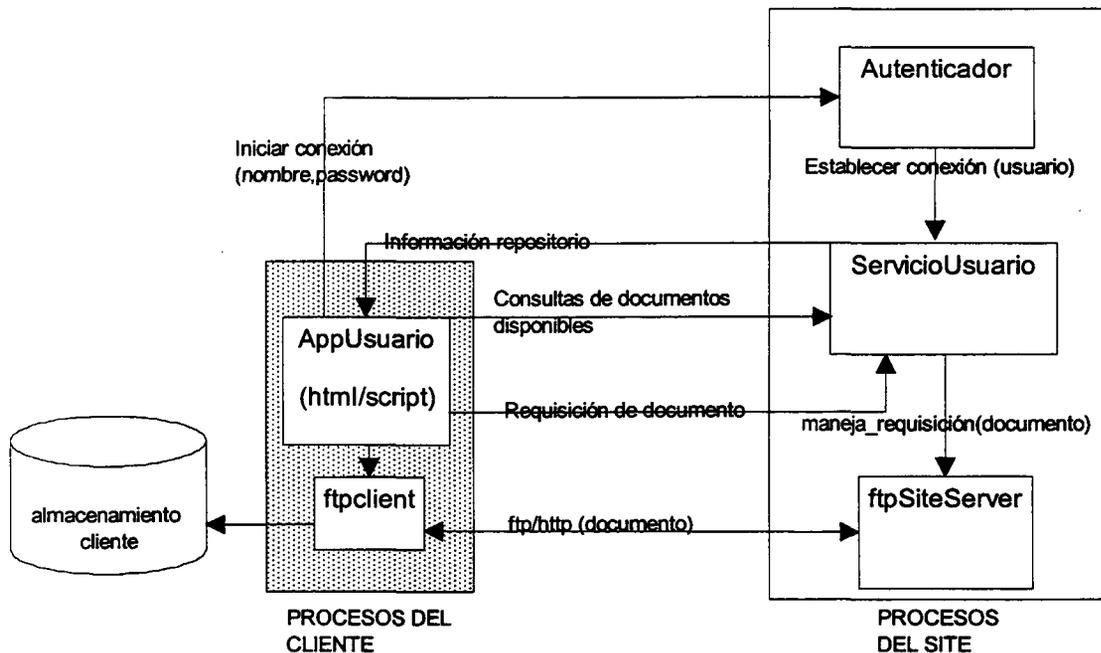


Figura 5.4. Interfaces con aplicación del usuario

### 5.2.2. Constitución interna del repositorio local

Esta es la parte más compleja del sistema total, pues lleva a cabo las labores administrativas, servicio al usuario y comunicación entre sitios. Por ello, se detallan primeramente los elementos internos y más adelante se detallan las interfaces necesarias para el comportamiento global del repositorio.

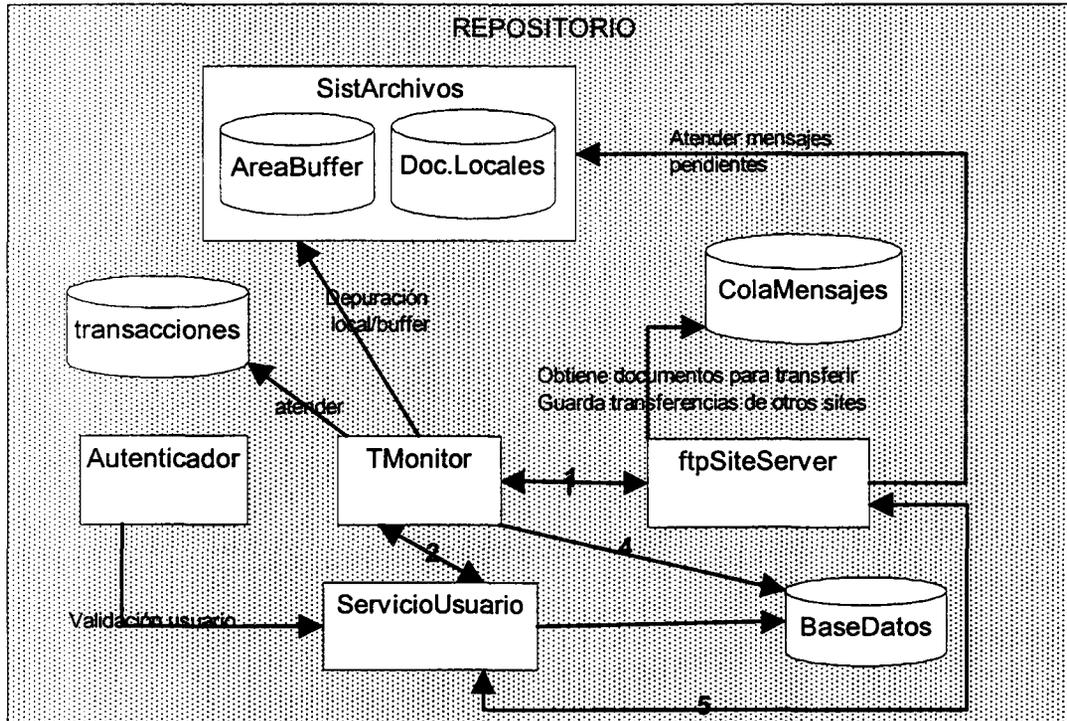
#### Constitución interna

El repositorio administrado por el sitio se puede subdividir en varios procesos y estructuras de datos. Las interacciones entre ellos se pueden ver en la figura 5.5.

- Tmonitor: proceso encargado de los procesos administrativos periódicos. Realiza las tareas de depuración, validaciones de consistencia de documentos

físicos contra documentos registrados, control de transacciones y mantenimiento de los parámetros generales de operación del repositorio. Este proceso deberá ejecutarse ininterrumpidamente.

- ServicioUsuario, que atiende toda interacción con el usuario acerca del contenido del repositorio durante las conexiones de los mismos, y le provee una visión transparente del mismo al abstraer detalles de localidad y mecanismos de comunicación. Los servicios que provee pueden ser consultas o recibir peticiones de creación, modificación o copia de algún recurso compartido. La única excepción a esto es el módulo autenticador.
- Autenticador, que valida que el usuario que intenta conectarse sea válido en RICA y para el sitio donde reside el servicio, mediante un esquema simple de palabra clave de acceso. Este módulo bien pudiera ser una parte del ServicioUsuario, pero se provee por separado ya que en un futuro pudiera ser sustituido por un mecanismo más robusto de seguridad.
- SistArchivos, que abstrae la representación física de los archivos de acuerdo a la plataforma específica, con el fin acceder a los recursos de manera uniforme.
- Cola de mensajes, que almacena los mensajes de entrada/salida desde y hacia la mensajería.
- Transacciones pendientes, que permiten llevar un control de las requisiciones pendientes de un cliente y permiten tomar cursos de acción ante fallas y/o demoras.
- Base de datos, que contiene información sobre documentos manejados, parámetros específicos de configuración y otros datos importantes. Los accesos a la base de datos se pueden hacer a través de "adaptadores de objetos" que encapsulen las consultas o estatutos SQL. Ver más adelante en la parte de diseño de la base de datos.
- FtpSiteServer, para el manejo de comunicaciones con la mensajería (y en algún momento dado con otros sitios) a través de mensajes. También recibe y envía archivos en comunicación directa con su contraparte. Si recibe un evento de la mensajería, lo redirige al módulo más adecuado para manejarlo, y éstos a su vez le entregan mensajes acerca de los acontecimientos. Este punto se detalla más a fondo en la sección de interfaces con otros sitios y la mensajería. Cabe aclarar que el nombre "ftpSiteServer" no implica que las transferencias sean a través del protocolo ftp necesariamente.



- 1.- ftpSiteServer indica a TMonitor que se ha completado la transferencia de un archivo requerido, para que éste pueda completar la transacción. También le indica acerca de eventos externos relativos a acciones administrativas.
- 2.- TMonitor refleja los resultados de procesos administrativos en ServicioUsuario, cuando sean relevantes al cliente. (Ej: depuración de un documento, transacciones fallidas, etc.)
- 3.- ServicioUsuario controla información relevante al usuario acerca del repositorio
- 4.- TMonitor accesa información de configuración/parámetros de administración/información de documentos
5. ServicioUsuario genera y envía mensajes sobre eventos ocurridos para su manejo en ftpSiteServer, y éste a su vez le redirige algún evento recibido desde el exterior

Figura 5.5. Constitución interna de un repositorio local

Con respecto a la figura, cabe aclarar que la base de datos a la que se hace referencia no contiene físicamente los archivos, sino SistArchivos, sino que más bien contiene los descriptores de documentos, locales o virtuales, así como datos acerca del funcionamiento del repositorio, uso de documentos, transacciones, datos del usuario específicos para el repositorio, etc. La separación entre estos se da porque los detalles de la implementación pueden variar de una plataforma a otra, por lo que se busca aislar estas cuestiones para aumentar la portabilidad.

### 5.2.3. Interfaces con mensajería y otros sitios

Las funciones de FtpSiteServer deberán ser manejar toda comunicación externa que implique comunicaciones con otros sitios o transferencias de archivos.

Cuando ocurre un evento del que otros sitios deben estar al tanto, un módulo interno creará un mensaje y lo entregará a ftpSiteServer para que lo envíe. Para enviar un mensaje simplemente los programa en la cola de acuerdo al orden en que fueron llegando y los atiende tan pronto como sea posible.

Si la comunicación de eventos se llevara a cabo con otros n sitios de manera directa, ftpSiteServer debería mantener n-1 líneas de comunicación y sincronización, y estar al tanto de las fallas de comunicación que pudieran ocurrir con todos ellos; en lugar de ello, uno de estos sitios se especializará como mensajería para desempeñar estas funciones, con la ventaja adicional de que un sitio no necesita conocer la ubicación directa de otros, dejando estos detalles a la mensajería, y permitiendo escalabilidad si se agrupan un número limitado de servidores por mensajería.

FtpSiteServer es a su vez un "escuchador" para mensajes de entrada, los cuales también entrarían a una cola de mensajes y dependiendo de su tipo se redirigirán al módulo más adecuado para su manejo.

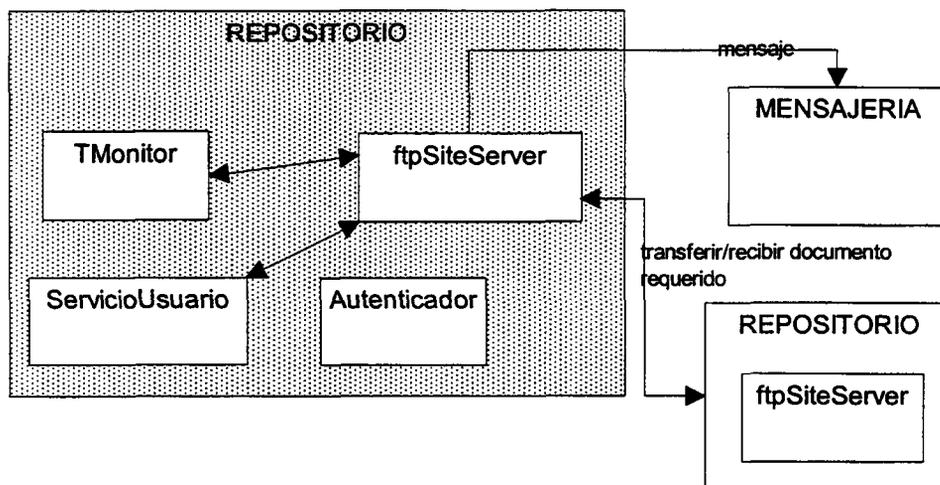


Figura 5.6. Interfaces con otros sitios y mensajería

Los tipos de mensajes que un sitio puede enviar y recibir son:

- **Broadcast:** su contenido se relaciona por eventos relativos a la visión global que debe tener un usuario del repositorio, por lo que deben ser conocidos por todos los demás sitios involucrados, es decir, se envía a la mensajería y ella deberá encargarse de distribuirlo y asegurar que todos reciban el comunicado.
- **Directos:** son mensajes que van dirigidos hacia un solo destinatario, ya sea la mensajería, o hacia otro sitio para solicitar se le envíe un archivo, donde se conoce al sitio destino de manera indirecta a través de la mensajería, y a través de la cual obtiene los datos que le permiten localizarlo y comunicarse. También se incluyen en este punto los mensajes de sincronización de tiempos, ya que es la

mensajería quien inicia estos mensajes con respecto a su propio tiempo. Todas las comunicaciones (a excepción del traspaso de archivos) se llevarían a cabo a través de la mensajería, cuando se desee distribuir un mensaje a todos los sitios, en modo broadcast, se cuenta con un tipo de mensaje directo (MsgComunica) que puede contener a su vez otros mensajes de tipo broadcast.

Cabe aclarar que la representación interna de los mensajes en un repositorio local será en forma de objetos para cada uno, de manera que se facilite su manipulación, pero su transmisión y el manejo que se les de dentro de la mensajería serán de acuerdo al estándar KQML [FIN 93].

KQML es un estándar usado en la comunidad de sistemas multiagentes para representar el contenido de un dominio, o para transmitir datos orientados a objetos, de manera autónoma y asíncrona. En este estándar, no se imponen restricciones en la forma de manejar los mensajes del lado del receptor, de manera que éste es libre de tomar un curso de acción adecuado a sí mismo. Está especificado en la forma un conjunto de mensajes con significados propios, en los cuales se pueden hacer consultas, aseveraciones, pedir realizar acciones, subscripciones y ofertas. Ya que es indiferente del contenido del mensaje, pueden tenerse subexpresiones dentro de los mismos. Las acciones deseadas a realizar que especifican los mensajes KQML son conocidos como performativas, los nombres de las mismas están reservados, de manera que se fomente la interoperabilidad del estándar.

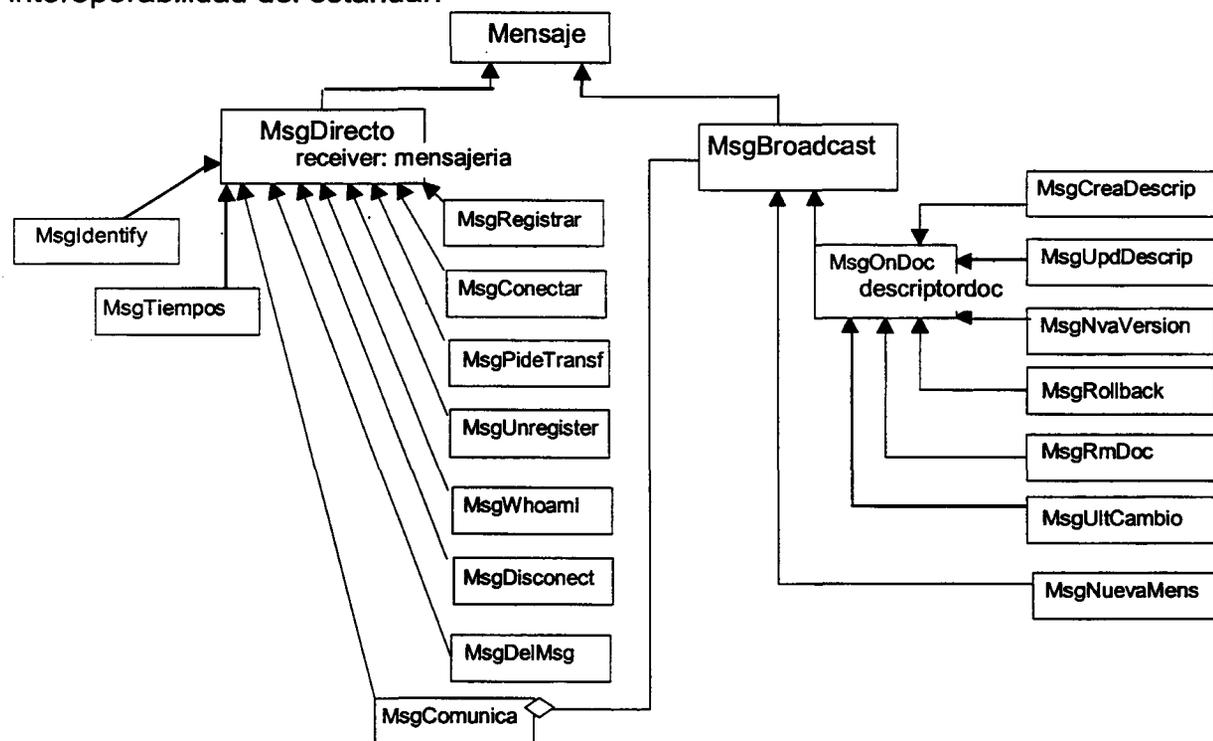


Figura 5.7. Clasificación de mensajes

El formato general de los mensajes se da de la siguiente forma:  
(Performativa serie de encabezados)

<b>Encabezado</b>	<b>Descripción</b>
: sender	Identifica a quien envía el mensaje (sitio local del repositorio)
: receiver	Identifica al receptor (ej: otro sitio, mensajería o sistema RICA)
: content	Contiene la performativa a realizar, o una subexpresión (otro mensaje)
:in-reply-to	Marca que el mensaje viene en respuesta de otro anteriormente recibido
:language	Identifica el lenguaje del contenido, ya sea un lenguaje computacional o algún otro
:ontology	Indica el dominio o tema del que se está tratando, dentro del cual tienen sentido los objetos y acciones a los que hace referencia el contenido.
:reply-with	Indica con que espera la respuesta (en caso de que la performativa lo solicite)

Tabla 5.1. Formato de un mensaje KQML estándar

Algunas de las performativas más comunes son:

<b>Performativa</b>	<b>Significado</b>
Tell	Para comunicar un hecho. Ej: notificaciones
Achieve	Pedir que el destinatario realice una acción. Ej: actualizar datos del documento, ajustar tiempo
Advertise	Anunciar que se tiene una cierta capacidad. Ej: promoverse como un sitio que puede proveer el servicio de mensajería de manera alterna
Stream-about	Preguntar acerca de un tema en particular a uno o varios sitios. Ej: localizar un documento
Register, unregister	Darse de alta-baja, en este caso, en el servicio de mensajería
Broadcast	Comunicar un hecho a múltiples sitios.
Reply	Contestar una pregunta

Tabla 5.2. Performativas más comunes en KQML

Para ilustrar el mecanismo de paso de mensajes, se dan los siguientes ejemplos:

Para mensajes directos, los mensajes pudieran ser como sigue:

Proceso de registro

1. Un sitio desea registrarse ante una mensajería
2. Forma el siguiente mensaje:  
(Register :sender me :receiver Mensajería :password xxxxxxxx)
3. La mensajería responde de esta manera,  
(identify :sender Mensajería :receiver reqsite)
4. El sitio se identifica a si mismo  
(whoiam :sender me :receiver Mensajería :message-method MessageRouter  
:content (contact-information :host lizt.mty.itesm.mx :port 1234 :email  
sef@lizt.mty.itesm.mx) :KQML-extensions http://java.sun.com/newProtocol  
:description (advertize :type sitioRepositorio))
5. En mensajes posteriores, la Mensajería podría retornar un número de  
identificador de sitio, datos para sincronizar relojes, etc.

Solicitar una transferencia de archivos

1. El sitio obtiene el identificador del sitio donde reside el último cambio de un  
documento (5, por ejemplo), entonces, solicita datos para su localización a la  
mensajería:  
(request-address :sender me :receiver Mensajería :content sitio5  
:reply-with idmsg01)
2. La mensajería contestaría con los datos del sitio no. 5  
(tell :sender Mensajería :receiver reqsite :content (contact-information :host  
lizt.mty.itesm.mx :port 1234 :email sef@lizt.mty.itesm.mx) :in-reply-to idmsg01)
3. El sitio solicitante establece una comunicación directa con el sitio 5 para pedirle  
la transferencia del archivo.

Un ejemplo con mensajes indirectos podría ser el siguiente:

2. Se crea en un sitio un nuevo documento, y desea comunicarlo a todos los  
demás sitios.

(Comunica :sender me :receiver Mensajería  
:content (CreaDescrip :content datos\_documento :sender sitio2 ))

2. La mensajería desempaquetaría el mensaje interno e identificaría que es un  
mensaje tipo broadcast, enviando a todos los sitios registrados un mensaje como  
el que sigue:

(CreaDescrip :sender Mensajería :receiver sitio<sub>i</sub> :content datos\_documento)

En cuanto a la transferencia de archivos, ftpSiteServer puede desempeñar los papeles de cliente o servidor. En el caso de ser la parte que requiere el documento, enviará la petición a la mensajería junto con el nombre del sitio al que hará la petición para que ésta lo localice y entregue el mensaje. Por su parte, un sitio que recibe una petición de enviar un documento, conoce a través del comunicado la dirección física del que lo requiere, por lo que puede establecer la comunicación directamente. En el caso del envío físico se busca evitar los intermediarios pues la longitud en bytes puede ser grande, traduciéndose en un mal tiempo de respuesta para el usuario posteriormente.

Como ya se comentó en la sección de interfaces con el usuario, ftpSiteServer se comunica con un módulo del usuario llamado ftpClient, que también será una instancia de la clase ftpProtocolLayer, pero sin comunicación con la mensajería u otros sitios diferentes al que lo atienden.

### **5.3 Interfaces con RICA**

El repositorio tiene como objetivo proveer una extensión a los servicios de RICA, por lo cual debe integrarse y complementarse con éste sistema. Las interfaces necesarias son básicamente de 2 tipos:

#### **1. Consultas de información:**

- Para validación de conexiones de usuarios (nombre, password)
- De usuarios locales (generales, áreas de interés, grupos a los que pertenece)
- Información general de grupos y su constitución
- Información de áreas de interés (jerarquía, clasificación)

Actualmente, los servicios de RICA utilizan una base de datos relacional basada en el producto miniSQL de Hughes [HUG 98]. El acceso a la información contenida en la misma se realizará mediante estatutos estándar de SQL, aunque se encapsulará este comportamiento mediante un "adaptador de objeto" dentro de ServicioUsuario.

#### **2. Petición de servicios a RICA:**

- Petición de notificación de eventos a usuarios/grupos/áreas de interés específicas
- Se asume que la replicación de información común entre el repositorio y otros servicios de RICA, información como la de usuarios, grupos y asociaciones a entre grupos y usuarios, en un esquema distribuido, deberá ser provisto por RICA. También es posible la incorporación de campos de fechas para registrar cuando se agregaron nuevos registros.
- Es necesario que antes de permitir la baja de un usuario, se asegure en RICA que, si éste cuenta con documentos de los cuales es el dueño, ceda sus derechos de administración a algún otro usuario o que los borre del repositorio.

### 5.3.1. Notificación de eventos

Los eventos de interés posibles pueden ser:

<b>EVENTO</b>	<b>CANDIDATOS A RECIBIRLO</b>
Creación de documento	Usuarios del mismo grupo que el creador Usuarios con áreas de interés del documento
Creación de una nueva versión	Mismo grupo que el creador Áreas del documento Usuarios actuales (lectores)
Eliminar documento	Grupo Usuarios actuales
Promoción (exportar documento)	Grupo Área
Creación comentarios o notas sobre documento	Creador del documento Grupo
Solicitar acceso a un documento	Lista de administradores y creador del documento
La programación de la depuración automática de un documento	Lista de administradores y creador del documento

Tabla 5.3 Eventos de notificación y candidatos de los mismos

### 5.3.2. Implementación

El servicio de notificación será provisto por RICA, al cual manejará los eventos arriba mencionados bajo una misma clasificación ("eventos del repositorio"), donde se recibirá el texto completo en un mensaje con todos los datos necesarios, necesitando de RICA la distribución del mensaje a los usuarios finales.

El módulo ServicioUsuario dirigirá esta petición hacia el sistema RICA. La implementación del mismo se realizará en JatLite, donde un módulo de software, objeto o agente envíe mensajes KQML a su contraparte en RICA para que interprete los mensajes enviados y los agregue a la tabla de donde más tarde un proceso monitor realizará las notificaciones.

El mensaje a enviar se implementará como una subclase del estándar KQML (la implementación en java de la clase KQMLMessage de JatLite lo permite), de manera que se incluyan encabezados específicos para manejar listas de grupos, listas de áreas y listas de usuarios específicos como destinatarios finales del mensaje.

Ejemplo:

```
tell
:sender Repositorio
:receiver NotificacionRICA
```

```
:grupos 1,2,3
:usuarios 1,7,10
:areas K.5,A.2.1,B.1
```

:content "Se ha creado una nueva versión para el documento 'Lógica Difusa', por Armando Perales, el 10-nov-1999 a las 4:55".

Una alternativa similar es manejar estas listas dentro del contenido, con algunas diferencias en cuestión de la facilidad de la implementación.

Para la recepción de la confirmación, es decir, saber si la notificación llegó a RICA y fue procesado por el sistema, se puede manejar de 2 formas:

- a) Pidiendo noticias explícitamente acerca del mensaje, por ejemplo, usando un mensaje KQML con :action ask-about.
- b) Un proceso dedicado a recibir confirmaciones de RICA

Para cada mensaje de notificación enviado, sería necesario mantener una lista de las mismas, de manera que, por cada confirmación de notificación obtenida por parte de RICA, se marcara como realizada en la lista, de lo contrario, reintentar en un tiempo posterior.

#### **5.4 Interfaces con otros medios digitales**

La intención del repositorio es ser una herramienta para soporte del trabajo temporal, sin embargo, los documentos que de este trabajo se generen pueden ser vistos como productos finales que es deseable divulgar. Por interfaces con otros medios digitales se entiende que algún otro sistema o medio digital puede ser un cliente potencial de los documentos generados o que pudieran ser vistos como "medios de distribución", como por ejemplo, una biblioteca digital. La interface con dichos medios puede requerir información extra, consideración explícita de protocolos para intercambio de información, etc.

Ya que no se conocen por adelantado aquellos sistemas externos con los que habrá la necesidad de establecer una comunicación, se plantea aquí un modelo general, el cual se validará con la implementación de un caso específico, como es la exportación de documentos a la biblioteca digital Phronesis [PHR 99], desarrollada en el Centro de Investigación en Informática del ITESM, en 1998.

El modelo genérico propuesto puede apreciarse en la figura 5.8.

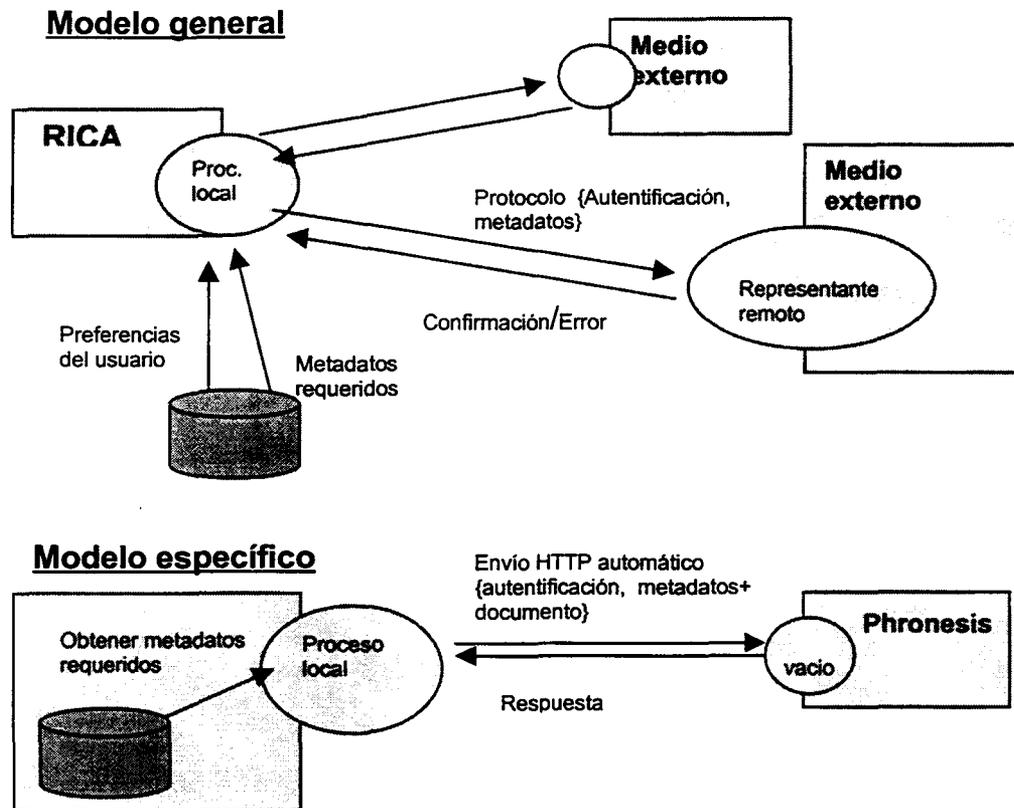


Figura 5.8. Modelo general y específico para interfaces con medios digitales externos.

En el diagrama puede apreciarse la existencia de un proceso local, encargado de iniciar la transferencia hacia el sistema externo, en base a preferencias del usuario y parámetros y metadatos necesarios para el procesamiento de ese medio externo. El proceso de transferencia puede ser automático o semiautomático, donde, en el último caso, pueda requerirse información extra del usuario antes de realizar físicamente el envío. En el medio externo debe existir un proceso representante que se encargue de complementar el envío de manera automática, que en ocasiones será un proceso vacío por no requerirse mayor procesamiento.

El proceso a llevar a cabo es el siguiente:

1. Agregar a las preferencias del usuario el tipo de bibliotecas digitales que le gustaría acceder, lo cual puede determinar el tipo de metadatos que debe introducir cuando crea un nuevo documento.
2. En el momento de un envío, deberá especificarse el medio al que se desea exportar y posiblemente deba llenar datos o confirmar algunos extraídos previamente de manera automática.

3. Agregar un proceso representante en la biblioteca que se desea acceder, con la cual se pueda comunicar de manera uniforme y que se encargue de manejar los detalles propios de cada situación, al servir de intermediario entre las rutinas reales que realizan los procesos y el repositorio. De cada medio digital con el que se haga interfaz debe tenerse la siguiente información:
  - metadatos necesarios
  - protocolo de comunicación o de envío (http, ftp, etc.)
  - formatos permitidos
  - tamaños de documentos permitidos
  - tipo de autenticación que llevan a cabo

Los metadatos necesarios para cada tipo de medio externo a acceder deben pedirse en el momento mismo del envío y no al momento de crear los documentos, por las siguientes razones:

- No siempre se puede saber por anticipado a que medio exportar un documento
- Llenar todos los metadatos de varios tipos de medios externos puede hacer que el formulario de registro inicial sea demasiado largo.
- Algunos datos no están completos o es deseable que sean confirmados antes de ser enviados.

#### 5.4.1. Interfaz con sistema Phronesis

En el caso específico de Phronesis, se tienen los siguientes requerimientos específicos para la interfaz:

- Proporcionar la información necesaria para procesar el documento en el otro sistema, esto es, en forma de metadatos.
- Enviar documento en línea de manera semiautomática, desde el sistema RICA, donde el usuario pueda confirmar la información proporcionada.
- Agregar un representante o proceso a Phronesis que permita enviar mensajes de manera transparente para el sistema Phronesis.

#### Proceso actual

Enseguida se describe el proceso que se lleva a cabo en Phronesis para el ingreso y búsqueda de documentos.

Para agregar nuevos documentos:

1. El usuario debe acceder la página WEB de la biblioteca que va a acceder y debe ser autenticado por el sistema, pues solo se permite añadir documentos a usuarios autorizados.
2. Se presenta un formulario acerca del documento en cuestión, el cual debe ser llenado manualmente. Al finalizar, se confirma al usuario el éxito o fracaso de la operación.
3. El documento será procesado de la siguiente manera:

- 3.1. Se verifica el formato del mismo
- 3.2. Se extrae el texto del mismo y se almacena de esta forma, siempre y cuando sea un documento de tipo texto, Postscript, PDF, HTML o RTF.
- 3.3. Se comprime y se almacena.
4. El documento no estará disponible de forma inmediata, pues deberá correrse un proceso de "reconstrucción de la biblioteca", el cual prepara índices para búsquedas posteriores.

Para realizar búsquedas:

4. El usuario accesa la página WEB de búsqueda. Es posible que la biblioteca esté configurada de manera que se requiera autenticación para la búsqueda.
5. Se presenta un formulario de búsqueda donde el usuario introduce las palabras claves y puede seleccionar entre 1 o varias bibliotecas registradas para hacer su búsqueda.
6. Se presentan los documentos encontrados, en forma compacta. El usuario puede, para cada uno de ellos, revisar sus metadatos, texto completo u obtener el documento original.

Como puede observarse, el sistema Phronesis requiere un conjunto de metadatos ("datos acerca de los datos") para procesar adecuadamente el archivo y para realizar procesos de búsqueda. De esta forma, la información requerida a enviar junto con el archivo es la siguiente:

*Título* –

Es el nombre del documento.

*Tema* –

Es la palabra o frase que especifica el área de especialidad del documento, o bien, una palabra clave del documento. Un documento puede tener uno o varios temas o palabras clave.

*Descripción* -

Se trata de un texto para describir el contenido del documento. En el caso de artículos se puede poner el resumen o "abstract".

*Idioma* –

Se trata del idioma en el que está escrito del documento.

*Autor* –

Nombre de la o las personas que crearon el documento.

*Publicado por* –

Casa editorial, departamento de universidad u organización encargado de la publicación del documento.

*Fecha de Creación* –

Fecha en la cual se escribió el documento. Para este dato se especifica solamente el mes y el año.

*Tipo de Documento* –

Especifica el tipo del documento, por ejemplo: artículo, reporte técnico, tesis, etc.

*Formato* –

Formato en el que está grabado el archivo que contiene el documento y que servirá para saber qué tipo de software es necesario para acceder el documento.

*Identificador -*

Clave única que identifica al documento dentro de la biblioteca digital, esta clave es generada automáticamente cuando se recibe un documento. Este identificador es único para todo el sistema de bibliotecas Phronesis.

Para facilitar el envío, es deseable recolectar la mayor parte de esta información de manera automática, pero aún se requiere la confirmación del usuario pues no todos los datos solicitados tienen una correspondencia directa con la información que se guarda en el repositorio, o para que efectúe el llenado manual de información no disponible.

Metadato	Disponible	Comentarios
Título	Si	
Tema	Si	Correspondería con las áreas de interés. En Phronesis, el usuario puede introducirlo libremente, por lo que en un determinado momento pudiera agregarse algún otro manualmente.
Autor	Si	AMBIGÜEDAD: Puede ser el primer autor o una composición de todos los autores.
Descripción	Si	AMBIGÜEDAD: Puede estar basado en la descripción inicial del documento, o en una composición de todas las notas.
Idioma	NO	
Publicado por	NO	En este caso, pudiera proveerse un valor por defecto que haga referencia al repositorio RICA (Proyecto RICA- repositorio).
Fecha	Si	Correspondería con la fecha de la última versión disponible
Tipo de documento	NO	En Phronesis se tienen algunos tipos predefinidos, aunque en un futuro se piensa permitir especificarlo al usuario.
Formato	Si	Cabe aclarar que el sistema Phronesis maneja ciertos formatos oficiales a los cuales da un tratamiento especial (texto, Postscript, HTML, PDF, RTF), y los demás son tratados como "Otros". Esta distinción es importante porque se basa en ella para búsquedas posteriores en base al contenido o solamente en relación a sus metadatos.

Tabla 5.4 Metadatos a exportar a Phronesis

La última consideración importante es la autenticación del usuario en el momento del envío. Se puede optar por una de estas alternativas:

- Crear un usuario genérico para acceso a Phronesis, representante del sistema RICA.
- Cada usuario debe personalmente pedir su autorización en Phronesis, y suministrarlo manualmente durante la transferencia.

### **5.5 Diseño detallado**

El diseño detallado se presenta en el anexo 2, y consiste en:

- la definición de la base de datos,
- pantallas del prototipo e interacción con el usuario,
- diagramas de especificación funcional (diagramas de interacción de objetos).

## **Capítulo 6.- Manejo de versiones**

### ***6.1 Alternativas para el manejo de versiones***

En muchos sistemas de manejo de versiones comerciales, como Microsoft Visual SourceSafe, CVS y otros, se tienen mecanismos de bloqueo pesimista mediante comandos para proteger-desproteger (“check in-check out”), donde un usuario separa para su uso exclusivo un documento, y durante este tiempo, todos los demás usuarios podrán, en el mejor de los casos, leer, más no realizar modificaciones, sobre el mismo. El tiempo de bloqueo es determinado por el usuario, y es posible que se realice aún cuando el usuario finalmente no efectúe ninguna modificación, tiempo durante el cual otros interesados deben esperar a que sea liberado. Esta forma de trabajar es adecuada en situaciones donde se trabaja de manera serializada y se desea evitar conflictos de concurrencia sobre los objetos compartidos, es decir, donde cada miembro del grupo realiza una aportación al trabajo en conjunto, sobre el cual otro miembro se basará para continuar con su propio trabajo, como por ejemplo, en el desarrollo de software.

Sin embargo, este tipo de control sobre el documento es considerado como restrictivo el contexto de la investigación colaborativa, donde las aportaciones pueden ser no-lineales y cada una de ellas es valiosa por lo que debe ser considerada y puesta a consideración entre los miembros del grupo. Mientras estas ocurran en secuencia, los miembros podrán leerlas y crear nuevas versiones a partir de las mismas, pero cuando 2 o más necesiten hacer alguna aportación, deberían poder hacerlo libremente sin tener que esperar un tiempo arbitrario.

A diferencia de estos sistemas, el repositorio RICA se implementa con una perspectiva de bloqueo optimista, es decir, se permite acceder y preparar modificaciones de los documentos a todos los usuarios autorizados, con el fin de maximizar la disponibilidad y evitar el monopolio de un determinado recurso. Para ello, se registran los accesos de lectura a cada versión del documento, y todo cambio que éste realice al mismo se solicita en el momento mismo en que va a efectuar físicamente la actualización (subir el archivo al servidor).

### ***6.2 Manejo de versiones concurrentes***

El caso de conflictos de concurrencia se ilustrará con un ejemplo: supongamos que el documento 1 versión 2 es leído por un grupo dentro del repositorio RICA. A partir de esta misma fuente, dos usuarios deciden de manera independiente y

simultánea agregar la versión 3, por lo que pueden implementarse estrategias diferentes:

a) Soporte a varias ramas de versiones del mismo documento. En este caso, todas las versiones se agregan bajo la misma etiqueta o número, y los miembros del equipo o un moderador deben decidir cual será la definitiva, o formar una nueva versión 4 con base a todas las aportaciones previas, con el apoyo de herramientas de comparación de dichas versiones. Esto representa las siguientes dificultades:

- Ya que el modelo de cooperación en el repositorio es asíncrono, puede tomar mucho tiempo decidir entre todos los miembros del grupo.
- En el caso más general, es decir, el repositorio distribuido entre varios sitios diferentes, se producen varios escenarios donde se pierde o confunde el significado de una versión común a partir de varias propuestas, pues es posible que varios nodos conozcan sólo parcialmente la información de versiones, de manera que hagan sus aportaciones de manera independiente y no sean consideradas en la solución común, y a la vez, usuarios que revisen la solución común podrían asumir que ésta considera todas las propuestas, cuando en realidad no fue así.
- La designación de un moderador da el control final a una sola persona, cuando se busca un consenso.

Aún resolviendo estos conflictos, siempre es posible querer reconsiderar una o varias versiones anteriores, como la 1 o la 2 del ejemplo. Su ventaja es la simplicidad y la certeza de que son aportaciones que se dan en respuesta a una misma fuente o planteamiento.

Pudiera plantearse la posibilidad de dejar evolucionar varias ramas del proyecto o investigación de manera independiente, pero esto pudiera provocar que nunca se llegue al consenso.

b) A cada versión se le ordena de acuerdo a su llegada, y mediante el mecanismo de notificación, los miembros están al tanto de los eventos ocurridos. Los documentos se presentan en pila, y se dan facilidades de comparación de textos entre versiones, siempre y cuando sean estos conformes a los formatos oficiales soportados, en otro caso, la comparación sería de forma manual.

Esta alternativa tiene la ventaja de que permite conservar todas las versiones para su análisis posterior y es posible implementar mecanismos de comparación de versiones generales. Su desventaja es que ahora se presenta el problema de asegurar que se hayan leído las aportaciones previas y señalar la versión de referencia de la cual se parte, de manera que es confuso distinguir si se está generado una conclusión del trabajo anterior o si se está presentando una alternativa.

c) Se pueden aprovechar las ventajas de ambas, es decir, se presentan las versiones de un documento tanto por el orden que le corresponde cronológicamente como por las fuentes o referencias de las que proviene, guardando todas las aportaciones con su número de versión relativa a la última que se accesó así como un consecutivo asignado y resuelto en base a la sincronización cronológica, con lo cual es posible obtener:

- Relación semántica de contenidos en base a su referencia
- Revisar aportaciones nuevas no leídas, sea cual sea su fuente
- Conocer que versiones pueden ser resultado de la conclusión de otras anteriores

Cabe aclarar que, siempre se tendrá conocimiento de todas las versiones nuevas a través de las notificaciones del sistema, pero se considera que aún el hecho de acceder un documento no es garantía de que se le vaya a leer, por lo que se deja a responsabilidad del usuario.

En este trabajo se tomó la decisión de seguir la última opción, de manera que se pueda trazar un historial completo de las versiones y disponer de ese conocimiento permanentemente, y a la vez poder establecer un modelo de las referencias que siguieron los usuarios para hacer sus aportaciones, apoyadas además con facilidades para su revisión y consideración como lo son la notificación de eventos, herramientas de comparación de versiones y la lectura virtual, que se detalla a continuación.

#### 6.2.1. Lectura virtual

Si un usuario accesa un documento y decide hacer una nueva aportación, al no estar bloqueado el acceso al mismo, es posible que en el tiempo transcurrido entre su lectura y el momento en que desea hacer la actualización, otro usuario haya agregado anteriormente otra versión al mismo. Es posible que el primer usuario desee considerar dar un vistazo a la contribución de su compañero antes de hacer la suya propia, pero tal vez prefiera agregar la propia como alternativa o propuesta a ser considerada, independientemente del contenido del documento del compañero. Si la intención fuera la segunda, se realiza una lectura virtual, es decir, bajo el consentimiento del usuario, el sistema considerará como leído el documento, pues no puede obligársele a leer una determinada versión del documento, pero se sabe que al menos ya es de su conocimiento que existe y en un futuro tal vez pueda hacer uso de las comparaciones entre versiones para conocer las diferencias.

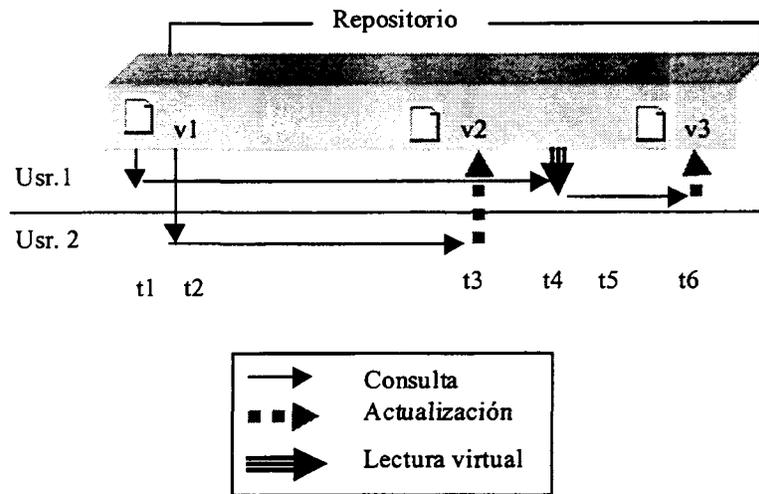


Figura 6.1. Mecanismo de lectura virtual

### 6.3 Manejo de versiones local

Desde una perspectiva local, esto es, en un solo sitio, a pesar de tener una estrategia no-bloqueante para el manejo de versiones, se debe hacer un bloqueo temporal a nivel documento desde el momento que se hace la petición de actualización hasta que se completa la transacción, con el fin de evitar conflictos de concurrencia al guardar el archivo físicamente, tanto para agregar nuevas versiones como para eliminación de las mismas.

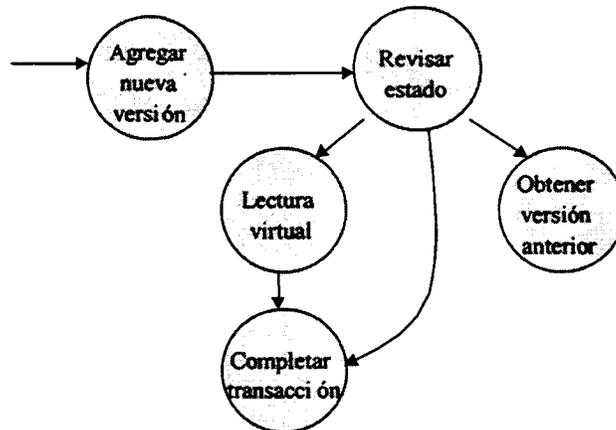


Figura 6.2. Proceso para actualización de versiones de manera local

### **1. Agregar/borrar versión -**

Petición de agregar nueva versión al repositorio, con el nuevo documento disponible para ser agregado en ese momento y los datos complementarios, o petición para efectuar el borrado de la versión superior ("rollback").

### **2. Revisar estado de versiones –**

Se revisa el efecto de operaciones anteriores sobre el documento de las cuales el usuario no ha tenido conocimiento, esto puede ser:

- El documento está siendo bloqueado por otro usuario que está realizando modificaciones, por lo que se le hace esperar en una cola.
- No ha accedido (leído) al menos la versión superior, lo que supone que no tiene el conocimiento del contenido de la misma. En este caso, se le pide confirmar al usuario si desea cancelar su petición y obtener el documento anterior para examinarlo, o si desea continuar con su proceso, en cuyo caso se efectúa una lectura virtual.
- Se revisa si el cambio operado fue un borrado de versión, en cuyo caso se hacen los ajustes necesarios de número de versión. Se asume que el usuario ha sido notificado de este suceso, pero si ocurre durante la espera, se hace aclarar al final del proceso.

En el caso excepcional que ocurra un acceso concurrente, se espera durante un tiempo predeterminado en una cola a que termine el bloqueo físico del documento, en orden de "primeras llegadas, primeras salidas". Si se da el caso de que se cumpla un tiempo de espera específico, se pide reintentar la operación.

### **3. Completar transacción –**

Cuando el usuario obtiene el control del documento, bloquea el documento para asegurar escritura única y realiza sus cambios. Enseguida se generarán los mensajes de notificación correspondientes.

## **6.4 Manejo de versiones distribuidas**

Cuando se incluye a todos los servidores participantes, el esquema cambia debido a que debe proveerse cierta tolerancia a fallas para maximizar la disponibilidad del servicio. Aún cuando un sitio haya podido agregar un cambio satisfactoriamente, aún es necesario que lo comunique a los demás y que éstos lo registren de manera local. El mecanismo de espera por turno para hacer modificaciones también cambia pues se debe notificar a otros sitios de esta intención. Problemas potenciales pueden ser que uno o varios de estos nodos podrían estar temporalmente fuera de servicio o desconectados, por lo que es necesario establecer un algoritmo que permita resolver conflictos posteriores de sincronización. En particular, si 2 sitios creen tener la última versión de un documento y no están en comunicación entre sí, debe ser posible agregar los

cambios y más tarde, cuando las condiciones lo permitan, se deberán reordenar adecuadamente los documentos.

Se pueden presentar los siguientes escenarios durante cada fase del proceso:

<b>Solicitud de servicio (agregar o borrar una versión)</b>	<b>Revisar estado de versiones</b>	<b>Lectura virtual - obtener versión</b>	<b>Completar transacción</b>
<p><b>Falla posible:</b> El servicio local del repositorio se ha interrumpido por alguna razón, se reinicia y se encuentra des-sincronizado.</p>	<p>La comunicación con otros sitios se pierde parcial o totalmente.</p>	<p>Se sabe que existen una o más versiones que no han sido leídas virtualmente, pero no es posible obtener físicamente el documento.</p>	<p>Se pierden las comunicaciones entre sitios o falla el servidor local donde reside la transacción, o no completa la fase de notificación a otros sitios.</p>
<p><b>Medida a tomar:</b> Generar nuevas versiones con base en las disponibles de manera local, y registrar el tiempo en que ocurren los eventos con la última diferencia conocida. Cuando se conozcan los parámetros de tiempo actualizados, ajustarlos en los documentos.</p>	<p>Generar nuevas versiones con base en las disponibles de manera local. Los conflictos de ordenamiento se resolverán cuando se restablezcan las comunicaciones.</p>	<p>Solo se dispone de las opciones de lectura virtual o cancelar.</p>	<p>Se programa el reenvío del mensaje de notificación a la mensajería hasta que esta de acuse de recibo, con lo cual se delega el problema de distribución del mensaje.</p>

Tabla 6.1 Eventualidades posibles en el manejo de versiones

En los primeros 2 escenarios, ya que las versiones se crean tomando en cuenta el criterio cronológico y lógico a la vez, es posible reordenar con base al orden real en que tuvieron lugar los eventos, para permitir a un usuario mantenerse actualizado consultado sólo los documentos más recientes, y a la vez establecer un modelo mental con relación a la última referencia que se tuvo para una determinada aportación.

## **Capítulo 7.- Evaluación comparativa de la herramienta**

### **7.1 Objetivos y métodos**

La evaluación conceptual de la herramienta propuesta se hará con base en la comparación con otras similares, es decir, herramientas dirigidas o con capacidades para trabajo colaborativo en grupos dispersos por medio de un repositorio electrónico de trabajo temporal. Los factores a comparar se concentrarán únicamente en los puntos de interés para la tarea y tipo de problema expuesto en esta tesis, en contra de aplicaciones con un alto o buen grado de aceptación.

Objetivos a evaluar:

- Características y servicios de apoyo al trabajo en grupo, para tareas poco estructuradas, a diferencia de otras situaciones específicas como workflow, por citar alguna.
- Estrategia de distribución
- Manejo de versiones y del historial de un documento
- Control de acceso, es decir, el modo como se comparten documentos dentro del repositorio.
- Notificación y facilidades para mantener a los usuarios al tanto de las actividades de otros miembros del grupo.

Las herramientas elegidas para dicha comparación son las estudiadas en el capítulo 2 como caso de estudio, BSCW y Lotus Notes, donde para éste último se considera solamente la parte de la funcionalidad que tiene relación con el espacio compartido y en cuanto a lo aplicable al contexto de esta tesis.

### **7.2 Comparación y evaluación**

#### **7.2.1. Comparación**

Habiendo estudiado las características de estas 2 herramientas desde el punto de vista de la investigación colaborativa, se compararán los puntos relevantes de todas ellas para obtener una evaluación de la herramienta propuesta.

### 1. Registro y notificación de eventos

<b>BSCW</b>	Es completa pues se extiende a todos los objetos y posibles eventos, incluyendo nuevos usuarios y otros. Es interna al repositorio aunque puede solicitarse el envío de correo electrónico de los mismos.
<b>Lotus Notes</b>	Por medio de correo electrónico. No se trata de un servicio básico, sino que, dependiendo de la aplicación, puede agregarse mediante programación.
<b>Repositorio En RICA</b>	Se incluye a nivel repositorio o externo, de acuerdo a las preferencias del usuario. Se complementará con otros servicios de notificación provistos por otros servicios de RICA, ya sea por medio de correo electrónico o en un visualizador, pero los eventos relevantes al repositorio se mostrarán en la aplicación de igual manera.

### 2. Integración con servicios complementarios

<b>BSCW</b>	Provee servicios básicos para enviar correo electrónico, crear grupos de notificación y añadir ligas para compartir.
<b>Lotus Notes</b>	Incluye diversos servicios complementarios importantes para la colaboración, como calendarización de actividades, correo electrónico, soporte a la edición colaborativa, etc.
<b>Repositorio En RICA</b>	Complementará otros servicios de RICA (chat, monitoreo de páginas web, etc.).

### 3. Control de acceso

<b>BSCW</b>	No cuenta con un enfoque muy flexible de control de acceso. Es adecuado el poder heredar privilegios de acceso sobre una jerarquía de documentos.
<b>Lotus Notes</b>	Es adecuado pues da soporte a varios roles y a cada uno se le definen capacidades de acceso de acuerdo a ello. Este control de acceso deberá ser programado, es decir, dependerá de cada tipo de aplicación.
<b>Repositorio En RICA</b>	El esquema actual no es demasiado flexible, pero incorpora la delegación de tareas al permitir uno o más administradores para ayudar a reducir la burocracia.

### 4. Control de modificaciones sobre los documentos

<b>BSCW</b>	Se bloquea completamente un documento para modificación, más no para lectura.
<b>Lotus Notes</b>	Es muy interesante la posibilidad de bloquear o esconder incluso porciones de documentos para su edición colaborativa, lo cual sirve

	en escenarios tanto síncronos como asíncronos.
<b>Repositorio En RICA</b>	El bloqueo "optimista" de los documentos maximiza la disponibilidad de acceso a personas autorizadas.

#### 5. Independencia de plataforma

<b>BSCW</b>	A través de tecnología web.
<b>Lotus Notes</b>	Existe la opción de utilizar tecnología web.
<b>Repositorio En RICA</b>	A través de tecnología web.

#### 6. Soporte al control de versiones

<b>BSCW</b>	Se soportan versiones, y es especialmente importante el hecho de que maneja estados de revisión sobre las mismas. También es posible crear ramificaciones de versiones, es decir, desarrollar varias rutas de trabajo a partir de un mismo documento.
<b>Lotus Notes</b>	No se provee esta funcionalidad.
<b>Repositorio En RICA</b>	Plantea la posibilidad explícita de manejar diferentes puntos de vista para una misma versión, buscando que se discutan todas las alternativas y finalmente se llegue a un acuerdo. Sin embargo, esto sería básicamente un protocolo social, el cual sería apoyado de mejor manera a través del manejo de estados de revisión de cada versión. Es importante tomar en cuenta que al proveer 2 vistas de las mismas, esto es, en orden cronológico y en orden de referencia, en complemento con la de notificación de eventos y la comparación de versiones, es posible mantener a los usuarios al tanto de sucesos nuevos y ubicarlos adecuadamente de acuerdo a su origen.

#### 7. Manejo del historial de documentos

<b>BSCW</b>	Provisto de manera muy completa, para todos los objetos del repositorio.
<b>Lotus Notes</b>	No se encuentra como una de las funcionalidades básicas, pero sería posible agregarla mediante programación dependiendo de cada tipo de aplicación.
<b>Repositorio En RICA</b>	Incluye la funcionalidad básica: lecturas y modificaciones sobre documentos. Los comentarios externos son visiblemente accesibles para todo documento.

## 8. Herramientas para soporte a la refinación social

<b>BSCW</b>	Es posible asignar una calificación con base a un criterio bien definido a los documentos (bien, regular, excelente, etc.), además de permitir hacer discusiones y comentarios externos a los mismos. También se apoya mediante la asignación de estados a las versiones de un documento (liberado, revisado, etc.)
<b>Lotus Notes</b>	No se detectó ninguna característica específica, pero es posible que notas y comentarios a un documento existan en su área de trabajo.
<b>Repositorio En RICA</b>	Se permite la coexistencia de una misma versión con visiones alternas para un mismo documento, y se permite la comparación de las mismas, al menos de formatos soportados oficialmente.

## 9. Consistencia y capacidad de distribución

<b>BSCW</b>	Carecen de una estructura replicada, por lo que están expuestos a demoras en redes y posibles fallas totales del servicio propias de internet. Control simplificado.
<b>Lotus Notes</b>	Es muy importante el soporte al trabajo móvil o fuera de línea mediante el mecanismo de replicación, aunque la replicación deberá ser explícita. Mediante la definición de dominios, se permite la escalabilidad de la red y dar soporte a un mayor número de usuarios.
<b>Repositorio En RICA</b>	Como cada sitio se encarga de dar atención a sus usuarios locales y se guardan copias de documentos localmente, se reduce la necesidad de hacer muchas transferencias lentas a través de la red. No se pierde la transparencia, pues aún en un modo desconectado (del sitio que da el servicio con respecto a otros sitios de RICA) es posible dar el servicio, y el método de control y la notificación de versiones permiten la resincronización posterior. También tiene posibilidades de crecimiento a través del esquema federativo propuesto.

## 10. Búsquedas y exploración del repositorio

<b>BSCW</b>	Mediante las búsquedas personalizadas se puede acceder a objetos de acuerdo a criterios de interés personal y se pueden configurar al gusto del usuario. Estas búsquedas se extienden a través de los espacios de trabajo.
<b>Lotus Notes</b>	Se permiten diferentes vistas de un mismo documento, agrupados de acuerdo a las preferencias del usuario y en base a criterios que pueda considerar importantes (prioridad, autor, etc.). También es posible indexar el contenido de texto para búsquedas textuales.

	Además, se da la opción de poder establecer enlaces entre documentos que permiten navegar más fácilmente entre ellos y obtener una visión más completa.
<b>Repositorio En RICA</b>	La búsqueda por el momento está restringida a aspectos tales como autores, grupos, áreas de interés y fechas de modificación, sin embargo, cabe aclarar que estos son aspectos muy comunes y relevantes. La búsqueda no limita el mostrar la existencia de aquellos objetos sobre los cuales se tiene acceso, de manera que pueda solicitarse permiso posteriormente a los administradores del mismo. En las otras alternativas es necesario pertenecer al dominio o ser "invitado" a compartir el área de trabajo, pero en este caso se pretende favorecer el establecimiento de contactos en base a intereses.

### 11. Contexto de los artefactos compartidos

<b>BSCW</b>	Los espacios de trabajo van asociados a un tema en particular, el cual relaciona los objetos que allí residen en torno a un tema en particular. Para usar un objeto en diferentes contextos, es necesario copiarlos a otros espacios de trabajo, y las 2 copias podrían tener una diferente evolución diferente a partir de este punto, lo cual puede ser deseable o no, según el caso.
<b>Lotus Notes</b>	También se usan áreas de trabajo y estas a su vez pueden anidarse en portafolios, lo cual permite la especialización.
<b>Repositorio En RICA</b>	El contexto lo proporcionan las áreas de interés, y también se incluye el concepto de especialización ya que estas áreas pueden estar organizadas jerárquicamente. Un objeto puede pertenecer a 1 o varias áreas de interés, pero existe como una sola unidad. En este caso, un problema potencial puede ser el que se deseara seguir diferentes caminos para un mismo documento dependiendo del área de interés involucrada, donde el soporte a la ramificación de versiones de un mismo documento pudiera ser útil.

### 7.2.2. Evaluación

A partir de la comparación realizada en el apartado anterior, se pueden obtener las siguientes conclusiones:

- La propuesta de esta tesis incorpora muchos de los conceptos de diseño válidos para el trabajo colaborativo, tal como las herramientas comparadas.
- Se difiere en puntos de vista relativos a como compartir los recursos, como hacerlos públicos, el acceso y organización de los mismos, y el nivel de control sobre los mismos.

- Entre las diferencias que presenta la propuesta con respecto a los casos estudiados, se detectaron características que se consideran como fortalezas, mientras que otras son debilidades potenciales y constituyen áreas de oportunidad para implementar mejoras.

**Fortalezas:**

- Existe la posibilidad de establecer contactos nuevos y favorecer la colaboración al permitir compartir documentos de acuerdo a áreas de interés y permitir que nuevos usuarios soliciten acceso, de manera que no solo se limita únicamente a grupos totalmente definidos y cerrados.
- El manejo de versiones, mediante vistas en orden cronológico y por referencia permite no perder el contexto y al mismo tiempo mantenerse actualizado.
- El manejo de versiones permite trabajar en modos en línea como fuera de línea a pesar de la distribución.
- El esquema distribuido permite escalabilidad, y al mismo tiempo, el uso de mensajerías permite tolerancia a fallas y reduce la carga de comunicaciones entre todos los sitios.

**Debilidades/oportunidades:**

- No se soporta la posibilidad de llevar a cabo evolución diferente de un documento a través de ramificación.
- Sería conveniente permitir la evaluación explícita de versiones, mediante categorías definidas y añadir estados de revisión. Sin embargo, cabe aclarar que ésto puede realizarse en el diseño propuesto mediante protocolos sociales y comunicación informal entre los participantes, o a través de comentarios externos a los documentos.
- Se puede incorporar permisos de acceso de acuerdo a la definición específica de roles de usuarios.

## **Capítulo 8.- Conclusiones y trabajos futuros**

### **8.1 Conclusiones**

Durante el transcurso de esta tesis se han definido y establecido una especificación arquitectónica, el diseño detallado y los mecanismos y herramientas tecnológicas que hacen factible implementar un repositorio virtualmente centralizado haciendo uso de recursos distribuidos, y que es, de acuerdo al marco teórico y la evaluación comparativa, adecuado para la tarea y usuarios finales a que se desea apoyar.

Una de las propuestas más importantes durante el desarrollo de este trabajo fue el manejo de versiones, el cual se presenta de forma muy diferente al manejo que se hace en otras herramientas similares. De manera conceptual, se ha probado que es valioso permitir esquemas no-bloqueantes para favorecer el libre flujo de ideas y democracia en la consideración de las mismas, así como la combinación con la notificación de eventos que permita mantener al día a los usuarios de los eventos relevantes, pero con la flexibilidad suficiente para explorar las versiones de la forma que más se adecue a sus necesidades, esto es, revisando en orden cronológico, en relación con la procedencia o referencia de una determinada contribución, mediante lecturas virtuales donde el usuario asume la responsabilidad de no consultar un determinado documento o mediante la comparación de documentos, que le permita ubicar más fácilmente las diferencias en contenido textual entre ellos.

Con respecto a la transparencia del servicio al usuario que debe ofrecer la herramienta, ésta puede apreciarse conceptualmente en el hecho de que el servicio no sería interrumpido por fallas en la red, y tampoco se vería afectado el flujo de contribuciones locales por el bloqueo de un documento en otro sitio remoto. Sin embargo, parte de la transparencia tiene que ser sacrificada en caso de no contar con una copia local de una versión específica para un documento cuyo origen no fue local, en caso de fallas en la red, ya que el sistema necesariamente tendrá que indicar al usuario que en este momento no puede atender a su petición para tomar algún curso de acción. En cuanto a la búsqueda de independencia de plataforma, se pudo aprovechar el hecho de que hay disponibles actualmente varias tecnologías que existen para la mayoría de las plataformas de software y hardware, entre ellas se encuentran lenguajes tales como java, html o perl, usadas en el prototipo, y herramientas comerciales o de licencia "gnu", como son la base de datos miniSQL. Se observaron también limitaciones que pueden imponer sistemas de archivos diferentes, pero estas

pueden ser aisladas con capas de software, lo cual se propuso en la forma de "adaptadores de objetos" en el diseño detallado (anexo 2).

Por otra parte, las interfaces con medios externos, como fue el caso de Phronesis, también se enfrentan a la limitación de ser muy específicas en cuanto a entradas y salidas se refiere, y por el hecho de que el significado de la información cambia de un sistema u otro, por lo que añadir el soporte a nuevos medios digitales externos puede requerir consideraciones especiales y programación para la integración con el repositorio. Aunque es difícil prever para cada tipo de herramienta externa, al menos en la implementación de la interfaz con Phronesis, el modelo general propuesto fue adecuado.

El objetivo de la integración con RICA, para proveer una plataforma más completa de colaboración, se ve cumplido al permitir compartir documentos u otro tipo de recursos temporales como un medio de comunicación directa entre miembros y no miembros con diferentes niveles de acceso y permitiendo establecer varios esquemas de trabajo, además, con la especificación del mecanismo de envío de eventos de notificación es posible mantener un nivel de conciencia de las actividades de otros aún de manera asíncrona. Con la definición de las mensajerías-repositorio local puede ser la base para la extensión de servicios ya existentes de forma centralizada a un esquema distribuido.

Finalmente, con la implementación de un prototipo se muestra la factibilidad de las funcionalidades más importantes de la propuesta de esta tesis, como son el control de acceso, manejo de versiones y mecanismos de lectura virtual, mientras otros tales como la notificación de eventos quedarán en especificación de diseño, como se menciona en la siguiente sección.

## **8.2 Limitaciones y trabajos futuros**

Por cuestiones de tiempo o restricciones del proyecto en general, el prototipo incluye solo la funcionalidad mínima indispensable. Entre las limitaciones más importantes del mismo, se encuentran el no implementar la interfaz para notificación de eventos, ya que este servicio se encuentra actualmente en desarrollo, o la implementación distribuida de la red de nodos "sitios" y mensajerías, pero se proporciona en su lugar una especificación a detalle de la misma en los Anexos 1 y 2, y se señala que JatLite provee un framework que proporciona el manejo de comunicaciones elemental entre los mismos.

Aunque la herramienta propuesta contiene sólo los elementos básicos que debería contar una herramienta de este tipo, es la base para futuros desarrollos en los cuales puede apoyarse otro tipo de objetos compartidos y nuevas relaciones y dinámicas que puedan establecerse entre los miembros de un grupo.

Existen muchas oportunidades de añadir nuevas funcionalidades al repositorio, así como existe la necesidad de validar más allá del nivel conceptual y de diseño la propuesta. Enseguida se presentan los puntos más importantes a este respecto:

- Es necesario conducir un estudio de usabilidad para validar en el campo el diseño y obtener retroalimentación de la facilidad y adecuación de su uso.
- Buscar un esquema más flexible de control de acceso que permita más fácilmente la evolución de roles de los participantes de un grupo. Esta es un área de investigación activa en la actualidad en el área de CSCW.
- Incluir soporte a la evaluación formal ("rating") de las versiones de un documento y poder asignarles estados o etiquetas de revisión.
- Estudiar la conveniencia de permitir crear varias ramas de un mismo documento.
- Incluir un manejo con mayor granularidad de los documentos, de manera que sea posible manipular partes de documento como objetos individuales y de esta forma poder llevar a cabo búsquedas, comparaciones u otras operaciones de forma más precisa y/o poderosa.

Además, una opción interesante a futuro por explorar sería el uso de "subrayado semántico" de documentos, como el que se propone en [APP 92], en donde se permite resaltar las partes más relevantes de un documento mediante subrayado electrónico y se pueden realizar búsquedas de acuerdo al contenido semántico del mismo.

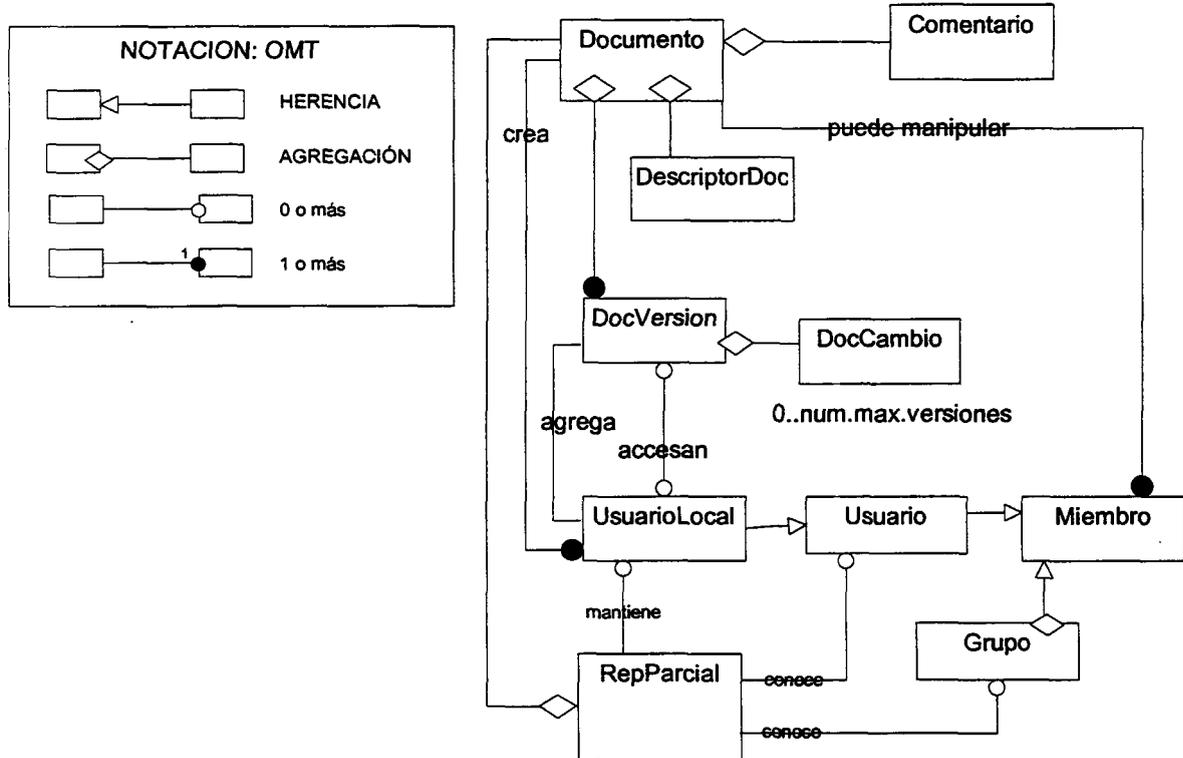
## Anexo 1 - Documentación del análisis

### Contenido:

- A1.1 Modelo de objetos
- A1.2 Ciclo de vida
- A1.3 Diccionario de datos
- A1.4 Invariantes y condiciones
- A1.5 Modelo de operaciones

Para el análisis de este sistema se tomó como base la metodología orientada a objetos FUSION.

### A1.1. Modelo de objetos



Modelo del sistema

### A1.2. Ciclo de vida

Simbología:

- secuencia, ej. Paso1.paso2
- | uno u otro
- || ocurre uno u otro, alternados
- \* 0 o más veces

Repositorio = SetupSite. (Levantar Servicio. Monitoreo. (Monitoreo || Atiende Solicitud Usuario || Recibir Mensajes || Atender mensajes) )\*

Atiende Solicitud Usuario = Crear Documento || Accesar documento || Actualizar Documento || Deshacer Version || Consultar Lista Documentos || Cambiar Permisos Documento || Atiende Transacciones

Monitoreo = Depuración de buffer. Depuración local.Revisar consistencia. Obtener Información de Consulta

Obtener Información de Consulta = Consulta Usuarios Registrados. Consulta Grupos Registrados

Mensajería = Setup Mensajería. (Registrar Sites || Registrar Mensajerías || Recibir Mensajes || Procesar Mensajes)

### A1.3 Diccionario de datos

#### REPOSITORIO

Representa al site donde se corre el servicio del repositorio.

Idsite	Identificador o número de site
NombreSite	Nombre del site
Numgruposite	Numero de grupo de sites
Mensajería	Nombre de la mensajería que le toca
DireccionMensajería	Dirección para localizar/comunicarse con la mensajería
Espacio_asignado_usuario	Espacio asignado a cada usuario, en bytes
Espacio_asignado	Espacio total asignado al repositorio para documentos, bytes
Espacio usado	Espacio ocupado en total, por todos los usuarios, bytes
Espacio_asignado_buffer	Espacio destinado para copias temporales, bytes
Espacio usado_buffer	Espacio usado de las copias temporales, bytes
→ Lista-documentos	Lista de objetos documento
→ Lista_grupos	Lista de objetos grupo que conoce
→ Lista_usuarios_locales	Lista de objetos usuario local
→ Lista_usuarios	Lista de objetos usuario conocidos
→ Lista_areas_interes	Lista de áreas de interés conocidas
Diferencial_hora	+/- días:horas:minutos:segundos que hay que sumar/restar a tiempos recibidos desde la mensajería
Freq_depuracion_buffer	Frecuencia con que se lleva a cabo la revisión de

	documentos antiguos en el área de buffer para depuración
Freq_depuracion_local	Frecuencia con que se revisan documentos antiguos para depuración, ya que la depuración local debería ser menos frecuente que la del buffer.
sig_depuracion_buffer	Después de realizar una depuración de buffer, recalcular la sig. fecha de depuración sumándole Freq_depuracion_buffer
sig_depuracion_local	Después de realizar una depuración de buffer, recalcular la sig. fecha de depuración sumándole Freq_depuracion_local
Antigüedad_max_locales	Antigüedad máxima establecida para un documento local sin utilizarse (leerse). En días
Antigüedad_max_buffer	Antigüedad máxima establecida para un documento local sin utilizarse (leerse). En días
Max_tiempo_esperar	Tiempo máximo a esperar la respuesta de otro site o la mensajería si se les requiere algo. Número de segundos
Max_num_intentos	Número máximo de veces que un site insiste en una petición antes de desistir
Freq_actualizacion	Frecuencia de actualización de información de RICA (carga de información de usuarios, grupos, etc.)
Fecha_ultima_actualización	Con respecto a buscar en RICA información actualizada
Dirsite	Dirección física del site, por si necesita comunicarla a algún otro site en un momento dado, o a la mensajería. Ej: cuando pide transferencia, cuando se registra.
FrecuenciaReintento	Frecuencia en que se revisa si hay transacciones pendientes por atender
MaxNumversiones	Número máximo de versiones a guardar por documento (viene de repositorio global)
TiempoReintento	Tiempo en que se van a reiniciar transacciones pendientes, que se obtiene de sumar tiemporeintento + frecuenciareintento
→ Lista_transacciones	Lista de transacciones que quedaron incompletas por algún motivo y que deben ser atendidas en tiemporeintento

### Áreas de interés

Representa áreas de interés según la clasificación de ACM para Ciencias Computacionales, se carga de la base de datos de RICA.

Nombre	Nombre del área de interés
Id_areaintereres	Clave del área de interés
SubclasificacionDe	Clave de área padCre

### DOCUMENTO

Representa a un documento que puede o no estar físicamente presente, en uno o varios formatos y en una o varias versiones.

Id_documento	Identificador único para un documento, formado con el número de grupo de su site, el número de su site y un consecutivo que se le asigne.
→ Registro_uso_local	Como cada site copia físicamente el documento, basta con registrar quien lo usa localmente para notificar cambios, ya que todos los sites siempre reciben noticias del último cambio

### DESCRIPTOR\_DOC

Datos de cabecera de un documento. Se necesita para transmitir información que describa a un documento.

Id_documento	Identificador del documento
Nombre	Título del documento
Id_usr_creador	Identificador del usuario creador
Fecha_ultima_version	Fecha de última modificación del documento
Fecha_ultima_modificacion	Fecha en que cambio del descriptor: ej-cambio de permisos
Site_ult_cambio	Nombre del site con el último cambio
NumVersion	Versión más alta que se conoce del documento
Fecha_creacion	Fecha en que se creó documento (en que se inscribió al repositorio)
Marca_depurar	Inicialmente falsa. Si se va a depurar, marcar. Esto es: antes de depurar un documento se notifica a su creador, y éste puede consultarlo para decidir si quiere conservarlo o no. Si quisiera conservarlo, entonces debería quitar la marca.
→ ListaAreasInteres	Lista de áreas de interés
→ ListaMiembroPermiso	Lista de permisos de acceso sobre el documento para usuarios y grupos
→ DocVersion	Lista de versiones conocidas
Notas	Notas o comentarios sobre el documento que pone el creador al mismo al momento de añadirlo

### DOCVERSION

Representa el conocimiento de que un mismo documento existe en diferentes versiones y formatos. No contiene físicamente los cambios, solo los registra.

ID_AutorCambio	Identificador del usuario que subió esta versión
NoVersion	Número de versión
Fecha_cambio	Fecha en que se subió el cambio
Formato	Formato físico del documento, es decir, en base a la herramienta que fue hecho (ej: PDF, DOC, etc.)
NotasVersion	Comentarios a la nueva versión

Site_ultimo_cambio_version	Nombre del Site con el último cambio de la versión, con la cual solicitar documentos (o cambios) de versiones específicas
----------------------------	---

### MiembroPermiso

Lista de privilegios de acceso otorgados a grupos o usuarios sobre un documento

Id_miembro	Identificador de miembro
Tipo_miembro	Grupo/usuario
Privilegio	R/RW/RWD/RWDA y owner

### UsoDocumento

Indica que usuarios están leyendo o modificando un documento en un momento dado.

Id_usuario	Identificador del usuario.
Fecha_acceso	Fecha en que se consultó el documento.
Version	Versión que se está utilizando. (Es posible que se haya requerido ver una versión anterior)
Formato	Formato que se está utilizando.

### MIEMBRO

Id_miembro	Identificador del miembro, que puede ser un usuario o grupo
Tipomiembro	G=grupo, U=usuario

### USUARIOCONOCIDO

Representa una lista de todos los usuarios conocidos, sean locales o remotos. La información de un usuario local se detalla más adelante.

Id_usuario	Identificador único de un usuario. Se puede formar con el grupo del site, el número del site y un consecutivo.
Nombre	Nombre del usuario
SiteBase	Indica nombre del site al que pertenece, vacío si fuera local

### USUARIOLOCAL

Id_usuario	Identificador único de un usuario. Se puede formar con el grupo del site, el número del site y un consecutivo.
EspacioOcupado	Espacio total ocupado por documentos que ha creado y sus modificaciones
fecha_ultima_consulta	Ultima fecha en que el usuario consultó algún documento en el repositorio.
→ Lista_areasinteres	Areas de interés que tiene registradas un usuario local
→ Lista_grupospertenece	Grupos a los que pertenece un usuario local

### GRUPO

IdGrupo	Identificador del grupo
NombreGrupo	Nombre del grupo
→ Lista_usuarios	Lista de identificadores de usuarios que pertenecen a un grupo.

#### Lista usuarios

Id_usuario	Identificador único de un usuario. Se puede formar con el grupo del site, el número del site y un consecutivo.
------------	--

#### MENSAJERIA

Sites que actúan como mensajerías para permitir comunicación entre sites. Cada mensajería atiende a un grupo de sites, de manera que no se sobrecargue de trabajo. Si un site no está en su grupo, conoce a través de que mensajería puede delegar el trabajo.

No_gruposite	Grupo de sites al que pertenece.
→ ListaSites	Lista de sites que tiene registrados para atender.
→ ListaMensajeriasGrupo	

#### MensajeriaGrupo

Nombremensajeria	Nombre de la mensajería
DirMensajeria	Dirección para localizar y comunicarse con otra mensajería
Estado	Activo/inactivo
No_gruposite	Numero de grupo de la mensajería

#### Sites

Sites conocidos por una mensajería, para poder redirigir un mensaje si no está en su grupo o comunicarse directamente si sí lo está.

Gruposite	Número de grupo al que pertenece
Id_site	Número de site
Dirsite	Dirección física de sites que están en el mismo grupo que la mensajería, para comunicarse o localizarlos.
Estado	Activo/inactivo

#### TRANSACCION (transferencia)

No.transacción	Identificador de la transacción para poder responder entre sites en base a ellos
Id_usuario	Al que se atiende
Site_transaccion	Site al que se le pide transacción
Estado	Completa/incompleta
Tiempoinicio	Tiempo en que se debe iniciar atender la transacción

Id documento	Identificador del documento a transferir
Formato	Formato del documento
Versión	Versión del documento
NumIntento	Registra el número de veces que se ha intentado cumplir dicha transacción

### MENSAJE

Id_site	Site que genera los mensajes
Idmensaje	Identificador del mensaje, compuesto con el número de site
Origen	Site de inicio
Destino	Indica si va hacia la mensajería, un site específico o es de tipo broadcast
TipoMensaje	Operación relacionada con el mensaje
Contenido	Texto que puede contener otro mensaje dentro

Tipos de mensajes:

#### MSGREGISTRAR:

Un site pide a la mensajería lo registre

#### MSGCONECTAR

Un site pide a la mensajería lo conecte a lista de sites activos

#### MSGUNREGISTER

Un site se da de baja de una mensajería

#### MSGWHOAMI

Un site envía sus datos de identificación a la mensajería

#### MSGIDENTIFY

Una mensajería solicita datos del site

#### MSGTIEMPOS

Una mensajería indica el tiempo oficial para marcar las transacciones.

#### MSGDISCONNECT

Una petición de desconexión del site a la mensajería

#### MSGDELMSG

Pedir borrar un mensaje

#### MSGCOMUNICACION

Envuelve un mensaje para n sites (msgbroadcast)

### MSGCREADESCRIPTOR

Pide agregar un descriptor de documento a la lista de documentos conocidos

### MSGUPDESCRIPTOR

Actualizar conocimiento sobre un documento registrado

### MSGNVAVERSION

Actualizar información de versiones de un documento

### MSGROLLBACK

Indica que se debe borrar ultima versión conocida de un documento y marcar como ultima versión la anterior.

### MSGDELDOC

Indica que se dé de baja un documento.

### MSGULTCAMBIO

Indica que se debe de actualizar al site de envío como poseedor del último cambio.

### MSGNUEVAMENSAJERIA

Indica a todos los sites que lo reciban que ese site es la mensajería la cual los atenderá temporalmente. Es indirecto pues el site actúa ya como mensajería.

### MSGSOLICITAPERMISOS

Indica que se necesita un permiso de acceso, va dirigido directamente a un usuario

### MSG USO DOCUMENTO

Indica que un usuario está leyendo o ha modificado un documento

### MSGCAMBIOPERMISOS

Indica cambio en permisos de acceso para un grupo o área o persona

## **A1.4 Invariantes y condiciones**

- Un usuario solo recibe notificaciones de su servidor local, las demás notificaciones relevantes a usuarios deberán ser manejadas por su site asignado.
- Todos los mensajes entre repositorios parciales se manejan a través del servicio de mensajería, exceptuando las transferencias de archivos y versiones.
- No depurar un documento si es el último cambio

## A1.5 Modelo de Operaciones

### SUPOSICIONES:

La administración de usuarios se da desde el proyecto RICA. Si es dueño de documentos en el repositorio en el momento de darse de baja, debe asegurarse desde RICA que se restrinja su baja hasta que no ceda derechos de administración a algún otro usuario, o los borre del repositorio.

Se asume que la replicación de usuarios, grupos y asociaciones a grupos de los usuarios en un esquema distribuido deberá ser provisto por RICA. También es posible la incorporación de campos de fechas para registrar cuando se agregaron nuevos registros.

### USUARIOS Y REGISTRO

<b>Operación:</b>	<b>Consultar Usuarios Registrados</b>
<b>Descripción:</b>	Consultar base de datos RICA para actualizar usuarios registrados localmente, en base a la última fecha de actualización del repositorio
<b>Lee:</b>	Fecha ultima de actualización de parámetros
<b>Cambia:</b>	Agrega nuevos usuarios locales Elimina usuarios
<b>Envía:</b>	Mensajería {registra_usuario_global, id_usuario, nombre, site usuario} Mensajería {elmina_usuario_global, id_usuario, nombre, site usuario}
<b>Asume:</b>	
<b>Resultado:</b>	Si a partir de última_fecha_actualización + frecuencia_actualizacion_en_días > fecha_hoy: Accesar RICA Copiar lista de usuarios registrados localmente Para todos los usuarios nuevos: Agregar a USUARIOSLOCALES Crear mensaje de notificación de alta de usuario global para otros sites. Enviar a mensajería  Para todos los usuarios que ya no existen: Borrar de USUARIOSLOCALES Crear mensaje de notificación de baja de usuario global para otros sites.

	Enviar a mensajería
--	---------------------

<b>Operación:</b>	<b>Inicio de sesión con el Usuario</b>
<b>Descripción:</b>	Iniciar sesión en repositorio de un usuario local
<b>Lee:</b>	Provee: id_usuario, password, Consulta lista de usuarios locales
<b>Cambia:</b>	Crea conexión del usuario
<b>Envía:</b>	Rechazo_conexión   OK.
<b>Asume:</b>	
<b>Resultado:</b>	Si usuario no se encuentra en lista_usuarios_locales O usuario_local.password<> password Envía Rechazo_conexión Else New Conexión_usr Accesar notificaciones de últimos eventos sucedidos desde la última visita Consultar área de trabajo del usuario

## DOCUMENTOS

<b>Operación:</b>	<b>Consultar Lista de Documentos</b>
<b>Descripción:</b>	Mostrar documentos disponibles
<b>Lee:</b>	Provee lista de criterios para ordenar (aplica default por autor), criterios de búsqueda (autor, área de interés, grupo o modificaciones a partir de una fecha)
<b>Cambia:</b>	
<b>Envía:</b>	Usuario: lista de descriptores
<b>Asume:</b>	Usuario válido y local, conectado
<b>Resultado:</b>	Nueva lista_descriptores que cumplen criterios de búsqueda, presentando: Titulo, autor, versión actual, fecha de última actualización, áreas de interés relacionadas

<b>Operación:</b>	<b>Crear documento</b>
<b>Descripción:</b>	Crear un nuevo documento de un usuario local
<b>Lee:</b>	Provee datos del documento, archivo, identificador del usuario
<b>Cambia:</b>	New descriptor_documento, new versión de documento, new permisos de acceso por omisión a grupos a los que pertenece
<b>Envía:</b>	Mensajería. {añadir documento,descriptor_documento}   Rechazo
<b>Asume:</b>	Usuario registrado y válido

<b>Resultado:</b>	<p>Si datos_documento.tamaño &gt; usuario_local.espacioasignado - usuario_local.espacioocupado Rechazo</p> <p>Else Transferir archivo de maquina cliente al servidor Creación de descriptor_documento: Por cada área de interés en documento añadir área de interés a listaAreasInteres</p> <p>New permiso a usuario, como owner y añadirlo a lista de permisos por documento Por cada grupo en usuario_local.Lista_gruposPertenece Crear permisos de escritura-lectura, añadirlo a lista de permisos por documento Crear docversion con v. 1.0</p> <p>Creación registro de uso de documentocon, con Fecha_ultimo_acceso =fecha_creación Actualizar usuario.espacioocupado + documento.tamaño Actualizar usuario.fechaultimaconsulta = fecha_hoy</p> <p>Crear mensaje de creación de nuevo documento Mensajería {nuevo_documento, descriptor_doc}</p> <p>Llamar a función añadir documento</p>
-------------------	--

<b>Operación:</b>	<b>Añadir documento</b>
<b>Descripción:</b>	Crear un nuevo registro de que existe el documento (descriptor). Cada site debe notificar a sus usuarios locales pertinentes este hecho.
<b>Lee:</b>	Provee descriptor_documento
<b>Cambia:</b>	New descriptor de documento
<b>Envía:</b>	
<b>Asume:</b>	No existe documento previamente
<b>Resultado:</b>	<p>Si no es documento local Añadir descriptor_documento a descriptores conocidos</p> <p>Por cada areainterés en descriptor.lista_areasinterés Crear notificación de creación de documento Pedir a RICA envío de notificación</p> <p>Por cada miembro en descriptor_doc.ListaMiembrosPermiso Crear notificación de creación de documento</p>

	Pedir a RICA envío de notificación
--	------------------------------------

<b>Operación:</b>	<b>Consultar información de documento</b>
<b>Descripción:</b>	Se hace petición para acceder metadatos de un documento
<b>Lee:</b>	Provee: id del documento, id del usuario
<b>Cambia:</b>	
<b>Envía:</b>	Rechazo   información del documento
<b>Asume:</b>	Usuario válido y local, documento existe
<b>Resultado:</b>	<p>Si el usuario tiene privilegios al menos de lectura ("R")</p> <p>Presentar información general del documento  Presentar áreas de interés relacionadas  Presentar versiones existentes por orden cronológico  Presentar versiones existentes por orden de referencia</p> <p>Presentar todas las notas acerca del documento</p> <p>Else  Rechazo: no tiene permiso  Se pregunta si desea solicitar permisos de acceso  Llamar a Solicitar Permisos de Acceso</p>

<b>Operación:</b>	<b>Accesar Documento</b>
<b>Descripción:</b>	Se hace petición para acceder (copiar) un documento
<b>Lee:</b>	Provee: id del documento, id del usuario, formato en que se desea, número de versión
<b>Cambia:</b>	
<b>Envía:</b>	Rechazo   Documento, Mensajería {usuario, uso de documento}
<b>Asume:</b>	Usuario válido y local, documento existe y se tiene registrado que está disponible en la versión y formato especificados
<b>Resultado:</b>	<p>Si el usuario tiene privilegios al menos de lectura ("R")</p> <p>Inicio de transacción marcada como pendiente</p> <p>Buscar si se tiene último cambio localmente,  Si no está disponible localmente,  Buscar site donde se tuvo el último cambio  Solicitar copia de documento a site remoto, indicando el número de transacción  Guardar localmente</p>

	<p>Iniciar transferencia a máquina cliente</p> <p>Si la transferencia fue exitosa</p> <p style="padding-left: 40px;">Registrar uso de documento local</p> <p style="padding-left: 40px;">Crear mensaje de notificación de uso de documento</p> <p style="padding-left: 40px;">Finaliza Transacción</p> <p>Else</p> <p style="padding-left: 40px;">Indicar problema al usuario</p> <p style="padding-left: 40px;">Pedir acción del usuario: reintentar, cancelar, programar para más tarde</p> <p style="padding-left: 80px;">Si programar para más tarde,</p> <p style="padding-left: 120px;">Agregar mensaje de petición a mensajes por enviar configurado con tiempos y frecuencias de reintento</p> <p style="padding-left: 120px;">Cancelar</p> <p style="padding-left: 40px;">Se envía mensaje de petición a mensajería</p> <p>Else</p> <p style="padding-left: 40px;">Rechazo: no tiene permiso</p>
--	---

<b>Operación:</b>	<b>Solicitar permisos de acceso</b>
<b>Descripción:</b>	Crea un mensaje de notificación para usuarios administradores de un documento
<b>Lee:</b>	Provee: id_documento, id_usuario, nivel deseado
<b>Cambia:</b>	
<b>Envía:</b>	Mensajería: {solicitud_permiso, id_documento, privilegio deseado, receiver: id_administrador, sender: id_solicitante}
<b>Asume:</b>	Documento existe
<b>Resultado:</b>	<p>Por cada usuario con permisos de administración sobre el documento</p> <p style="padding-left: 40px;">New mensaje de petición de permisos para el usuario solicitante</p> <p style="padding-left: 40px;">Si es usuario remoto</p> <p style="padding-left: 80px;">Paso a mensajería para manejo del mensaje</p> <p>Else</p> <p style="padding-left: 40px;">Pedir a RICA efectúe la notificación</p>

<b>Operación:</b>	<b>Consultar permisos de acceso</b>
<b>Descripción:</b>	Los usuarios con privilegios de administración pueden consultar los privilegios de acceso sobre un determinado documento.

<b>Lee:</b>	Provee: id_usuario, id_documento Consulta: lista de miembros-permiso
<b>Cambia:</b>	
<b>Envía:</b>	Información de permisos   Rechazo
<b>Asume:</b>	
<b>Resultado:</b>	Si el usuario tiene permisos de administrador, Por cada miembro en listamiembrospermiso Mostrar miembro y nivel de privilegio

<b>Operación:</b>	<b>Modificar Permisos Documento</b>
<b>Descripción:</b>	Cambiar permisos acceso documento, a partir de la especificación de un usuario.
<b>Lee:</b>	Provee: miembro_usuario miembro_grupo, id_documento, nuevo privilegio
<b>Cambia:</b>	ListaMiembrosPermiso
<b>Envía:</b>	Mensajería {cambio permisos,id_documento,listamiembrospermiso}
<b>Asume:</b>	El usuario es el creador del documento o administrador del mismo
<b>Resultado:</b>	Actualiza lista de miembros-permiso Enviar a Mensajería {cambio_permisos,id_documento,listamiembrospermiso}

<b>Operación:</b>	<b>Crear comentario</b>
<b>Descripción:</b>	Crea una nota o comentario externo a un documento
<b>Lee:</b>	Provee: id_documento, id_usuario, texto comentario
<b>Cambia:</b>	Nuevo comentario
<b>Envía:</b>	Mensajería { nuevo_comentario, id_usuario, id_documento, fecha_creacion}
<b>Asume:</b>	Va ligado a un documento existente; usuario existe y tiene capacidades de lectura
<b>Resultado:</b>	Agrega comentario a lista_comentarios Crea mensaje y lo envía a mensajería

<b>Operación:</b>	<b>Registro de transacciones</b>
<b>Descripción:</b>	Marca el inicio de una transferencia de archivo
<b>Lee:</b>	Provee: id_documento, id_usuario, siteservidor, no. Versión para transferencia,
<b>Cambia:</b>	New transacción con fecha inicio y tiempo inicio
<b>Envía:</b>	
<b>Asume:</b>	

<b>Resultado:</b>	Crea transacción y se marca como pendiente
-------------------	--

<b>Operación:</b>	<b>Completa Transacción</b>
<b>Descripción:</b>	Completa la transferencia de un archivo
<b>Lee:</b>	provee id_documento, id_usuario, transaccion
<b>Cambia:</b>	Delete transacción
<b>Envía:</b>	
<b>Asume:</b>	Existía un registro de transacción anterior
<b>Resultado:</b>	Delete transacción para el documento

<b>Operación:</b>	<b>Actualizar Documento</b>
<b>Descripción:</b>	Crear nueva versión documento
<b>Lee:</b>	Provee: id_documento, id_usuario, datos de la versión, archivo, versión anterior
<b>Cambia:</b>	New versión
<b>Envía:</b>	Mensajería: {nueva versión, descriptor doc}   Rechazo
<b>Asume:</b>	Usuario válido y local, documento existe, usuario tiene un registro en registrousolocal para la última versión del documento
<b>Resultado:</b>	<p>Si usuario tiene permiso de escritura ("W")</p> <p style="padding-left: 40px;">Verificar última versión leída del documento  Si faltan versiones por leer anteriores  Preguntar si desea cancelar, dar documento por leído u obtener versiones anteriores  Si pide versiones anteriores,  Por cada versión no leída  Llamar a obtener documento</p> <p style="padding-left: 40px;">Obtener archivo de máquina cliente  Crear nuevo docversion y añadirlo a lista_docversion, con no. Versión = versión anterior +1  Y consecutivo = max( consecutivo para no.versión +1)</p> <p style="padding-left: 40px;">Actualizar descriptor del documento con no.versión y site del último cambio igual al local  Actualizar EspacioOcupado para el usuario local  Actualizar EspacioOcupado total</p> <p style="padding-left: 40px;">Agregar modificación en Usodocumento</p> <p style="padding-left: 40px;">Crear mensaje de nueva versión  Crear mensaje de uso documento</p>

	<p>Enviar mensajes a mensajería</p> <p>Else</p> <p>Rechazo: no tiene permiso</p>
--	--

<b>Operación:</b>	<b>Eliminar documento</b>
<b>Descripción:</b>	Borrar documento a petición del usuario creador
<b>Lee:</b>	ListaMiembrosPermiso
<b>Cambia:</b>	
<b>Envía:</b>	Mensajería: {eliminar documento, id_documento}
<b>Asume:</b>	El usuario debe ser el dueño del documento o tiene permisos de administración
<b>Resultado:</b>	Si id_usuario en ListaMiembrosPermiso y CONTIENE(privilegio,"D") Borra documento Notifica Mensajería: {eliminar documento, id_documento}

<b>Operación:</b>	<b>Deshacer Versión</b>
<b>Descripción:</b>	"Rollback" a una versión
<b>Lee:</b>	Provee id_documento, id_usuario
<b>Cambia:</b>	
<b>Envía:</b>	Mensajería { RegresarVersiónAnterior, id_documento, id_usuario}
<b>Asume:</b>	Que el usuario es el autor de la versión y que es la versión registrada más recientemente
<b>Resultado:</b>	Si docversion.noversion = documento.descriptor.version Y autorcambio = id_usuario Borrar docversion Eliminar archivo físicamente. Si no hay más documentos con el mismo número de versión, Actualizar descriptor de documento Crear mensaje para regresar a versión anterior (msgRollback) y enviar a mensajería Pedir a RICA notificación de este evento

<b>Operación:</b>	<b>Promover Documento</b>
<b>Descripción:</b>	Envía documento a una biblioteca digital o medio externo
<b>Lee:</b>	Provee: metadatos del documento revisados
<b>Cambia:</b>	
<b>Envía:</b>	Mensajería {notifica_promoción, id_documento}
<b>Asume:</b>	
<b>Resultado:</b>	Obtiene parámetros de medio externo

	Enviar documento Generar mensaje notificación Envía Mensajería {notifica_promoción, id_documento}
--	---

## COMUNICACIÓN Y SERVICIO

<b>Operación:</b>	<b>SetupSite</b>
<b>Descripción:</b>	Inicializa site
<b>Lee:</b>	Provee datosmensajería, parámetros_site
<b>Cambia:</b>	
<b>Envía:</b>	Mensajería {Petición_Registrar, parametrosite}
<b>Asume:</b>	
<b>Resultado:</b>	Registra datosmensajería Establecer conexión con mensajería Requerir registro con parámetros del site Llamar a Sincronizartiempos

<b>Operación:</b>	<b>Sincronizartiempos</b>
<b>Descripción:</b>	Actualiza el diferencial de tiempo del servidor local con respecto al de la mensajería
<b>Lee:</b>	Tiempolocal
<b>Cambia:</b>	DiferencialHora
<b>Envía:</b>	Mensajería {Solicita conocer tiempo}
<b>Asume:</b>	
<b>Resultado:</b>	Enviar a mensajería solicitud de conocer tiempo_mensajería Recibe respuesta Ajusta DiferencialHora = tiempo_mensajería - tiempo_local

<b>Operación:</b>	<b>LevantarServicio</b>
<b>Descripción:</b>	Reconecta el servicio después de una falla o desconexión
<b>Lee:</b>	Dirmensajería, supp parametros_site
<b>Cambia:</b>	
<b>Envía:</b>	Mensajería {RequisitoConexion, parametrosite}
<b>Asume:</b>	
<b>Resultado:</b>	Sincronizartiempos Establecer conexión con mensajería mediante dirmensajería Llamar a PedirMensajesPendientes Llamar a ProcesarMensajesPendientes

<b>Operación:</b>	<b>Enviar Mensaje</b>
<b>Descripción:</b>	Se encarga de añadir mensaje a una cola de envío y prepararlo

	para ser enviado.
<b>Lee:</b>	Provee: Tipomensaje, mensaje, dirección de mensajería
<b>Cambia:</b>	
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	Añade un identificador único compuesto con el número de site al mensaje Añadir mensaje a cola de envío para pasar a la mensajería

<b>Operación:</b>	<b>Procesa Mensaje</b>
<b>Descripción:</b>	Realiza transmisión a destinatario, que siempre es una mensajería si se trata de un site, si es una mensajería, puede ser otra mensajería o alguno de sus sites registrados.
<b>Lee:</b>	Provee: Direccion_destinatario
<b>Cambia:</b>	
<b>Envía:</b>	Destinatario {Mensaje}
<b>Asume:</b>	
<b>Resultado:</b>	Mientras no recibido y no max_num_intentos Intentos=intentos+1 Iniciar transmisión Espera respuesta un tiempo max_tiempo_esperar

<b>Operación:</b>	<b>Recibir Mensajes</b>
<b>Descripción:</b>	Recibe Mensajes y los añade a una lista de mensajesXatender
<b>Lee:</b>	Provee: Mensaje
<b>Cambia:</b>	
<b>Envía:</b>	Mensajería {confirmación_recibido,identificador de mensaje}
<b>Asume:</b>	
<b>Resultado:</b>	Añade mensaje a cola de mensajesXatender

<b>Operación:</b>	<b>Atender Mensajes</b>
<b>Descripción:</b>	Saca un mensaje de la cola de mensajesXatender y lo decodifica, dirige la acción
<b>Lee:</b>	
<b>Cambia:</b>	
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	Mientras haya mensajes en cola Obtiene primer mensaje pendiente en cola Extraer contenido mensaje Invocar acción requerida por tipo de mensaje:

	<p>Si es nuevo_documento, llamar a ActualizarNuevoDocumento</p> <p>Si es nueva_version, llamar a Actualizar Versión Documento</p> <p>Si es promoción_documento, llamar a Actualiza Promoción Documento</p> <p>Si es nuevo_comentario, llamar a Actualiza Comentario</p> <p>Si es uso_documento, llamar a Actualiza uso documento</p> <p>Si no hay más mensajes en cola Esperar</p>
--	--

<b>Operación:</b>	<b>Actualizar Nuevo Documento</b>
<b>Descripción:</b>	Crea un nuevo descriptordoc por un nuevo documento
<b>Lee:</b>	Provee; descriptordoc
<b>Cambia:</b>	Nuevo Descriptordoc, usodocumento
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	<p>Crear nuevo descriptordoc para documento</p> <p>Crear nuevo usodocumento para el autor</p> <p>Por cada miembro en listamiembrospermiso</p> <p style="padding-left: 20px;">Si miembro.tipo = grupo</p> <p style="padding-left: 40px;">Por cada usuario_local en grupo.listausuarios</p> <p style="padding-left: 60px;">Pedir a RICA notifique promoción a este usuario_local</p> <p>Else</p> <p style="padding-left: 20px;">Si miembro en repositorio.lista_usuarioslocales</p> <p style="padding-left: 40px;">Pedir a RICA notifique promoción</p> <p>Por cada areainterres en listaareainterres del documento</p> <p>Por cada usuariolocal en repositorio.listausuarioslocales</p> <p style="padding-left: 20px;">Si areainterres en usuariolocal.listaareasinteres</p> <p style="padding-left: 40px;">Pedir a RICA notifique promoción</p>

<b>Operación:</b>	<b>Actualizar Versión Documento</b>
<b>Descripción:</b>	Actualizar descriptordoc por una nueva versión de un documento
<b>Lee:</b>	Provee; Mensaje NuevaVersion, que contiene descriptordoc actualizado
<b>Cambia:</b>	Descriptordoc, documento
<b>Envía:</b>	
<b>Asume:</b>	

<b>Resultado:</b>	<p>Documento.descriptordoc = descriptordoc</p> <p>Crear mensaje para notificación de cambios:  a usuarios con áreas de interés iguales a las del documento,  grupos a los que pertenece el usuario del documento,  usuarios locales que lo usan actualmente</p> <p>Enviar mensaje a servicio de notificación en RICA</p>
-------------------	--

<b>Operación:</b>	<b>Actualiza Promoción Documento</b>
<b>Descripción:</b>	Notifica a usuarios interesados
<b>Lee:</b>	Provee: id documento, contenido del mensaje de notificación
<b>Cambia:</b>	
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	<p>Por cada miembro en listamiembrospermiso</p> <p>Si miembro.tipo = grupo</p> <p>    Por cada usuario_local en grupo.listausuarios</p> <p>        Pedir a RICA notifique promoción a este usuario_local</p> <p>Else</p> <p>    Si miembro en repositorio.lista_usuarioslocales</p> <p>        Pedir a RICA notifique promoción</p> <p>Por cada areainterres en listaareainterres del documento</p> <p>    Por cada usuariolocal en repositorio.listausuarioslocales</p> <p>        Si areainterres en usuariolocal.listaareasinteres</p> <p>            Pedir a RICA notifique promoción</p>

<b>Operación:</b>	<b>Actualiza Nuevo Comentario</b>
<b>Descripción:</b>	Copia localmente notas o comentarios acerca de un documento
<b>Lee:</b>	Provee; id_documento, id_usuario, comentario
<b>Cambia:</b>	New comentario
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	<p>Crea nuevo comentario</p> <p>Enviar petición de notificación en RICA a usuarios en usodocumento, dueño del documento, a autores de versiones y a miembros del grupo del documento y del usuario</p>

<b>Operación:</b>	<b>Actualiza Uso Documento</b>
<b>Descripción:</b>	Actualizar registro de uso de documentos (historial)
<b>Lee:</b>	Provee; id_usuario, id_documento, tipo_acceso –lectura o actualización-, fecha y hora
<b>Cambia:</b>	New uso documento

<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	Crea nuevo registro de usodocumento

<b>Operación:</b>	<b>DepuraDocumentos</b>
<b>Descripción:</b>	Borrar documentos antiguos de buffer o locales, en base a la frecuencia de depuración de cada site
<b>Lee:</b>	Provee: borrarbuffer=s/n, borrarlocales=s/n
<b>Cambia:</b>	
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	<p>Si fecha_última_depuración_local + frecuencia_depuración local &gt; fecha_hoy</p> <p>    Por cada documento local</p> <p>        Si documento marcado para depuración y fecha_hoy - fecha_marca &gt; tiempo_tolerancia</p> <p>            Borrar documento</p> <p>        Si Fecha_hoy – fecha_ultimo_acceso a documento &gt; Antigüedad_max_locales</p> <p>            Crear mensaje notificación de depuración para los administradores</p> <p>            Documento.marcaDepurar = true</p> <p>Si fecha_última_depuración_buffer + frecuencia_depuración buffer &gt; fecha_hoy</p> <p>    Por cada documento en copia temporal</p> <p>        Si Fecha_hoy – fecha_ultimo_acceso a documento &gt; Antigüedad_max_buffer</p> <p>            Borrar documento</p>

<b>Operación:</b>	<b>CancelarDepuración</b>
<b>Descripción:</b>	El usuario creador quita la marca de depuración del documento local
<b>Lee:</b>	Provee: id_documento

<b>Cambia:</b>	Documento.marca=false
<b>Envía:</b>	
<b>Asume:</b>	Archivo marcado para depuración, solo cancelan usuarios administradores
<b>Resultado:</b>	Documento.marca=false

<b>Operación:</b>	<b>RecibeTransferencia</b>
<b>Descripción:</b>	Se recibe un documento requerido previamente
<b>Lee:</b>	Supp No. Transacción
<b>Cambia:</b>	New docbase,new registrousolocal, delete transaccion
<b>Envía:</b>	
<b>Asume:</b>	Documento ha sido previamente solicitado
<b>Resultado:</b>	Se recibe archivo se guarda en área buffer de documentos  Se completa transacción (llamar a CompletaTransacción) Se actualiza espacio ocupado buffer del repositorio

<b>Operación:</b>	<b>AtiendeTransacciones</b>
<b>Descripción:</b>	Saca un mensaje de la cola de tareas pendientes y lo ejecuta
<b>Lee:</b>	Transaccion
<b>Cambia:</b>	
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	Obtiene siguiente transacción pendiente de la cola de transacciones pendientes (si tiempoinicio + maximotiempoesperar >= tiempoactual) Verifica si aplica reintento (si transaccion.numintento < max_num_intentos) Si aplica reintento Crear mensaje para pedir transferencia, con site_ultimo_cambio Enviar a mensajería Else Avisar a usuario que no es accesible el documento, preguntar si desea: programar para más tarde, o cancelar transferencia

<b>Operación:</b>	<b>RegistrarSite</b>
<b>Descripción:</b>	Una mensajería registra sites de su grupo y otras mensajerías
<b>Lee:</b>	Provee: nombresite, supp dirsite

<b>Cambia:</b>	
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	New site Añadir site a listasites

<b>Operación:</b>	<b>Registrar Mensajería</b>
<b>Descripción:</b>	Una mensajería registra otras mensajerías
<b>Lee:</b>	Provee: nombremensajería, dirmensajería, numgrupo
<b>Cambia:</b>	
<b>Envía:</b>	
<b>Asume:</b>	
<b>Resultado:</b>	New mensajería Añadir mensajería a listamensajerías

## ANEXO 2- Diseño detallado

### Contenido:

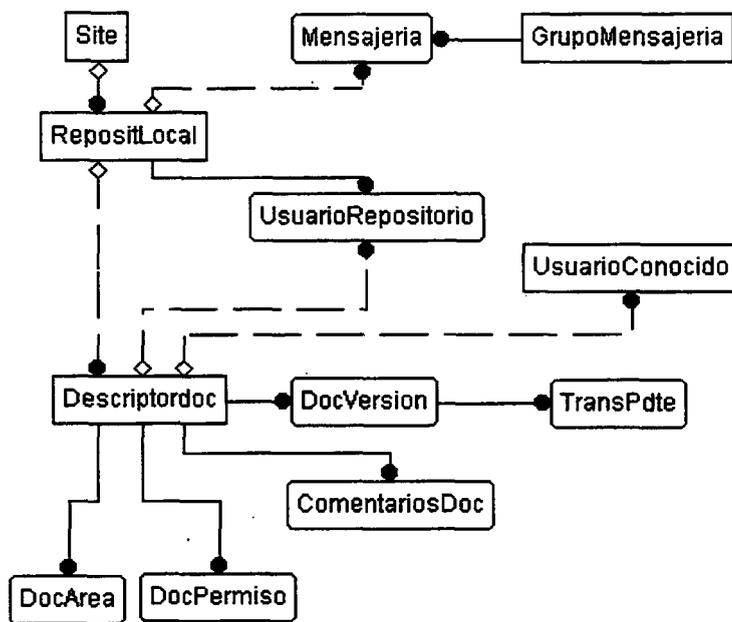
A2.1 Diseño de datos

A2.2 Diagramas de interacción de objetos

A2.3 Diagrama de navegación

### A2.1 Diseño de datos

#### Modelo Entidad-Relación



Notación:

- Relación 1 a muchos
- Relación 1 a 1
- ◆- - -● Relación indirecta

#### Detalle de Entidades

Enseguida se muestra la definición de atributos y llaves de cada una de las tablas. La implementación está realizada en MiniSql, base de datos relacional.

Table = DESCRIPTORDOC

	Field	Type	Length	Not Null	Unique Index
1	Id_documento	int	4	Y	N/A
1	Id_sitedoc	int	4	Y	N/A
	Nombre	char	100	N	N/A
2	Id_usuario	int	4	Y	N/A
2	Id_siteusr	int	4	Y	N/A
	Fecha_ult_versi	date	4	N	N/A
	Fecha_ult_modif	date	4	N	N/A
	Id_site_ult_cam	int	4	N	N/A
	Noversion	int	4	N	N/A
	Fecha_creacion	date	4	N	N/A
	marca_depurar	int	4	N	N/A
	notas	text	255	N	N/A
	idxDESCRIPTORDOC	index	N/A	N/A	Y (1)
	idxUSRCREADOC	index	N/A	N/A	N (2)

Table = DOCAREA

	Field	Type	Length	Not Null	Unique Index
	id_documento	int	4	Y	N/A
	id_sitedoc	int	4	Y	N/A
	cvearea	char	10	N	N/A
	idxDOCAREA	index	N/A	N/A	Y

Table = DOCVERSION

	Field	Type	Length	Not Null	Unique Index
1	id_documento	int	4	Y	N/A
1	id_sitedoc	int	4	Y	N/A
1	noversion	int	4	Y	N/A
1	consecutivo	int	4	Y	N/A
1	cveformato	char	4	Y	N/A
	fecha_cambio	date	4	N	N/A
	notasversion	text	255	N	N/A
	id_usuario	int	4	Y	N/A
	id_siteusr	int	4	N	N/A
	nombrelocal	text	255	N	N/A
	Hora	Time	4	Y	N/A
	Fecha	date	4	Y	N/A
	nombreamchivo	text	255	N	N/A
	idxDOCVERSION	index	N/A	N/A	Y

Table = DOCPERMISO

	Field	Type	Length	Not Null	Unique Index
1,2	id_documento	int	4	Y	N/A
1,2	id_sitedoc	int	4	Y	N/A

1

2

tipoPermiso	char	5	N	N/A
tipo_miembro	char	1	N	N/A
id_usuario	int	4	N	N/A
id_siteusr	int	4	N	N/A
id_grupo	int	4	N	N/A
cvearea	char	10	N	N/A
idxDOCPERMxUSR	index	N/A	N/A	N (1)
idxDOCPERMxGPO	index	N/A	N/A	N (2)

Table = COMENTARIOSDOC

Field	Type	Length	Not Null	Unique Index
id_documento	int	4	Y	N/A
id_sitedoc	int	4	Y	N/A
noversion	int	4	Y	N/A
consecutivo	int	4	Y	N/A
fecha	date	4	N	N/A
comentario	text	255	N	N/A
id_usuario	int	4	Y	N/A
id_siteusr	int	4	N	N/A

Table = REPOSITLOCAL

Field	Type	Length	Not Null	Unique Index
idsite	int	4	Y	N/A
nombreSite	char	30	N	N/A
numgruposite	int	4	N	N/A
mensajeria	char	30	Y	N/A
dirMensajeria	text	250	Y	N/A
esp_asignado_us	int	4	N	N/A
esp_asignado_lo	int	4	N	N/A
espacio_usado_l	int	4	N	N/A
esp_asignado_bu	int	4	N	N/A
esp_usado_buff	int	4	N	N/A
freq_depur_buff	int	4	N	N/A
freq_depur_loc	int	4	N	N/A
sig_depuracion_	date	4	N	N/A
sig_depuracion_	date	4	N	N/A
antiguedad_max_	int	4	N	N/A
antiguedad_max_	int	4	N	N/A
max_secs_espera	int	4	N	N/A
max_num_intento	int	4	N	N/A
freq_actualizac	int	4	N	N/A
fecha_ult_actua	date	4	N	N/A
dirsite	text	250	Y	N/A
freqreintense	int	4	N	N/A
tiemporeintento	time	4	N	N/A

pathscripts	text	250	N	N/A
pathimages	text	250	N	N/A
multdifftime	int	4	N	N/A
difftime	time	4	N	N/A
pathlocalfiles	text	250	N	N/A
pathcachefiles	text	250	N	N/A
pathtmpfiles	text	250	N	N/A

Table = USUARIOREPOSITORIO

Field	Type	Length	Not Null	Unique Index
id_usuario	int	4	Y	N/A
espacio_ocupado	int	4	N	N/A
fecha_ult_consu	date	4	N	N/A

Table = USUARIOCONOCIDO

Field	Type	Length	Not Null	Unique Index
id_usuario	int	4	Y	N/A
id_siteusr	int	4	Y	N/A
nombre	char	100	N	N/A
idxUSUARIOSITE	index	N/A	N/A	Y

Table = SITE

Field	Type	Length	Not Null	Unique Index
no_gruposite	int	4	Y	N/A
id_site	int	4	Y	N/A
dirsite	text	250	N	N/A
nombresite	char	30	N	N/A
idxSITE	index	N/A	N/A	Y

Table = GRUPOMENSAJERIA

Field	Type	Length	Not Null	Unique Index
no_gruposite	int	4	Y	N/A
nombremensajeri	char	30	N	N/A
dirmensajeria	text	250	N	N/A
idxGRUPOMENSAJE	index	N/A	N/A	Y

Table = TRANSPDTE

Field	Type	Length	Not Null	Unique Index
id_documento	int	4	Y	N/A

id_sitedoc	int	4	Y	N/A
noversion	int	4	N	N/A
cveformato	char	4	Y	N/A
id_usuario	int	4	Y	N/A
fecha	date	4	N	N/A
hora	time	4	N	N/A
tipotrans	char	10	N	N/A
datostrans	text	100	N	N/A
frequintento	int	4	N	N/A
numreintentos	int	4	N	N/A
sigreintento	time	4	N	N/A
idxTransDoc	index	N/A	N/A	N
idxTransUsr	index	N/A	N/A	N

## A2.2 Diagramas de interacción

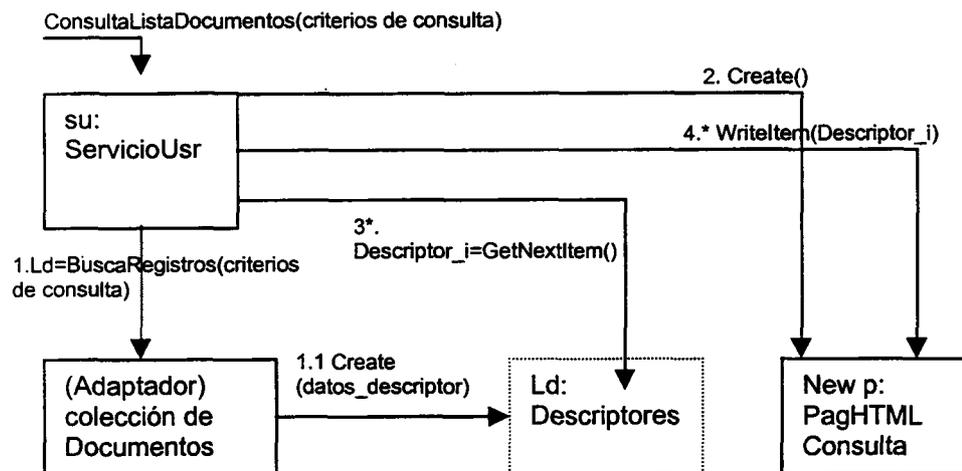
Enseguida se muestra una representación gráfica de las interacciones entre los componentes del sistema en un estilo semejante a un diagrama de objetos, aunque no todos los componentes seguirán necesariamente este tipo de implementación.

Simbología y notación:

Se tomó la notación para el diseño FUSION como base, donde:

- Las flechas representan el flujo de control, ya sean llamadas a métodos o funciones
- Los rectángulos representan objetos o componentes, y aquellos que tienen líneas punteadas, colecciones o listas.
- Los números indican secuencia, donde un \* indica repetición y un ' indica una alternativa en la secuencia.
- El símbolo adaptador o (adap) se utiliza para señalar aquellos componentes que encapsulan de alguna manera el acceso a una base de datos.

### CONSULTA LISTA DOCUMENTOS

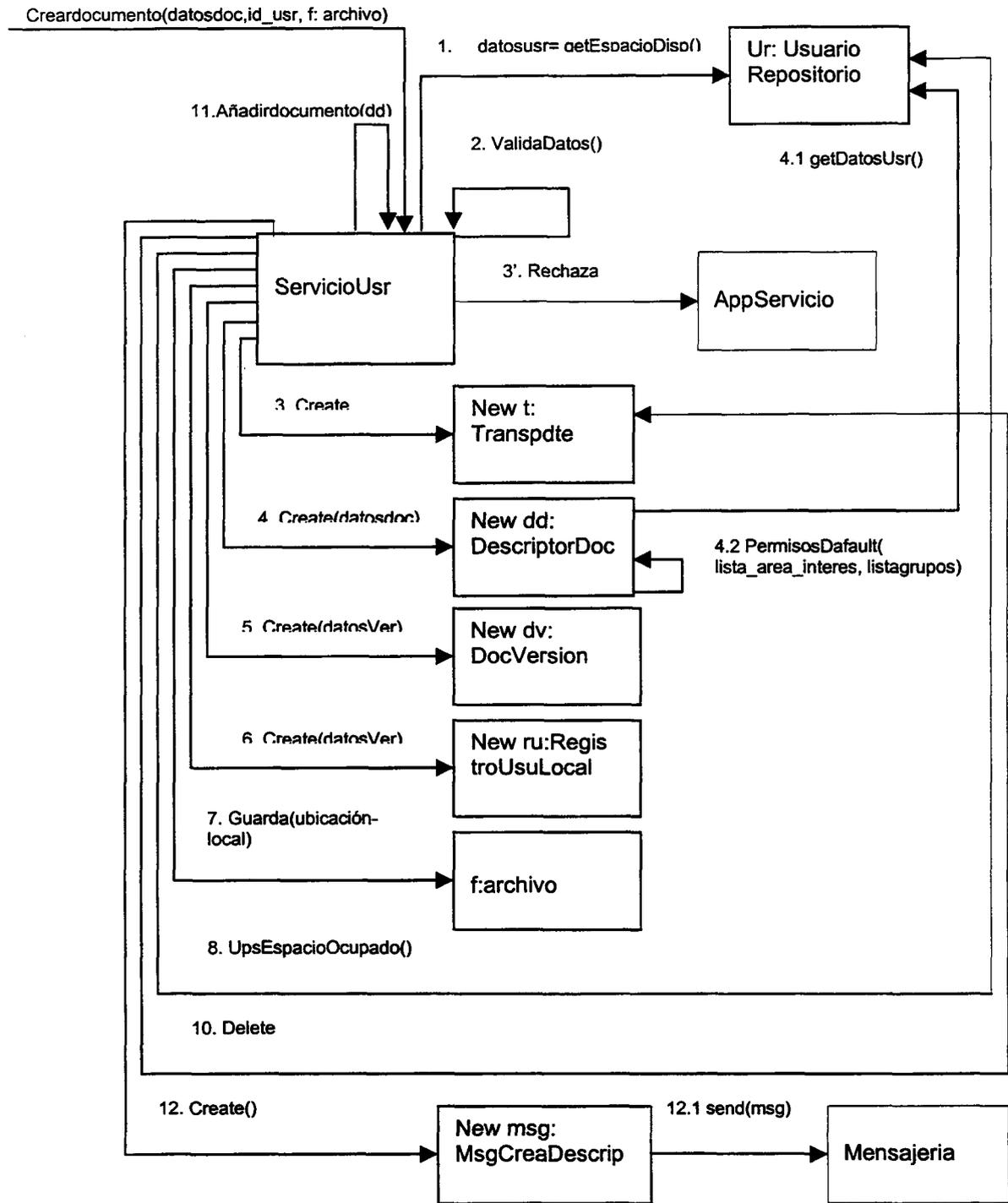


### DESCRIPCIÓN:

#### Operación ConsultarListaDocumentos

Crea una página de consulta de documentos conocidos de acuerdo a un criterio de consulta, los cuales pueden ser: área de interés, autor, a partir de una fecha o por grupo. Se crea una página HTML que contendrá el resultado de la consulta, obtiene una lista de descriptores de documentos ya ordenada y extrayendo cada uno, genera el contenido de la página, el cual envía. Finalmente se envía a la aplicación del cliente.

# CREAR DOCUMENTO



## DESCRIPCIÓN:

Operación: CreaDocumento

Crea documento asociado al usuario, de acuerdo con un formulario de entrada en la página de servicio al usuario.

Primero valida la información proporcionada y revisa el tamaño del archivo y lo valida contra el espacio disponible del usuario. (1,2)

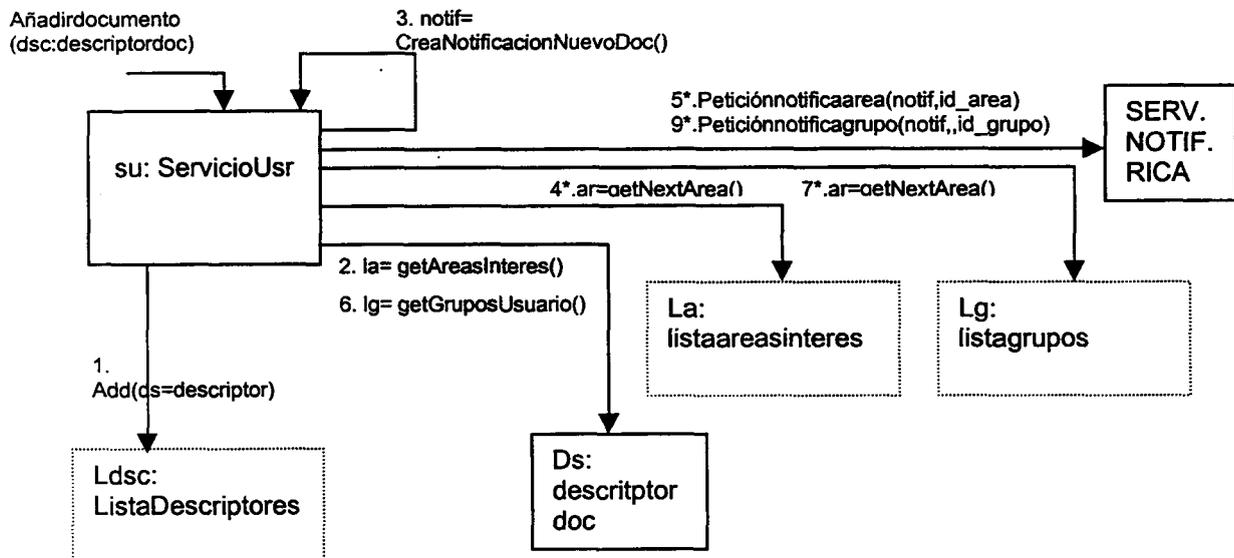
Crea los descriptores del documento y versión de los mismos, y asigna permisos por defecto en Permisos Default.

Obtiene archivo físico del cliente, dejándolo en un área protegida destinada a documentos locales, dándole una representación interna y comprimiéndolo.  
Crea documento y un mensaje de notificación que envía al monitor.

Finalmente llama a descripción de AñadirDocumento, que agrega el descriptor a la lista de documentos conocidos.

Descripción de PermisosDefault(listagrupos) da permisos por default a grupos a los que pertenece el usuario (RW) y permisos totales para el creador del documento (RWPDM).

## AÑADIR DOCUMENTO



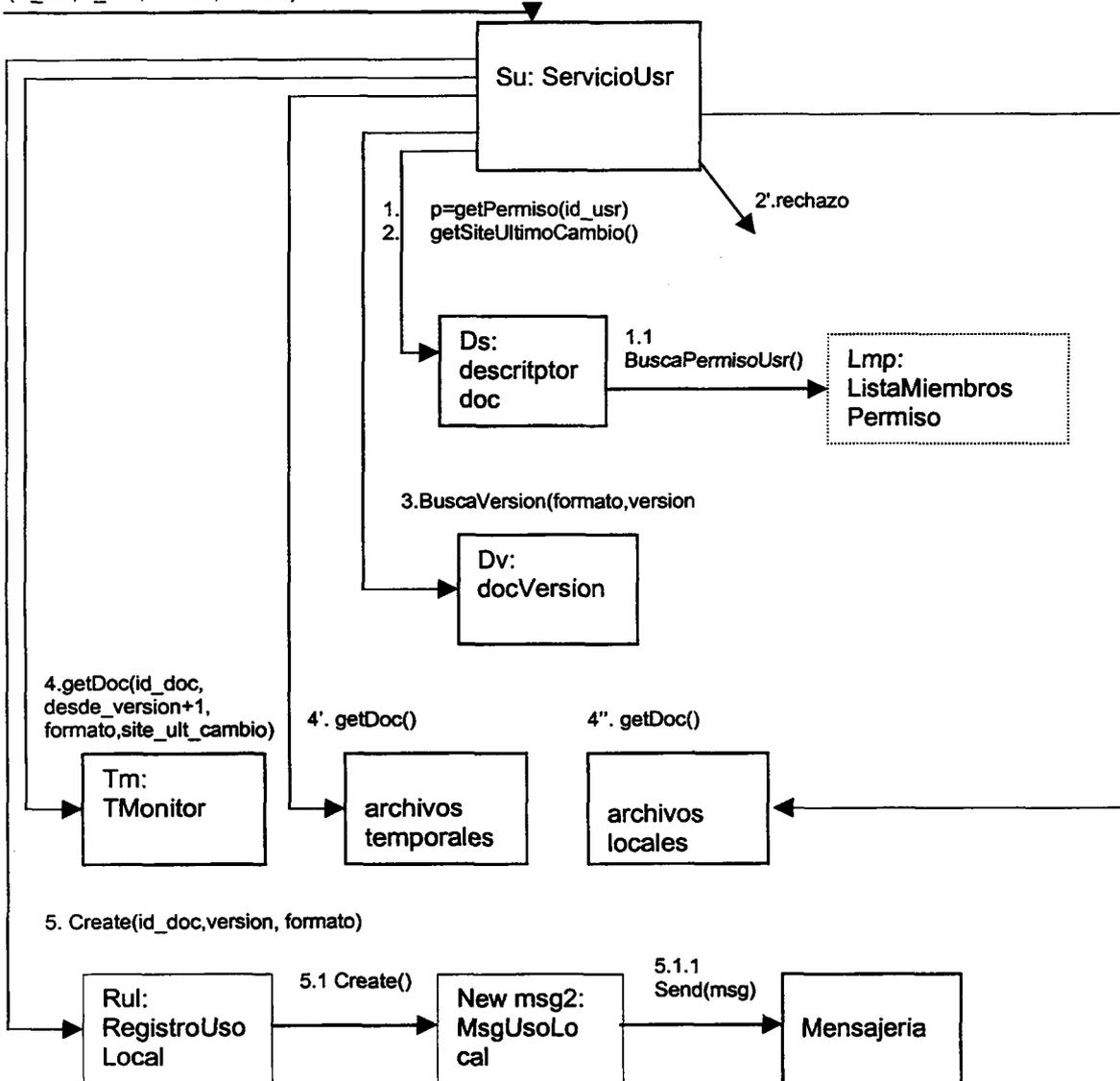
## DESCRIPCIÓN:

### Operación: AñadirDocumento

Se añade un nuevo descriptor de documento. Se crea un mensaje de notificación acerca de la creación del mismo y se busca a que áreas de interés a que pertenece. Por cada área, se pide al servicio de notificación de RICA envíe el mensaje. Se buscan los grupos a los que pertenece el documento (que tienen permiso) y por cada uno, se procede a enviar el mensaje de notificación a través de RICA.

## ACCESAR DOCUMENTO (GetVersion())

Accesardocumento  
(id\_usr, id\_doc, version, formato)



## DESCRIPCIÓN.

Operación- AccesarDocumento

Se pide el acceso a un documento para un usuario. Primero se revisa que tenga los privilegios de lectura (R ).

Se busca si se tiene una copia disponible en buffer del formato específico.

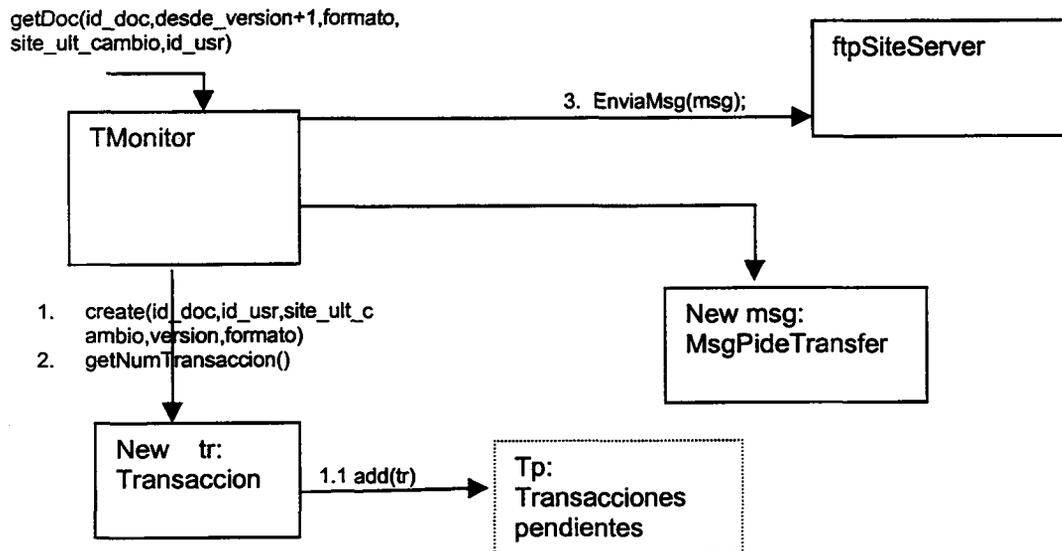
Se obtiene el site del último cambio, y se envía petición de pedir ese documento al monitor.

Finalmente, es necesario propagar el registro de uso del documento para formar el historial

Descripción de- GetDoc

Se detalla más adelante.

## GETDOC (OBTENER DOCUMENTO)



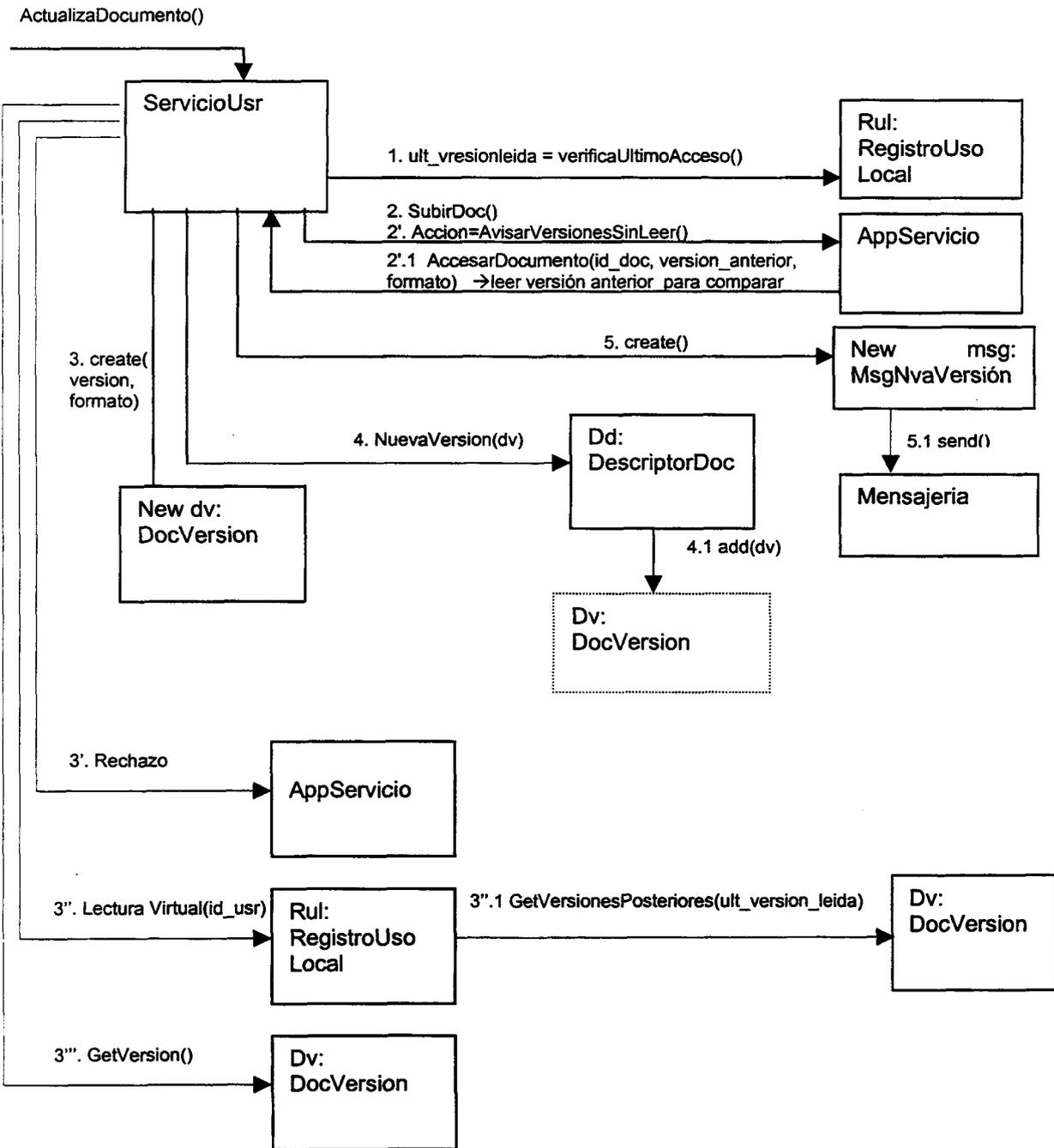
## DESCRIPCIÓN

Operación Getdoc

Realiza la petición de realizar una transferencia al módulo de transferencias.

Registra el inicio de una transacción (como pendiente).

# ACTUALIZA DOCUMENTO



## DESCRIPCIÓN

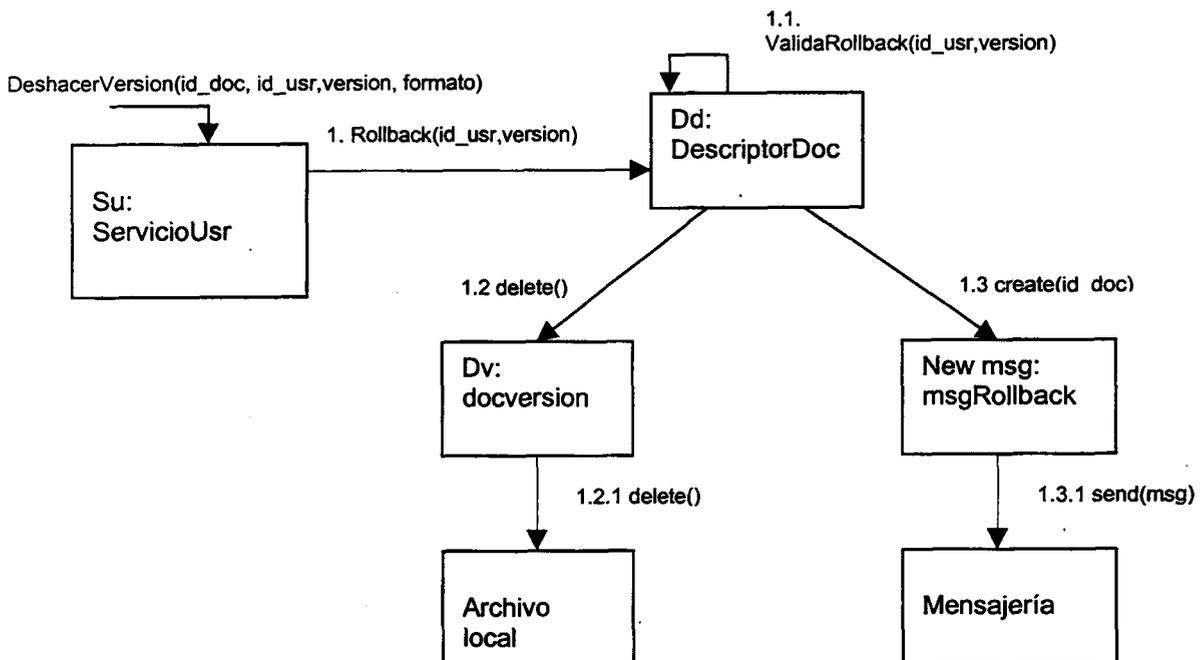
Operación: Actualiza documento

Crea una nueva versión del documento a petición del usuario.

Descripción de:

VerificaUltimoAcceso: revisa que no haya versiones nuevas entre la última consulta del documento y el tiempo registrado de lectura del archivo para ese usuario.

## DESHACER VERSION



## DESCRIPCIÓN:

Operación DeshacerVersion:

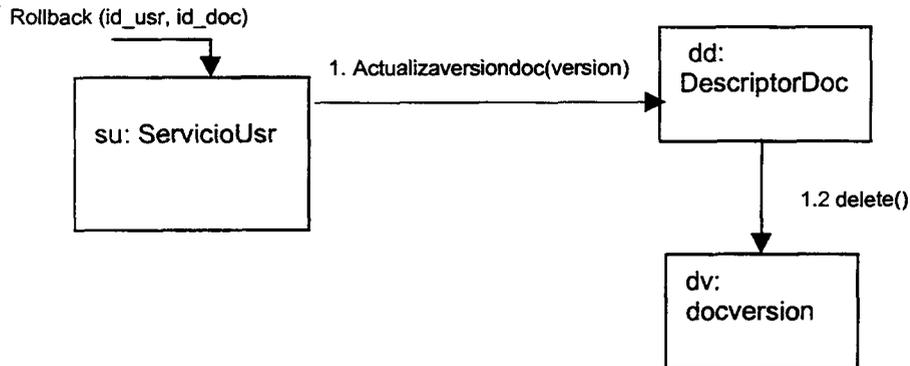
Se elimina la versión del documento y el formato, siempre y cuando el usuario tenga permisos de regresar esa versión (en este caso, haber sido el autor de la misma). Se notifica a los demás sites este hecho.

ValidaRollback(id\_usr, version)

Valida que `docversion.noversion = documento.descriptor.version` (última versión) y que

`docversion.autorcambio = id_usuario`

## REGRESAR VERSIÓN ANTERIOR (ROLLBACK)

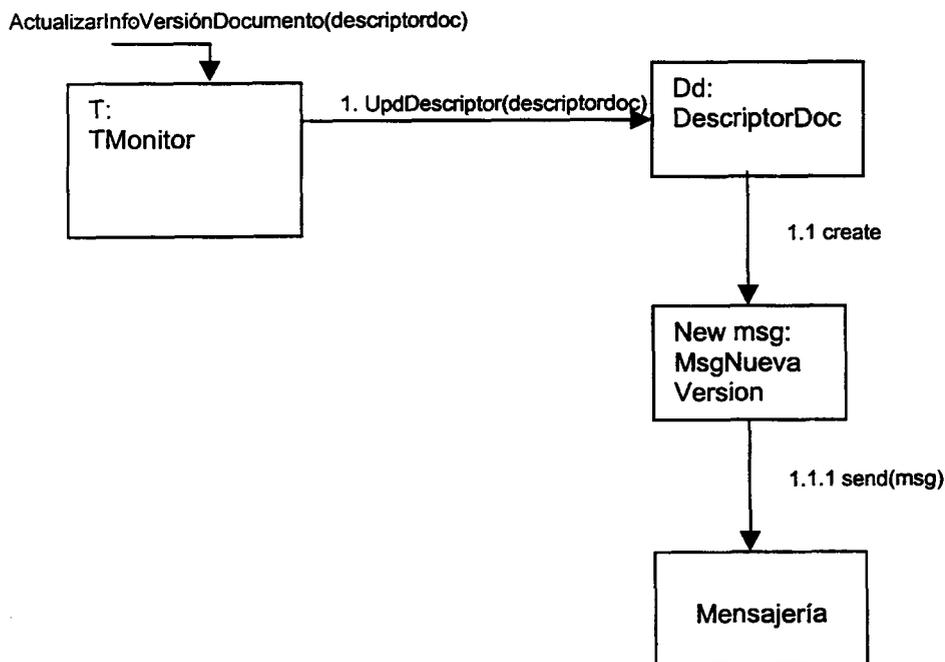


Descripción:

Operación rollback – actualiza el registro del descriptor del documento y elimina la versión correspondiente.

Esta operación se lleva a cabo en todos los sites. Si existe una copia física del archivo, será depurada automáticamente por hallarse en área temporal.

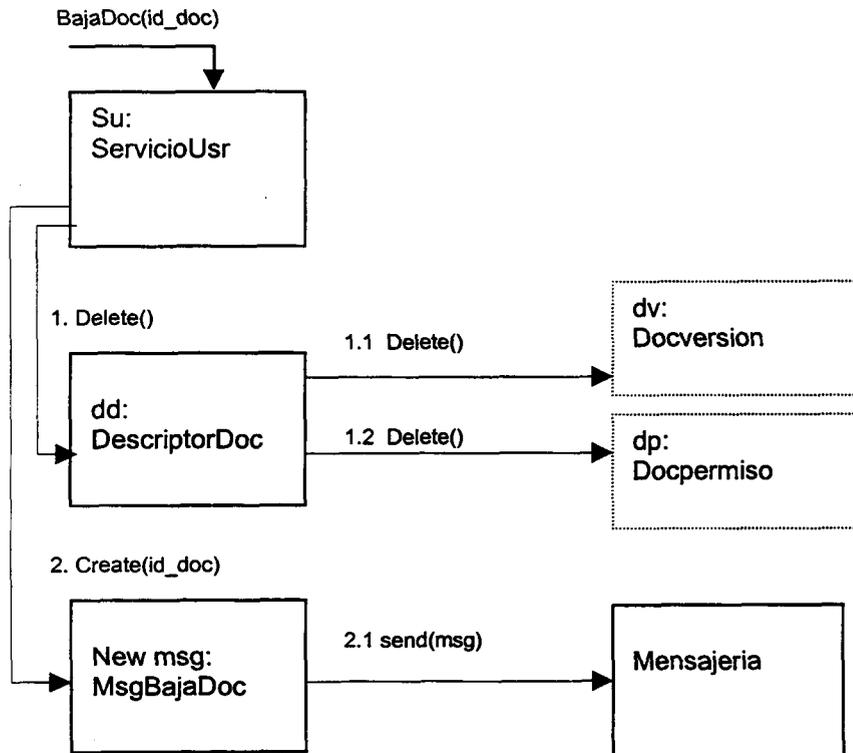
## ACTUALIZAR INFORMACION DE VERSIÓN-DOCUMENTO



**Descripción:**

Operación ActualizaInfoVersionDocumento – Se realiza en todos los sites, para actualizar la información de versiones y descriptores de documento cuando se ha modificado en otro site, para mantener la consistencia.

**BAJADOC**

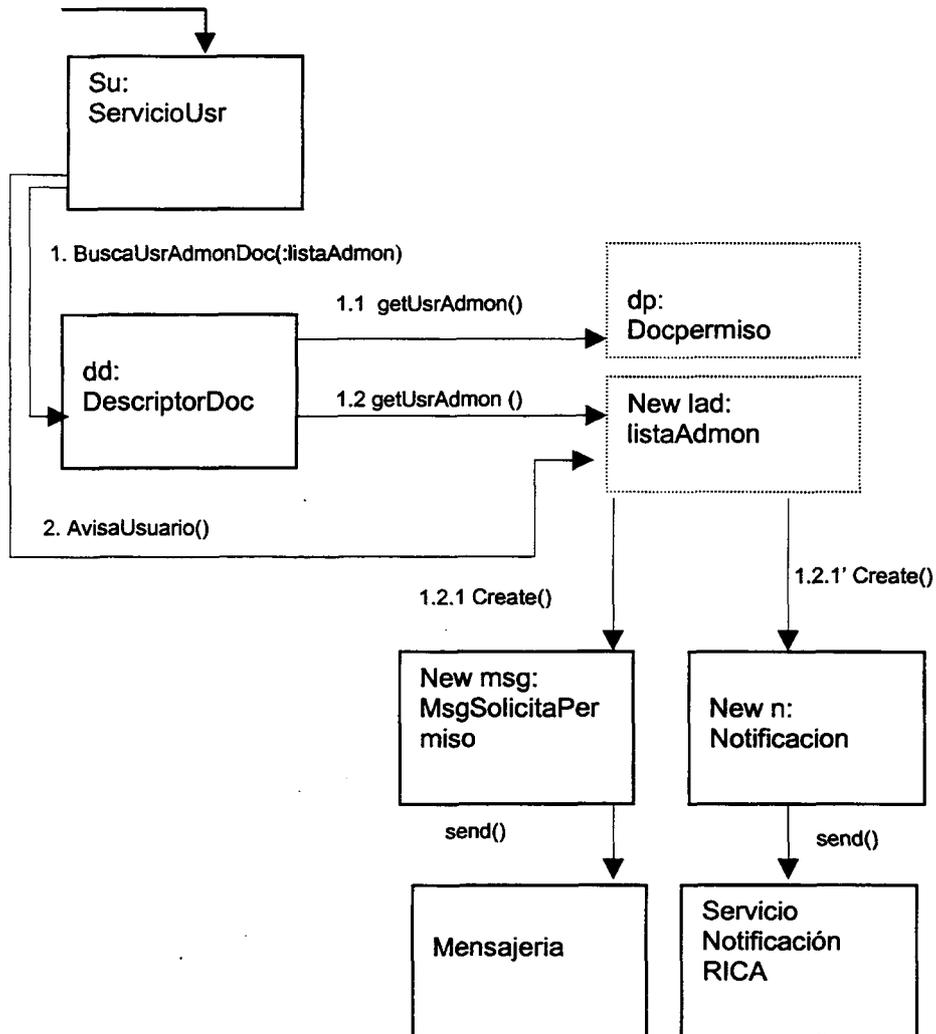


**DESCRIPCIÓN:**

Operación BajaDoc- Se borra tanto el descriptor del documento como los registros de versiones y los permisos asociados. Finalmente, debe transmitirse este cambio a través de MsgBajaDoc a los otros sites.

## SOLICITAR PERMISO SOBRE DOCUMENTO

SolicitarPermisosSobreDocumento(id\_doc)

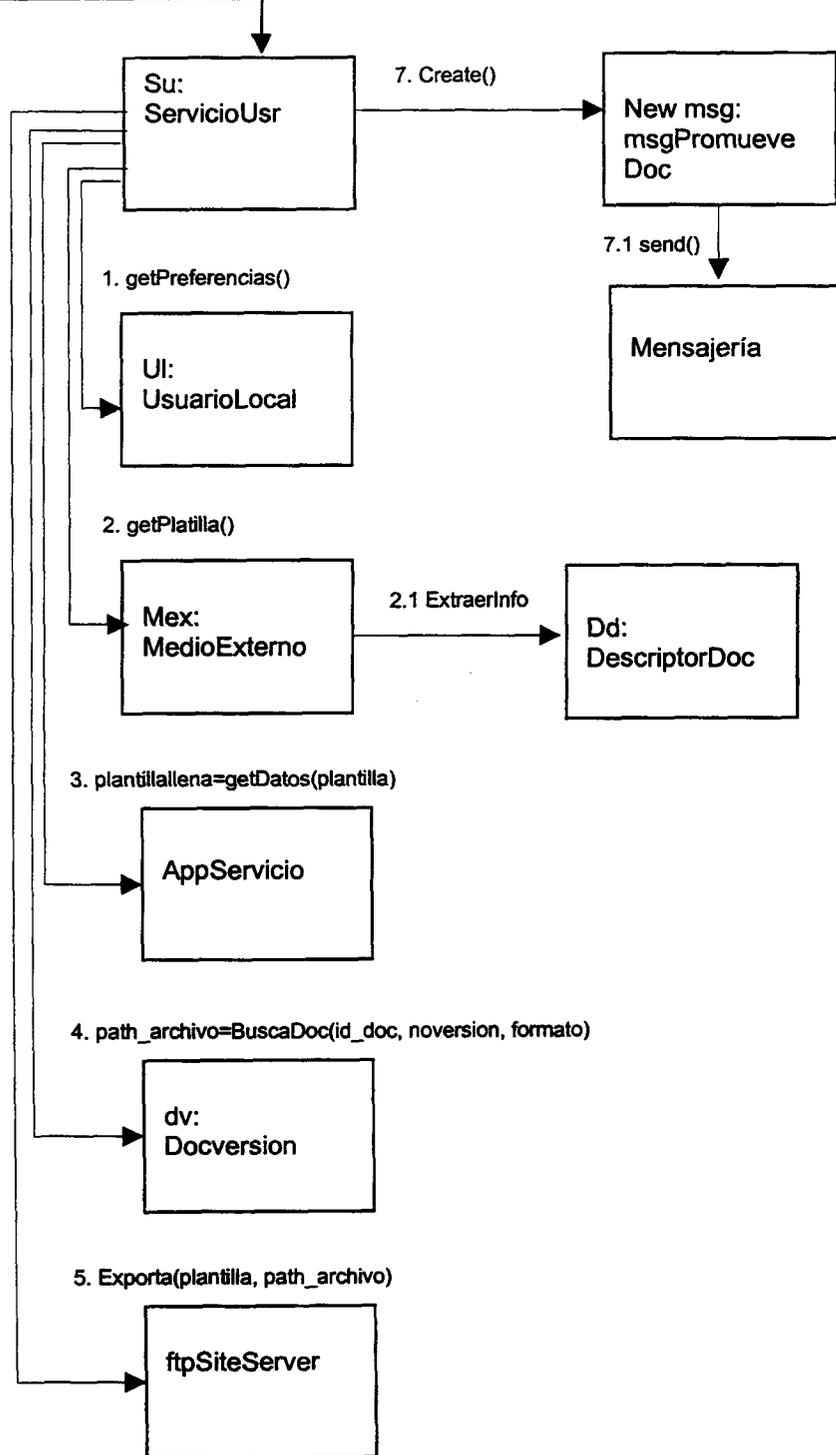


### DESCRIPCIÓN:

Operación SolicitaPermisosDocumento – Se buscan todos los usuarios que son administradores (o el owner del mismo). Por cada uno de ellos, si es usuario local, se crea un mensaje de notificación para RICA, en otro caso, se crea un mensaje que se transmite a otros servidores.

# PROMOVER DOC

PromoverDoc(id\_doc, noversion, formato, medio\_hacia)



## DESCRIPCIÓN:

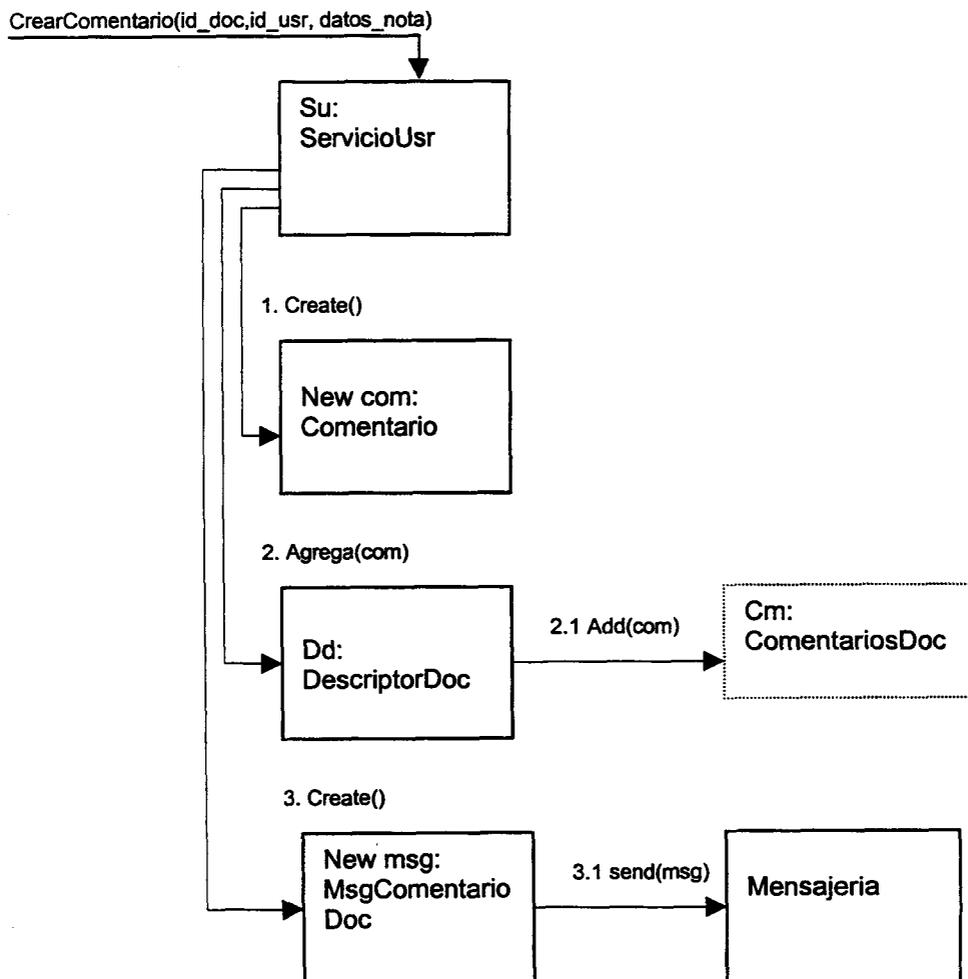
Operación PromueveDoc – Esta es una descripción genérica de lo que podría ser el proceso para exportar un documento a un medio externo.

Operación getPreferencias – busca en datos del usuario preferencias acerca de que medio acceder, o buscar información específica del usuario.

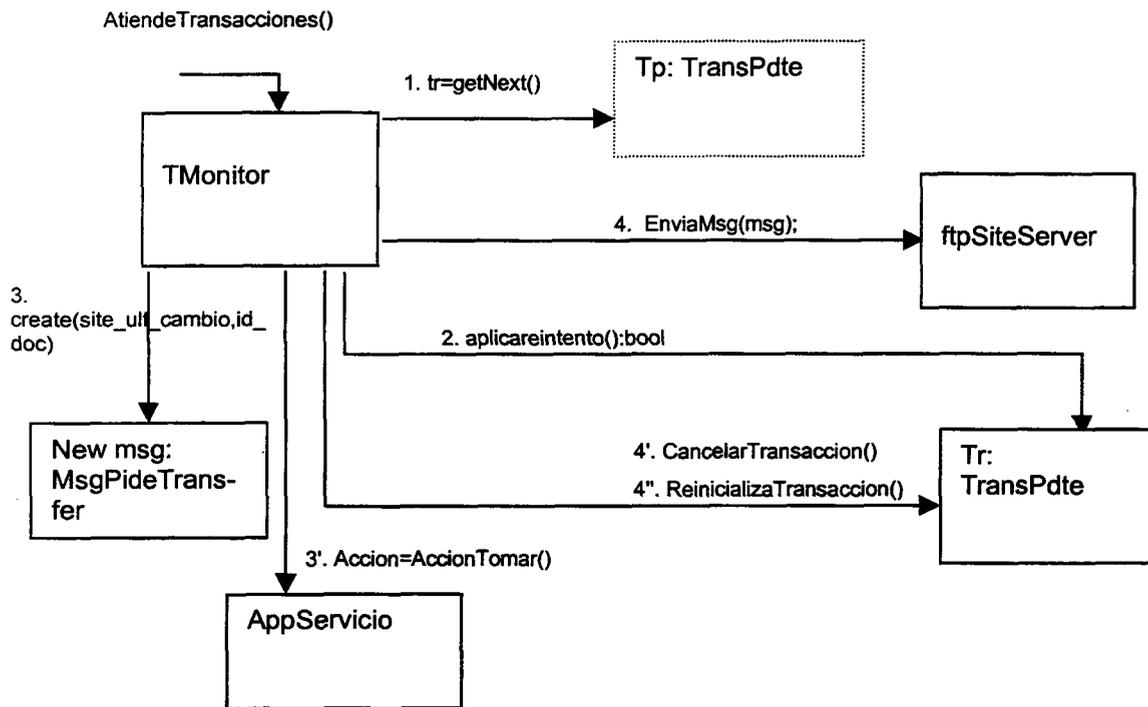
Operación getPlantilla – buscaría una forma prediseñada o “template” que incluye la información que se envía al medio externo. También puede incluir la búsqueda de un script o programa que procese el llenado de la forma, o puede buscar un archivo de texto que contenga el formato en que debe construirse la forma o plantilla para llenar los datos. En este punto es posible que la operación llene automáticamente algunos de los datos ya disponibles en la base de datos.

Operación getDatos – presenta los datos y la plantilla para que el usuario los complete o confirme, y finalmente envíe el archivo.

## CREAR COMENTARIO



## ATIENDE TRANSACCIONES

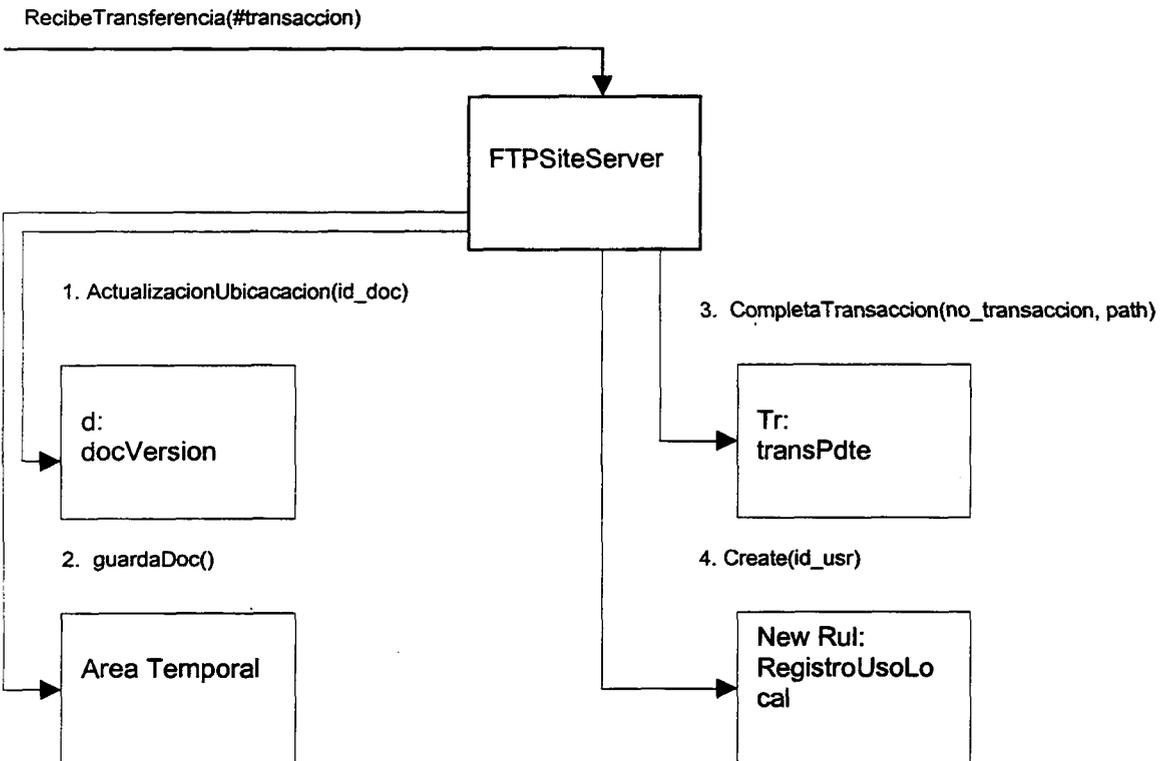


### DESCRIPCIÓN:

Operación AtiendeTransaccionesPendientes

Busca dar atención a transacciones pendientes, y verifica la acción a realizar en caso de falla.

## RECIBE TRANSFERENCIA

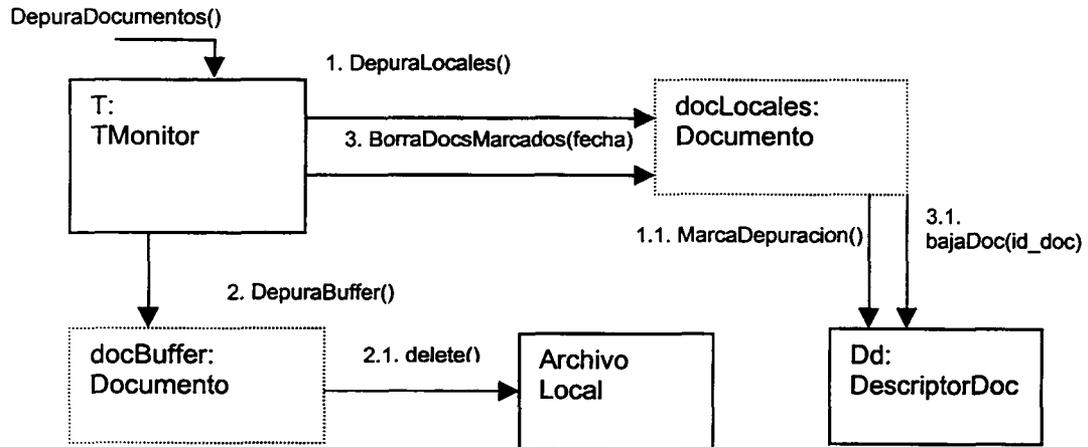


### DESCRIPCIÓN:

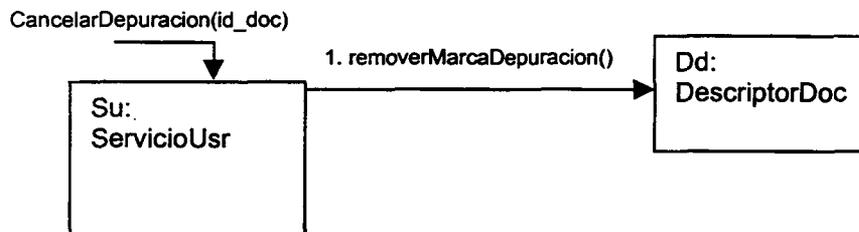
#### Operación RecibeTransferencia:

Una vez que llega un archivo solicitado, se completa la transacción iniciada con Tmonitor y este a su vez se encarga de dar almacenamiento y de crear los registros correspondientes tanto en el área de documentos copia (buffer) como de crear un registro de uso local. También se actualizan los parámetros de espacio ocupado tanto para el usuario como para el repositorio.

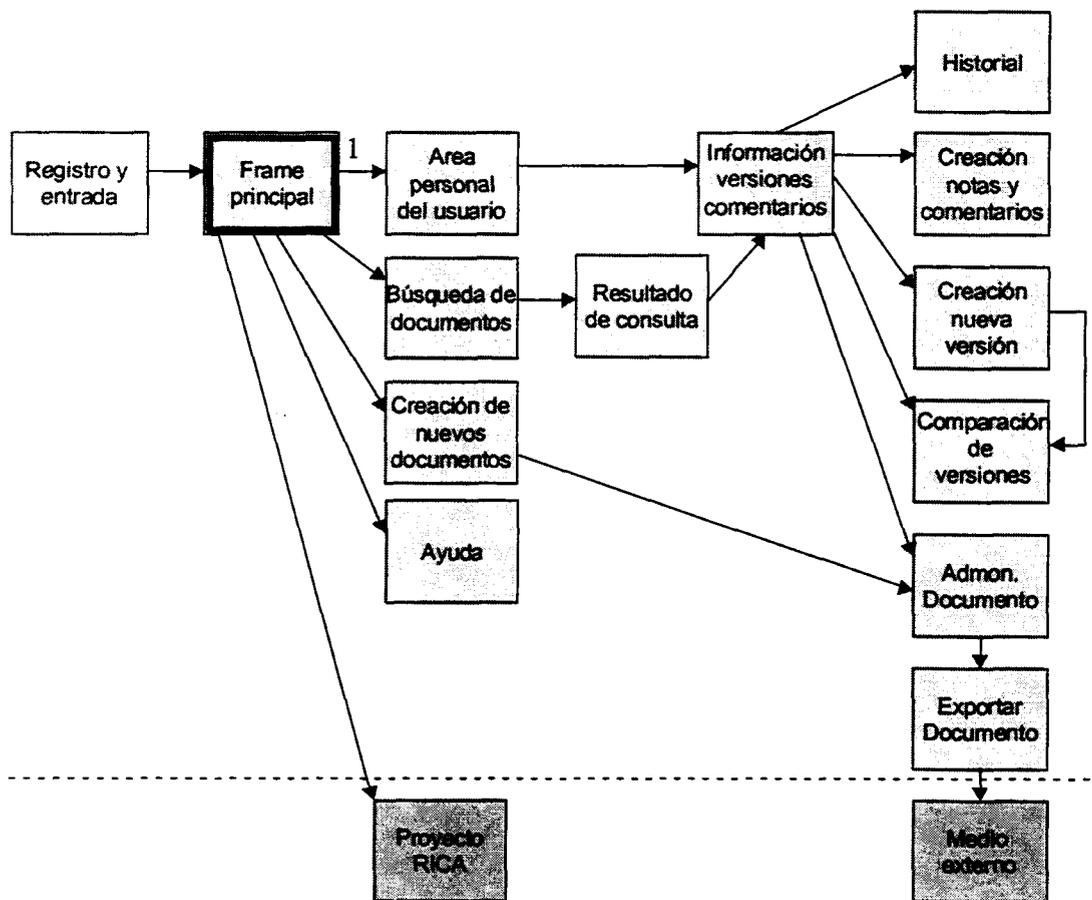
## DEPURAR DOCUMENTOS



## CANCELAR DEPURACIÓN



### A2.3 Diagrama de navegación



**Notas:**

Todas las pantallas están contenidas en el frame principal, de manera que siempre es posible regresar a este punto, a excepción de la liga al proyecto RICA y otros medios digitales, que son ligas externas al repositorio.

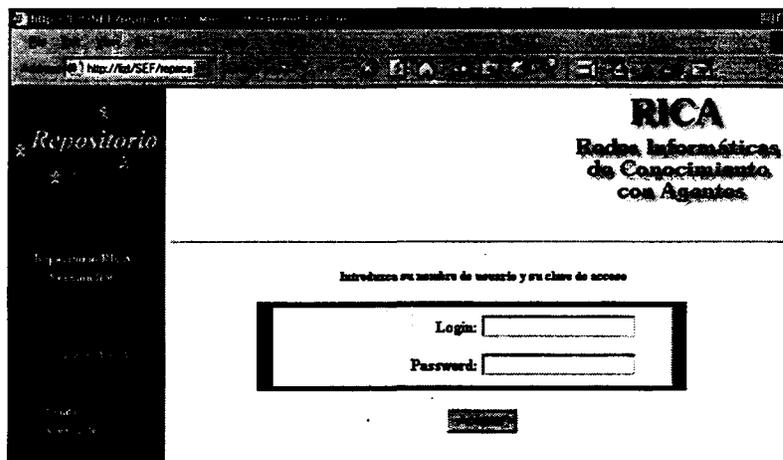
## Anexo 3 – Prototipo

El propósito de este anexo es mostrar las pantallas del prototipo, mostrando a manera de una visita guiada los aspectos principales de funcionalidad del mismo. En el anexo 2 se puede consultar el diagrama de navegación del mismo.

Como requisito previo al uso del sistema, el usuario deberá estar registrado en el sistema RICA, y haber declarado sus áreas de interés. El usuario puede o no pertenecer a uno o más grupos.

### 1. Acceso y entrada al sistema.

El usuario hará referencia a la liga donde se encuentre el repositorio local del sitio al que está inscrito, y deberá introducir su clave de usuario y contraseña.



2. Una vez dentro, se presenta al usuario una pantalla dividida en 2 secciones, donde la sección izquierda presenta el menú de opciones disponibles en el sistema.



4. Lo primero que se presenta al usuario es el área personal, donde se muestran los documentos en uso actualmente (acceso reciente) y noticias acerca de eventos relevantes dentro del repositorio. Es posible ir hacia el detalle de cada documento apuntando en el icono correspondiente.

**Area personal**

Usuario: Susana Eguia Franco  
Ultima consulta realizada: 01-Dec-1999

*Ultimos eventos de interés*

No hay nuevos eventos

*Documentos usados recientemente*

	<b>DOC PDTE</b> versión 3, Autor: Susana Eguia Franco
	<b>Manual de mysql</b> versión 1, Autor: Susana Eguia Franco

3. Para crear un nuevo documento, debe apuntarse hacia el botón de "Nuevo documento", y llenar la forma que se presenta. Esto se lleva a cabo en tres pasos, siendo el primero el alta de la información general,

**Nuevo Documento**

Paso 1 de 3: Gracias por dedicar parte de su tiempo a llenar este pequeño formulario.

Título:

Descripción:

Formato:

Ex caso de no utilizar los formatos de arriba, conteste lo siguiente:

Otro tipo de formato:  extensión:

**Paso 2 de 3:** Indique la categoría(s) del documento utilizando y para mayor o menor detalle. Presione para agregar a la lista. Para cancelarla, seleccione la categoría de la lista y presione .

Area: Computing Methodologies  
 Subarea:

**Paso 3 de 3:** Ahora ya puede proceder a enviar el archivo.

Pases a seguir: (1) Presionar "browser", (2) Presionar enviar

En esta sección pueden agregarse o removerse varias de las áreas seleccionadas.

Cuando se tiene la lista definitiva, se puede proseguir a enviar el archivo.

Después de presionar el botón de enviar, se indica si se pudo completar con éxito la operación, o se manejan los casos de error. En caso de éxito, se puede pasar directamente a ajustar los permisos de acceso para el documento.

4. Para la administración del documento se presentan las siguientes opciones:

**Administración del Documento**

Título: Exposición de QOML

Permisos de Acceso  
 Depurar  
 Exportar  
 Remover Marca Depuración

**Permisos de acceso:**

**Modificar**

Tipo usuario	Nombre	Permisos de acceso
	LIZT	Lectura
	Susana Egusa Franco	Administrador

Ir permisos por usuarios

**Permisos para GRUPOS:**

CETEClab	<input type="radio"/> Negar acceso
Gpo. desarrollo Alfa	<input type="radio"/> Solo consulta
Inv. Sistemas Multiagentes	<input type="radio"/> Permitir modificaciones
LIZT	
Prueba	
gpo. clase SMA	

5. Para consultar o buscar documentos, debe apuntarse al botón de "Búsqueda" en la barra izquierda, y prosigue la búsqueda de esta manera:

Buscar por: <input type="text" value="Area"/> <input type="button" value="Buscar"/>	Buscar por: <input type="text" value="Fecha modificación"/> Desde: <input type="text" value="25"/> <input type="text" value="11"/> <input type="text" value="1999"/> Ordenar por: <input type="text" value="Area"/> <input type="button" value="Buscar"/>
--	--

 **BUSQUEDA DE DOCUMENTOS MODIFICADOS DESDE: '25-Nov-1999'**

 **Manual de miniSQL** por *Susana Egua Franco*

Area:	D. Software, Computer Applications
Versión:	1
Fecha:	01-Dec-1999
Notas:	Manual del release anterior.

 **Exposicion de KOML** <sup>NEW</sup> *Susana Egua Franco*

Area:	I.2 ARTIFICIAL INTELLIGENCE
Versión:	1
Fecha:	01-Dec-1999

6. En el icono que aparece al lado del titulo de los documentos localizados, es posible ir al detalle del documento. Esta sección no se presentará a detalle, solo en lo referente al control de versiones.

 **Información del Documento**

- ▷ [Información general](#)
- ▷ [Áreas académicas relacionadas](#)
- ▷ [Versiones registradas: Orden Cronológico | Orden de Referencia](#)
- ▷ [Notas/Discusión](#)
- ▷ [Historial de Use](#)
- ▷ [Administrar Documento](#)
- ▷ [Agregar Versión](#)

**Información general**

Título:	Manual de miniSQL
Autor:	Susana Egua Franco

7. En la pantalla anterior, se muestran ligas hacia la consulta de versiones registradas, donde la información de versiones se presenta de la siguiente manera:

### Versiones registradas por Orden Cronológico

Se presentan primero las versiones más recientes.

<b>01-Dec-1999 04:55:43</b>	<a href="#">↓ obtener</a> <a href="#">↑ comentar</a> <a href="#">⊕ exportar</a> <a href="#">↔ comparar con...</a>
<b>No versión:</b> 3	
<b>Autor:</b> Susana Egua Franco	
<b>Formato:</b> HTML	
<b>Descripción:</b> seguimos haciendole cambios	

<b>01-Dec-1999 04:34:13</b>	<a href="#">↓ obtener</a> <a href="#">↑ comentar</a> <a href="#">⊕ exportar</a> <a href="#">↔ comparar con...</a>
<b>No versión:</b> 2	
<b>Autor:</b> Susana Egua Franco	

### Versiones registradas por Orden de Referencia

**Documentos bajo la versión 3**

Para este documento: [↓ obtener](#) [↑ comentar](#) [⊕ exportar](#) [↔ comparar con...](#)

<b>Fecha:</b> 01-Dec-1999 04:55:43
<b>Autor:</b> Susana Egua Franco
<b>Formato:</b> HTML
<b>Descripción:</b> seguimos haciendole cambios

**Documentos bajo la versión 2**

Para este documento: [↓ obtener](#) [↑ comentar](#) [⊕ exportar](#) [↔ comparar con...](#)

<b>Fecha:</b> 01-Dec-1999 04:34:13
<b>Autor:</b> Susana Egua Franco

En este punto son accesibles funciones como exportar a otro medio digital la versión seleccionada, comparar con otra (mediante un diálogo posterior y solo para formatos oficiales soportados como son HTML, PDF, PS y texto).

8. Volviendo a la página de información general, es posible agregar una versión (siempre y cuando se trate de un usuario autorizado), donde aparece un diálogo similar pero más breve al del alta del documento. En caso de ser necesaria una acción posterior del usuario, como por ejemplo una lectura virtual o cancelación, se indicará de esta manera. En caso contrario, se presenta un mensaje de confirmación.

La pantalla que se muestra a continuación ilustra un caso donde se presenta la posibilidad de hacer una lectura virtual, pues no se ha consultado la versión 3.

**⚠ Información**

Existen versiones de este documento no-leídas desde su última consulta:

Versión	Modificado por	Fecha	Formato
3	Susana Eguia Franco	01-Dec-1999	HTML

seguimos haciendole cambios

Si desea continuar agregando esta nueva versión, siga [esta liga](#). Presiones [aquí](#) para volver a la información del documento.

9. Finalmente, se presenta un ejemplo para exportar un documento a un medio digital. Esto puede hacerse desde 2 puntos, ya sea, desde la consulta para versiones, o desde la administración de documentos, donde se permite seleccionar la versión a exportar. Una vez seleccionada, se podrá seleccionar el medio de salida para el mismo:

**Exportar Documento**

Exportar hacia:

Medios registrados:

 **Phronesis**  
Biblioteca Digital  
Biblioteca digital Phronesis  
DGHII Campus Monterrey ITESM  
Area de conocimiento: Ciencias Computacionales

Una vez seleccionado el medio a que se desea exportar, se presenta una forma con los metadatos requeridos por la contraparte, algunos de los cuales se proponen de manera automática, pero deben ser corroborados por el usuario, así como completar los campos restantes.

**Paso 1 de 1: Gracias por dedicar parte de su tiempo a llenar este pequeño formulario.**

Los datos que aquí se solicitan son necesarios para exportar el documento exitosamente. Verifique que la información presentada sea correcta, y complete la que hace falta.

Título:

Tema (s):

(Separelos por comas)

Autor (es):

(Separelos por comas)

Descripción:

10. Finalmente, se presenta una propuesta para facilitar la comparación de versiones, la cual se encuentra actualmente en fase de desarrollo.

Comparación de versiones						
← V. 1.0	F Siguiente	↑ Anterior	Adiciones en verde 	Modificaciones en gris 	Lineas borradas en rojo 	V. 2.0 →
<ol style="list-style-type: none"> <li>1. Este es el texto que es igual en sus 2 versiones</li> <li>2. continuando con lo que tenemos, esto debe ser un cambio</li> <li>3. y esto texto normal otra vez</li> <li>4.</li> <li>5.</li> <li>6. texto normal otra vez</li> <li>7. mas texto normal</li> <li>8. esto esta borrado</li> <li>9. es texto normal otra vez</li> </ol>			<ol style="list-style-type: none"> <li>1. Este es el texto que es igual en sus 2 versiones</li> <li>2. este era otro texto que habia</li> <li>3. y esto texto normal otra vez</li> <li>4. aqui hay una adiccion</li> <li>5. de 2 lineas</li> <li>6. texto normal otra vez</li> <li>7. mas texto normal</li> <li>8.</li> <li>9. es texto normal otra vez</li> </ol>			

## Referencias bibliográficas

- [APP 92] **Apple Computer, "Developing a Community Internet", video, 1992.**
- [BAE 93] **Baecker, Ronald M.; "Readings in Groupware and Computer-Supported Cooperative Work: Assisting in Human-Human collaboration"; Morgan Kaufman, 1993**
- [BAN 91] **Bannon, Liam J.; Schmidt, Kjeld; "CSCW: Four characters in search of a context"; Studies in Computer Supported Cooperative Work; J.M. Bowers y S.D. Benford (Editores), Elsevier Science Publishers B.V., Holanda del Norte, 1991.**
- [BEN 97] **Bentley, R.; Applet, W.; Busbach, U.; Hinrichs, E.; Kerr, D; Sikkel, K.; Trevor, J.; Woetzel, G.; "Basic support for cooperative work on the world wide web", CSCW Group, Instituto de Tecnología Informática Aplicada (GMD FIT), Centro de Investigación Nacional Alemán para la Tecnología Informática), International Journal of Human Computer Studies, Primavera, 1997**
- [BEN2 97] **Bentley, Richard; Applet, Wolfgang; "Designing a system for cooperative work on the World-Wide Web: Experiences with the BSCW system", (CSCW Group, Instituto de Tecnología Informática Aplicada (GMD FIT), Centro de Investigación Nacional Alemán para la Tecnología Informática)**
- [BEN3 97] **Bentley, R., Horstmann, T. and Trevor, J.; "The World WideWeb as enabling technology for CSCW: The case of BSCW", Computer Supported Cooperative Work: The Journal of Collaborative Computing. Special issue on CSCW and the Web, Vol. 6, Nos 2-3, Kluwer Academic Publishers, 1997**
- [BEN4 97] **Bentley, R., Appelt, W., Busbach. U., Hinrichs, E., Kerr, D., Sikkel, S., Trevor, J. And Woetzel, G.; "Basic Support for Cooperative Work on the World Wide Web", International Journal of Human-Computer Studies 46(6): Special issue on Innovative Applications of the World Wide Web, Academic Press, Junio 1997, p. 827-846**
- [BEN5 97] **Bentley,R., Täschner, T. and Appelt, W.; "The freedom to choose: Transforming content on-demand in the BSCW Shared Workspace system", Proceedings of WebNet 97, Noviembre 1997**
- [BEN6] **Bentley, R. and Appelt, W.; "Designing a System for Cooperative Work**

- 97] on the World-Wide Web: Experiences with the BSCW System  
Proceedings of HICSS'30: The Hawaii International Conference on the  
System Sciences", Maui, Hawaii, 7-10 Enero 1997, IEEE Computer  
Society Press.
- [BER 98] Berg, Marc; **"The politics of technology: On bringing social theory  
in technologicval design"**; Science, Technology & Human Values;  
Cambridge; Sage Publications, Otoño 1998.
- [BRE 98] Brena, Ramón; Página del doctor Ramón Brena- "Temas de tesis";  
<http://www-cia.mty.itesm.mx/~rbrena>, 1998
- [BRE2 98] Brena, Ramón; Aguirre, José Luis; Franco, Ricardo; Armero, Alejandro;  
"COoRdinación de Recursos de Educación e investigación mediante  
Agentes", <http://lizt.mty.itesm.mx/RICA/Reporte98.htm>
- [COU 96] Coulouris, George; Dollimore, Jean; Kindberg, Tim; "Distributed  
systems: concepts and design", Addison -Wesley, 2da. Edición,  
Londres, 1996
- [DIX 98] Dix, Finlay, Abowd & Beale, **"Human-Computer Interaction"**, Prentice  
Hall Europe, 1998.
- [FIN 93] Finin, Tim; Weber, Jay; Wiederhold, Gio; Genesereth, Michael; Fritzon,  
Richard; McKey, Donald; McGuire, James; Pelavin, Richard; Shapiro,  
Stuart; Beck, Chris; **"Draft specification of the KQML Agent-  
Communication Language"**; The Darpa Knowledge Sharing Initiative,  
External Interfaces Group; Junio 1993.
- [GMD 99] **"BSCW Homepage"**; GMD; Julio 1999; <http://bscw.gmd.de/>
- [GMD 99] CoopWWW Project  
<http://orgwis.gmd.de/projects/COOPWWW/>
- [GYR 95] **"La industria electrónica y las políticas de desarrollo tecnológico"**,  
GYROS: Revista de ingeniería y ciencias básicas, Año 2, nº. 2, 1995,  
pp. 90
- [HUG 98] Hughes, David; **Página de Minisql**, <http://www.Hughes.com.au>
- [KHA 94] Khanna, Raman (editor); "Distributed computing: Implementations and  
Management strategies"; Prentice Hall; Nueva Jersey, EEUU 1994
- [KOC 97] Koch, Michael; "CSCW, Computer Supported Collaborative Work";  
Technische Universität München, Alemania, en  
<http://www.telekooperation.de/cscw/cscw.html>, Abril 1997

- [MAL 90] Malone, T.W., Crowston, K., "What is coordination theory and how can it help design cooperative work systems?", Proceedings on CSCW, Oct. 1990, pp. 357-370
- [MAR 92] Marca, David; Bock, Geoffrey; "**Groupware: Software for Computer-Supported Cooperative Work**"; IEE Computer Society Press; 1992
- [PHR 99] Documentación del sistema Phronesis  
<http://copernico.mty.itesm.mx/~tempo/Proyectos/>
- [RAN 97] Tamura, Randall; "**Lotus Notes and Domino 4.6 Unleashed**"; SAMS Publishing Co.; 1997
- [ROD 91] Rodden, Tom; Blair, Gordon; "CSCW and Distributed Systems: The problem of control"; Proceedings of the Second European Conference on Computer-Supported Cooperative Work; Bannon, L., Robinson M. & Schunidt, K. (Editores). Sept. 1991, Amsterdam, Holanda
- [SCH 92] Schatz, Bruce R.; "**Building an electronic Community System**"; Journal of Management Information Systems; Invierno 1991-1992, Vol. 8, no. 3, pp. 87-107.
- [SCH 97] Schlichter, Johann; Koch, Michael, Bürger, Martin; "**Workspace awareness for distributed teams**", para Workshop Coordination Technology for Collaborative Applications, Singapur, 1997, W. Conen (ed), Lecture Notes on Computer Science, Springer, <http://www11.informatik.tu-muenchen.de/publications/html/schlichter97.htm>
- [SHN 97] Shneiderman, Ben; "**Codex, memex, genex: The pursuit of transformational technologies**", International Journal of Human-Computer Interaction, Diciembre 22, 1997
- [SIK 97] Sikkel, K.; "A Group-based Authorization Model for Cooperative Systems", Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'97), Lancaster, Septiembre. 1997, Kluwer Academic Publishers
- [STI 98] Stiemerling, Oliver; Cremers, Armin B.; "**The use of cooperation scenarios in the design and evaluation of a CSCW System**"; IEEE Transactions on Software Engineering, Vol. 24, 12, 1998 pp. 1171-1181.
- [TUM 97] "**CSCW- Computer Supported Cooperative Work**", Tum Informatik XI; <http://www.telekooperation.de/cscw/cscw.html> ; Abril de 1997
- [HEE 98] Heecheol Jeon, **JatLite**, <http://java.stanford.edu/>, CDR, Stanford

University, 1998

Centro de Información-Biblioteca



30002006792239