



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO

Departamento de Computación

Presentación gradual de información de conciencia de grupo

Tesis que presenta:

Abel Cortazar May

para obtener el Grado de:

**Maestro en Ciencias
en Computación**

Directora de Tesis:

Dra. Sonia Guadalupe Mendoza Chapa

Resumen

Actualmente, las grandes organizaciones tienden a mejorar sus servicios debido a la creciente competencia y a los mercados cambiantes, lo cual requiere que los diferentes departamentos de una organización cooperen para reducir costos, mediante el uso compartido de recursos, y para reducir las barreras de tiempo/espacio, mediante el acercamiento de información relevante a los miembros de la organización. Algunas organizaciones gubernamentales se apoyan en sistemas de información (e.g., POLITeam [Sohlenkamp et al., 2000]) para acelerar algunos procesos cooperativos, como la toma de decisiones. Sin embargo, estos sistemas se centran principalmente en entornos virtuales que proveen a los colaboradores información de conciencia de grupo sobre sus colegas o sus recursos compartidos, aun si ellos no están interesados en dicha información. Por medio de estos sistemas, los colaboradores no pueden acceder progresivamente a la información de conciencia grupo a medida que la necesitan. Ni los productores ni los consumidores de información pueden definir y controlar la clase de información de conciencia de grupo que requieren enviar y recibir respectivamente. En esta tesis de maestría, se propone un sistema de información que proporciona a los miembros de una organización información gradual de conciencia de grupo, proveniente de entornos tanto virtuales como físicos. La información gradual varía desde información de conciencia de grupo periférica o de nivel macro (e.g., presencia) hasta información de conciencia enfocada o de nivel micro (e.g., disponibilidad y acciones detalladas) pasando por niveles intermedios de conciencia (e.g., ubicación y proximidad). De esta manera, los miembros de una organización, inicialmente pueden acceder desde una PDA a la información mínima sobre otros o sobre algunos recursos de acuerdo a su rol. Después, si es necesario, pueden acceder gradualmente a información más detallada dependiendo de sus necesidades. Por medio de esta propuesta de solución, los colaboradores pueden adquirir el nivel de conciencia de grupo requerido para evitar pérdidas de oportunidades o saturación de información irrelevante.

Palabras clave: trabajo cooperativo asistido por computadora, conciencia periférica y enfocada, conciencia gradual, entornos virtuales y físicos.

Abstract

Nowadays, large organizations tend to improve their services due to competition and changing markets, which requires that the different departments of an organization cooperate to reduce costs by sharing resources and time/space barriers by bringing outstanding information closer to the organization staff. Some organizations rely on information systems (e.g., POLITeam [Sohlenkamp et al., 2000]) to speed up some cooperative processes such as the decision making. However, these systems mainly focalize on virtual environments that provide collaborators with group awareness information about their colleagues and their shared resources, even if they are not interested in such information. By means of these systems, the collaborators cannot progressively access to the group awareness information as they need. Neither the information producers nor the consumers can define and control the kind of group awareness information they require to send and to receive, respectively. In this master thesis, we propose an information system that provides the organization members with gradual group awareness information coming from both virtual and physical environments. Gradual information varies from peripheral or macro-level awareness information (e.g., presence) to focused or micro-level awareness information (e.g., availability and detailed actions) going by intermediate awareness levels (e.g., location and proximity). In this way, the staff of an organization initially can access from a PDA to the minimal information about others or about some resources according to their role. Afterwards, if necessary, they can gradually access to more detailed information depending on their needs. By means of this proposed solution, collaborators can acquire the required group awareness level to avoid loss of opportunities or saturation of irrelevant information.

Keywords: cooperative work, peripheral and focused awareness, gradual awareness, physical and virtual environments.

Agradecimientos

Doy gracias a Dios el único sabio, Rey de los siglos, inmortal, invisible, sea honor y gloria por los siglos de los siglos, por darme la fuerza, el conocimiento y la sabiduría para concluir el presente trabajo de tesis.

A mis asesores por su apoyo incondicional mis más sinceros agradecimientos.

Dra. Sonia Mendoza Chapa:

Gracias por darme la oportunidad de trabajar con usted y por su apoyo en los momentos difíciles que se tornó en en el desarrollo de este trabajo, así como por su paciencia y dedicación durante todo el tiempo de mis estudios. Principalmente le agradezco por transmitirme sus conocimientos que serán útiles en mi vida profesional.

Dr. José Guadalupe Rodríguez:

Le agradezco por su tiempo y por su aporte en el desarrollo de este trabajo de tesis. También, por la impartición de sus clases, las cuales serán de provecho en mi vida profesional.

Dr. Joaquín Sergio Zepeda Hernández:

Gracias por invertir su tiempo en la revisión de este trabajo de tesis y por aportar sus ideas en el enriquecimiento de dicho trabajo.

Agradezco al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional por darme la oportunidad de concluir los estudios y apoyarme durante toda mi estancia en el Departamento de Computación.

Gracias a la Universidad Juárez Autónoma de Tabasco por darme la oportunidad y las facilidades durante todo el tiempo del posgrado y por apoyar mi superación académica en el CINVESTAV.

Agradezco a todos los profesores que compartieron sus conocimientos durante la estancia en el departamento.

Gracias a toda mi familia por preocuparse en todo momento de mí.

A mi amada esposa: *Gracias por su paciencia, por su amor incondicional y por animarme en momentos difíciles. Te agradezco ¡amor! por esperarme en las madrugadas y por confiar en mí.*

A mis hijos:

Les agradezco todos esos momentos felices que pasamos, pero también por no quejarse en los momentos de mi ausencia.

A mis padres:

Gracias porque siempre estuvieron a mi lado, sin importar la distancia. Principalmente por brindarme su apoyo incondicional y por que sus consejos siempre me ayudaron a tomar decisiones correctas en mi vida.

A mi hermana Sara:

Le agradezco por brindarme su apoyo en realizar todos los trámites en mi centro de trabajo, pero más que nada gracias por tenerte como mi hermana.

A mis hermanos en Cristo:

Gracias por preocuparse y por sus rogativas ante el Padre celestial, ya que siempre me impulsaron a no caer en los tiempos de debilidad.

Agradezco a todos mi compañeros de estudios por todos los momentos de alegría.

Índice general

Resumen	iii
Abstract	v
1 Introducción	1
1.1 Contexto de investigación	1
1.2 Planteamiento del problema	4
1.3 Objetivos del proyecto	6
1.4 Organización del documento	7
2 Estado del arte	9
2.1 Modelos relevantes en sistemas colaborativos	9
2.1.1 Modelo de observabilidad publicada y filtrada	10
2.1.2 Modelo de interacción espacial	10
2.2 Sistemas conciencia periférica y enfocada	18
2.2.1 Sistemas de conciencia periférica	18
2.2.2 Sistemas de conciencia enfocada	24
2.3 Descripción del sistema Community Bar	35
2.4 Discusión sobre el trabajo relacionado	38
3 Análisis y diseño del sistema SIGCo	43
3.1 Escenario de uso del sistema SIGCo	43
3.2 Análisis del sistema SIGCo	47
3.2.1 Descripción de recursos	47
3.2.2 Casos de uso del sistema SIGCo	48
3.3 Clases del sistema SIGCo	51
3.4 Arquitectura de <i>software</i> del sistema SIGCo	52
3.4.1 Capa de Servidor	54
3.4.2 Capa de comunicación	56
3.4.3 Capa de cliente	60
3.5 Arquitectura de distribución del sistema SIGCo	67
4 Implementación del sistema SIGCo	71
4.1 Tipos de recursos	71
4.1.1 Recursos humanos	71

4.1.2	Recursos físicos	72
4.1.3	Recursos virtuales	73
4.2	Herramientas utilizadas en el desarrollo de SIGCo	75
4.2.1	Base de datos	75
4.2.2	Lenguaje de programación	76
4.2.3	Hardware	77
4.3	Implementación de las clases del sistema SIGCo	77
4.3.1	Núcleo funcional	78
4.3.2	Interfaces	78
4.4	Pruebas y Resultados del sistema SIGCo	83
4.4.1	Interfaz de usuario para la administración del sistema SIGCo	84
4.4.2	Interfaces de usuario del cliente SIGCo	89
4.4.3	Resultados	94
5	Conclusiones y trabajo futuro	97
5.1	Síntesis de la problemática	97
5.2	Conclusión	98
5.3	Contribuciones	99
5.4	Implicaciones	99
5.5	Limitaciones	100
5.6	Trabajo futuro	100
A	Instalaciones y configuraciones	103
A.1	Instalación de QT C++ <i>Embedded open source</i>	103
A.2	Instalación del sistema SIGCo	104
	Bibliografía	105

Capítulo 1

Introducción

1.1 Contexto de investigación

La presente tesis de maestría se inscribe en el campo de investigación denominado Trabajo Cooperativo Asistido por Computadora (TCAC), el cual estudia los aspectos sociales de las actividades individuales y colectivas, así como los aspectos tecnológicos de la información y de las comunicaciones, con el fin de soportar la colaboración entre personas potencialmente distribuidas.

Debido al carácter multidisciplinario del TCAC, esta tesis se centra en los aspectos tecnológicos de este campo de investigación, con el fin de diseñar e implementar un sistema colaborativo que presente gradualmente a los colaboradores, información de conciencia de grupo proveniente de entornos físicos y virtuales. El concepto de “información gradual” se refiere al tipo y al volumen de información que los colaboradores pueden visualizar desde sus dispositivos de cómputo.

Un sistema colaborativo es un conjunto de programas informáticos que soporta grupos de personas involucradas en una tarea u objetivo común y que provee una interfaz a un ambiente compartido [Ellis and Wainer, 1994]. Desde el punto de vista funcional, un sistema colaborativo está constituido por tres espacios que abordan los aspectos esenciales de las actividades colectivas [Calvary et al., 1997]:

1. El **espacio de trabajo** denota el conjunto de objetos del dominio de la tarea que modela la producción colaborativa de entidades compartidas. Los editores de documentos como AllianceWeb descrito en [Decouchant et al., 2008] y las herramientas de soporte para la toma de decisiones como OpenSpace-online [Ender, 2009] son algunos ejemplos de sistemas que ofrecen espacios de trabajos.
2. El **espacio de coordinación** abarca las dependencias lógicas, temporales y espaciales de las actividades colaborativas. Además, este espacio provee los medios para permitir que los colaboradores trabajen eficientemente y en conjunto para alcanzar una meta. Algunos ejemplos de sistemas que privilegian el espacio de coordinación son las herramientas de flujo de trabajo tal como Kepler/Pegasus en [Mandal et al., 2007] y los sistemas de gestión de proyectos tal como es P3M3 [Maqsood and Javed, 2007].

3. El **espacio de comunicación** es el medio por el que se lleva acabo la comunicación entre los colaboradores. Los sistemas de espacios de media como JamSpace [Gurevich, 2006] y de correo electrónico AlphaMail [Kay, 2007] son algunos ejemplos de sistemas que apoyan la comunicación.

La interfaz del usuario no constituye un tercer espacio funcional, sino que está implícita en los espacios de trabajo, de coordinación y de comunicación. Como cualquier función específica del dominio de la tarea, los servicios que proveen los tres espacios funcionales deben ser observables y accesibles a los colaboradores para coadyuvar a la efectividad de la colaboración.

Las investigaciones en TCAC han dejado ver la importancia de proveer a colaboradores físicamente distribuidos con información que:

- a) substituya la carencia de señales que se presentan en ambientes cara a cara,
- b) que provea en ellos una homogeneidad de fundamentos.

Precisamente, la coordinación se centra en promover bases comunes y motivar las actividades personales [Domingos et al., 1998]. Estos aspectos hacen énfasis en la necesidad de proveer mecanismos flexibles de coordinación que faciliten el trabajo colaborativo a través de la conciencia de grupo.

El concepto de “conciencia de grupo” se define como *un conocimiento de las actividades de otros que provee un contexto para las actividades propias* [Dourish and Bly, 1992]. De acuerdo con Dourish, la conciencia de grupo asegura que las actividades personales serán relevantes a las actividades colectivas [Dourish, 1997]. Además, la conciencia de grupo garantiza una evaluación implícita de la relevancia de las actividades personales con respecto a los objetivos comunes y al progreso del trabajo colaborativo. Entre la información de conciencia de grupo que es relevante para los colaboradores que interactúan en un entorno virtual, se puede identificar:

- a) la presencia,
- b) la localización en el espacio de trabajo,
- c) las acciones que realizan,
- d) los objetos que manipulan y
- e) sus habilidades.

Las investigaciones sobre la captura y provisión de información de conciencia de grupo se han limitado principalmente a espacios virtuales. En este ámbito, varios intentos han sido realizados por definir mecanismos de captura como el sistema de reconocimiento de caras del sistema RAMS [García et al., 2008a] [García, 2009] y por construir componentes gráficos experimentales como la herramienta Artifact-Awareness [Tee et al., 2009] que representen la información de conciencia de grupo en sistemas colaborativos.

Sin embargo, un sistema colaborativo puede proveer:

- 1) **Información excesiva:** ocasiona conflictos de privacidad, intrusión, ancho de banda y de escalabilidad.
- 2) **Información escasa:** causa problemas de contactos inapropiados o pérdida de oportunidades.

En consecuencia, existe un vacío entre la **conciencia periférica** o de **nivel macro**, que se define como el conocimiento de las actividades en un contexto global, y la **conciencia enfocada** o de **nivel micro**, que se define como el conocimiento preciso de las actividades del grupo [Vertegaal, 1997].

La Figura 1.1 muestra algunos ejemplos representativos de aplicaciones que ofrecen conciencia periférica o conciencia enfocada, así como el vacío que existe entre ambos tipos de conciencia de grupo. Primeramente, se presenta un ejemplo de una aplicación de conciencia periférica (ver lado izquierdo de la Figura 1.1), que muestra información general de los participantes en una sesión de *chat*, como a continuación se lista:

- a) nombre e imagen de cada participante;
- b) estado (activo o inactivo) de cada participante y
- c) mensajes de los participantes (i.e., la conversación).

Asimismo, se presenta un ejemplo de una aplicación de conciencia enfocada (ver lado derecho de la Figura 1.1), lo cual muestra la siguiente información:

- a) **Vista principal (*Main view*):** espacio de trabajo, donde los objetos pueden ser manipulados.
- b) **Vista miniatura (*Miniature view*):** espacio de trabajo, que representa a los objetos compartidos en forma de rectángulos coloreados.
- c) **Vista radar (*Radar view*):** espacio de trabajo, donde se muestra el telepuntero y el recuadro de la vista en curso de cada participante.
- d) **Vista WYSIWID¹:** área alrededor del cursor de otro participante.
- e) **Telepunteros (*Telepointer*):** posición del cursor de cada participante en una vista principal.
- f) **Portales (*Teleport*):** vista principal de otros participantes.
- g) **Barras multi-usuario coloreadas (*Multi-user Scrollbars*):** indican la posición relativa en el espacio de trabajo de la vista en curso de cada participante.

¹ *What You See Is What I Do (lo que tú ves, es lo que yo hago)*

La mayoría de los sistemas que ofrecen conciencia de grupo se ha limitado a la administración de información proveniente de ambientes virtuales. Por lo general, la información producida en cada sitio participante es duplicada en todos los demás sitios, con el fin de que los colaboradores puedan acceder de una sola vez a toda la información, aun si no están interesados. En consecuencia, ni el productor ni el consumidor de información de conciencia de grupo pueden decidir qué tipo de información y volumen desean respectivamente enviar y recibir.

Estos problemas han motivado el estudio de la presentación gradual de información de conciencia de recursos humanos, virtuales y físicos. El ámbito de estudio de esta tesis de maestría es el entorno de una organización como pueden ser hospitales o universidades, donde sus recursos humanos están distribuidos en diferentes lugares públicos o privados como: salas de reunión, pasillos u oficinas. Este entorno cuenta con diferentes tipos de recursos:

1. **Físicos:** impresoras, pizarrones interactivos y escáneres.
2. **Virtuales:** expedientes médicos, expedientes estudiantiles o laborales.

Cada uno de estos recursos previamente mencionados, puede ser compartidos eventualmente entre los recursos humanos de una organización.

1.2 Planteamiento del problema

El concepto de “información gradual” en espacios físicos no ha sido considerado previamente en el dominio de investigación del TCAC. El estado del arte nos muestra que el único sistema que provee información gradual de conciencia de grupo, es la herramienta colaborativa “*Community Bar*” [Romero et al., 2007]. Por medio, de esta herramienta se emite información de la presencia de usuarios en una sesión síncrona o en tiempo real. La Figura 1.2 nos muestra una ventana del sistema “*Community Bar*” que muestra automáticamente a los colaboradores instantáneas de video de cada participante; si un usuario eventualmente decide no mostrar instantáneas de video, tiene la opción de mostrar su foto, su nombre o incluso una caja coloreada indicando que está activo. La herramienta también proporciona una barra de deslizamiento (*horizontal slider*) que permite al receptor controlar el tipo de información a visualizar. La principal limitación de esta propuesta se debe a que la información emitida de manera gradual es sólo de la presencia de usuarios en espacios meramente virtuales.

Los sistemas de espacios de media (*mediaspaces*), desarrollados a principios de la década de los noventas, permitieron comprender algunos efectos tecnológicos y sociales de la provisión de conexiones de audio y video para poder visualizar y conversar con personas localizadas en otros lugares. Algunos de estos sistemas tales como *Video Walls* [Abel, 1990], *Portholes* [Dourish and Bly, 1992] y *CoMedi* [Coutaz et al., 1999] que captan áreas públicas y áreas privadas mediante cámaras. Sin embargo, las principales desventajas de estos sistemas son:

- 1) alto consumo de ancho de banda,

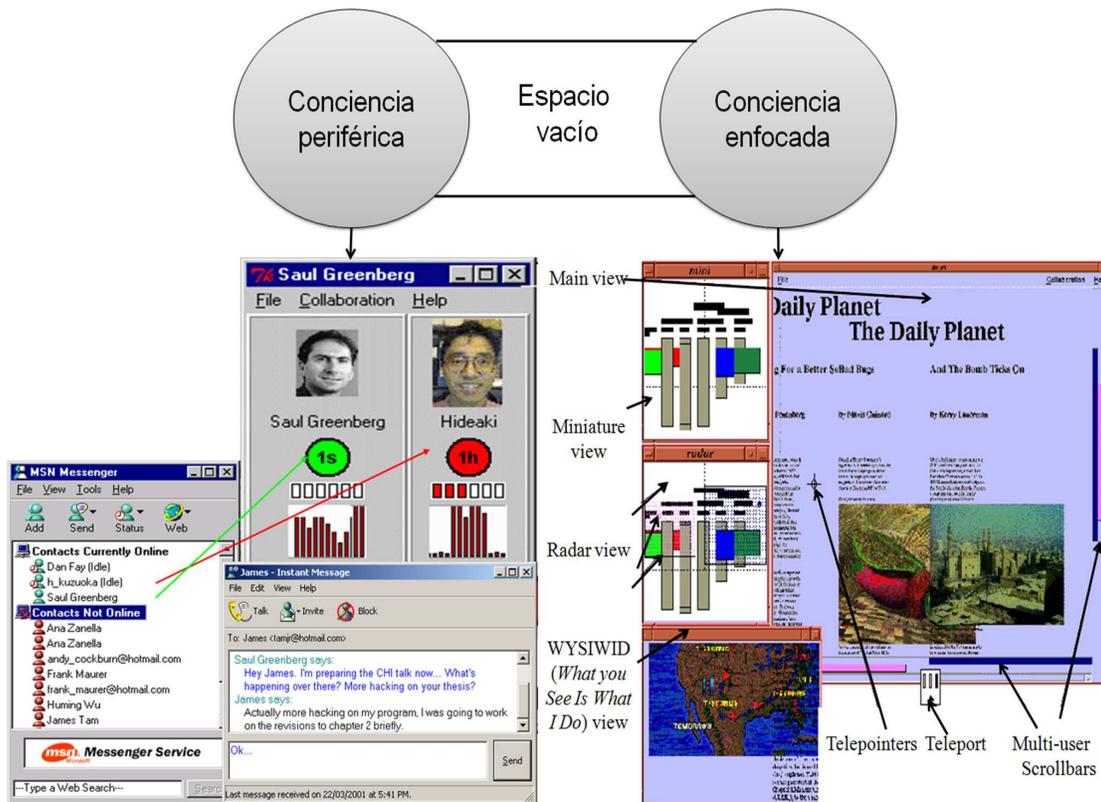


Figura 1.1: Ejemplos de sistemas colaborativos que ofrecen conciencia periférica y enfocada

- 2) problemas de privacidad al no poderse ocultar de los demás y
- 3) escalabilidad.

Algunas de las preocupaciones principales de los diseñadores de estos sistemas en aquellos tiempos eran casi exclusivamente los requerimientos excesivos de tecnología como la administración de video, de audio y de ancho de banda. Si bien, hoy en día, estos problemas de carácter tecnológico están relativamente resueltos, sin embargo **las violaciones de privacidad e intrusión** son temas de investigación abiertos. Estos problemas de carácter social se deben principalmente a que los **diseñadores de sistemas colaborativos carecen de una metodología para determinar con precisión qué información requieren las personas para facilitar el contacto entre ellas** [Greenberg and Johnson, 1997].

Estos problemas han llevado a los diseñadores de sistemas colaborativos a confiar en la perspicacia y en la creatividad. Además, el control del uso compartido de los recursos y la privacidad de la información asociada pueden cambiar, debido a que las **necesidades de un grupo puede variar con el tiempo** [Bérard and Coutaz, 1997].

Los recursos físicos y virtuales de una organización facilitan la realización de sus actividades diarias, pero **no son aprovechados de manera adecuada**. Por lo tanto,

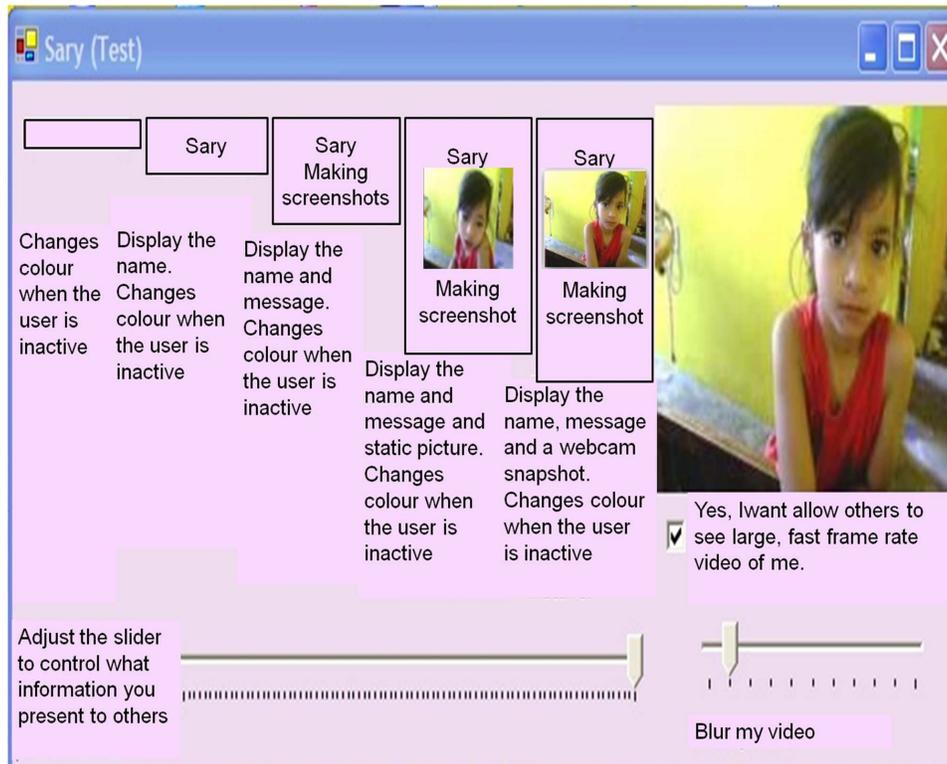


Figura 1.2: Despliegue gradual de información de conciencia de grupo en *Community Bar*

se necesita una herramienta que promueva el uso compartido de recursos y que agilice algunos procesos organizacionales. En esta tesis de maestría, se propone una herramienta que presenta de manera gradual la información de conciencia de grupo proveniente de espacios virtuales y físicos.

1.3 Objetivos del proyecto

En la sección precedente, se mencionaron algunos ejemplos de sistemas colaborativos que proporcionan información de conciencia de grupo, así como los problemas de carácter social que conlleva el diseño de estos sistemas. Además, la información de presencia, localización, disponibilidad de recursos humanos y físicos en dispositivos de cómputo y salas de reunión han sido poco consideradas en trabajos previos. Por lo tanto, se han tomado en cuenta las relaciones entre las personas y sus recursos físicos, e.g., si una persona está ausente, entonces sus recursos no están disponibles [García et al., 2008b].

Objetivo general

- Desarrollar el prototipo de un sistema de información, accesible desde PDA, que presente gradualmente la información de conciencia de grupo proveniente de entornos virtuales y físicos.

Objetivos particulares

- Diseñar e implementar los componentes gráficos (*widgets*) que muestren gradualmente la ubicación, la presencia y la disponibilidad de recursos humanos y físicos localizados en un edificio.
- Diseñar e implementar una herramienta que permita a los colaboradores controlar la difusión de su información y la de sus recursos.
- Diseñar e implementar una herramienta que permita a los colaboradores visualizar sólo la información de conciencia de grupo que desean acceder.

Mediante este sistema, los colaboradores podrán visualizar desde dispositivos de mano (PDA's) la información acerca de sus colegas y de sus recursos virtuales (e.g., multimedia) y físicos (e.g., impresoras, escáneres y pizarrones interactivos).

1.4 Organización del documento

En esta sección se describe la manera en que esta organizada la presente tesis para obtener una mejor comprensión de la misma. Una vez descrito el contexto de investigación, así como el problema que se pretende resolver y los objetivos que se persiguen en este trabajo de investigación.

La tesis está dividida en cinco capítulos como lo muestra la Figura 1.3, a continuación se describe el contenido de cada capítulo: el capítulo 2 describe el *modelo de observabilidad* para sistemas colaborativos, lo cual propone una forma de controlar la información obtenida de un usuario, y también se describe el mecanismo *Focus y Nimbus* para modelar la información de usuarios y objetos en un espacio virtual, estos mecanismos forman parte del *modelo de interacción espacial*. Después, se presentan tres sistemas de conciencia periférica y tres sistemas de conciencia enfocada, mencionando sus características, ventajas y desventajas. Seguidamente se analiza el sistema *Community Bar*, el cual implementa una herramienta para visualizar y emitir de manera gradual información de la presencia de usuarios en espacios virtuales. Finalmente se presenta una conclusión de los sistemas analizados.

El capítulo 3 presenta el diseño del sistema SIGCo (Sistema de Información Gradual de Conciencia), el cual ofrece información gradual sobre la disponibilidad, la ubicación y la presencia de los recursos humanos, físicos y virtuales de una organización. En primer lugar, se presenta un escenario que pone en evidencia la necesidad de contar con el sistema propuesto. Después, se describen las diferentes clases que forman parte del sistema. Por último, se describe la arquitectura de distribución utilizada en el desarrollo del sistema SIGCO.

En el capítulo 4 se detalla la implementación del sistema SIGCo. Primeramente, se describen los tipos de recursos que pueden haber en una organización. Después, se describen las herramientas utilizadas para la implementación de SIGCo. Consecuentemente, se presenta los algoritmos implementados en el desarrollo del sistema SIGCo. Finalmente, se presentan las pruebas y resultados del sistema SIGCo.

Finalmente, en el capítulo 5 se presenta las conclusiones, contribuciones y las perspectivas futuras de este trabajo de investigación.



Figura 1.3: Organización del documento

Capítulo 2

Estado del arte

Este capítulo inicia por la descripción de dos modelos relevantes que han sido utilizados para el diseño de sistemas colaborativos (sección 2.1). El primero es denominado modelo de observabilidad, el cual provee una metodología para publicar las actividades de cada colaborador, a la vez que se preserva su privacidad. El segundo es el modelo de interacción espacial, el cual permite representar objetos, como individuos o grupos, en un espacio compartido. Después, se describen algunos ejemplos representativos de sistemas colaborativos (sección 2.2), los cuales han sido clasificados en dos tipos: 1) sistemas de conciencia periférica o de **nivel macro**, que proveen información general del grupo y 2) sistemas de conciencia enfocada o de **nivel micro**, que ofrecen información detallada del grupo. Asimismo, se describe el sistema *Community Bar* (sección 2.3) que es el único sistema que proporciona información gradual de conciencia de grupo en ambientes virtuales. Por último, se presenta un análisis comparativo de los sistemas colaborativos estudiados en este capítulo (sección 2.4).

2.1 Modelos relevantes en sistemas colaborativos

Las investigaciones sobre el proceso de desarrollo de sistemas colaborativos (e.g., editores de documentos, editores de mapas conceptuales, espacios de media y juegos) se han incrementado en los últimos años. Particularmente, algunos modelos, e.g., *clover*, *workflow* e interacción espacial, han sido propuestos para facilitar el diseño de sistemas colaborativos. Cada uno de estos modelos es significativo, sin embargo se ha hecho poco por emplearlos para satisfacer las necesidades reales del trabajo en grupo.

En esta sección se describen dos modelos relevantes que han sido utilizados en el diseño de sistemas colaborativos. Particularmente, en la sección 2.1.1, se describe el modelo de observabilidad publicada y filtrada, el cual es importante en este trabajo de investigación porque propone una forma de controlar la difusión de la información obtenida de un colaborador. Posteriormente, en la sección 2.1.2, se describe el modelo de interacción espacial, el cual utiliza los mecanismos *Focus* y *Nimbus* para presentar información de conciencia de grupo en un espacio compartido.

2.1.1 Modelo de observabilidad publicada y filtrada

La “observabilidad” [Salber et al., 1995] es la capacidad de un sistema de hacer perceptible al grupo las variables de estado personales de cada uno de sus miembros, que resultan pertinentes para la actividad en curso. La pertinencia se estudia y se define a partir del análisis de dicha actividad.

En una actividad grupal, la observabilidad se opone al concepto de “protección del espacio privado”, e.g., el conocimiento de que una persona está leyendo su correo electrónico en su oficina. El carácter personal de una variable de estado se especifica a partir del análisis del problema. Por lo tanto, la observabilidad de las variables de estado personales de un colaborador puede ser conveniente para el grupo, pero potencialmente contraria al principio de respecto de la privacidad. También resulta importante introducir el concepto de “observabilidad publicada” [Salber et al., 1995] según el cual dichas variables se volverán observables por el grupo solamente si el propietario autoriza su publicación. La autorización de publicación puede ser estática o dinámica.

La dualidad autorización/rechazo de la publicación de una variable personal es conveniente para el carácter binario de las computadoras, pero refleja parcialmente la complejidad de las relaciones sociales. Por lo tanto, para responder a esta necesidad, se introduce el concepto de “filtro de publicación” [Salber et al., 1995]. Un filtro está asociado a una variable de estado personal publicable P , la cual está relacionada, por lo menos, con una variable de presentación R . Como se muestra en la Figura 2.1, una variable de estado personal publicable P es puesta en correspondencia con las variables de presentación R y R' , asociadas respectivamente a los usuarios k y j . La dependencia entre P y R es mantenida por dicho filtro, cuya naturaleza puede depender de P , del rol del propietario, del rol del observador, de sus relaciones socio-profesionales, etc.

Si se retoma el ejemplo del colaborador que está ocupado leyendo su correo electrónico, entonces su actividad en curso “leer correo” es una variable de estado personal. El colaborador puede publicar dicha variable para que sea visible por los miembros de su equipo, pero puede no compartirla con los demás. De esta manera, estos últimos saben que dicho colaborador está presente y no disponible, pero ignoran el detalle de sus actividades. Por lo tanto, un filtro de publicación permite que los usuarios decidan explícitamente qué información desean compartir y con quién.

2.1.2 Modelo de interacción espacial

Este modelo tiene la finalidad de obtener información del entorno que rodea a un colaborador por medio de los mecanismos *Focus* y *Nimbus*. La conciencia cuantifica el grado, la naturaleza y la calidad de la interacción entre dos objetos, cuya relación es bidireccional.

El mecanismo *Focus* determina la zona del espacio sobre la cual un objeto recibe información, en tanto que el mecanismo *Nimbus* determina la zona del espacio sobre la cual un objeto proyecta su información. Tanto *Focus* como *Nimbus* son específicos para cada medio de transmisión de información. Además, la conciencia entre objetos (e.g., personas e información) se puede cambiar mediante estos mecanismos, cuando los objetos muestran su presencia en algún área del espacio compartido. Por lo tanto, *Focus* y *Nimbus* modelan

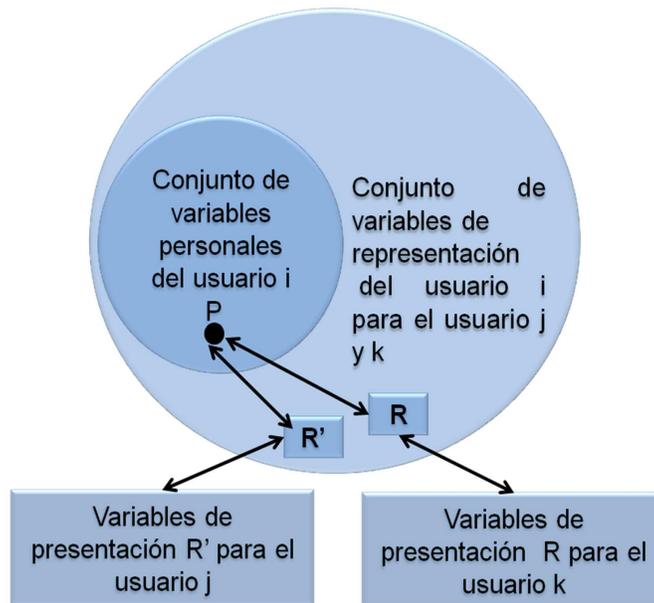


Figura 2.1: Modelo de “observabilidad” publicada y filtrada

la interacción entre objetos, de manera que se puede calcular el nivel de conciencia entre pares de objetos cuando sus respectivos *Focus* y *Nimbus* se intersectan.

Benford y Fahlén [Benford and Fahlén, 1993] definen el nivel de conciencia como “*El grado de conocimiento que adquiere un objeto A de un objeto B en un medio de transmisión de información M es alguna función del Focus de A en M en relación al Nimbus de B en M*”.

En la Figura 2.2, se muestra la intersección entre el *Focus* y el *Nimbus* de los objetos *A* y *B*, la cual representa la interacción entre ambos objetos. El área gris oscuro representa el *Nimbus* de *B* mientras que el área gris claro representa el *Focus* de *A*, lo que indica qué tanto conocimiento puede adquirir *A* de *B*. De acuerdo con este modelo de interacción, el espacio es representado como un grupo de objetos compartidos por una comunidad de usuarios. La noción de “espacio compartido” se centra en modelar la presencia de objetos, así como su proximidad.

El espacio compartido se representa por un par $SS(U, O)$, que tiene asociado un conjunto de usuarios U y un conjunto de objetos O . El conjunto de usuarios U es una colección heterogénea de objetos que pueden representar usuarios, grupos o cualquier objeto capaz de ejercer una presencia activa, e.g., un avatar (representación gráfica, generalmente humana, que se asocia a un colaborador para su identificación). Esta asociación entre usuarios y objetos en un espacio compartido da lugar a una serie de correspondencias, que ayudan a comprender las relaciones entre los usuarios que comparten un conjunto de objetos.

La presencia P de un colaborador en un espacio compartido se representa por un

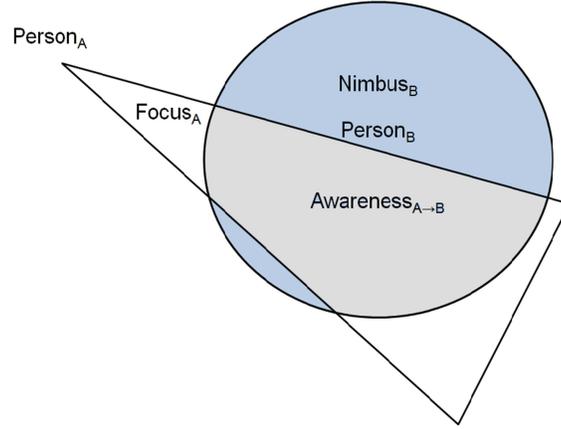


Figura 2.2: Intersección del *Focus* del usuario *A* y del *Nimbus* del usuario *B*

objeto posición *pos* y un conjunto adyacente de objetos *Adj*:

$$P(pos, Adj) \quad (2.1)$$

Donde $pos \in O$ y $Adj \in \wp O^1$.

Un objeto posición es una forma heterogénea de objeto dentro de un espacio compartido, en tanto que un conjunto adyacente es un conjunto de objetos alrededor de un objeto posición. La Figura 2.3 muestra el objeto posición *A* y su conjunto adyacente de objetos *Adj*. Esta representación de la presencia de un colaborador en un espacio compartido forma la base para describir *Focus* y *Nimbus*.

El conjunto presencia o espacio presencia representa el conjunto de todas las posiciones de presencia de un colaborador en un espacio compartido:

$$PP(O) = \{P(o, Adj) \mid o \in O \text{ and } Adj \in \wp O\} \quad (2.2)$$

Donde *P* es la presencia del usuario *o* en el espacio compartido y *Adj* es el conjunto adyacente de objetos.

A continuación, se expresa *pos* y *Adj* en función de la presencia *P*, debido a que *P* puede ser un objeto posición o un elemento del conjunto adyacente de objetos:

- $pos(P)$ es el objeto posición
- $Adj(P)$ es el conjunto adyacente

La definición matemática de *Nimbus* es la siguiente:

$$Nimbus : U \times Id \rightarrow PP(O) \quad (2.3)$$

Donde *Id* representa el conjunto *Nimbi* asociado a un usuario, el cual se refiere a diferentes instancias de la función *Nimbus*.

¹ \wp representa el conjunto potencia

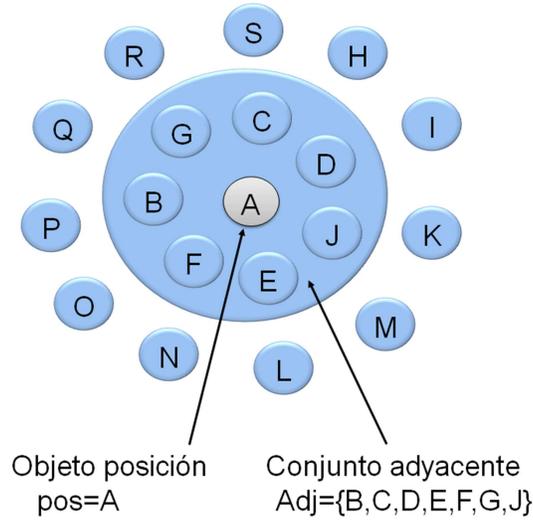


Figura 2.3: Representación del objeto posición A en un conjunto adyacente de objetos Adj

La definición matemática de *Focus* es la siguiente:

$$Focus : U \times Id \rightarrow PP(O) \quad (2.4)$$

Donde Id representa el conjunto *Foci* asociado a un usuario, el cual se refiere a diferentes instancias de la función *Focus*.

La vinculación de los usuarios en un espacio compartido se lleva a cabo mediante las funciones *Nimbus* y *Focus*, las cuales exploran los id para identificar los diferentes *Nimbi* y *Foci* asociados a cada usuario. Cabe mencionar que un usuario puede tener uno o más id asociados a diferentes instancias de las funciones *Nimbus* y *Focus*.

La Figura 2.4 muestra un ejemplo del *Nimbus* y *Focus* del usuario *Tom*, quién está representado en un espacio compartido por el objeto posición A , así como sus respectivos conjuntos adyacentes de objetos. La intersección entre *Nimbus* y *Focus* representa el nivel de conciencia que un usuario tiene de un colega. Por lo tanto, el *Nimbus* del usuario *Tom* es el objeto posición A , el cual está asociado al conjunto adyacente de objetos $\{C, G, B, F, E\}$ que representa otros *Nimbus* que no son de interés para el usuario *Tom*. Cabe mencionar que, en este caso, se escoge un conjunto arbitrario de objetos, debido a que el modelo de interacción espacial no impone ninguna restricción respecto a su composición.

Asimismo, han sido definidas tres funciones que ayudan a determinar la ubicación de un usuario en un espacio compartido:

1. La función *location* retorna un conjunto de objetos posición:

$$location : U \rightarrow \wp O \quad (2.5)$$

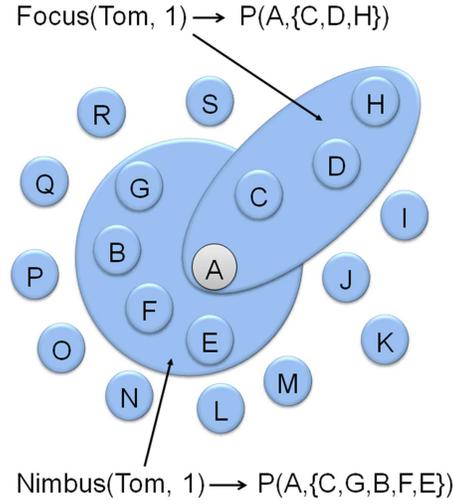


Figura 2.4: *Focus* y *Nimbus* del usuario *Tom* en un conjunto adyacente de objetos

$$location(u) = \{g \mid g \in O \text{ and } (pos(Nimbus(u, id)) = g \text{ or } pos(Focus(u, id)) = g)\}$$

Donde $u \in U$ e $id \in Id$.

2. La función *Agg_Nimbus* devuelve todos los objetos que están dentro de los *Nimbus* de los usuarios:

$$Agg_Nimbus : U \rightarrow \wp O \quad (2.6)$$

$$\{Agg_Nimbus(u) = \{g \mid g \in O \text{ and } (g \in Adj(Nimbus(u, id)))\}$$

3. La función *Agg_Focus* devuelve todos los objetos que están dentro de los *Focus* de los usuarios:

$$Agg_Focus : U \rightarrow \wp O \quad (2.7)$$

$$\{Agg_Focus(u) = \{g \mid g \in O \text{ and } (g \in Adj(Focus(u, id)))\}$$

La Figura 2.5 muestra un ejemplo de dos *Focus* y dos *Nimbus* del usuario *Tom*, que ejemplifican las funciones previamente mencionadas.

La función *location* (expresión 2.5) retorna el conjunto de objetos posición del usuario *Tom*:

$$location(Tom) \longrightarrow \{B, A, K\}$$

La función *Agg_Nimbus* (expresión 2.6) devuelve el conjunto de todos los *Nimbus* del usuario *Tom*:

$$Agg_Nimbus(Tom) \longrightarrow \{C, G, B, F, E, J, I\}$$

La función *Agg_Focus* (expresión 2.7) regresa el conjunto de todos los *Focus* del usuario *Tom*:

$$Agg_Focus(Tom) \longrightarrow \{C, D, H, F, N\}$$

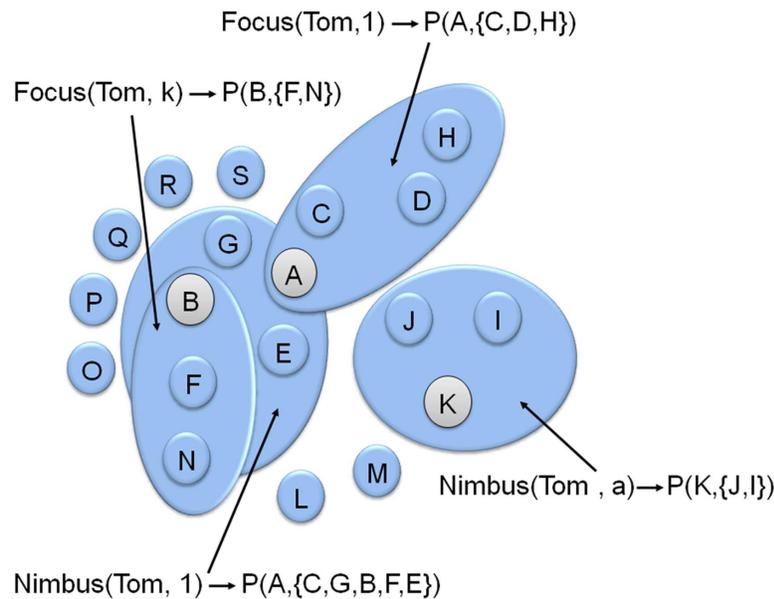


Figura 2.5: Dos Focus y dos Nimbus del usuario Tom

Finalmente, cabe mencionar que la conciencia puede ser modelada como una función discreta o continua, dependiendo de la naturaleza de la aplicación.

Estructuras de gráficos

En esta subsección, se presentan algunas estructuras de gráficos que representan el modelo de interacción espacial. Estas estructuras han permitido modelar una gama de aplicaciones cooperativas, e.g., aplicaciones de hipertexto tales como SEPIA [Haake and Wilson, 1992] y EGRET [Johnson, 1992] y sistemas de flujos de trabajos (*Workflow*) como Milano [De Michelis and Grasso, 1994].

Un grafo G es una estructura discreta, que consiste de un conjunto V de vértices unidos por medio del conjunto E de aristas. En la sección 2.1.2, se mencionó que el conjunto adyacente de objetos se tomaba de manera arbitraria, sin embargo las estructuras de gráficos permiten encontrar dicho conjunto por medio de una función de incidencia i . Por lo tanto, un grafo consiste de una tupla $G(V, E, i)$.

La función de incidencia i puede explorar un grafo, relacionando cada arista del conjunto E con dos vertices del conjunto V :

$$i : E \rightarrow \{\{u, v\} \mid u, v \in V\} \quad (2.8)$$

Donde cada arista del conjunto E está asociada a dos vértices del conjunto V .

En la Figura 2.6, se presenta un ejemplo de un gráfico con cinco vértices y cinco aristas.

$V(G)$ es el conjunto de vértices del grafo G de la Figura 2.6:

$$V(G) = \{v1, v2, v3, v4, v5\}$$

$E(G)$ es el conjunto de aristas del grafo G de la Figura 2.6:

$$E(G) = \{a, b, c, d, e\}$$

La función de incidencia (expresión 2.8) para cada arista del conjunto E , correspondiente al ejemplo ilustrado en la Figura 2.6, es la siguiente:

$$i(a) \mapsto \{v1, v2\}, i(b) \mapsto \{v2, v3\}, i(c) \mapsto \{v2, v5\}, i(d) \mapsto \{v5, v4\}, i(e) \mapsto \{v1, v4\}$$

Donde \mapsto representa la asociación de una arista con un par de vértices.

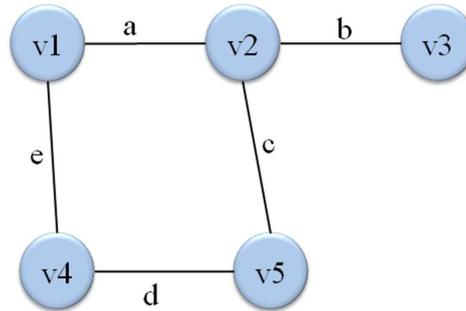


Figura 2.6: Representación de una estructura de grafo

En el modelo de grafos, no se indica cómo se construye el conjunto adyacente. Por lo tanto, permite que los conjuntos sean conformados a partir de conjuntos arbitrarios de vértices. Esta libertad permite considerar selecciones no continuas de vértices, mediante criterios sensibles al contexto, e.g., consultas a bases de datos.

A continuación, se define la función *linked*, la cual prueba si el conjunto adyacente de objetos está vinculado a una posición de presencia:

$$linked : PP(G) \mapsto Boolean \quad (2.9)$$

Donde PP es el conjunto de todas las posiciones de presencia del grafo G .

La función *linked* puede ser utilizada para especificar de manera particular el *Focus* o el *Nimbus* de un usuario, el cual está formado por los vértices que están directamente ligados a una ubicación en el espacio compartido.

En la Figura 2.7, se muestra un ejemplo de la función *linked* con un usuario $u1$, el cual tiene un $id=1$. Esta función prueba el conjunto adyacente de objetos $\{v2, v5, v6, v7\}$ para el usuario $u1$, con la finalidad de indicar que el vértice $v0$ tiene asociado un conjunto adyacente de objetos. El *Focus* del usuario $u1$ está formado directamente por los vértices vinculados al vértice $v0$, el cual representa al usuario en el espacio compartido. Por otra parte, el conjunto adyacente de objetos está constituido por todos los vértices

vinculados con el vértice $v0$. Por lo tanto, se puede especificar el *Focus* del usuario $u1$ (expresión 2.4) mediante la función *linked*, de la siguiente manera:

$$Focus(u1, 1) \longrightarrow P(v0, \{v2, v5, v6, v7\})$$

La forma general de conocer el estado de todos los *Foci*, asociados con el usuario $u1$, está expresada de la siguiente manera:

$$\forall id \in Id \text{ linked}(Focus(u1, id))$$

La expresión anterior indica que el estado de todos los *Foci* está formado por todos los vértices que están asociados al vértice $v0$.

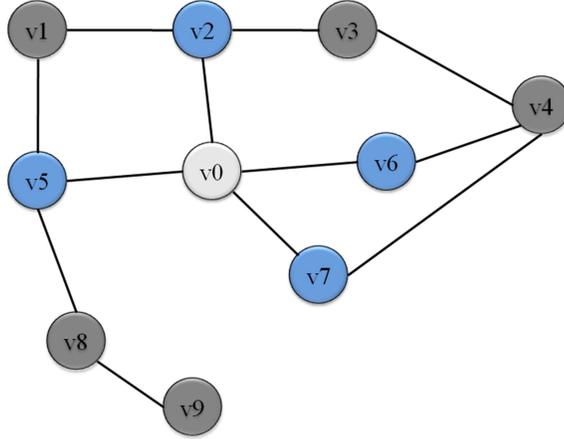


Figura 2.7: *Focus* de $v0$ en una estructura de grafo

La función *sphere* prueba si la distancia del *Focus* o del *Nimbus* de un usuario es verdadera, dado un valor de distancia. La distancia representa el número de vértices asociados al *Focus* o al *Nimbus*. La definición de la función *sphere* es la siguiente:

$$sphere : PP(G) \times \mathbb{N} \mid \rightarrow Boolean$$

$$sphere(pp, id) = \forall v \in Adj(pp) \text{ and } in_distance(location(pp), v, id) \quad (2.10)$$

$$in_distance(v, g, id) = len(path(v, g)) \leq Id \quad (2.11)$$

Donde \mathbb{N} es el conjunto de números naturales, pp es una **posición de presencia** en un grafo o espacio compartido, $in_distance(v, g, id)$ es una función que obtiene la distancia dados los siguientes parámetros: 1) la ubicación de un objeto $location(pp)$, 2) una arista del conjunto adyacente de objetos y 3) un id .

Por ejemplo, si se quiere probar la distancia del vértice $v0$ (i.e., número de vértices asociados a $v0$) dado un rango de cuatro, esta se puede calcular mediante la función *sphere*

recorriendo cada par de vértices asociados a $v0$. El recorrido se lleva a cabo mediante la función $in_distance(v, g)$ (expresión 2.11), la cual hace uso de la función $len()$ para realizar el conteo de la distancia del grafo. La Figura 2.7 ilustra un gráfo que muestra el *Focus* del usuario $u1$ (representado por el vértice $v0$) con distancia de cuatro. Por lo tanto, la función *sphere* de este ejemplo es la siguiente:

$$\forall id \in Id \text{ sphere}(Focus(u1, id), 4) \quad (2.12)$$

Las estructuras de grafos, mencionadas previamente, permiten modelar la conciencia de grupo (*Focus* y *Nimbus*) de diversas aplicaciones colaborativas.

2.2 Sistemas conciencia periférica y enfocada

En esta sección, se describen algunos sistemas colaborativos relevantes que han sido propuestos en el dominio de investigación del Trabajo Colaborativo Asistido por Computadora (TCAC). Estos sistemas ofrecen algún tipo de conciencia, por lo tanto han sido clasificados en TCAC como: *Sistemas de conciencia periférica* y *Sistemas de conciencia enfocada*. En la sección 2.2.1, se describen los sistemas de conciencia periférica Doc2U, Peeholes, JamSpace y mensajeros instantáneos, los cuales ofrecen información general de los miembros del grupo. Posteriormente, en la sección 2.2.2, se analizan los sistemas de conciencia enfocada Calliope, CoWord, GrewpEdit y Concept Mapper, los cuales proveen información específica de los miembros del grupo mientras interactúan en un espacio compartido.

2.2.1 Sistemas de conciencia periférica

Estos sistemas se centran en un solo objeto en un espacio virtual. Este objeto podría contener suficiente información de conciencia para algún colaborador y para otro colaborador tal vez no tendría suficiente información para tomar alguna decisión. Por ejemplo, si se trata de un documento compartido, donde el objeto sea el nombre del documento. Para un colaborador el conocer solo el nombre significaría poder tomar la decisión de colaborar en la edición de un documento. Sin embargo para otros colaboradores el nombre del documento no significaría nada. Por lo tanto, la información de conciencia de grupo que adquiere un colaborador de un objeto depende de su capacidad innata.

A continuación, se describen algunos sistemas que presentan esta granularidad de información de conciencia de grupo.

Doc2U

Es un sistema que provee, por medio de la notificación de eventos, conciencia de la presencia de recursos almacenados en la Web [Moran et al., 2004]. Doc2U utiliza el concepto de presencia de documentos como una analogía de la presencia de usuarios en los mensajeros instantáneos o páginas Web.

Tabla 2.1: Analogía entre los conceptos de conciencia de presencia de usuarios en mensajeros instantáneos y conciencia de presencia de documentos en Doc2U

Criterios	Información de usuarios	Información de documentos
Presencia y disponibilidad	Registrado No registrado Disponible (en línea) Indisponible (fuera de línea) Normal en línea Disponible para <i>chat</i> Ausente Ausente indeterminado No molestar	Registrado No registrado Disponible (en línea) Indisponible (fuera de línea) Vacante (acceso libre) Parcialmente disponible Bloqueado o desbloqueado Reservado y bloqueado (o desbloqueado)
Actividad e interacción del usuario	Leyendo Inscribiéndose Escribiendo Solicita suscripción Envía mensaje a usuarios Inicia la aplicación para comunicarse con otros usuarios (<i>chat</i> o conferecia de audio/video)	Siendo leído Siendo anotado Siendo escrito Solicita suscripción Envía comandos a documentos Inicia la aplicación para tratar documentos (editar, anotar o buscar)

La tabla 2.1 muestra una comparación de los conceptos de conciencia de presencia de documentos y conciencia de presencia de usuarios. Esta comparación se realiza en términos de la presencia, disponibilidad, actividad e interacción entre usuarios y objetos. Por ejemplo, los mensajeros instantáneos consideran los estados de la presencia de usuarios, mediante los términos **en línea** o **fuera de línea**, pero en el caso de Doc2U, los estados de la presencia de documentos está definida como **bloqueado** o **desbloqueado**. La presencia de conciencia de documentos se consideran importantes en la administración de sesiones para la provisión de conciencia de grupo.

La Figura 2.8 muestra la ventana del sistema Doc2U, donde se aprecia la información de conciencia de grupo que un colaborador puede adquirir. En este ejemplo, el colaborador Bill carga un documento para que su asesor, el colaborador Hellen, lo revise. Cuando Hellen inicia la edición del documento, el sistema bloquea el acceso para que nadie pueda utilizarlo, pero Bill tiene la posibilidad de unirse a Hellen para hacer modificaciones de acuerdo a lo que ella le indique. Por lo tanto, el documento puede pasar del estado **bloqueado** al estado **desbloqueado** o Bill puede esperar a que su asesor termine de revisar el documento. Además, otros colaboradores (e.g., revisores) pueden cargar este documento en su espacio privado para hacer anotaciones, de tal manera que pueden realizar notificaciones de nuevas versiones del documento al colaborador Bill. Estas notificaciones

son efectuadas a través de mensajes instantáneos (ver Figura 2.8).

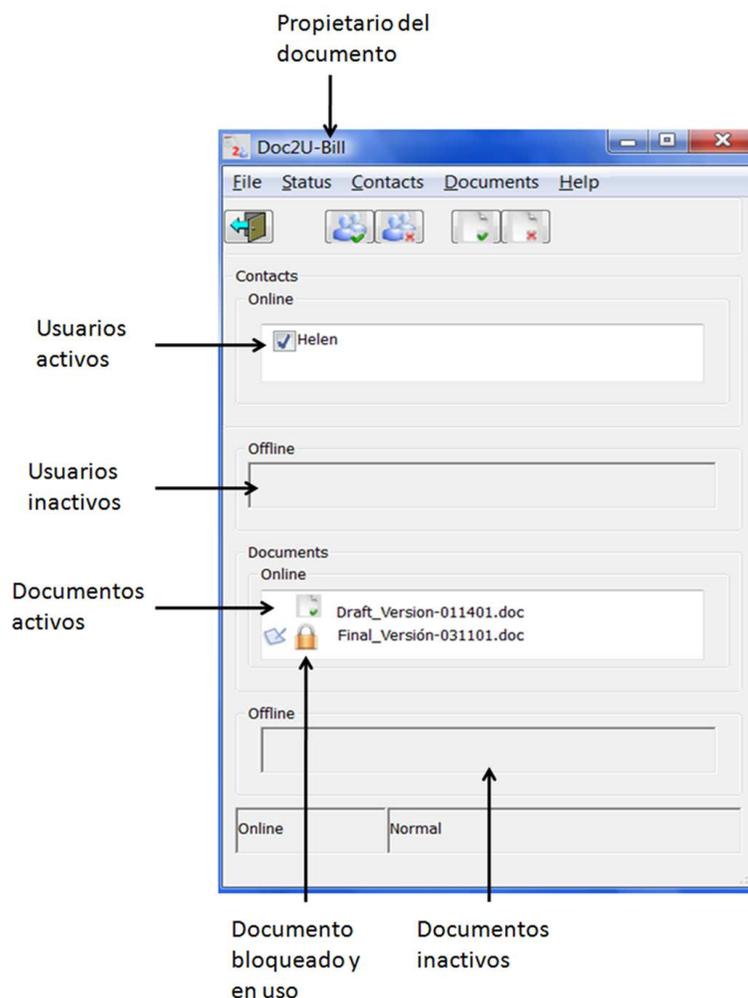


Figura 2.8: Información de conciencia de grupo proporcionada por el sistema Doc2U

Doc2U provee información de conciencia de usuarios y documentos mediante íconos. Actualmente Doc2U está implementado como un servicio extendido de un servidor Web, el cual provee notificación de eventos. Este sistema forma parte del modelo de administración de entidades compartidas de la plataforma de PIÑAS [Decouchant et al., 2001] [Decouchant et al., 2010].

Peepholes

Es un sistema de agenda electrónica que permite crear comunidades virtuales. La Figura 2.9 muestra una comunidad virtual con cinco colaboradores, cuya información general (e.g., nombre, teléfono y oficina) se toma de sus respectivas agendas electrónicas [Greenberg, 1996]. Peepholes actualiza automáticamente la agenda electrónica de cada

colaborador que conforma la comunidad virtual. Por lo tanto, revisa continuamente la llegada de correos electrónicos, para tomar los nombres y direcciones de correos electrónicos y agregarlos a la lista de direcciones.

Este sistema provee información de conciencia de grupo, a través de íconos que cambian de color y de forma cuando un colaborador pasa de un estado a otro. Cabe mencionar que cada colaborador está representado por un ícono que porta su nombre, así como el estado en el que se encuentra (e.g., en línea, fuera de línea, inactivo, ausente y desconectado). Por ejemplo, cuando un colaborador pasa del estado en línea al estado fuera de línea, Peepholes coloca un ícono de color gris. Por el contrario si pasa del estado indisponible al estado disponible, el sistema cambia el ícono de color gris a color negro y además envía un sonido para notificar la presencia del colaborador a sus colegas del sistema.

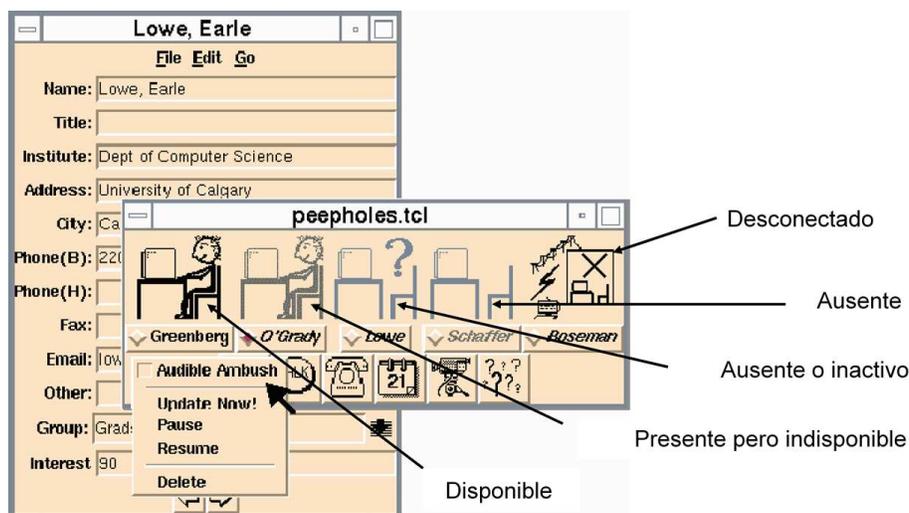


Figura 2.9: Interfaz de usuario del sistema Peepholes

Peepholes tiene las siguientes características:

- **Indicadores visuales de actividad:** son utilizados para mostrar el estado de actividad de cada miembro de una comunidad virtual. La Figura 2.9 muestra que el colaborador Greenberg está **en línea** (ícono de color negro con un personaje sentado en una silla); O'Grady ha estado **fuera de línea** durante unos minutos (ícono de color gris con un personaje sentado en una silla); Lowe está conectado pero no ha utilizado su computadora durante un tiempo (ícono de color gris con signo de interrogación sobre una silla vacía); Schaffer acaba de cerrar su sesión (ícono de color gris con una silla vacía), y Roseman es **inalcanzable** (ícono de color negro con tache).
- **Indicadores sonoros de actividad:** cuando un colaborador quiere pasar del estado **fuera de línea** al estado **en línea**, el sistema Peepholes envía un sonido a los contactos remotos para que sepan que el colaborador nuevamente regresó y

está activo. Este sonido atrae la atención de los colaboradores, quienes pueden en cualquier momento iniciar una conversación con dicho colaborador si lo desean.

- **Información gratuita:** la información utilizada en Peepholes se encuentra de manera libre en la red y es obtenida mediante consultas continuas al proceso *ruser daemons*, el cual está disponible en la mayoría de los servidores Unix y permite acceder de forma razonable a la información de muchos usuarios que hacen uso de servidores Unix.

La desventaja de Peepholes es que no provee ninguna diferencia entre un colaborador ausente y uno activo (ver Figura 2.9).

JamSpace

Es un sistema formado por *hardware* y *software* que permite a los usuarios componer música [Gurevich, 2006]. JamSpace tiene un *hardware* que consiste en una superficie plana con doce teclas sensibles a la presión. Una tecla puede ser presionada con la mano para provocar una nota con intensidad proporcional a la presión que se ejerce en dicha tecla. Las teclas son asignadas a los instrumentos de percusión o a las notas de la escala musical, dependiendo del instrumento elegido por el colaborador. El *software* consiste en una aplicación cliente/servidor, cuya interfaz gráfica de usuario consta de cinco componentes:

1. un marcado de pista para el colaborador local,
2. un conjunto de pistas JamSpace,
3. una interfaz de comunicación para establecer las conexiones con el servidor,
4. un metrónomo (i.e., compás de las composiciones musicales) o un *tempo* (i.e., velocidad con que debe ejecutarse una pieza de música) y
5. una matriz para administrar las sesiones en tiempo real de otros clientes (*jams*).

Las actividades que un colaborador puede realizar en cualquier momento son las siguientes:

- grabar y reproducir de manera privada un fragmento de una pista,
- marcar los ciclos de mezclas de una pista o personalizar una pista conjuntamente con otros colaboradores que están presentes en el espacio compartido,
- enviar un marcado de pista a todos los colaboradores, en el caso de que estos estén escuchando la pista en tiempo real, para compartir una mezcla de música, y
- elegir uno o varios íconos para representar una lista predefinida de marcados de pistas en su espacio compartido.

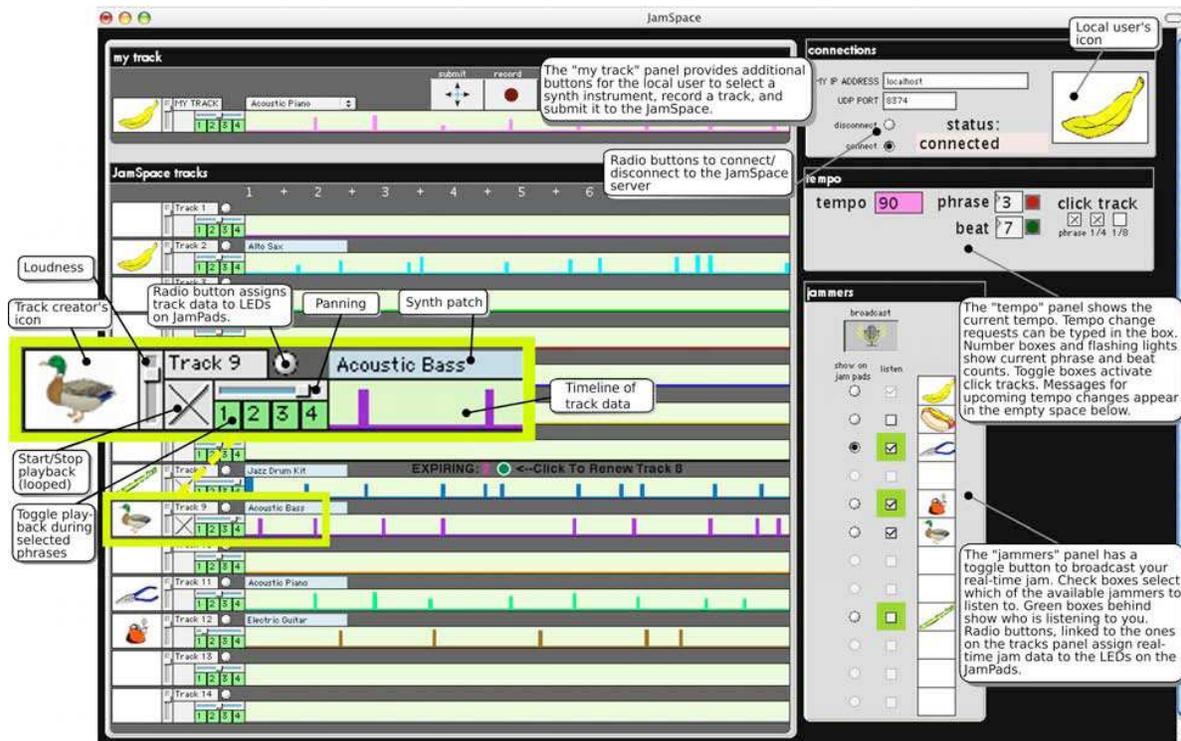


Figura 2.10: Interfaz de usuario cliente JamSpace

Estas actividades permiten los colaboradores desarrollar pruebas, estilos y gustos sobre una música, al tiempo que se protege su privacidad y anonimato ya que se desconocen sus nombres reales.

JamSpace provee la siguiente información de conciencia (ver Figura 2.10):

- **Matriz de conexión:** provee un conjunto de marcados de pistas, en el que aparecen los diferentes colaboradores que están realizando el marcado de un fragmento de pista (ver parte inferior derecha de la Figura 2.10);
- **Lista de colaboradores:** provee la lista de alias de los colaboradores que están escuchando una pista o presionando las teclas (ver parte inferior derecha de la Figura 2.10);
- **Representación del colaborador:** provee al colaborador diferentes íconos para representar su presencia ante los demás colaboradores;
- **Espacios que posee un colaborador:** provee espacios personales, públicos, compartidos y privados:
 - a) el **espacio personal** permite escuchar una pista y crear una mezcla de pistas, así como escuchar otras pistas que se están reproduciendo en tiempo real;
 - b) el **espacio público** está representado por los diferentes marcados de pistas realizados por los colaboradores remotos;

- c) el **espacio compartido** permite enviar pistas a los colaboradores remotos y posteriormente saber quién las está escuchando y
- d) el **espacio privado** está destinado a un colaborador que no está conectado al servidor JamSpace (estado de desconexión). En este modo, el colaborador puede reproducir, grabar y escuchar una pista en su propio JamPads (ver lado derecho de la Figura 2.10). La identidad y la presencia del colaborador y son invisibles para los demás y él tiene la seguridad de que su entorno no puede ser invadido.

JamSpace tiene las siguientes desventajas:

1. no soporta explícitamente la interacción directa,
2. no provee información específica de los colaboradores (e.g., nombre de la persona que realiza una mezcla),
3. no muestra el título de las pistas que están siendo escuchadas,
4. el número de colaboradores está limitado a catorce y
5. no provee ninguna vista WYSIWIS² ni WYSIWID³.

Mensajeros instantáneos

Los mensajeros clásicos, como *Instant Messenger*, permiten iniciar una conversación entre dos o más colaboradores a mediante cadenas de textos. En general, estos sistemas muestran información global de los colaboradores, e.g., la presencia (mediante nombres) de quienes están participando en la conversación. Algunos mensajeros instantáneos tienen la opción de mostrar la foto de los colaboradores, así como los colaboradores activos e inactivos mediante íconos de estado (ver Figura 2.11).

2.2.2 Sistemas de conciencia enfocada

Estos tipos de sistemas ofrecen un conocimiento preciso de las actividades del grupo para permitirles coordinar sus esfuerzos con otros en aras de lograr un objetivo común. Asimismo, estos sistemas suelen ofrecer una vista común llamada WYSIWIS² sobre las ubicaciones y acciones de los colaboradores. Además, algunos sistemas proveen vistas WYSIWID³ que dan información precisa de los objetos que están manipulando los colaboradores.

A continuación, se describen algunos sistemas que ofrecen este tipo de información de conciencia de grupo.

² *What You See Is What I See: lo que tú ves es lo que yo veo*

³ *What You See Is What I DO: lo que tú ves es lo que yo hago*



Figura 2.11: Interfaz de usuario de *Instant Messenger*

Calliope

Es un editor compartido que provee: 1) un espacio de texto principal, el cual permite que múltiples colaboradores puedan trabajar sobre un mismo documento, 2) un conjunto de herramientas para proveer conciencia de grupo y 3) una variedad de espacios de texto alternativos para proporcionar a los colaboradores un medio de comunicación que reduzca los costos de conexión [Mitchell, 1996].

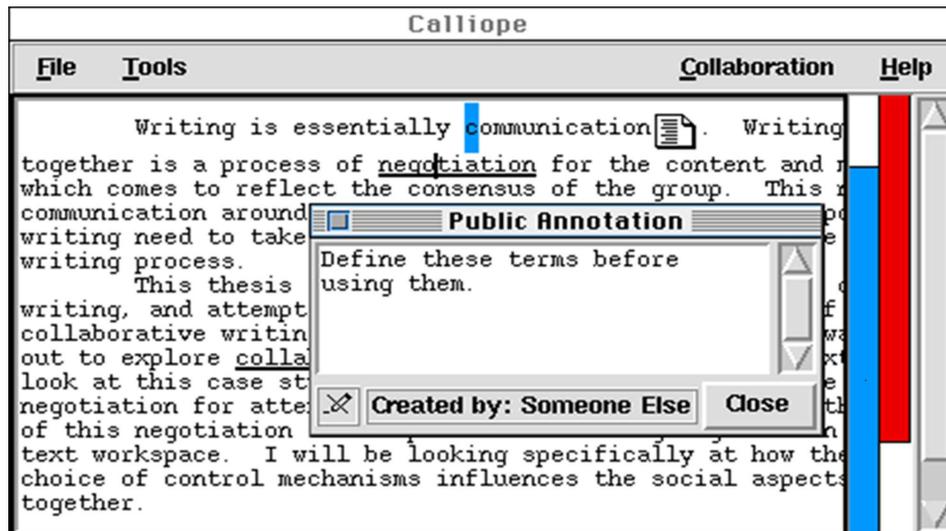


Figura 2.12: Anotaciones públicas en Calliope

Calliope soporta el uso de comunicación anotativa en la forma de ventanas de texto, que pueden ser adheridas a secciones arbitrarias del texto principal y visualizadas inmediatamente por todos los colaboradores. Las anotaciones públicas pueden ser modificadas por todos los colaboradores, pero sólo por uno a la vez. Sin embargo, para evitar el costo potencial de visualizar el texto de manera inmediata, las anotaciones pueden ser creadas en la ventana privada de texto y después adheridas al texto principal (ver Figura 2.12).

Calliope, provee un número de espacios de texto alternativos tales como una ventana privada de texto, un cuaderno de notas no estructurado, así como anotaciones públicas y externas, que se diseñaron para proporcionar una vía de comunicación entre los colaboradores. Particularmente, este sistema permite a un colaborador introducir texto en una ventana privada, con el fin de formar y pulir sus ideas antes de exponerlas a sus colegas. Asimismo, Calliope permite mover el texto seleccionado en el espacio de texto principal hacia una ventana privada de texto y viceversa, mediante un *click* del *mouse*. Una ventana privada, en combinación con el mecanismo de candado a nivel de selección, permite a un colaborador trabajar de manera exclusiva y privada en una sección de texto y mostrarla a sus colegas cuando lo desee. El espacio de texto privado toma la forma de una ventana separada para enfatizar que este espacio es diferente del espacio de texto principal (ver Figura 2.13).

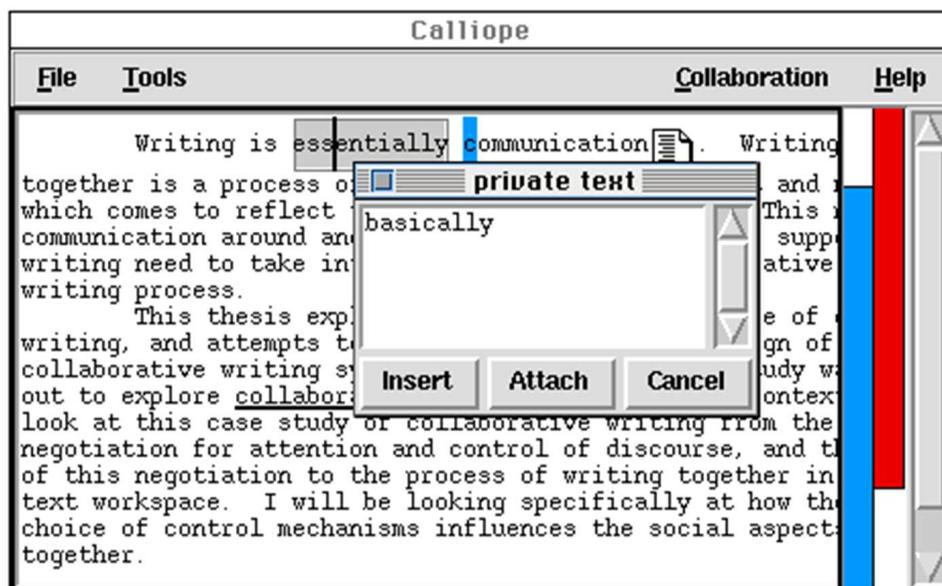


Figura 2.13: Ventana privada de texto en Calliope

Calliope provee un cuaderno de notas, en forma de una ventana compartida no estructurada, que permite introducir y manipular libremente unidades de texto. También, permite hacer señalamiento de texto con movimientos de telepuntero y marcado de texto con anotaciones a mano alzada. Las unidades de texto son visibles de manera inmediata y pueden ser modificadas por cualquiera. Para prevenir problemas de consistencia, Calliope provee candados, a nivel de unidades de texto, indicados por medio de selecciones con

color. El texto puede arrastrarse, cortarse y pegarse entre el cuaderno de notas y el espacio de texto principal (ver Figura 2.14).



Figura 2.14: Cuaderno de notas no estructurado en Calliope

Los objetivos principales del diseño de cuaderno de notas son, por un lado, reducir el costo de introducción de ideas por parte de los miembros no dominantes del grupo y, por otro lado, estimular la discusión de ideas, antes de finalizar un párrafo en el espacio de texto principal. Las ideas plasmadas por los diferentes colaboradores tienen el mismo rango, puesto que el cuaderno de notas no tiene una estructura predeterminada.

El cuaderno de notas toma la forma de una ventana separada para lograr dichos objetivos. Dado que es un espacio de texto menos formal que el espacio de texto principal, resulta más fácil para los colaboradores introducir ideas alternativas y jugar con diferentes enfoques cuando escriben texto. No obstante, también introduce el problema de relación entre el espacio de texto principal y las ideas experimentales introducidas en el cuaderno de notas.

Calliope provee la siguiente información de conciencia:

- **Código de colores:** están asociados a los diferentes colaboradores remotos que trabajan en el documento y se utilizan de manera consistente para facilitar la identificación de sus contribuciones en un documento.
- **Barras de desplazamiento (*scrollbar*):** permiten ubicar a los colaboradores remotos para saber en qué parte del documento están trabajando (vistas de objetos que manipulan). Estas barras están colocadas paralelamente a la barra de desplazamiento regular y están indentificadas con color para facilitar a los colaboradores la ubicación relativa de sus colegas en el documento (ver Figura 2.12).
- **Telepunteros:** permiten ubicar a los objetos en el espacio de texto principal y en el cuaderno de notas mediante movimientos de su cursor. Además, estimulan el

sentido de un contexto compartido porque cada colaborador puede visualizar las acciones de sus colegas. Al igual que las barras de desplazamiento compartidas, los telepunteros están indicados con color para permitir a los colaboradores identificar quién ejecuta una acción.

- **Vista panorámica con lupa:** muestra las selecciones de texto y las ubicaciones de todos los colaboradores en el documento, manteniendo sus identidades por medio del color. Además, las selecciones de texto tienen una granularidad variable, que puede fluctuar desde un carácter hasta el documento completo. Por estas razones, a través de una vista panorámica con lupa, se pueden identificar a aquellos colaboradores que trabajan conjuntamente en una parte determinada del documento. Las vistas panorámicas combinan las siguientes ideas:
 1. **vista *gestalt*:** provee un sentido de la estructura y del contenido del documento completo. Asimismo, permite a un colaborador saber cómo está cambiando el texto y dónde están ubicados sus colegas en relación a cada uno y al documento.
 2. **ventana de punto de vista:** permite a un colaborador saber lo que uno de sus colegas está haciendo, con quién está trabajando y cómo está modificando el documento.
- **Acciones y cambios:** mantiene un registro histórico de cambios en el documento. A medida que los colaboradores introducen texto, la información sobre quién lo introdujo es almacenada en un formato de texto con color. Este formato puede ser desplegado o escondido por cada colaborador en su vista local del documento, proporcionando pistas sobre quién ha contribuido en ciertas partes del documento. Además, Calliope permite a un colaborador “interrogar” a un fragmento de texto para obtener información sobre quién lo escribió.
- **Vistas WYSIWIS:** proveen un marco de referencia común; un colaborador puede crear una vista WYSIWIS, utilizando una de las barras de desplazamiento compartidas para alinear su vista principal con la de uno de sus colegas. Una vez que sus vistas están conectadas, cualquier desplazamiento efectuado por el colaborador remoto se refleja en la vista del colaborador local. El colaborador local es libre de desconectar las vistas en cualquier momento para desplazarse a otra parte del documento. La vista del colaborador remoto no es afectada por la conexión.

CoWord

Este sistema es una extensión de Microsoft Word, la cual convierte al procesador Word (aplicación mono-usuario) en un editor colaborativo que soporta múltiples colaboradores, múltiples vistas y la edición concurrente de cualquier objeto de un documento por medio de Internet [Xia et al., 2004].

Los objetivos de CoWord son los siguientes:

- **Compatibilidad de aplicaciones:** conservar la interfaz de usuario y las funcionalidades que proporciona Word, e.g., el formato del documento.
- **Colaboración sin restricciones:** permitir que múltiples colaboradores editen y deshagan los cambios realizados en un documento.

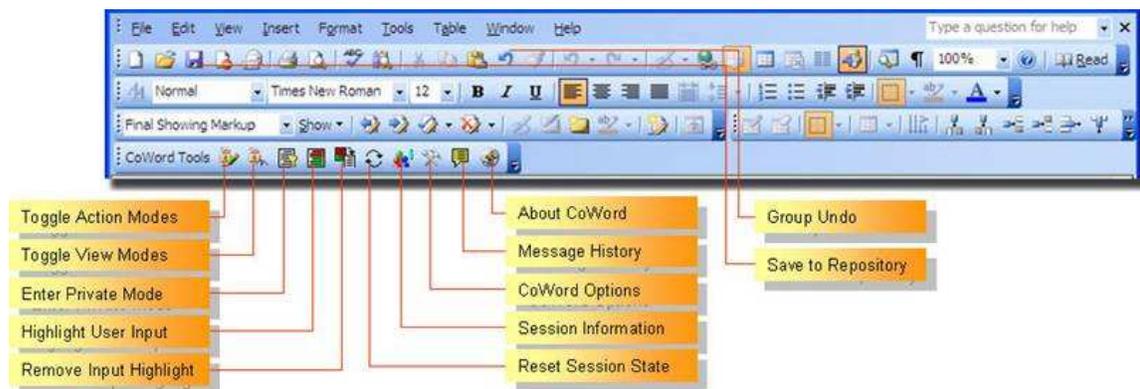


Figura 2.15: Barra de heramientas de CoWord

La compatibilidad de aplicaciones asegura que CoWord herede la funcionalidad convencional de Word (funcionalidad mono-usuario), ya que CoWord puede trabajar de manera colaborativa y mono-usuario. CoWord permite la colaboración sin restricción, i.e., no especifica estilos de colaboración ni estructuras impuestas a los usuarios, lo cual les da una completa libertad en el uso de CoWord. La Figura 2.15 muestra los componentes de CoWord anexados a Word.

Caraterísticas de CoWord:

- **Modo de colaboración asimétrico:**
 - **modo público:** las operaciones de un colaborador local son propagadas a todos los colaboradores remotos (modo estándar);
 - **modo privado:** las operaciones son ejecutadas localmente y no son propagadas a los sitios remotos, pero los colaboradores privados pueden recibir operaciones públicas. Este modo es opcional.
- **Control de interacción flexible:**
 - **modo de actores múltiples:** varios colaboradores pueden editar un documento al mismo tiempo (modo estándar);
 - **modo de actor-simple:** solo un colaborador puede editar un documento en un momento dado (los colaboradores deben turnarse para actuar);
 - **modo de vistas-múltiples:** varios colaboradores pueden ver: 1) diferentes partes del documento, 2) un documento en formatos diferentes (e.g., página Web o documento XML Word) y 3) los modos de interfaz de usuario (e.g., vista en miniatura, vista en esquema o en diseño Web) que utilizan sus colegas;

- **modo de vista-simple:** todos los colaboradores pueden ver la misma porción del documento en el mismo formato y en el mismo modo de interfaz del colaborador.

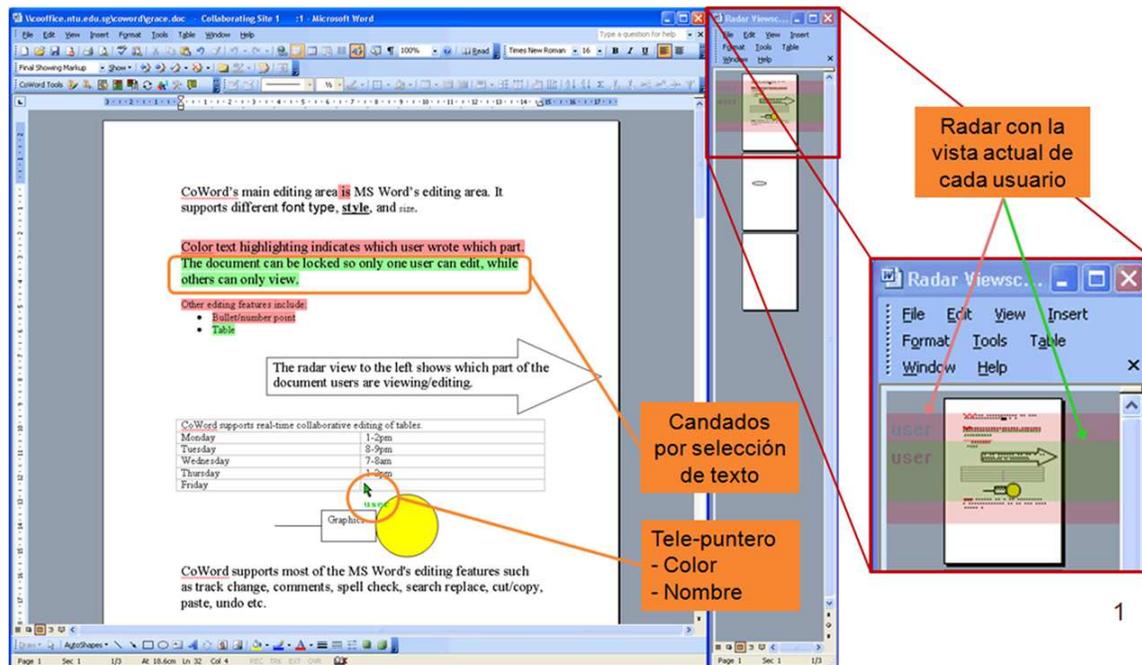


Figura 2.16: Información de conciencia de grupo en CoWord

CoWord provee la siguiente información de conciencia de grupo (ver Figura 2.16):

- Identidades de los colaboradores:** los nombres de los colegas se conocen mediante un telepuntero el cual tiene asociado un color diferente para cada colaborador (ver Figura ??).
- Telepunteros (*telepointers*):** indican las posiciones de los punteros de los colaboradores remotos (ver Figura ??).
- Vista radar (*radar view*):** muestran las ubicaciones de las vistas en curso de los colaboradores remotos (ver Figura ??).
- Notificación de operaciones remotas:**
 - **notificación de operaciones sonora:** el sistema envía diferentes sonidos para indicar la ejecución de diferentes tipos de operaciones remotas;
 - **notificación visual:** el sistema muestra una imagen que parpadea para indicar la región del documento y el tipo de operación que un colaborador remoto está realizando.

- d) **Entradas de texto coloreado:** los colaboradores remotos insertan texto, el cual es resaltado por el sistema mediante diferentes colores asociados a dichos colaboradores. El texto coloreado se puede guardar y cancelar en diferentes sesiones. Esta característica puede ser considerada como una versión ligera de la inserción de textos en tiempo real.
- e) **Seguimiento de cambios en tiempo real:** se puede realizar un seguimiento de las operaciones, de inserción, eliminación y actualización, así como mantener las identidades de todos los colaboradores y las fechas/horas de cambios.

La desventaja de CoWord es que necesita una aplicación mono-usuario para implementar las características colaborativas, lo cual depende del entorno de ejecución que proporcione la API adecuada para implementarla.

Las limitaciones en cuanto a la información de conciencia son las siguientes:

- no ofrece vistas WYSIWID²,
- no soporta vistas de otros colaboradores (*view-ports*),
- no provee barras múltiples de desplazamiento (*multiuser scroll bars*) para localizar a los diferentes colaboradores en el espacio de trabajo, y
- no impone restricciones en la edición de documentos, e.g., un colaborador puede trabajar libremente en cualquier parte del documento, sin que se bloquee la parte que está modificando.

GrewpEdit

Es un sistema para la edición colaborativa de documentos de texto y de programas en lenguaje Java. GrewpEdit se enfoca en apoyar a cientos de colaboradores que trabajan en cientos de documentos en una organización [Granville and Hickey, 2005]. Esta sistema proporciona un método llamado *Rendezvous* (tamaño de grupo arbitrario e independiente de su localización) y una arquitectura denominada toma-enchufable (*plugin-like*) para distribuir un conjunto de herramientas diversas de colaboración, utilizando una interfaz común denominada conciencia/encuentro (*awareness/rendezvous*). Además, introduce un concepto llamado *Grewplet* o *Grewp* (forma de colaboración), el cual provee un mecanismo para crear fácilmente nuevas aplicaciones colaborativas y extender la capacidad del sistema GrewpEdit.

GrewpEdit tiene las siguientes características:

- ofrece cuatro formas de realizar la colaboración:
 - 1) por edición de documentos,
 - 2) por edición de dibujo mediante un pizarrón,
 - 3) por mensajero instantáneo, y

- 4) por edición de Grewplets en Java (i.e., permite extender las aplicaciones de GrewpEdit);
- ofrece un lenguaje Java para crear Grewplets y así extender la capacidad del sistema GrewpEdit;
 - implementa el método *Rendezvous* para grupos de tamaños arbitrarios, independientemente de su ubicación;
 - guarda todas las acciones para proporcionar un soporte a las conexiones lentas; y
 - mantiene la privacidad de los colaboradores (un sitio puede ser privado o público).

Cada forma de colaboración mencionada previamente, puede ser utilizada independientemente de las demás, de tal manera que un colaborador puede emplear un forma cualquiera o todas dependiendo de sus necesidades.

GrewpEdit provee la siguiente información de conciencia de grupo:

- cajas textos para mostrar la lista de los colaboradores,
- cajas textos para mostrar los grupos creados,
- botones de colores para indicar los nombres de los colaboradores que editan documentos,
- cajas de colores para indicar en que parte del documento trabajan los colaboradores (en las cuatro formas de colaboración),
- una lista de los *Grewps* presentes en el sistema,
- una lista de *Grewps* creados por el colaborador local y por los colaboradores remotos,
- un conjunto de pestañas *Grewp*, que indican los diferentes espacios de trabajo del colaborador local y de los colaboradores remotos y
- una lista de los colaboradores y de los grupos a los que pertenecen.

La Figura 2.17 muestra el editor GrewpEdit donde se observa cada una de las herramientas que ofrece.

La desventaja principal de GrewpEdit es que no provee ninguna diferencia entre un colaborador activo y uno inactivo. Después de un tiempo se desconoce qué parte del documento ha modificado cada colaborador.

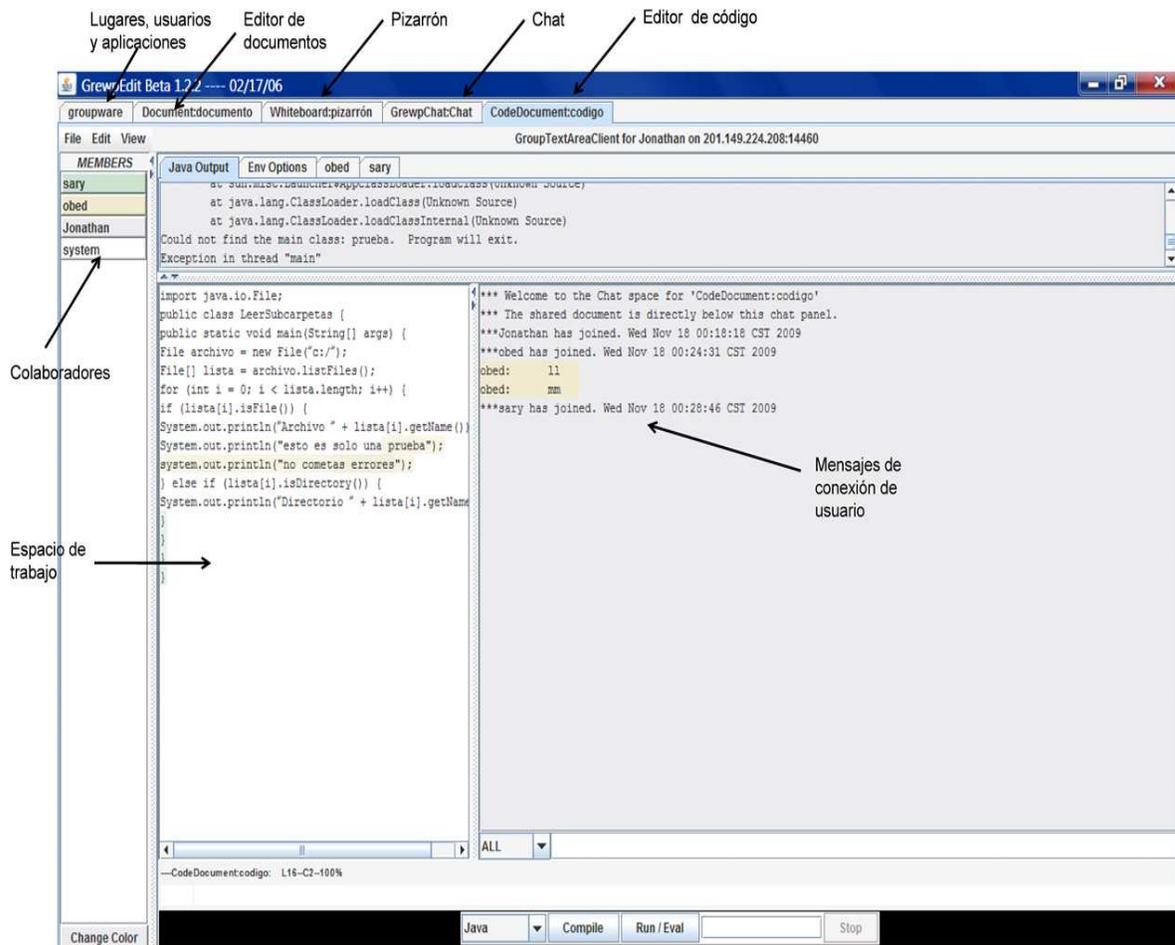


Figura 2.17: Interfaz de usuario del editor GrewpEdit

Concept Mapper

Es un sistema colaborativo para la edición de mapas conceptuales. Este sistema permite a los colaboradores manipular objetos en su espacio de trabajo y llevar a cabo alguna tarea individual o grupal. Además, este sistema permite visualizar las actividades de sus colegas dependiendo de sus necesidades (individuales o grupales) [Gutwin and Greenberg, 1998].

Concept Mapper tiene las siguientes características:

- **navegación en espacios de trabajos:** los colaboradores pueden moverse libremente en su espacio de trabajo y al mismo tiempo visualizar la información de sus colegas sobre su presencia, su ubicación, sus actividades y sus acciones en curso;
- **percepción de acciones:** son las acciones que realizan los colaboradores sobre los objetos mediante manipulación directa. Estas acciones se realizan por medio de la retroalimentación de procesos, la cual se lleva a cabo cuando un colaborador

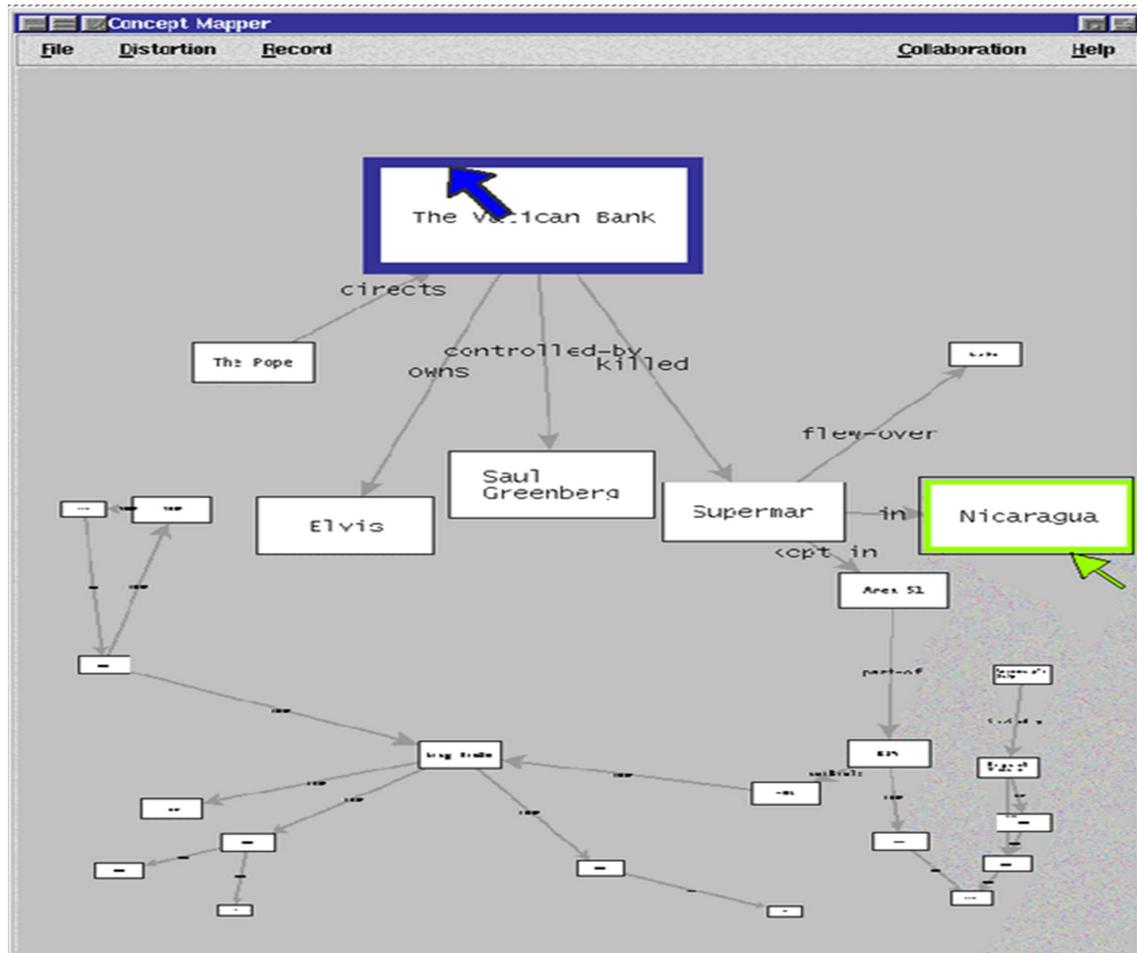


Figura 2.18: Interfaz de usuario del editor de mapas Concept Mapper

ejecuta algún comando, e.g., mover el cursor sobre un botón o cuando un colaborador presiona un botón para enviar los efectos de alguna acción a los colaboradores remotos.

- **representación de vistas:** las vistas permiten a los colaboradores visualizar la presencia, la ubicación y las acciones de los colaboradores remotos en el espacio compartido.

Las características previamente mencionadas proporcionan el soporte de información de conciencia de grupo a los colaboradores del sistema Concept Mapper.

A continuación, se presenta los diferentes tipos de información de conciencia de grupo que provee este sistema:

- **telepunteros** (*telepointer*): muestran información sobre la ubicación de los colaboradores en el espacio compartido;

- **vista WYSIWIS¹**: los colaboradores pueden observar las acciones que están realizando sus colegas (ver Figura 2.18);
- **vistas radar** (*radar view*): permiten ver la presencia de los colaboradores, la localización de los colaboradores y sus acciones. Las vistas son representadas por pequeños recuadros, que corresponden a la vista principal de cada colaborador remoto.

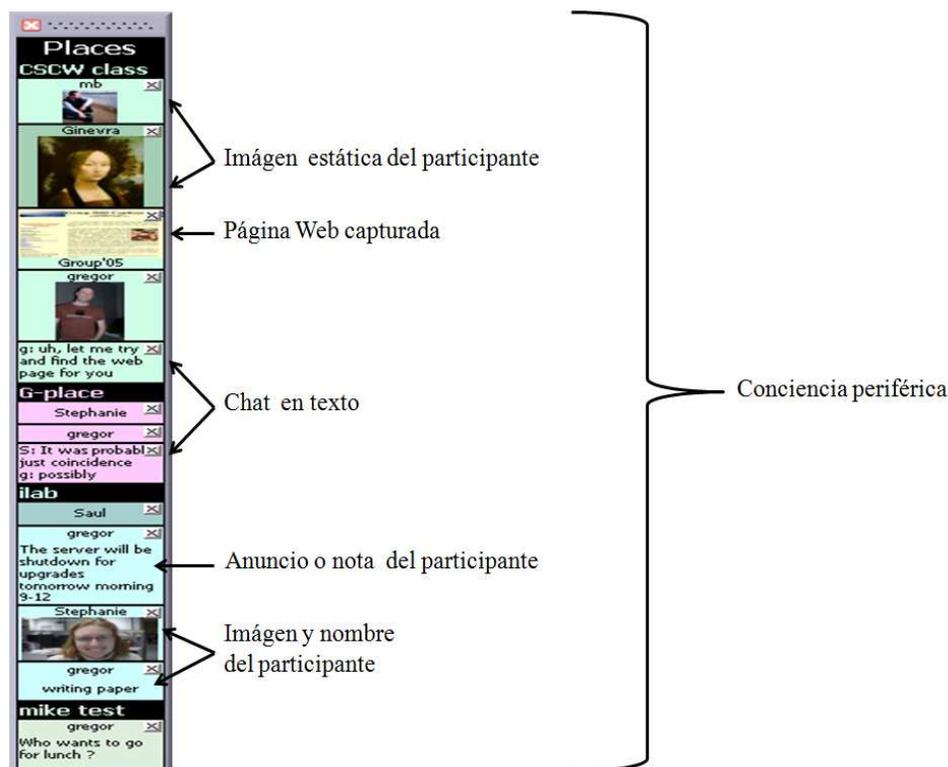


Figura 2.19: Información de conciencia de grupo en Community Bar

2.3 Descripción del sistema Community Bar

Community Bar (CB) [Romero et al., 2007] es un sistema colaborativo que soporta la interacción (son planeadas, breves y frecuentes) casual entre grupos sociales pequeños (2 a 15 colaboradores). Estas interacciones son estimuladas por la proximidad física de los usuarios, los cuales adquieren información de conciencia (presencia, disponibilidad y actividad) de los demás miembros del grupo. Por lo tanto, para satisfacer estos requerimientos Community Bar ofrece las siguientes características como soporte a grupos sociales:

1. **Información privada visible en la periferia:** la información de conciencia tiene que ser constante y dinámica para mantener la información de conciencia de los colaboradores. Community Bar muestra la información de los colaboradores mediante una barra, de tal manera que un colaborador puede seleccionar un colaborador de forma rápida con el fin de pasar fácilmente de conciencia de grupo a interacción (ver Figura 2.19).

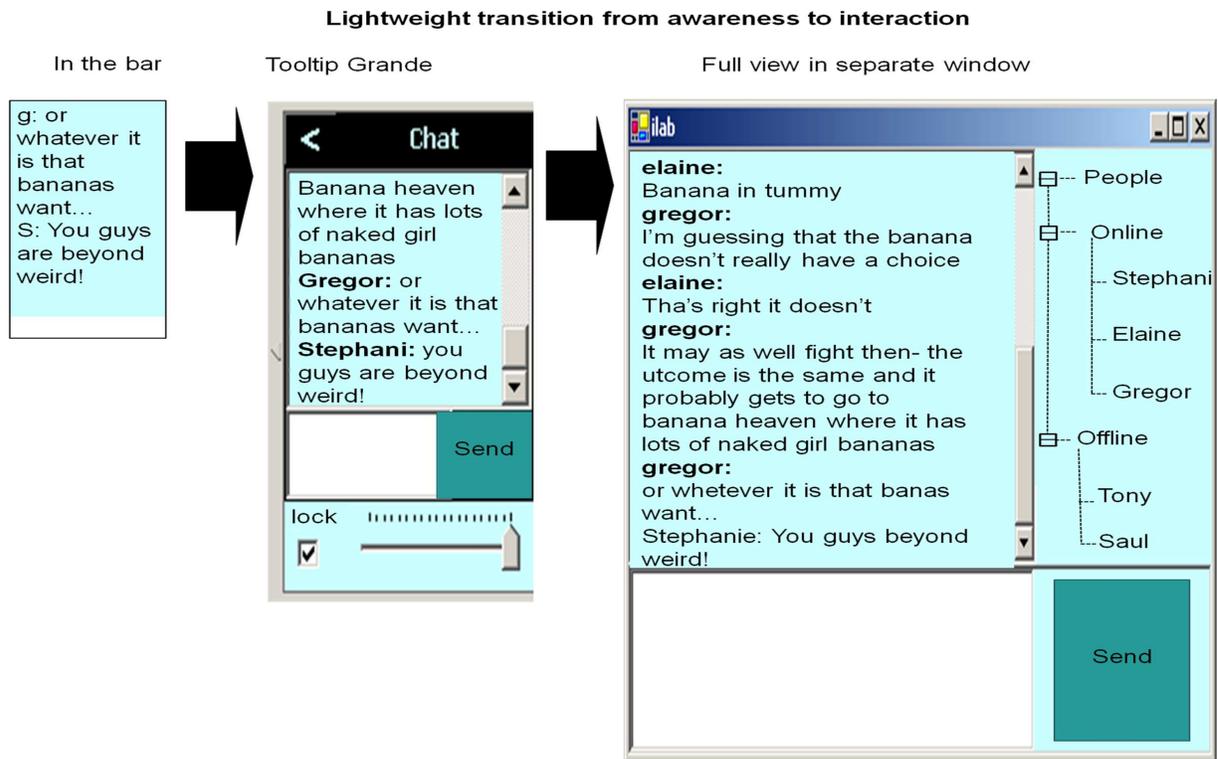


Figura 2.20: Transiciones ligeras en Community Bar

2. **Transiciones ligeras de conciencia periférica a interacción:** las transiciones son cambios que un colaborador realiza cuando existe algún tipo de información de su interés en los *tooltips* (cajas que muestran consejos a los colaboradores). De esta manera, un colaborador podría en algún momento interactuar con otros, pasando de un *tooltip* pequeño (vistas parciales) a una ventana de interacción. Community Bar implementa las transiciones ligeras mediante *tooltips* pequeños (vistas parciales) y *tooltips* grandes (vistas completas). Los *tooltips* grandes permiten al colaborador ampliar su ventana de interacción para tener una vista completa de la conversación (ver Figura 2.20).
3. **Soporte a grupos *ad-hoc* de colaboradores:** una de las prioridades de Community Bar es el soporte para la formación rápida de mundos sociales, e.g., cuando se

crea un mundo social, un colaborador puede configurar sus sitios o lugares y unirse a corto y largo plazo sin importar el tiempo de creación (ver Figura 2.21).

4. **Control de *Nimbus*:** permite ocultar o hacer visible la información de un colaborador (e.g., su nombre o su foto) a los demás miembros del grupo mientras se conserva su privacidad. Community Bar implementa una barra de desplazamiento para limitar la información de conciencia, tal como una instantánea de video o una foto (ver Figura 2.22).
5. **Control de *Focus*:** es la manera en que las personas se dirigen o toman cierto tipo de información para determinar su propia conciencia. Community Bar ofrece a los colaboradores un control explícito para reducir o ampliar su vista en un lugar. Cada *tooltip* posee una barra de desplazamiento (similar al control de *Nimbus*) para ajustar el *Focus* (ver la barra de desplazamiento de las Figuras 2.20 y 2.21).

Community Bar ofrece a los colaboradores la siguiente información de conciencia de grupo:

- información de presencia mediante su foto o instantánea de video,
- una lista textual de los lugares creados (ver Figura 2.21),
- un conjunto de vistas (*viewset*) de los diferentes colaboradores y lugares creados (ver Figura 2.19),
- una vista individual para interactuar con un colaborador específico.



Figura 2.21: Soporte de grupos *ad-hoc* en Community Bar

2.4 Discusión sobre el trabajo relacionado

Los sistemas colaborativos previamente descritos ofrecen información de conciencia sobre la presencia, la ubicación y las actividades de los colaboradores.

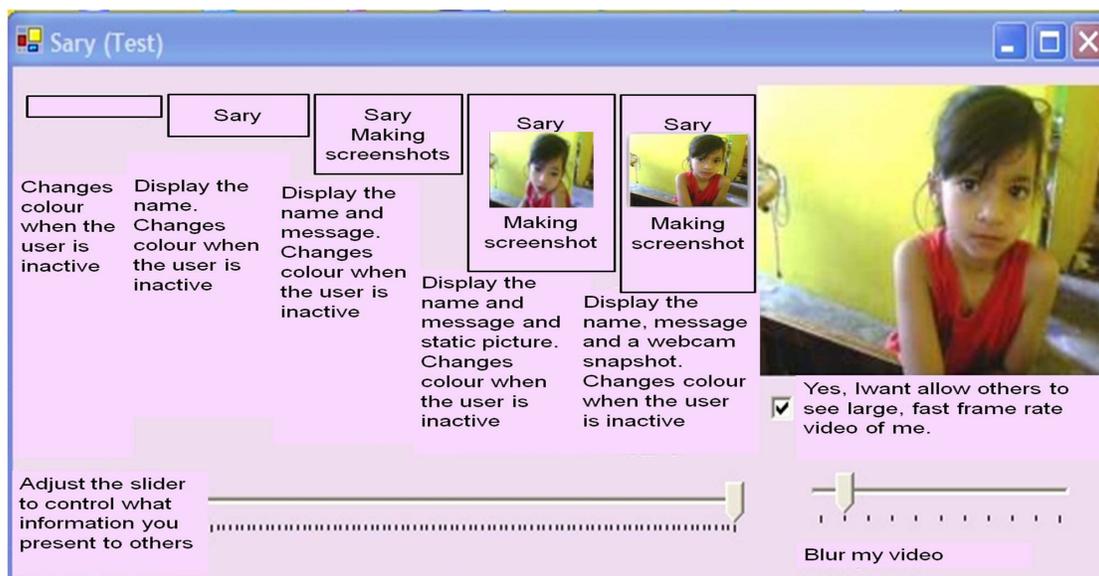


Figura 2.22: Control de la información de conciencia mediante *Nimbus* en Community Bar

El sistema Peepholes utiliza una alternativa de bajo costo para mostrar la presencia de los colaboradores mediante indicadores icónicos, pero tiene la desventaja de que no provee ninguna diferencia entre un usuario ausente y uno activo.

Por su parte, los mensajeros instantáneos pueden compartir recursos mediante el concepto de petición/respuesta entre dos colaboradores, pero no contemplan la intervención de todo el grupo de usuarios.

El sistema JamSpace permite crear música y realizar mezclas de pistas de manera colaborativa. Esta última funcionalidad es una ventaja para los colaboradores, debido a que aprenden sobre los gustos, costumbres y composiciones de otras personas. La desventaja de JamSpace es la falta de información de conciencia de grupo, e.g., el nombre de una pista y los nombres de los colaboradores que modifican o mezclan alguna pista en el espacio compartido son desconocidos por los usuarios (ver sección 2.2.1).

Por último, el sistema Doc2u provee información de conciencia de documentos, tal como la presencia, la disponibilidad y la actividad. La desventaja de Doc2U radica en las anotaciones, ya que estas no están asociadas directamente al documento, sino que se envían al propietario del documento, mediante mensajes.

Estos cuatro sistemas de conciencia periférica están enfocados en ambientes virtuales y ofrecen poca información de conciencia a los colaboradores. En algunos casos, los colaboradores requieren información detallada de las acciones de sus colegas (e.g., conocer

la habilidad de un colaborador), pero estos sistemas no poseen este tipo de información de conciencia grupo.

Por otro lado, los sistemas de conciencia enfocada ofrecen información de conciencia a nivel de grano fino. En particular, el sistema CoWord provee a los colaboradores información de conciencia de grupo mediante la interfaz de Microsoft Word, ya que convierte a Word (aplicación mono-usuario) en un editor colaborativo (ver sección 2.2.2). Una de las principales desventajas de CoWord radica en la dependencia de Word, lo que implica que no se pueda extender o agregar nuevas funcionalidades a CoWord.

Por otro lado, el sistema GrewpEdit ofrece varios tipos de aplicaciones tales como: a) la edición de documentos, b) la edición de código en Java (llamado Grewplets), c) la interacción mediante un pizarrón y d) la interacción entre colaborador mediante conversaciones (*chat*). Todas estas aplicaciones colaborativas de GrewpEdit ofrecen versatilidad, pero no proveen información de conciencia de grupo (e.g., la ubicación del colaborador en el documento compartido) ni imponen restricciones de acceso (e.g., dos colaboradores pueden modificar la misma parte de un documento).

El sistema Calliope es un editor de documentos colaborativos en el que cada colaborador puede escribir una parte del documento, como el resumen o la bibliografía de un artículo. Las acciones realizadas en este editor no están dirigidas explícitamente a una persona específica, sino que se publican de manera intencional.

Por su parte, el sistema Concept Mapper posee un mecanismo de vista global, la cual permite a los colaboradores ver los objetos que manipulan sus colegas en el espacio de trabajo por medio de vistas WYSIWID².

Los cuatro sistemas mencionados, ofrecen información de conciencia enfocada dentro de su entorno de trabajo. Esta información en ocasiones es útil para algunos, pero para otros puede resultar excesiva. Además, se pueden reducir eventualmente algunas ventajas de estos sistemas porque ocasionan problemas de privacidad y en algunos casos inconsistencia de la información compartida.

Estos problemas se deben principalmente a que los diseñadores de estos sistemas colaborativos no saben con exactitud qué tipo de información es relevante para las personas. Consecuentemente, un sistema puede proveer mucha información, ocasionando problemas de privacidad, ancho de banda y escalabilidad, o bien puede ofrecer poca información, causando problemas de contactos inapropiados o pérdidas de oportunidades.

Cada uno de los sistemas previamente mencionados están diseñados para proveer información de entornos meramente virtuales.

La Tabla 2.2 muestra los diferentes tipos de información de conciencia de grupo que proveen estos sistemas, los cuales están basados en tres tipos principales: a) presencia, b) actividad y c) disponibilidad. Por ejemplo, se puede observar en la tabla 2.2 que los sistemas DOC2U y Peepholes proveen los tres tipos de información de conciencia de grupo. Por otro lado, los sistemas como Instant Messenger y JamSpace ofrecen solo información de presencia y disponibilidad.

La Tabla 2.3 muestra una comparación entre los diferentes sistemas de conciencia enfocada previamente analizados. Esta comparación se basa en tres tipos de información y ocho tipos de *widgets* de conciencia de grupo. Los tipos de información se refieren a: 1) presencia, 2) actividad, 3) disponibilidad, en tanto que los tipos de *widgets* corre-

sponden a: 1) vistas WYSIWIS, 2) vistas WYSIWID, 3) telepunteros, 4) vistas múltiples (mutiusuarios), 5) vistas radar, 6) barras multiusuario (*multi-user scrollbars*), 7) portales (*teleport*) y 8) vistas panorámicas. Cabe mencionar que cada tipo de *widget* ofrece un subconjunto de información de conciencia de grupo, e.g., una vista radar muestra el conjunto de objetos que está manipulando un colaborador. En la Tabla 2.3 también se puede observar que ninguno de los sistemas provee los tres tipos de información ni los ocho tipos de *widgets* mencionados, e.g., Calliope es el único sistema que provee la mayoría de estas características, aunque no ofrece vistas de otros participantes (portales) ni vistas WYSIWID. Por su parte, GrewpEdit provee información de presencia, actividad y disponibilidad mediante vistas WYSIWIS y telepunteros.

Tabla 2.2: Tipos de información de conciencia de grupo que ofrecen los sistemas conciencia periférica

Sistemas Colaborativos de Conciencia Periférica	Objetivo	Información de Conciencia de Grupo
Doc2U	Proveer conciencia de presencia de documentos	Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad <input checked="" type="checkbox"/>
Peepholes	Administrar agendas electrónicas	Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad <input checked="" type="checkbox"/>
JamSpace	Mezclar música	Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad <input type="checkbox"/>
Instant Messenger	Iniciar conversaciones	Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad <input type="checkbox"/>

Por otra parte, el sistema Community Bar ofrece información de conciencia de grupos sociales bajo el concepto sociológico de agrupación. Community Bar presenta la información de conciencia de grupo mediante una herramienta (ver Figura 2.22) que permite decidir qué información enviar (lado del emisor) y qué información visualizar (lado del receptor). Además, este sistema proporciona información de conciencia enfocada y periférica. Sin embargo, Community Bar no permite compartir archivos y además el número de colaboradores que soporta está reducido a un máximo de quince.

Se concluye que los sistemas de conciencia periférica ofrecen poca información en comparación con los sistemas de conciencia enfocada (ver Tabla 2.2 y Tabla 2.3), sin embargo los sistemas de conciencia enfocada corren el riesgo de ofrecer demasiada información. Por lo tanto, no existe un equilibrio entre la información que desea el colaborador y la

información que proveen los dos tipos sistemas (i.e., existe un vacío entre ellos). Por otro lado, Community Bar no provee información de conciencia de entornos físicos.

Tabla 2.3: Tipos de información y de *widgets* de conciencia de grupo que ofrecen los sistemas de conciencia enfocada

Sistemas Colaborativos de Conciencia Enfocada	Objetivo	Tipos de Información y de <i>Widgets</i> de Conciencia de Grupo
Calliope	Editar documentos	<input checked="" type="checkbox"/> Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad
		<input checked="" type="checkbox"/> Vista WYSIWIS <input checked="" type="checkbox"/> Vista WYSIWID
		<input type="checkbox"/> Telepunteros <input checked="" type="checkbox"/> Vistas múltiples
		<input checked="" type="checkbox"/> Vistas Radar <input checked="" type="checkbox"/> Vistas con lupa
		<input checked="" type="checkbox"/> Portales <input type="checkbox"/> Barras multiusuario
CoWord	Editar documentos en Word	<input checked="" type="checkbox"/> Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad
		<input type="checkbox"/> Vista WYSIWIS <input checked="" type="checkbox"/> Vista WYSIWID
		<input checked="" type="checkbox"/> Telepunteros <input type="checkbox"/> Vistas múltiples
		<input checked="" type="checkbox"/> Vistas Radar <input type="checkbox"/> Vistas con lupa
		<input checked="" type="checkbox"/> Portales <input type="checkbox"/> Barras multiusuario
GrewpEdit	Editar documentos (edición de dibujo, edición de programas en Java, edición de documentos y establecer conversaciones)	<input checked="" type="checkbox"/> Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad
		<input checked="" type="checkbox"/> Vista WYSIWIS <input type="checkbox"/> Vista WYSIWID
		<input checked="" type="checkbox"/> Telepunteros <input type="checkbox"/> Vistas múltiples
		<input type="checkbox"/> Vistas Radar <input type="checkbox"/> Vistas con lupa
		<input type="checkbox"/> Portales <input type="checkbox"/> Barras multiusuario
Concept Mapper	Editar mapas conceptuales	<input checked="" type="checkbox"/> Presencia <input checked="" type="checkbox"/> Actividad <input checked="" type="checkbox"/> Disponibilidad
		<input checked="" type="checkbox"/> Vista WYSIWIS <input checked="" type="checkbox"/> Vista WYSIWID
		<input checked="" type="checkbox"/> Telepunteros <input checked="" type="checkbox"/> Vistas múltiples
		<input type="checkbox"/> Vistas Radar <input type="checkbox"/> Vistas con lupa
		<input checked="" type="checkbox"/> Portales <input checked="" type="checkbox"/> Barras multiusuario

En el siguiente capítulo se pretende hacer frente a estos problemas, con el fin de llenar

el vacío existente entre dichos tipos de sistemas, mediante la provisión de información gradual de conciencia de grupo sobre los recursos humanos, físicos y virtuales de una organización.

Capítulo 3

Análisis y diseño del sistema SIGCo

El objetivo del presente capítulo es describir el sistema propuesto, denominado Sistema de Información Gradual de Conciencia (SIGCo), el cual permite controlar gradualmente la información de conciencia de grupo proveniente de espacios físicos y virtuales. Primero, se presenta un escenario que pone en evidencia los beneficios de la implantación de este sistema en una organización, con el fin de proveer información gradual al personal (sección 3.1). Después, se describe los casos de uso del sistema SIGCo (sección 3.2), así como las principales clases que lo conforman (sección 3.3). Posteriormente, se explica la arquitectura de *software* del sistema SIGCo y la interacción de los objetos de dichas clases mediante diagramas de secuencia (sección 3.4). Finalmente, se presenta la arquitectura de distribución de dicho sistema (sección 3.5).

3.1 Escenario de uso del sistema SIGCo

Hoy en día muchas organizaciones (e.g., bancos, hospitales y dependencias de gobiernos) cuentan con recursos físicos tales como equipos de comunicación para redes, equipos de ultrasonidos y equipos de cómputo, así como recursos virtuales tales como aplicaciones de software, base de datos y hojas de cálculos. Estos recursos han permitido a las organizaciones acelerar algunos de sus procesos cooperativos, pero la administración ineficiente de dichos recursos ha generado problemas tales como: a) la saturación del ancho de banda de la red (e.g., multimedia), b) la duplicación de información y c) la provisión de información excesiva. Estos problemas han ocasionado que los colaboradores pierdan oportunidades y se viole su privacidad.

Para resolverlos, se propone el sistema SIGCo, el cual puede ser implantado en un entorno hospitalario o en cualquier organización donde el personal requiera acceder a la información de manera gradual. Con el fin de ilustrar las funcionalidades que ofrece este sistema, a continuación se describe un escenario de uso en un entorno hospitalario.

Las diversas áreas del hospital están relacionadas con los diferentes servicios que ofrecen a sus pacientes. En este escenario, se contemplan las siguientes cinco áreas:

- Área de recepción (A1): a) lleva el control de citas de los pacientes, b) proporciona información del estatus de los pacientes a sus familiares y c) canaliza los pacientes

a las diferentes áreas;

- Área de enfermería (A2): a) verifica la medición de peso y talla de los pacientes, b) prepara a los pacientes para consulta u hospitalización y c) lleva una bitacora de los medicamentos suministrados a cada paciente hospitalizado;
- Área de consultas (A3): proporciona un tratamiento a los pacientes de acuerdo a sus síntomas, pero si el padecimiento es grave (e.g., cáncer de hígado o leucemia), canaliza los pacientes a otras áreas;
- Área de especialidades (A4): se encarga de tratar a los pacientes con padecimientos graves que provienen de otras áreas;
- Área de diagnósticos (A5): realiza los diferentes estudios (e.g., rayos X, mastografía, resonancia magnética y tomografía) a los pacientes provenientes de las diferentes áreas del hospital y de hospitales externos.

En las áreas públicas del edificio, se han colocado cámaras con la finalidad de que el sistema RAMS (*Resource Availability Management System*) [García, 2009] pueda identificar y localizar al personal del hospital. Cabe mencionar que a cada usuario se le atribuye un rol (e.g., recepcionista, enfermera, médico general y especialista), el cual corresponde al puesto asignado por el área de Recursos Humanos.

En las diversas áreas existen recursos físicos (e.g., equipos de tomografía, de rayos X y de ultrasonido) e informáticos (e.g., expedientes, diagnósticos, radiografías y ultrasonidos) que permiten realizar las actividades cotidianas del hospital. Particularmente, el equipo del área de diagnósticos es compartido con otros hospitales. Por lo tanto, en cualquier momento, dicho equipo puede ser utilizado por especialistas externos.

Por otra parte, los recursos virtuales (e.g., imágenes de rayos X y de ultrasonido) pueden ser generados por entidades externas (e.g., equipo de rayos X y de ultrasonido) o por el propio sistema SIGCo (e.g., un historial de citas o una imagen de ultrasonido). Ambos tipos de recursos pueden ser empleados por otros usuarios, si el encargado otorga derechos de acceso.

El personal del hospital puede conocer la ubicación de sus colegas por medio de un mapa que muestra su identificación (e.g., nombre o foto). Cabe mencionar que el sistema SIGCo da la posibilidad de ocultar la ubicación de los usuarios, pero por tratarse de un entorno hospitalario la ubicación del personal siempre es visible a todos sus colegas.

El sistema propuesto también facilita la creación de grupos *ad-hoc* con la finalidad de permitir que los médicos compartan con sus colegas información clínica de sus pacientes (e.g., análisis sanguíneos, radiografías y ultrasonidos) y colaboren conjuntamente en la determinación de diagnósticos.

El sistema SIGCo requiere que todos los usuarios se autentifiquen para poder proveerles información de acuerdo a su rol (ver tabla 3.1).

Para mostrar las funcionalidades del sistema SIGCo, suponga que un paciente (cuyos datos han sido registrados previamente) hace su llegada al hospital para tratar su enfermedad.

Este paciente se presenta primeramente en la recepción (área A1) en el horario señalado (i.e., previa cita) para que pueda ser atendido por su médico (ver el paso 1 de la Figura 3.1).

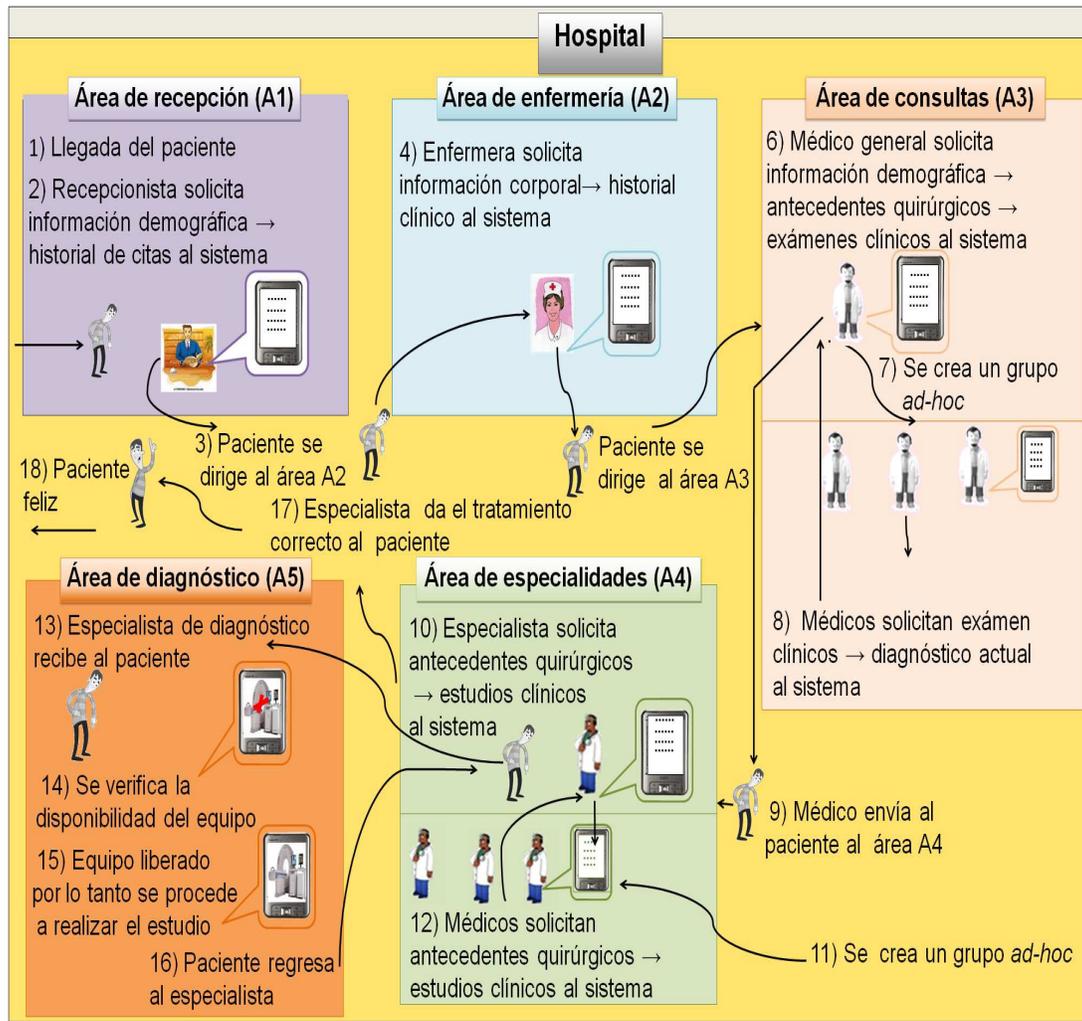


Figura 3.1: Escenario de uso del sistema SIGCo en un hospital

La recepcionista procede a verificar la cita del paciente y el nombre del médico que lo atenderá; esta información es mostrada conforme la recepcionista la solicite al sistema SIGCo (ver el paso 2 de la Figura 3.1). Después de haber verificado la información del paciente, la recepcionista lo envía al área de enfermería (área A2) para ser atendido (ver el paso 3 de la Figura 3.1).

La enfermera recibe al paciente y verifica su información corporal (e.g., peso) y su historial clínico (e.g., receta médica) comparándola con la información registrada en el sistema SIGCo (ver el paso 4 de la Figura 3.1). Después, la enfermera envía al paciente al área de consultas (área A3) para que sea atendido (ver el paso 5 de la Figura 3.1).

Tabla 3.1: Tipos de información accesibles de acuerdo al rol del usuario

Rol	Información gradual de conciencia de grupo
Recepcionista	Información demográfica (e.g., nombre, edad y ocupación) → Historial de citas → Estatus del paciente
Enfermera	Información corporal → Historial clínico (e.g., una prescripción médica) → Alergias (e.g., mariscos y penicilina)
Médico general	Antecedentes quirúrgicos → Historial familiar → Exámenes médicos (e.g., glucosa y química sanguínea) → Diagnóstico actual
Médico especialista	Antecedentes quirúrgicos → Historial familiar → Estudios clínicos (e.g., imágenes de ultrasonidos y de rayos X) → Diagnóstico actual
Especialista del área de diagnóstico	Información demográfica → Antecedentes de síntomas (e.g., dolor de cabeza y diarrea frecuente) → Recursos físicos (e.g., ubicación y disponibilidad)
Colegas de los médicos generales	Antecedentes quirúrgicos → Historial familiar → Exámenes médicos (e.g., glucosa y química sanguínea) → Diagnóstico actual
Colegas de médicos especialistas	Antecedentes quirúrgicos → Historial familiar → Estudios clínicos (e.g., imágenes de ultrasonidos y de rayos X) → Diagnóstico actual

El paciente llega al área de consultas, donde es atendido por el médico general. Una vez analizada la información clínica del paciente, el médico solicita al sistema SIGCo los datos generales del paciente (ver el paso 6 de la Figura 3.1), después sus análisis clínicos y por último el diagnóstico inmediato anterior para estar conciente de su estado actual.

A continuación, el médico general emite la información de su paciente a los colegas (médicos generales) que estén interesados en participar y analizar conjuntamente dicha información (ver el paso 7 de la Figura 3.1).

Después de un tiempo, algunos colegas notifican su interés al médico general. Como resultado se crea un grupo *ad-hoc* para que colaboren en el diagnóstico del enfermo y den un juicio colectivo (ver el paso 8 de la Figura 3.1). Más tarde, el grupo *ad-hoc* emite su juicio sobre el diagnóstico del enfermo. El médico general analiza el diagnóstico que han dado sus colegas y toma la decisión de enviar el paciente a un médico especialista. Por lo tanto, el paciente se dirige al área de especialidades (área 4) para que sea tratado de manera adecuada (ver el paso 9 de la Figura 3.1).

El médico especialista del área A4, que recibe al paciente, solicita información al sistema SIGCo (ver el paso 10 de la Figura 3.1). A continuación, el médico especialista notifica a sus colegas la información del paciente (e.g., estudios clínicos y diagnóstico anterior)

para que puedan colaborar con él y establecer un diagnóstico de dicho paciente (ver el paso 11 de la Figura 3.1). Por lo tanto, se crea un grupo *ad-hoc* para discutir la situación del paciente y emitir conjuntamente un diagnóstico (ver el paso 12 de la Figura 3.1). El grupo *ad-hoc* notifica el resultado al colega encargado del paciente. Este resultado sugiere que se realice un estudio de médula ósea para comprobar dicho diagnóstico (ver el paso 12 de la Figura 3.1). El médico especialista procede a emitir una orden al paciente para que se le practiquen los estudios indicados.

El médico del área de diagnóstico recibe al paciente y procede a solicitar su información (e.g., tipos de estudios y diagnóstico) al sistema SIGCo por medio de una PDA. Este médico verifica la disponibilidad de los equipos, pero el sistema le informa que alguien externo está utilizando el equipo de tomografía (ver el paso 14 de la Figura 3.1). Por lo tanto, el paciente tiene que esperar a que el equipo sea liberado. Después de algún tiempo, el sistema SIGCo notifica al médico del área de diagnóstico que el equipo ha sido liberado y entonces procede a realizar el estudio al paciente (ver el paso 15 de la Figura 3.1). Una vez finalizado dicho estudio, el paciente regresa nuevamente al área de especialidades.

El médico especialista emite nuevamente la información del paciente a sus colegas (ver el paso 11 de la Figura 3.1), quienes le responden con un resultado satisfactorio (ver el paso 12 de la Figura 3.1). En ese momento, se le proporciona el tratamiento correcto al paciente (ver el paso 16 de la Figura 3.1).

Finalmente, el paciente satisfecho con sus resultados procede a abandonar el hospital (ver el paso 17 de la Figura 3.1).

3.2 Análisis del sistema SIGCo

Esta sección se describe los recursos humanos, físicos y virtuales que pueden existir en una organización. También, se pone en evidencia los requisitos funcionales del sistema SIGCo mediante casos de uso, así como los diferentes actores que intervienen en dicho sistema [Lerman, 2004].

3.2.1 Descripción de recursos

El sistema SIGCo administra los siguientes tres tipos de recursos:

- **Recursos físicos:** son los dispositivos computacionales (e.g., PDAs, escáneres y cámaras digitales) y recursos médicos (e.g., equipo de tomografía y de ultrasonido) que los colaboradores pueden utilizar y compartir con sus colegas. Además, los recursos físicos pueden ser compartidos con otras organizaciones externas.
- **Recursos humanos:** son los miembros de una organización, que hacen uso de la información de conciencia de grupo, pero que además pueden generar recursos virtuales. Una de las características de los recursos humanos es la movilidad que tienen dentro de la organización, por lo que en cualquier momento podrían no encontrarse en su lugar de trabajo (e.g., oficina). Por lo tanto, se vuelve necesario localizar a los colaboradores para administrar su información (e.g., la presencia) haciendo uso del sistema RAMS [García, 2009].

- **Recursos virtuales:** son recursos informáticos tales como archivos multimedia y *software*, los cuales pueden ser generados por entidades externas (e.g., equipo de ultrasonido o tomógrafo), por el personal de la organización (e.g., receta médica) y por el propio sistema SIGCo (e.g., historial de citas).

Los recursos físicos y virtuales pueden ser utilizados por otros usuarios, si el propietario da la autorización. En particular, los recursos virtuales están limitados por los roles establecidos por el Departamento de Recursos Humanos de una organización.

Los recursos físicos y virtuales pueden ser públicos o privados. Los recursos públicos (e.g., una copiadora o un estetoscopio) son aquellos que pueden ser utilizados por todos los miembros de una organización. En cambio, los recursos privados (e.g., equipo de tomografía o de rayos X) son aquellos que están asociados (i.e., bajo responsabilidad) a un rol. Asimismo, los recursos virtuales pueden estar distribuidos en varios sitios, por lo cual son administrados de manera independiente.

3.2.2 Casos de uso del sistema SIGCo

En esta sección, se describen los casos de uso del sistema SIGCo. En particular, se presentan los casos de uso de los productores y consumidores que intervienen en dicho sistema.

Casos de uso de los productores

Los productores son los diferentes actores que intervienen en el sistema SIGCo para realizar algunas tareas tales como agregar, modificar o publicar recursos, con el fin de compartirlos con sus colegas.

La Figura 3.2 muestra un diagrama de los casos de uso del sistema SIGCo, que describen las operaciones de sus diferentes actores:

- **Administrador:** es un usuario encargado de realizar la administración de los recursos humanos, físicos y virtuales de la organización (e.g., asignar un rol a un recurso humano o modificar un dato de un paciente):
 - **Generar información de recursos físicos:** dado que un recurso puede ser modificado a causa de un imprevisto (e.g., una urgencia de un paciente), el administrador puede realizar cambios en cualquier momento (e.g., un recurso físico puede pasar del estado en línea al estado en mantenimiento) por medio de la operación `modificar recurso`.
 - **Generar información de recursos humanos:** el administrador puede agregar un recurso humano y describirlo por medio de las siguientes operaciones:
 - a) **agregar un rol** en términos de la función que desempeña el recurso humano dentro de la organización,
 - b) **describir los permisos** del recurso humano para acceder a la información de conciencia de grupo (e.g., una enfermera no puede acceder a la misma información de conciencia que un cirujano).

- **Generar información de recursos virtuales:** el administrador puede agregar recursos y restringir la información accesible a los usuarios de acuerdo a su rol dentro de la organización. Además, el administrador puede crear grupos *ad-hoc* para colaborar en la determinación de los diagnósticos de los pacientes.

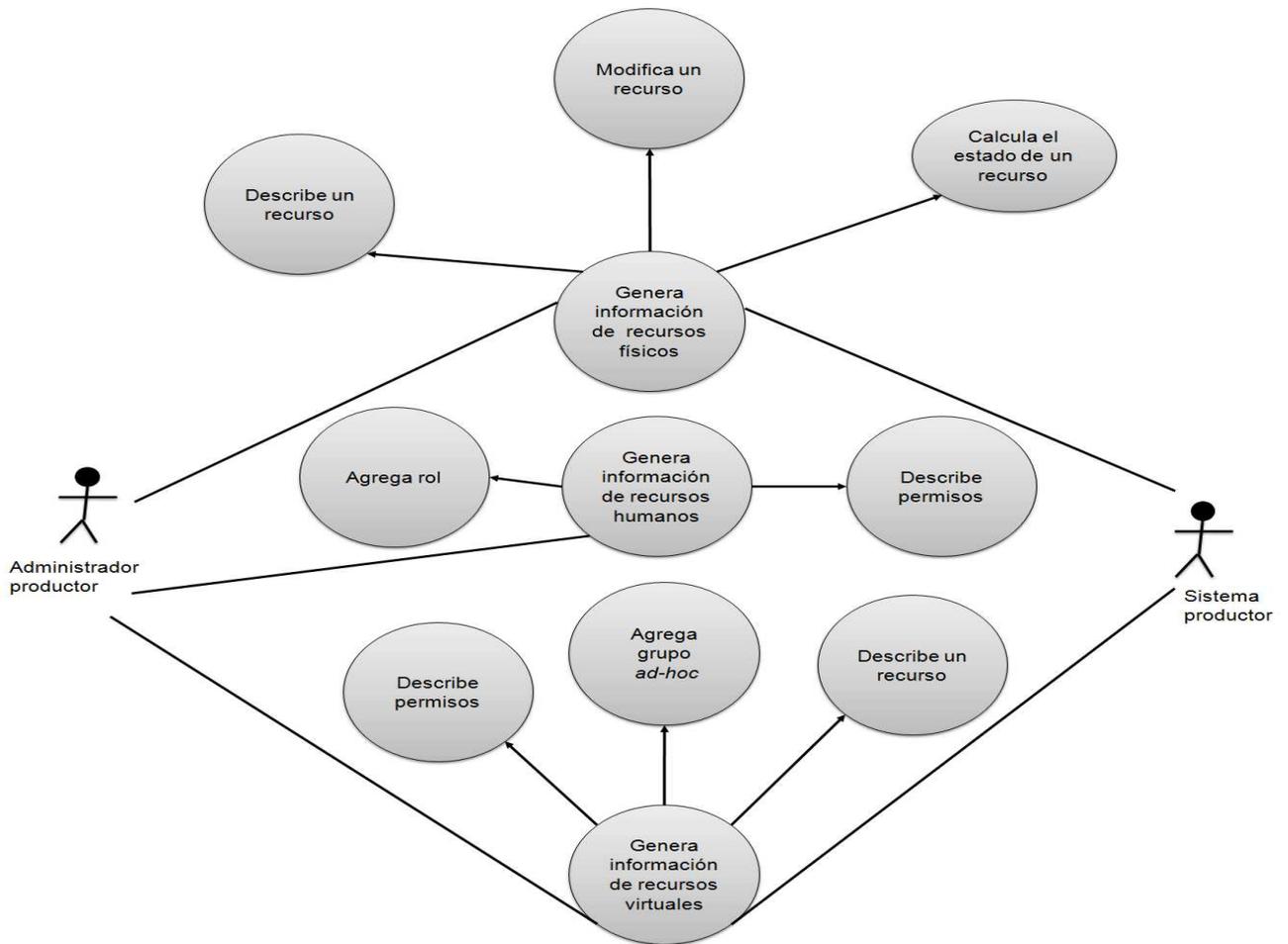


Figura 3.2: Casos de uso de los productores del sistema SIGCo

- **Sistema SIGCo (productor):** este sistema puede agregar y modificar información de un recurso de manera automática (e.g., una característica de un recurso físico o virtual).
 - **Generar información de recursos físicos:** el sistema SIGCo puede compartir un recurso con otras organizaciones mediante las siguientes operaciones:
 - a) describir las características de un recurso,
 - b) describir el estado de un recurso en términos de su disponibilidad (e.g., si el recurso está siendo utilizado, su estado es ocupado), y

- c) **modificar las restricciones de uso y acceso** de un recurso en función de los permisos atribuidos a los usuarios.
- **Generar información de recursos virtuales:** el sistema SIGCo puede agregar recursos y restringir su acceso mediante los siguientes operaciones:
 - a) **describir las características** de un recurso de acuerdo al tipo de enfermedad (e.g., cáncer de hígado),
 - b) **describir las restricciones de uso y acceso** de un recurso en función del rol de los usuarios.

Casos de uso de los consumidores

Los consumidores son los clientes que reciben información de conciencia de grupo, cuando se autentican en el sistema SIGCo. La Figura 3.3 ilustra los casos de uso, los cuales describen las operaciones que pueden realizar los consumidores del sistema SIGCo.

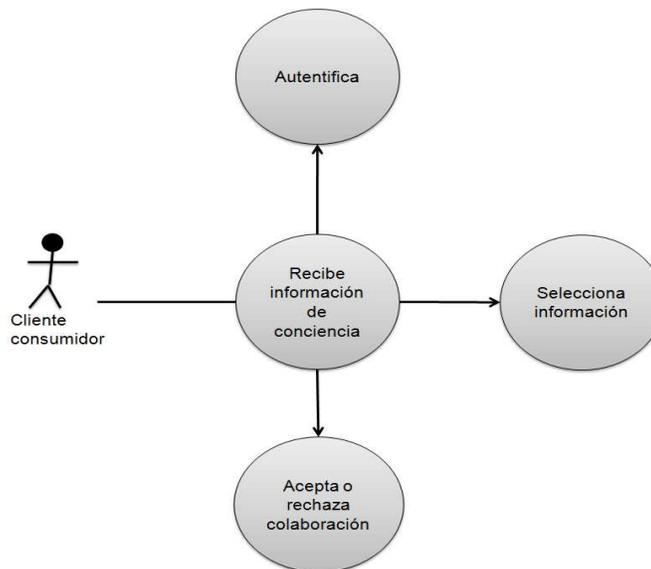


Figura 3.3: Casos de uso del consumidor del sistema SIGCo

Un consumidor pueden realizar las siguientes operaciones:

- **Recibir información gradual de conciencia de grupo:** si un consumidor desea recibir información gradual de conciencia, debe proporcionar información al sistema SIGCo mediante las siguientes operaciones:
 - a) **autenticarse** en el sistema SIGCo.
 - b) **seleccionar información** de acuerdo a su interés.

- **Aceptar o rechazar colaboración:** un consumidor puede, en cualquier momento, recibir una invitación de colaboración. Esta invitación puede ser aceptada o rechazada.

3.3 Clases del sistema SIGCo

En secciones precedentes se describieron los recursos existentes en una organización. Como se mencionó anteriormente, el sistema SIGCo permite administrar recursos humanos, físicos y virtuales.

Las clases que componen al sistema SIGCo son las siguientes:

- **Recurso:** provee sus propiedades (e.g., el método `agrega_información`) a otras clases tales como `RecursoFísico`, `RecursoVirtual` y `RecursoHumano`, de tal manera que cada una de ellas puede extender sus características dependiendo del tipo de recurso.
- **RecursoFísico:** almacena y provee información de recursos físicos a los colaboradores.
- **RecursoVirtual:** almacena la información de los recursos virtuales en una base de datos y provee dicha información a los colaboradores.
- **RecursoHumano:** almacena información sobre los miembros de una organización, así como información de pacientes. También, permite proveer información de dichos miembros y de pacientes.
- **Operación:** controla las diferentes operaciones que un colaborador puede hacer.
- **GrupoAd-hoc:** administra grupos de usuarios.
- **Rol:** controla el acceso de información sobre los recursos físicos y virtuales a los colaboradores.

A continuación, se describen las relaciones entre las diferentes clases que componen al sistema SIGCo (ver Figura 3.4):

- Entre las clases `RecursoHumano` y `Operación` existe una relación de n a m , ya que un recurso humano puede realizar múltiples operaciones (e.g., consulta) sobre recursos virtuales y, físicos, a su vez, una operación pueden ser efectuadas por múltiples recursos humanos;
- Las clases `Operacion` y `Rol` están asociadas mediante una relación de n a m , puesto que una misma operación pueden ser realizada por múltiples roles y, a su vez, un rol puede realizar una operación (e.g., un miembro de la organización puede consultar el historial de citas o la receta médica de algún paciente).

- Entre las clases `RecursoHumano` y `Rol` hay una relación de n a 1 , ya que un recurso humano puede jugar un rol dentro de la organización, pero un rol puede ser asignado a múltiples recursos humanos.
- Las clases `RecursoHumano` y `GrupoAd-hoc` establecen una asociación de n a m , puesto que un recurso humano puede formar parte de varias agrupaciones, e.g., un médico general puede participar en un estudio sobre la tifoidea y a su vez formar parte del grupo de estudios clínicos sobre la hepatitis. Por otra parte, un grupo *ad-hoc* está formado por varios recursos humanos.
- Entre las clases `GrupoAd-hoc` y `Rol` existe una relación de n a 1 , ya que un grupo *ad-hoc* puede ser creado de acuerdo a un tipo de rol (e.g., enfermera o médico general), pero un rol puede estar asociado a varios grupos *ad-hoc*, e.g., un médico especialista decide crear un grupo llamado `medico_especialista_hígado` y otro colega decide crear un grupo llamado `medico_especialista_mama`. Estos dos grupos creados pertenecen a un mismo rol de médicos especialistas, pero son diferentes grupos.

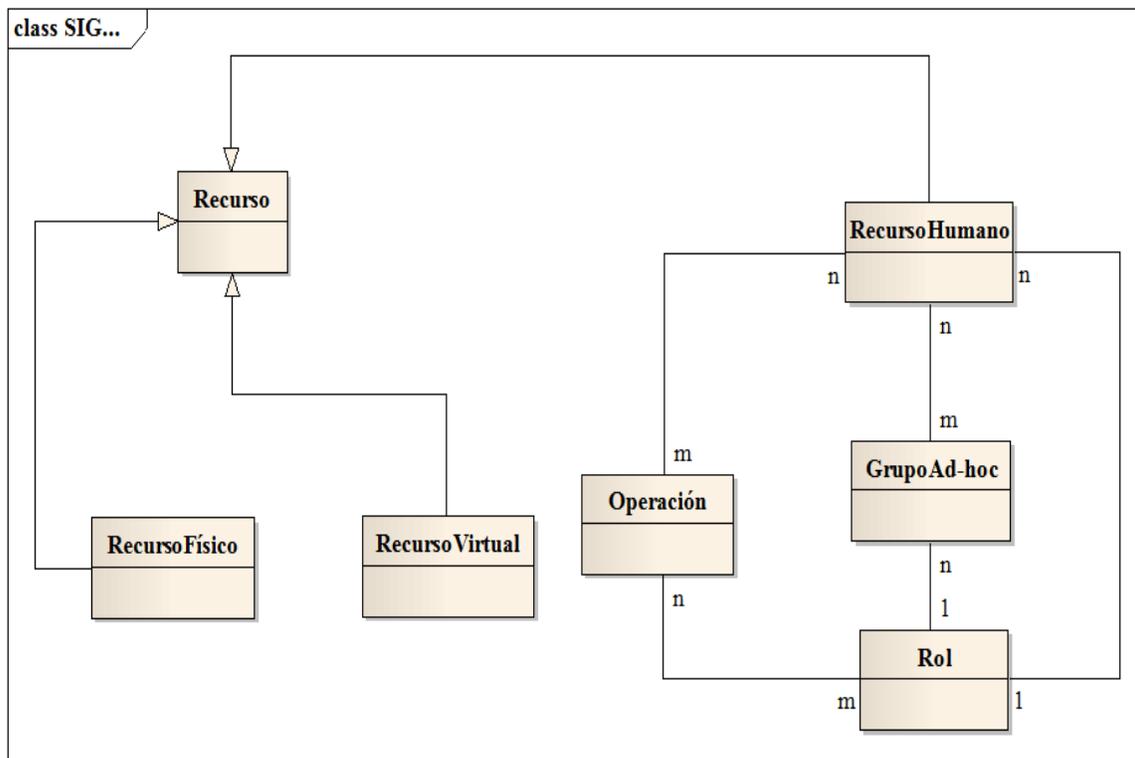


Figura 3.4: Diagrama de clases del sistema SIGCo

3.4 Arquitectura de *software* del sistema SIGCo

La arquitectura de *software* del sistema SIGCo está formada por tres capas 3.5:

- **capa de Servidor:** se encarga de almacenar la información de conciencia de grupo referente a los recursos humanos, físicos y virtuales y de proveer a los colaboradores, de manera gradual, dicha información mediante filtros (rol e interés);
- **capa de Comunicación:** administra las conexiones de los clientes y provee los mecanismos para la distribución de la información de conciencia a los grupos de colaboradores.
- **capa de Cliente:** se encarga de presentar la información de conciencia de grupo en las diferentes interfaces de usuario.

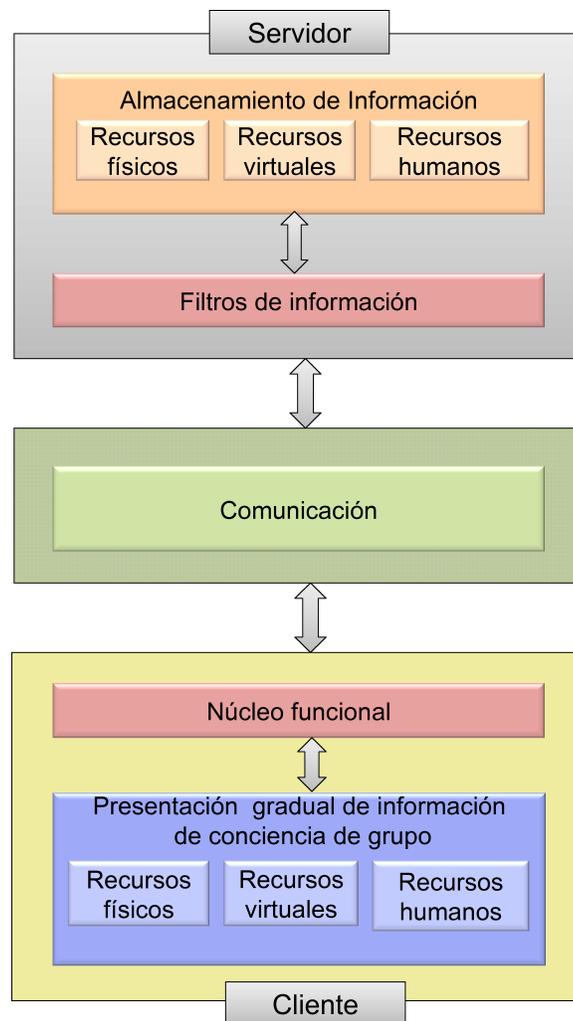


Figura 3.5: Arquitectura de *software* del sistema SIGCo

A continuación, se describe detalladamente cada una de las capas de la arquitectura de *software* del sistema SIGCo.

3.4.1 Capa de Servidor

Esta capa está formada por los componentes de **almacenamiento de información** y **filtro de información**, los cuales están encargados de procesar toda la información de conciencia de grupo (ver Figura 3.5). Los componentes de la capa de servidor se ilustran en la Figura 3.5

Almacenamiento de información

Este componente está formado por los módulos **recursos humanos**, **recursos físicos** y **recursos virtuales**, los cuales permiten administrar el almacenamiento de la información de los diferentes recursos de una organización.

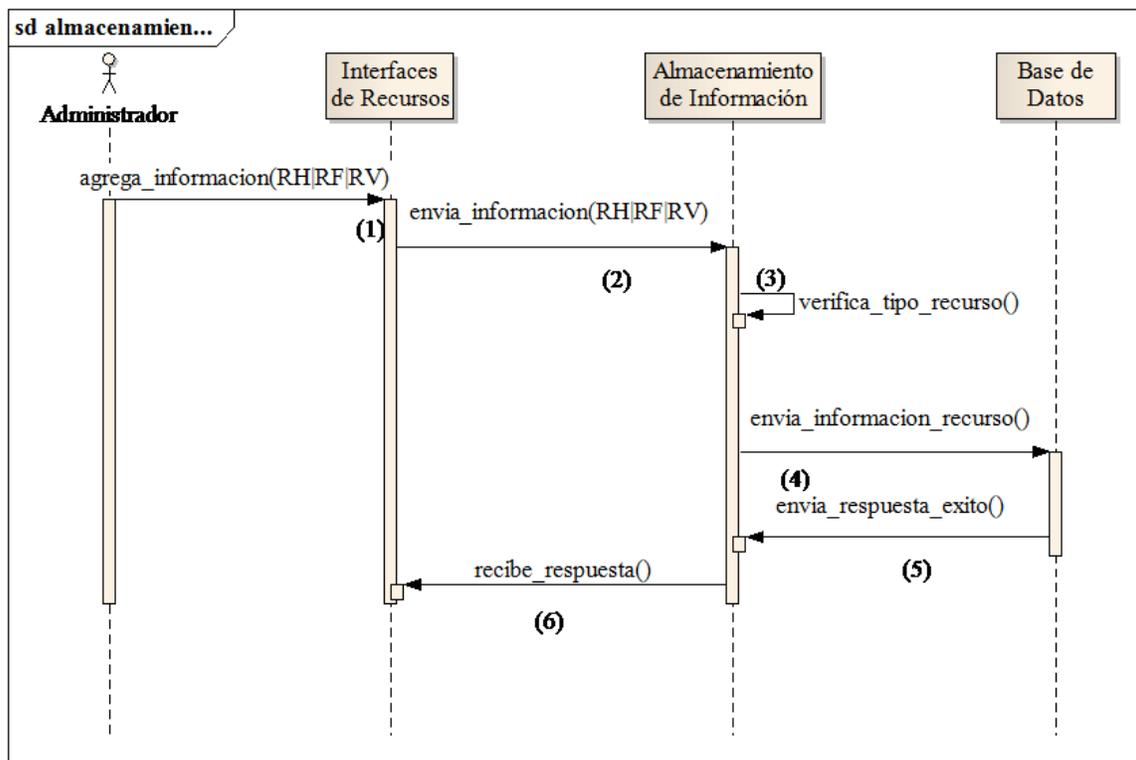


Figura 3.6: Diagrama de secuencia para almacenamiento de recursos

La Figura 3.6 muestra un diagrama de secuencia que indica las diferentes invocaciones de métodos que son llevadas a cabo para almacenar la información de los recursos humanos, físicos y virtuales que pueden existir en una organización. Un objeto de la clase **Interfaces de recursos** ofrece tres interfaces de usuarios que permiten al administrador **agregar información** de dichos tipos de recursos y almacenarla en un objeto de la clase **Base de Datos**.

Primeramente, el administrador **agrega_informacion** (RH|RF|RV) al sistema SIGCo; la información que puede ser ingresada se refiere a un recurso humano (RH), a un recurso

físico o a un recurso virtual (ver paso 1 de la Figura 3.6). Después esta información es enviada a un objeto de la clase **Almacenamiento de Información** (ver paso 2 de la Figura 3.6), el cual realiza el proceso de verificación de la información para saber que tipo de recurso va a ser almacenado (ver paso 3 de la Figura 3.6). Seguidamente, este objeto envía la información a un objeto de la clase **Base de Datos** para ser almacenada (ver paso 4 de la Figura 3.6). Este objeto envía un acuse al objeto de la clase **Almacenamiento de Información** (ver paso 5 de la Figura 3.6). Este objeto reenvía el acuse al objeto de la clase **Interfaces de Recursos**, el cual presenta el mensaje contenido en dicho acuse en la interfaz correspondiente (ver paso 6 de la Figura 3.6).

Filtros de información

Este componente se encarga de filtrar toda la información proveniente de los colaboradores. El filtrado es el proceso de seleccionar la información conciencia de grupo de acuerdo de interés de los colaboradores. Esta selección se realiza en dos formas:

- 1) Por **rol**: la selección está limitada de acuerdo al rol otorgado por la organización a cada colaborador. Un rol involucra un conjunto de privilegios y responsabilidades. El concepto de rol permite expresar la organización social del grupo de trabajo y resguardar los recursos físicos y virtuales de acciones no autorizadas.
- 2) Por **interés**: la información es publicada de acuerdo al criterio de cada colaborador (i.e., recibe sólo lo que solicita). Por lo tanto, la información no se duplica en los demás sitios de los colaboradores. El sistema SIGCo no necesita estar enviando información innecesaria a todos los clientes conectados, sino que el envío está regido por una petición particular de un cliente, mas no por el propio servidor.

A continuación, se describe el proceso de filtrado de información mediante un diagrama de secuencia.

Primeramente, el usuario **realiza una consulta** a un objeto de la clase **Consulta** del sistema SIGCo (ver paso 1 de la Figura 3.7). Después, este objeto envía la solicitud de información a un objeto de la clase **Roles** (ver paso 2 ver de la Figura 3.7), el cual a su vez solicita información sobre el usuario (usuario y contraseña) a un objeto de la clase **Base de Datos** (ver paso 3 de la Figura 3.7). Como resultado, este objeto le retorna la información solicitada (ver paso 4 de la Figura 3.7). El objeto de la clase **Roles** verifica la información del usuario (usuario y contraseña) (ver paso 5 de la Figura 3.7). Si la información es incorrecta, la solicitud del usuario es rechazada (ver paso 8b de la Figura 3.7). En caso contrario, el objeto de la clase **Roles** solicita el tratamiento de la consulta del usuario a un objeto de la clase **Interés** (ver paso 6 de la Figura 3.7). Este objeto verifica el tipo de información solicitada por el cliente (ver paso 7 de la Figura 3.7), la obtiene de la base de datos y finalmente la envía al objeto de la clase **Consulta** (ver paso 8a de la Figura 3.7).

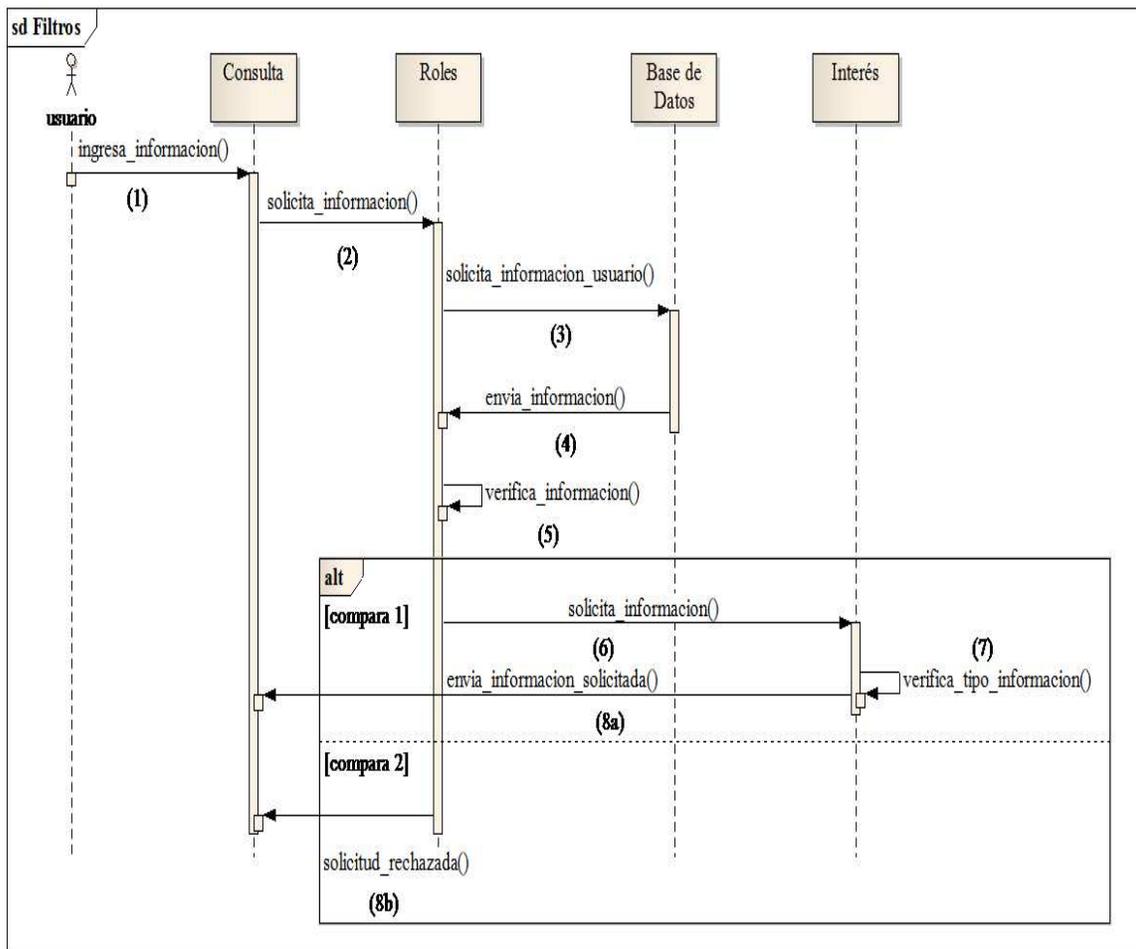


Figura 3.7: Diagrama de secuencia del filtro de información

3.4.2 Capa de comunicación

Esta capa utiliza dos formas de establecer la comunicación entre los clientes y el servidor. La primera forma utiliza una arquitectura *peer-to-peer* (P2P) [James and Keith, 2004] para permitir que los clientes puedan comunicarse entre sí cuando se crean grupos *ad-hoc* de colaboración. Después, está la segunda forma de comunicación, que emplea una arquitectura cliente/servidor para proveer a los colaboradores información gradual de conciencia de grupo. La arquitectura *peer-to-peer* (P2P) se basa en los protocolos TCP y UDP, en tanto que la arquitectura cliente/servidor utiliza únicamente el protocolo TCP.

La Figura 4.8 muestra los diferentes comandos que pueden ser utilizados por los clientes para comunicarse con el servidor o con otros clientes. En particular, los siguientes comandos permiten a un usuario iniciar una sesión (ver Figura 3.8(a)):

- **Ping:** es utilizado por el servidor para difundir (*broadcast*) un mensaje en una red de área local mediante el protocolo UDP.

- **Pong:** es empleado por un cliente en respuesta a un comando **Ping**.
- **Saludo:** es utilizado por un cliente para indicar al servidor que un colaborador está listo.
- **Login y Password:** son empleados por un cliente para autenticar un colaborador.

La Figura 3.8(b) muestra un grupo de comandos, cuyo objetivo es la obtención de información mediante peticiones al servidor:

- **Mensaje:** es utilizado por un cliente junto con el comando **Pong** para notificar al servidor la información sobre el *host*(puerto y dirección IP).
- **Rol:** es empleado por el servidor en respuesta al comando **Login y Password** de un cliente. Este comando contiene información sobre el rol de un miembro de la organización.
- **Paciente:** es utilizado por un cliente para solicitar al servidor la lista de pacientes.
- **Información:** es empleado por un cliente para solicitar al servidor información sobre un paciente específico (e.g., una imagen de un ultrasonido o una receta médica).

Los comandos **Grupos**, **Usuarios**, **Bloqueados**, **Indefinido** se utilizan para crear grupos *ad-hoc* y los comandos **Recursos**, **Estado**, **Información** para conocer las características de los recursos físicos de una organización (ver Figura 3.8(c) y 3.8(d)):

- **Recursos:** es utilizado por un cliente para solicitar al servidor los nombres de los recursos disponibles en una organización (e.g., equipo de ultrasonido o de rayos X).
- **Estado:** es empleado por un cliente para solicitar al servidor el estado de un recurso específico (e.g., libre o ocupado).
- **Información:** es utilizado por un cliente para solicitar al servidor las características de un recurso particular.
- **Grupos:** es utilizado por el servidor para enviar a los clientes involucrado el nombre de un grupo *ad-hoc*.
- **Usuarios:** es empleado por el servidor para enviar a los clientes la lista de usuarios asociados a un grupo.
- **Mensaje:** es utilizado tanto por el servidor como por un cliente para el intercambio de información (modo conversacional).
- **Bloqueados:** es empleado por el servidor para informar a un cliente el rechazo de un colaborador de formar parte de un grupo *ad-hoc* .
- **Indefinido:** es utilizado por el servidor para informar a un cliente que no está registrado en el servidor.

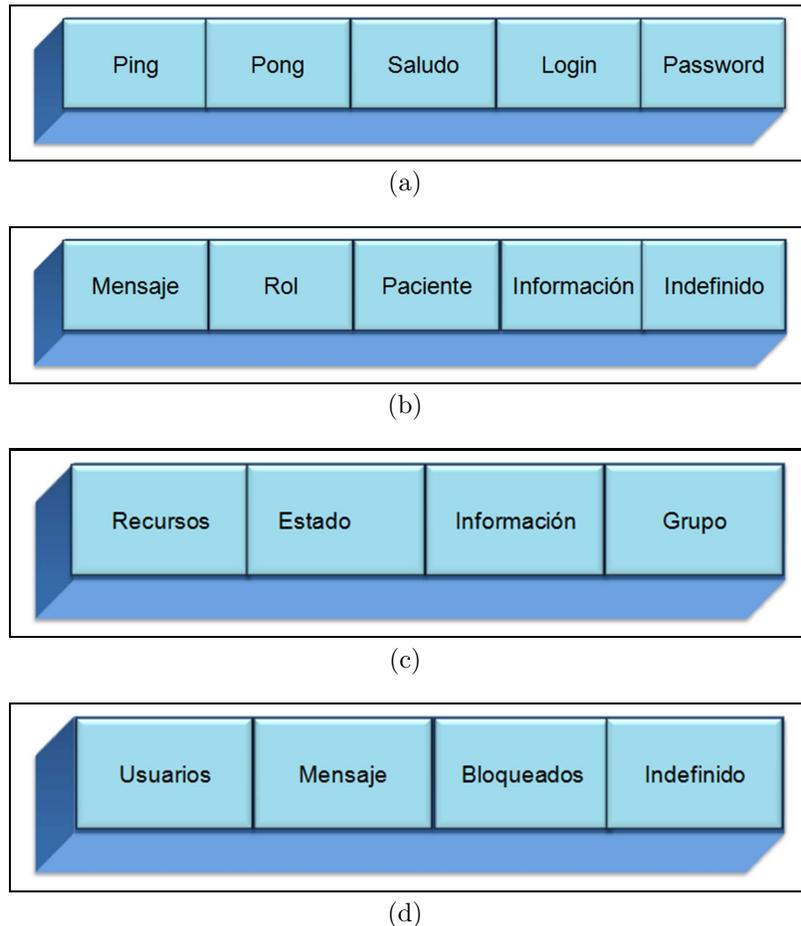


Figura 3.8: Protocolo de mensajes

Protocolos de comunicación

En este apartado se describen las diferentes arquitecturas de comunicación utilizadas en la capa de comunicación.

Como ya mencionó, el sistema SIGCo emplea las arquitecturas cliente/servidor y *peer to peer* (P2P), las cuales hacen uso de los protocolos TCP/IP y UDP. Un protocolo de comunicación es un conjunto de reglas normalizadas para la representación, señalización, autenticación y detección de errores. Este conjunto de reglas permite a los dispositivos de cómputo enviar información a través de un canal de comunicación.

El protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*) es un conjunto de protocolos a nivel de la capa de red, el cual es utilizado por Internet para la transferencia de datos entre redes de computadoras. Por su parte, el protocolo UDP (*User Datagram Protocol*) es un protocolo a nivel de la capa de transporte, el cual permite el envío de datagramas a través de la red, sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

A continuación se presenta un análisis comparativo entre UDP y TCP:

- **UDP:** proporciona un nivel de transporte no fiable de datagramas, ya que apenas añade la información necesaria para la comunicación extremo a extremo al paquete que envía al nivel inferior. Es utilizado por aplicaciones como NFS (*Network File System*) y RCP (comando para copiar archivos entre computadoras remotas), pero sobre todo se emplea en tareas de control y en la transmisión de audio y video a través de la red. No introduce retardos para establecer una conexión. Tampoco mantiene estado de conexión alguno ni realiza seguimiento de estos parámetros. Así, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de TCP.
- **TCP:** es el protocolo que proporciona un transporte fiable de flujo de bits entre aplicaciones. Está pensado para poder enviar grandes cantidades de información de forma fiable, liberando al programador de la dificultad de gestionar la fiabilidad de la conexión (retransmisiones, pérdida de paquetes, orden de llegada de los paquetes y duplicación de paquetes). Sin embargo, la complejidad de la gestión de la fiabilidad tiene un costo en eficiencia, ya que para llevar a cabo las gestiones anteriores se tiene que añadir bastante información a los paquetes a enviar. Debido a que los paquetes tienen un tamaño máximo, cuanto más información añade el protocolo para su gestión, menos información que proviene de la aplicación podrá contener ese paquete (el segmento TCP tiene una sobrecarga de 20 bytes en cada segmento, mientras que UDP solo añade 8 bytes).

En consecuencia, cuando es más importante la velocidad que la fiabilidad, se utiliza UDP. En cambio, TCP asegura la recepción en el destino de la información a transmitir [Stallings, 2004].

Arquitectura cliente/servidor

En el apartado anterior se describió el protocolo diseñado para llevar a cabo la comunicación entre las diferentes arquitecturas. En este apartado se describe la arquitectura cliente/servidor utilizada en el sistema SIGCo. Este tipo de arquitectura permite que dos aplicaciones, cliente y servidor, puedan comunicarse entre sí, e.g., un navegador Web implementa el lado cliente de HTTP, en tanto un servidor Web implementa el lado servidor de HTTP.

El sistema SIGCo está compuesto por una aplicación cliente y una aplicación servidor. Cada aplicación posee una clase de comunicación mediante la cual se establece el intercambio de mensajes. El servidor se encarga de atender las peticiones de los clientes y de entregar las respuestas correspondientes. Por otra parte, el cliente realiza peticiones de servicio mediante mensajes a la aplicación servidor, la cual recibe y contesta cada petición del cliente. Los mensajes que envía un cliente o un servidor se realizan mediante comandos, los cuales indican las acciones que debe realizar un cliente o un servidor (ver sección 3.2.2).

La Figura 3.9 muestra la manera en la que se lleva a cabo la comunicación entre un **cliente** y un **servidor** dentro del sistema SIGCo. El proceso de comunicación, tiene lugar cuando un usuario ingresa nombre de **usuario** y contraseña. El **cliente** envía al **servidor** los comandos *Login* y *Password*, los cuales indican que un usuario solicita autenticación para ingresar al sistema SIGCo. El **servidor** recibe el mensaje y verifica que el **usuario** sea válido. Si los datos del usuario son correctos, el **servidor** envía al **cliente** el *Rol* del usuario y la *lista de pacientes*. En caso contrario, envía el comando *undefine*, el cual indica que el **usuario** no existe. Una vez autenticado el **usuario**, el **cliente** envía un comando de *Saludo* indicando que está listo (*online*) para iniciar el intercambio de mensajes. Así, por ejemplo, si un colaborador necesita información de un paciente el **cliente** envía un comando *Información* al **servidor** solicitarla. El **servidor** recibe este comando y envía una consulta a la base de datos para solicitar dicha información. Finalmente, el **servidor** retorna el comando *Información*, así como la información solicitada por el **cliente**.

Por otra parte, un usuario puede estar interesado en conocer cierta característica de algún recurso físico. En este caso el **cliente** envía un comando *Informacion_recurso*, el cual indica el nombre del recurso y la característica solicitada. El **servidor** recibe el comando y realiza la consulta correspondiente a la base de datos. Después, el **servidor** vuelve a retornar el comando *Informacion_recurso*, al **cliente**. De igual manera, el usuario podría estar interesado en el estado de algún recurso (*libre* o *ocupado*). En este caso, el **cliente** envía los comandos *Estado* y *Recurso* al **servidor**, los cuales indican el nombre del recurso y el estado. En ese momento el **usuario** solicita un recurso virtual (e.g., una imagen de rayos X o un video de un ultrasonido) y envía el comando *Recurso* al **servidor**. En respuesta, el **servidor** verifica el comando, realiza la acción correspondiente y retorna al **cliente** el comando *Recurso* junto con la información del recurso virtual. Finalmente, el usuario abandona la sesión, por lo cual el **cliente** envía un *Mensaje de cierra sesión* al **servidor** para terminar la comunicación.

3.4.3 Capa de cliente

Esta capa se encarga de enviar peticiones a la capa de comunicación para solicitar información gradual de conciencia de grupo y presentarla en las interfaces de usuarios. La capa cliente está compuesta por las clases *Núcleo Funcional* y por la clase *Presentación Gradual* de conciencia de grupo.

A continuación, se describen las clases de la capa cliente mediante un diagramas de secuencia.

Clase Núcleo Funcional

Esta clase encarga de traducir todas las peticiones de los clientes en comandos que pueden ser interpretados por la clase *Capa de Comunicación* y viceversa (ver Figura 3.2). Para mostrar la funcionalidad de esta clase, a continuación se describe el proceso para llevar la interpretación de comandos (ver Figura 3.10).

Primeramente, el usuario ingresa sus datos en un objeto de la clase *Interfaz de*

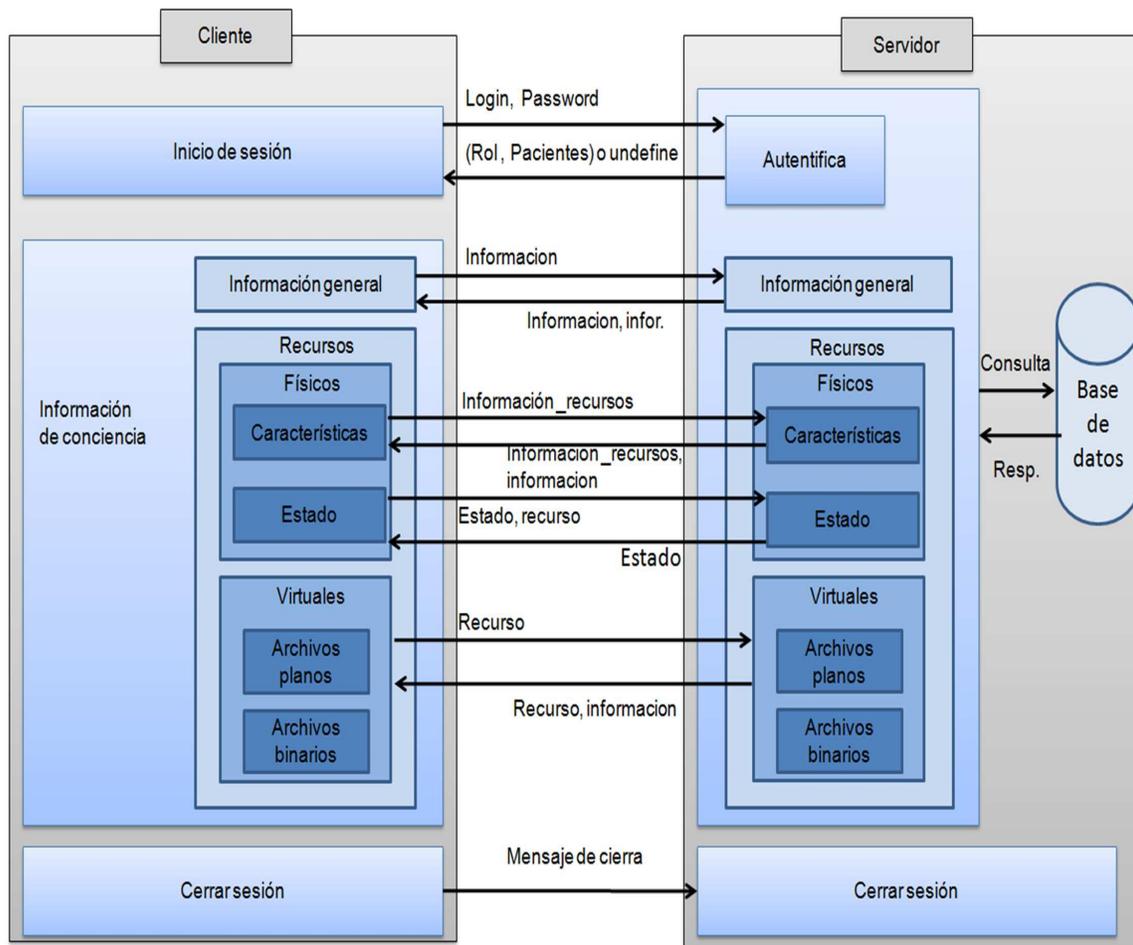


Figura 3.9: Intercambio de mensajes entre cliente y servidor en el sistema SIGCo

Usuario (ver paso 1 de la Figura 3.10). Este objeto envía estos datos a un objeto de la clase **Núcleo Funcional** (ver paso 2 de la Figura 3.10), el cual traduce dichos datos en comandos entendibles por un objeto de la clase **Capa de comunicación** (ver paso 3 de la Figura 3.10). El objeto de la clase **Núcleo Funcional** envía dichos comandos al objeto de la clase **Capa de Comunicación** (ver paso 4 de la Figura 3.10), el cual establece la comunicación con un objeto de la clase **Servidor** (ver paso 5 de la Figura 3.10). A continuación, el objeto de la clase **Capa de Comunicación** envía la petición del usuario al objeto de la clase **Servidor** (ver paso 6 de la Figura 3.10). Después, este último objeto obtiene la información solicitada de una base de datos y la envía junto con los comandos enviados por el objeto de la clase **Núcleo Funcional** (ver paso 4 de la Figura 3.10) al objeto de la clase **Capa de Comunicación** (ver paso 7 de la Figura 3.10). Este objeto envía la información solicitada por el cliente junto con el comando de petición al objeto de la clase **Núcleo Funcional** (ver paso 8 de la Figura 3.10), el cual se encarga de traducir los comandos en información que sea entendible por el objeto de la clase **Interfaz de Usuario** (ver paso 9 de la Figura 3.10). Finalmente, el objeto de la clase **Núcleo**

Funcional envía la información solicitada al objeto de la clase Interfaz de Usuario para que la presente (ver paso 10 de la Figura 3.10).

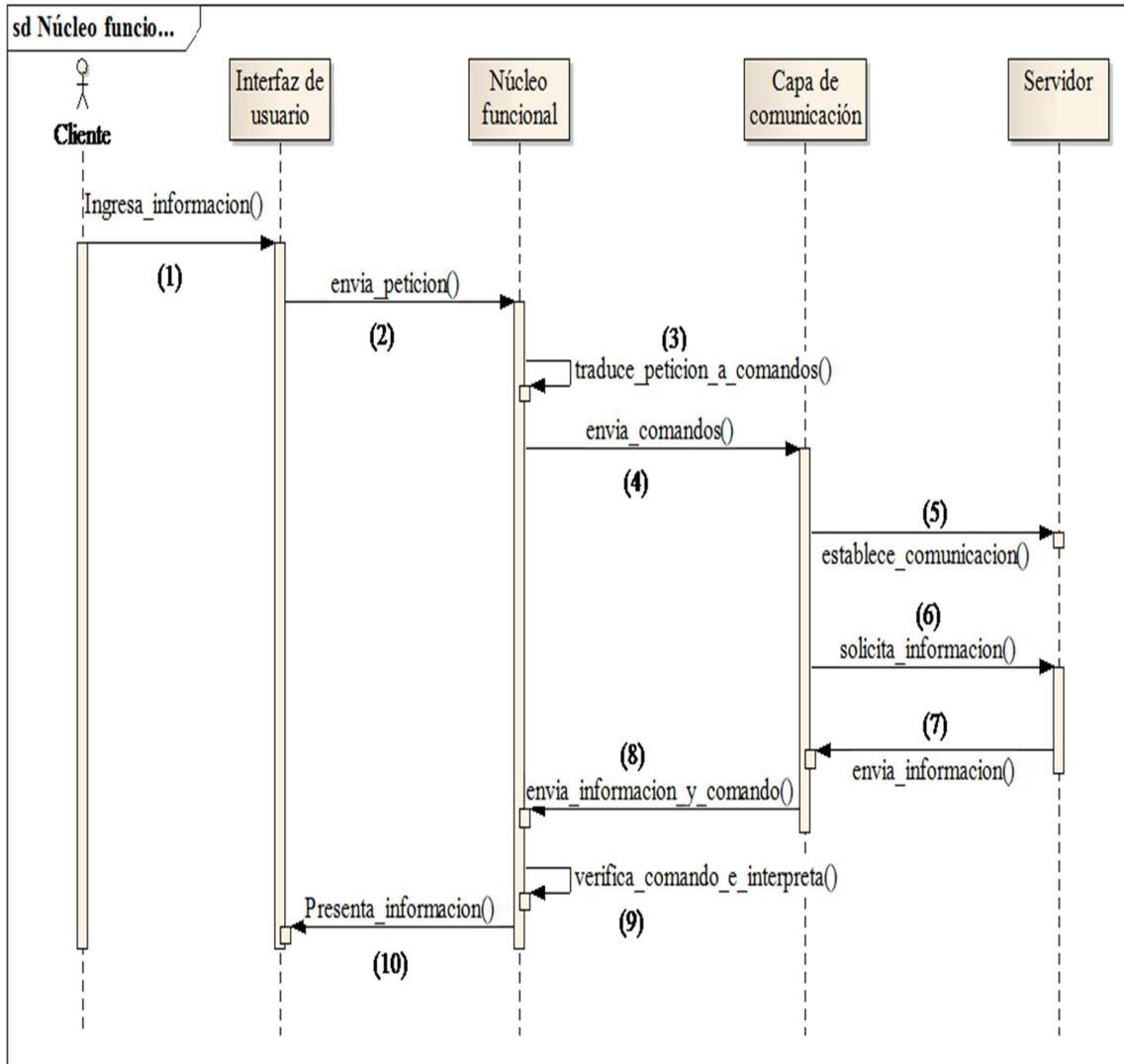


Figura 3.10: Diagrama de secuencias del núcleo funcional de SIGCo

Clase Presentación de Información

Esta clase se encarga de presentar la información gradual de conciencia de grupo en las diferentes interfaces de los colaboradores; esta clase requiere los servicios de las siguientes clases:

1. clase **Recursos Físicos**: presenta en la interfaz del usuario la información (e.g., las características de un recurso o si está libre u ocupado)

2. clase **Recursos Humanos**: se encarga de recibir la información (e.g., la lista de pacientes o el nombre de un paciente) proveniente del núcleo funcional y de presentarla en la interfaz de usuario;
3. clase **Recursos Virtuales**: se encarga de presentar la información (e.g., información demográfica o estudios clínicos de un paciente) en la interfaz del usuario.

Cada clase permite distinguir las diferentes peticiones que los clientes hacen al objeto de la clase `servidor`.

Existen dos tipos de interfaces de usuario principales. La primera interfaz permite la autenticación de usuario y la segunda muestra la interfaz de información de conciencia de grupo. A continuación se describe cada una de estas interfaces:

- Clase **Interfaz de Autenticación**: permite a un miembro de la organización ingresar sus datos para poder entrar al sistema SIGCo.

La Figura 3.11 muestra un diagrama de secuencias, el cual indica las diferentes etapas del proceso de autenticación de un usuario, así como el proceso de solicitud de información de conciencia que un cliente puede hacer.

Para iniciar el proceso de autenticación, el usuario ingresa su nombre y su clave en su **Interfaz de Autenticación** (ver paso 1 de la Figura 3.11); después estos datos son enviados a un objeto de la clase **Núcleo Funcional** (ver paso 2 de la Figura 3.11). Este objeto traduce la petición del usuario en comando que sea entendible por un objeto de la clase **Capa de Comunicación** (ver paso 3 de la Figura 3.11, dicho objeto establece la comunicación con el objeto de la clase **Servidor** (ver paso 4 de la Figura 3.11). Después, el objeto de la clase **Capa de Comunicación** envía la información del usuario al objeto de la clase **Servidor** (ver paso 5 de la Figura 3.11) para realizar el proceso de autenticación. Este objeto obtiene los datos del usuario de la base de datos y las envía al objeto de la clase **Capa de Comunicación**, el cual se encarga de traducirlos a comandos (ver paso 6 de la Figura 3.11). Estos comandos junto con la información del usuario son enviados por el objeto de la clase **Capa de Comunicación** al objeto de la clase **Núcleo Funcional** (ver paso 7 de la de la Figura 3.11). Este objeto verifica dicha información del usuario (ver paso 8 de la Figura 3.11). Si la información es incorrecta, indica que el usuario es desconocido, entonces el objeto de la clase **Núcleo Funcional** envía un mensaje de error al objeto de la clase **interfaz de autenticación** (ver paso 9b de la Figura 3.11). En caso contrario (ver paso 9a de la Figura 3.11), si el usuario es conocido se destruye el objeto de la clase **Interfaz de Autenticación** (ver paso 10 de la Figura 3.11) y se crea un objeto de la clase **Interfaz de Información**, el cual presenta la lista de los pacientes (ver paso 11 de la Figura 3.11) al usuario.

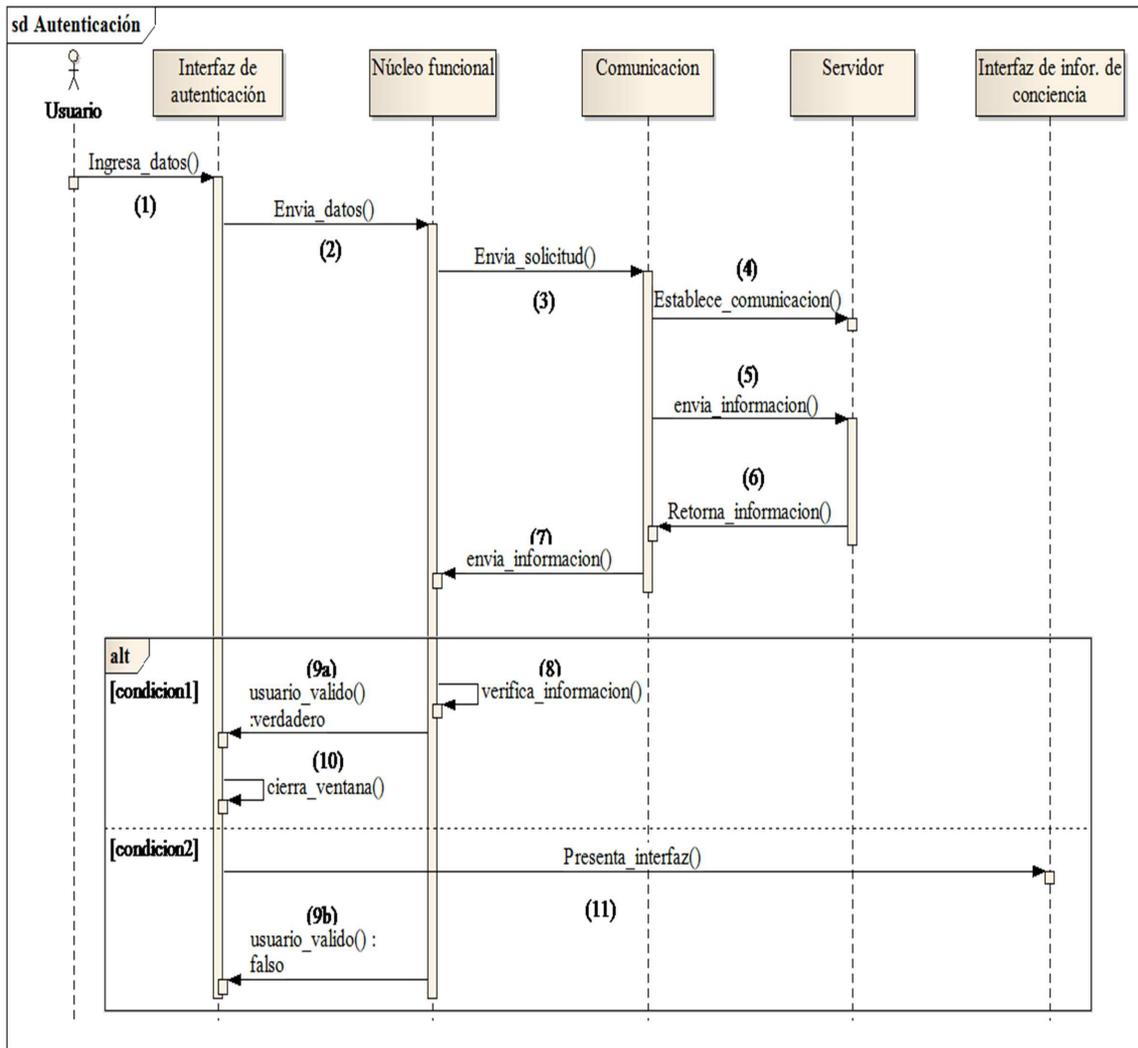


Figura 3.11: Proceso de autenticación

- Clase **Interfaz de Información de Conciencia**: permite visualizar información de los recursos humanos, virtuales y físicos de la organización, de acuerdo al interés del usuario. La información de conciencia está basada en los mecanismos *Focus* y *Nimbus* descritos en la sección 2.1.2 del capítulo 2. Estos mecanismos se ha considerado en el desarrollo del sistema SIGCo, empleando *widgets* (e.g., botones o paneles) que permiten al usuario solicitar información sobre los recursos de una organización y visualizarla de acuerdo a su interés. El usuario puede pasar de un nivel mínimo de información de conciencia a niveles intermedios hasta llegar a un nivel máximo de información de conciencia de grupo (ver Figura 3.12 y Figura 3.13).

La Figura 3.12 muestra un ejemplo de un diagrama que muestra seis cajas que corresponden a seis niveles de información de conciencia los cuales constituyen el conjunto universal permitida a un usuario. Cada nivel tiene su propio conjunto de

información, e.g., $N1 = \{n_1, n_2, n_3, n_4\}$, donde $N1$ representa un conjunto de información que podría referirse a los datos generales de un paciente y $n_1..n_4$ representa cada elemento de información, e.g., nombre, edad, fecha de nacimiento y CURP. Asimismo, estos niveles de información de conciencia permiten al usuario controlar la obtención de la información que desea visualizar de acuerdo a su propio interés.

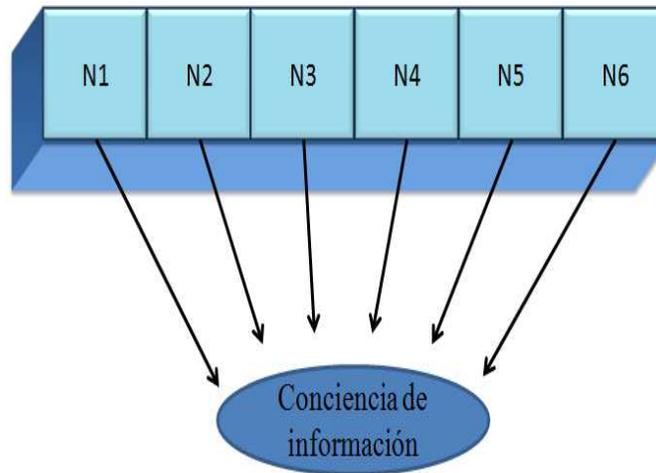


Figura 3.12: Niveles de conciencia que ofrece el sistema SIGCo

A continuación, se muestra un diagrama de secuencia que muestra el proceso de solicitud de información para un usuario.

Un usuario del sistema SIGCo puede controlar la información relevante por medio del control *Focus* (i.e., información relevante) y *Nimbus*. Este control permite que los usuarios puedan solicitar la información de conciencia de grupo conforme a su interés y visualizarla.

La Figura 3.13 muestra un diagrama que indica cómo el usuario adquiere un nivel de conciencia conforme explora la información. Suponga que un paciente llega al hospital con previa cita al área de Recepción, por lo que un empleado lo recibe y consulta su información en el sistema SIGCo. El empleado se autentifica mediante *password* y nombre de usuario. El sistema SIGCo dispone información específica de acuerdo al rol del usuario (i.e., no se muestra información que no le corresponde). Por lo tanto, un empleado de la organización (e.g., Recepcionista) puede navegar por la información y solicitarla de acuerdo a su conveniencia. En este caso, el recepcionista desea ver el historial de citas para saber si el paciente ha cumplido con todas las citas que le ha indicado el médico y darle algunas sugerencias. Por lo tanto, el recepcionista puede navegar del nivel uno (nombre del rol del empleado) hasta el nivel tres (cita actual) hasta llegar al nivel 4 (historial de citas).

La Figura 3.13 muestra los diferentes niveles de información a los que tiene acceso el recepcionista. Cabe mencionar que cada nivel de información de conciencia representa una petición al sistema SIGCo y un nivel de conciencia para el usuario, partiendo

de un primer nivel (nombre del rol de recepcionista) pasando por niveles intermedios de información de conciencia (Estado del paciente, Cita actual y Historial de citas) hasta llegar al máximo nivel permitido (Información general del paciente) de acuerdo a su rol.

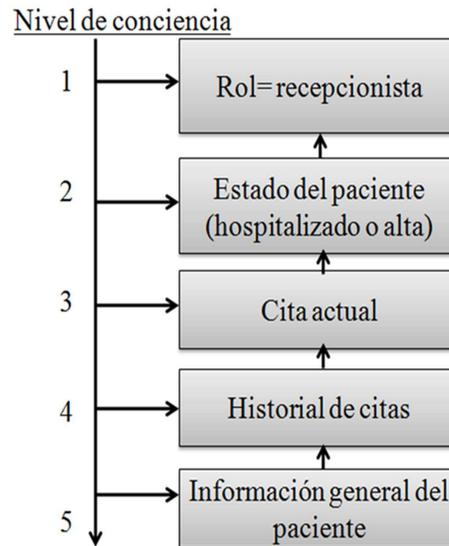


Figura 3.13: Ejemplo de conciencia de grupo en SIGCo

La Figura 3.14 muestra el proceso que un usuario puede realizar en la interfaz de información de conciencia para visualizar la información que el pueda requerir en cualquier momento.

Una vez que el usuario haya sido autenticado de manera exitosa se muestra la lista de todos los pacientes en la interfaz de usuario. Los pasos 1, 2 y 3 (ver Figura 3.14) muestran la secuencia del proceso de autenticación hasta presentar la lista de los pacientes en la interfaz de usuario. Después, un usuario selecciona un paciente (ver paso 5 de la Figura 3.14) e inmediatamente un objeto de la clase **Interfaz de Información de Conciencia** solicita la información al objeto de la clase **Núcleo Funcional** (ver paso 6 de la Figura 3.14) como ya se mencionó anteriormente. Este objeto traduce la solicitud del usuario en comandos entendibles por el objeto de la clase **Capa de Comunicación** (ver paso 7 de la Figura 3.14). Después, el objeto de la clase **Núcleo Funcional** envía dicha solicitud al objeto de la clase **Capa de Comunicación** (ver paso 8 de la Figura 3.14) el cual se encarga de establecer la comunicación con un objeto de la clase **Servidor** (ver paso 9 de la Figura 3.14). El objeto de la clase **Capa de Comunicación** solicita la información del paciente al objeto de la clase **Servidor**, la cual obtiene y la envía al objeto de la clase **Capa de Comunicación** (ver paso 10 de la Figura 3.14). Después, este objeto envía dicha información ya traducida en comandos al objeto de la clase **Núcleo Funcional** (ver paso 11 de la Figura 3.14). En respuesta, este objeto traduce dichos comandos en

información entendible por el usuario (ver paso 12 de la Figura 3.14) y envía dicha información al objeto de la clase Interfaz de Información de Conciencia para presentarla (ver paso 13 y 14 de la Figura 3.14).

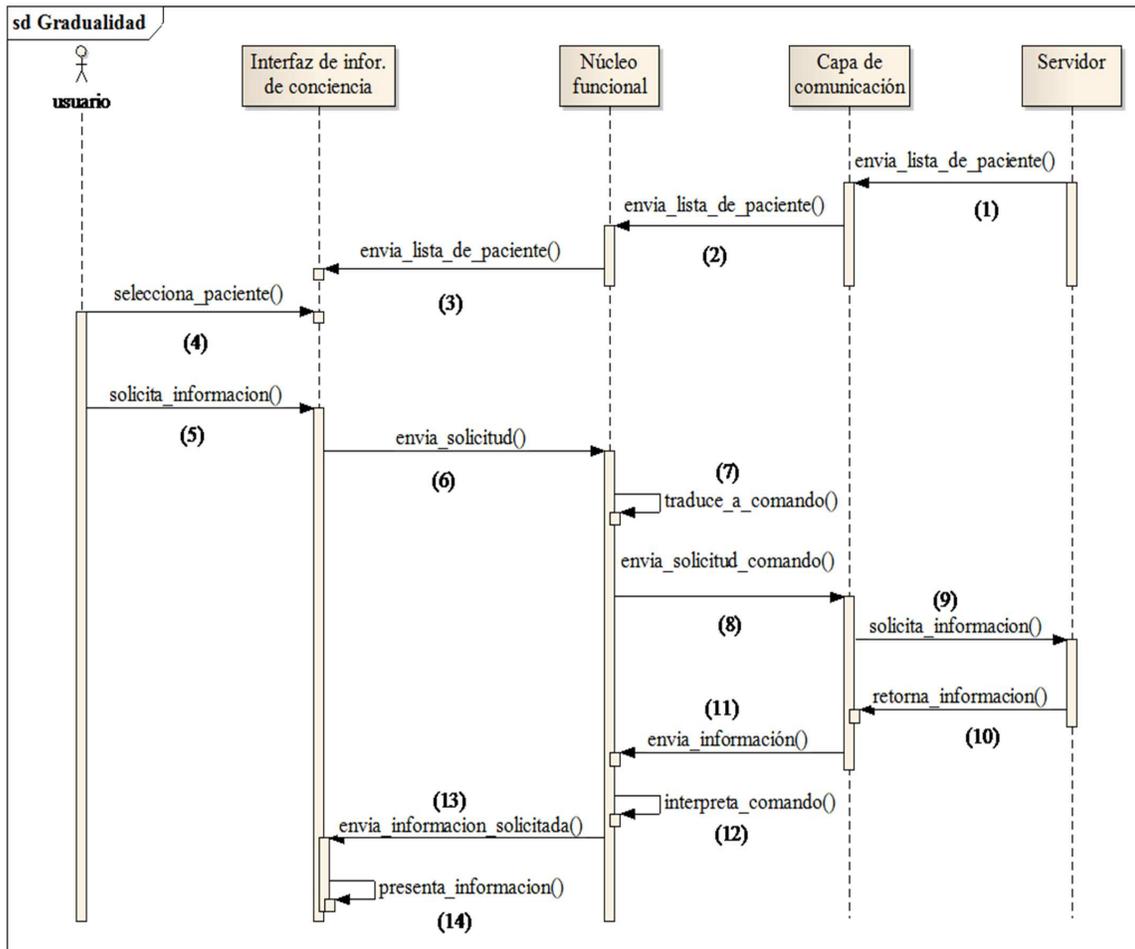


Figura 3.14: Proceso de solicitud de información en SIGCo

3.5 Arquitectura de distribución del sistema SIGCo

Un esquema de distribución especifica cómo los componentes definidos en una aplicación están distribuidos entre los sitios participantes. Se pueden diferenciar cinco tipos básicos de arquitecturas de distribución a partir de los cuales se pueden generar otros más [Roth and Unger, 2000]:

- **Arquitecturas con componentes centralizados:** tienen al menos un componente centralizado sobre el cual trabaja un grupo de colaboradores. En consecuencia, cada sitio participante interactúa únicamente con el componente compartido ubicado en el servidor.

- **Arquitecturas con comunicación directa (replicada):** no tienen componentes centralizados. Por el contrario, cada participante está conectado con todos los demás que forman parte del grupo, lo que resulta en una ejecución multi-sitio.
- **Arquitecturas híbridas:** tienen al menos un componente central y una comunicación directa entre los sitios participantes. Es la combinación de las arquitecturas de componentes centralizados y de las que tienen comunicación directa.
- **Arquitecturas asimétricas:** se generan con base en la combinación de la arquitectura centralizada, descentralizada e híbrida. Las arquitecturas asimétricas no tienen componentes centrales, sino que están distribuidos entre los sitios participantes de forma no simétrica, i.e. cada participante puede o no tener una réplica de algún componente. Una arquitectura asimétrica está integrada por varios sitios, los cuales se agrupan en pares. Así, cada par tiene características específicas de alguna de las arquitecturas de distribución (centralizada, replicada e híbrida) y dispone de sus propias entidades compartidas.
- **Múltiples servidores:** estas arquitecturas utilizan más de un servidor i.e., existen componentes centrales y distribuidos en varios sitios. La principal ventaja de esta arquitectura es que los usuarios pueden acceder rápidamente al servidor de una red local.

El sistema SIGCo utiliza una arquitectura híbrida, ya que contiene un componente centralizado y además hay una comunicación directa entre los colaboradores que forman parte de un grupo. Este tipo de arquitectura tiene un coordinador centralizado, el cual asegura el mantenimiento de la consistencia y la secuencia de los eventos provenientes de los sitios participantes.

La Figura 3.15 muestra dos clientes y el sistema SIGCo etiquetado como (administrador), así como los comandos mediante los que se comunica el cliente y el coordinador. Primeramente, se muestra una caja etiquetada como **información de conciencia** (ver lado cliente de la Figura 3.15), la cual representa la información que un **cliente** solicita al **servidor** mediante petición/respuesta. Después, aparece un componente central llamado **Coordinador**, que se encarga de notificar la **presencia** del **servidor** a los diversos **clientes**, crea los diversos grupos de colaboración y notifica a los colaboradores el fin de la sesión. A continuación se muestra la interacción del componente **Coordinador** entre los diferentes clientes (ver Figura 3.15).

El componente **coordinador** se encarga de notificar la presencia del **servidor** a los **clientes**, que por primera vez inician sesión. Esta notificación de la presencia se realiza enviando un comando **Ping** por toda la red, de tal forma que cuando un usuario inicia sesión recibe este comando. En respuesta, el **cliente** también notifica su presencia al coordinador enviando un comando **Pong**. Después, que el cliente conozca la dirección del **servidor** y viceversa (IP y Puerto), ahora el **cliente** puede realizar peticiones al **servidor**. Posteriormente, si un **cliente** desea establecer colaboración con otros colegas, primeramente se debe de crear el grupo, en este caso se crea un grupo llamado **GrupoA**. Seguidamente, de haber creado el grupo, el **Coordinador** envía un comando **Grupos** para

notificar *Grupos ah-hoc* a los miembros de dicho grupo. Posteriormente, los miembros del **grupoA** notifican al **Coordinador** mediante el comando **Mensaje** la unión al grupo o el rechazo. Si algún cliente acepta colaborar, entonces el **Coordinador** envía el nombre del grupo y la lista de los colaboradores. Apartir de los comando **Usuarios** y **Mensajes** se entablan las conversaciones entre los diferentes **clientes**. Finalmente, si un colaborador abandona la sesión el **Coordinador** notifica a todos los clientes de un grupo la salida de dicho colaborador mediante un comando **Mensaje** de **cierra sesión** (ver Figura 3.15).

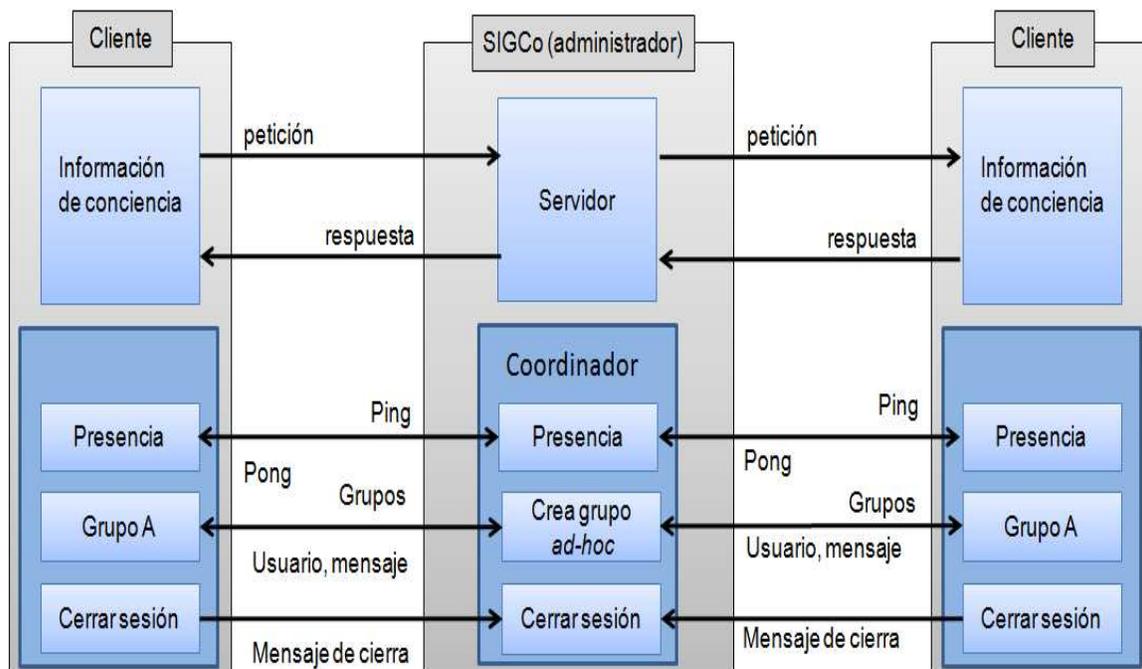


Figura 3.15: Comunicación entre clientes mediante un coordinador

El presente capítulo presentó el análisis y el diseño del sistema SIGCo. Dicho sistema inicio con un escenario, la cual puso en evidencia la necesidad de proveer información gradual de conciencia de grupo en un entorno hospitalario. Después, se describieron los casos de uso del sistema SIGCo, del cual surgieron las operaciones y los diferentes actores que intervienen en el sistema SIGCo. Seguidamente, se mostraron las clases del sistema SIGCo y luego se explicaron mediante diagramas de secuencia cada clase contenida en cada capa de la arquitectura del sistema SIGCo. Finalmente, se presentó la arquitectura de comunicación del sistema SIGCo mediante un diagrama de interacción. En el siguiente capítulo se pretende implementar cada una de las clases contenida en la arquitectura de *software*, así como probar la funcionalidad del sistema SIGCo.

Capítulo 4

Implementación del sistema SIGCo

En el capítulo anterior se describieron las capas que componen la arquitectura del sistema SIGCo y la manera en cómo interactúan dichas capas. Asimismo, se hace una descripción de los recursos humanos, físicos y virtuales, estos recursos forman la información de conciencia de grupo en el sistema SIGCo. Después, se mencionan las herramientas utilizadas para el desarrollo de dicho sistema y los principales algoritmos utilizados en su implementación. Posteriormente, se presentan las diferentes pruebas realizadas al sistema SIGCO. Finalmente, se mencionan las ventajas principales de dicho sistema.

4.1 Tipos de recursos

En esta sección se muestran ejemplos de los recursos humanos, físicos y virtuales que forman la información de conciencia de grupo del sistema SIGCo (ver capítulo 3 de la sección 3.2) mediante ejemplos que denotan las características de cada recurso. El sistema SIGCo tiene como objetivo proveer a los colaboradores de información gradual de conciencia de grupo, sobre los recursos previamente mencionados. Asimismo, cada recurso se administra desde un servidor central, que provee de información gradual a los colaboradores.

4.1.1 Recursos humanos

Los recursos humanos son el motor principal de toda organización, ya que se encargan de realizar sus actividades dentro de una organización, estos recursos pueden ser administrados por el departamento de recursos humanos, quienes se encargan de contratar y asignar los **roles** a los empleados o miembros de una organización. Por lo tanto, se ha considerado el **rol** de los miembros de la organización para definir el tipo de información al cual pueden acceder dichos miembros. También, se incluyó como parte de los recursos humanos a los clientes que llegan a una organización para solicitar algún servicio. Cabe mencionar que los clientes son el principal recurso activo de toda empresa. Dicho lo anterior, se vuelve necesario administrar la información de los recursos humanos de una organización para brindar a los miembros información de conciencia de grupo de manera oportuna y en cualquier lugar donde los clientes se encuentren (movilidad).

En el sistema SIGCo un administrador se encarga de agregar información sobre los miembros de una organización. La información sobre los recursos humanos son: a) el nombre del recurso humano, b) rol dentro de la organización, c) horario de entrada y salida del recurso humano y d) área de trabajo del recurso humano. En la Tabla 4.1 se muestran dos ejemplos que describen a dos miembros de una organización.

Tabla 4.1: Descripción de un empleados de un hospital

Nombre	Rol	Horario	Ubicación
Antonio Rodríguez Pérez	Cirujano	9:00 p.m. a 6:00 a.m	EC1
María Antonia Hernández Suárez	Ginecólogo	6:00 a.m. a 2:00 p.m	EC2

4.1.2 Recursos físicos

Los recursos físicos como impresoras, copiadoras, equipos de cómputo y equipos médicos son herramientas que ayudan a los miembros de una organización a realizar algunas actividades (e.g., procesar ordenes o imprimir algún reporte).

En la implementación del sistema SIGCo se consideró la información de dichos recursos para ofrecer a los miembros de una organización información gradual de conciencia grupo, que les permita aprovechar dichos recursos de manera eficiente. Además, estos recursos pueden ser compartidos con los miembros de una organización o de organizaciones externas, el cual pueden ayudar a que organizaciones externas con bajo capital económico puedan hacer uso de los equipos físicos que posee dicha organización. Por lo tanto, es necesario que sean administrado para ofrecer de manera oportuna información que sea de interés para los colaboradores.

Tabla 4.2: Descripciones de equipos de tomografías

Modelo	Tipos de escaneos	Velocidad de Escaneo	Estudios por día	Ubicación
HISPEED CT/I	Helicoidal y axial	Multicorte de 10 a 60 seg., y axial de 3 seg.	20	DT1
Gemini PEC-TAC CT/I	Helicoidal y esperial multicorte	Dos cortes (4 imágenes por seg.)	20	DT2

Se ha considerado las siguientes características de los recursos físicos: a) el modelo del equipo b) los tipos de escaneos, c) la velocidad de escaneo (e.g., helicoidal o espiral) y d) el número de estudios por día. Estas características permitirán que los miembros de una organización puedan seleccionar los equipos de acuerdo a su interés.

Tabla 4.3: Descripciones de equipos de ultrasonidos

Modelo	Aplicación	Color	Tipos de imágenes	Ubicación
GE LOGIQ-5 BT03	Abdomen, Gineco, Urología	SI	3D	DT1
DP-220	Obstetricia, Gineco, Abdominal, Partes pequeñas	no	-	DT2

La Tabla 4.2 muestra las principales características de un tomógrafo, las cuales son almacenadas en el sistema SIGCo. Por ejemplo, un dato importante es la velocidad de corte, la cual indica el tiempo con el que se toma la muestra de un paciente; entre más rápido sea la velocidad de corte más estudios por día se pueden realizar. Cabe mencionar que la toma de muestra de un paciente, ya sea en el torax o en la cabeza es un proceso lento. Otro, ejemplo de muestra en la Tabla 4.3 en las que se indican las características relevantes de los equipos de ultrasonidos como *el modelo del equipo, las aplicaciones, tipos de imágenes y ubicación de los equipos en la organización*. Una característica importante en la elección de dichos equipos se refiere a las **aplicaciones**, e.g., si pueden ser utilizados para realizar tomografías en **Abdomen, Ginecología o Urología**, de acuerdo a esta característica se pueden seleccionar algunos equipos de ultrasonido que cumplan con las exigencias de algún miembro de una organización.

4.1.3 Recursos virtuales

Al igual que los recursos físicos, los recursos virtuales pueden ser compartidos con los diferentes miembros de una organización. Los recursos virtuales pueden ser generados por los colaboradores o por el propio sistema (ver capítulo 3 sección 3.2), e.g., puede ser una receta médica o una imagen de rayos X. Cabe mencionar que la información compartida de dichos recursos, es de acuerdo a los roles otorgados a los miembros de una organización, e.g., un expediente clínico expedido por un especialista no puede ser visto por una enfermera.

La Tabla 4.4 muestra la información de una receta médica expedida por un médico, donde se muestra el tipo de información, que puede ser accedida, en algún momento por un miembro de la organización. Esta receta pueda ser generada después de una consulta general o intervención quirúrgica.

A continuación, se muestran varias tablas que describen recursos virtuales con características diferentes.

La Tabla 4.5 muestra los tipos de alergias que un paciente puede presentar ante un medicamento; esta información se obtiene mediante encuesta al paciente, al momento de entrar al hospital.

Tabla 4.4: Descripciones de recetas médicas

Medicamentos	Pacientes	Estatus
Cefalaxina Tomar una capsula c/8h, kentomicina tomar una c/12, metanzol sodico tomar 2 capsula c/8	Juanito	Falso
Soltrim tomar 3ml c/12 x 4dias, Espabion tomar 2.5 c/8hr x 7dias, Lacteol fort polvo tomar un sobre diluido en jugo de naranja c/12h x 5 dias	Marina	Verdadero

Tabla 4.5: Descripciones de alergías

Medicamentos	Latex	Medicamento Actual	Motivo	Última toma	Característica
Penicilína	si	Paracetamol	temperatura	2 días	comezón
rributan	no	ninguno	ninguno	ninguno	ninguno

La Tabla 4.6 muestra información de conciencia sobre los antecedentes quirúrgicos de un paciente, e.g., si el paciente ha sido intervenido quirúrgicamente en algún momento (operación de prostata o de mama).

Tabla 4.6: Descripciones de antecedentes quirúrgicos

Paciente	Tipo de cirugía	Cirujano	Hospital	Fecha
Julanito	Op. Vesicula	Mariano Pérez	Clínica del sureste	01-11-2007
Menganito	Apéndice	Miguel Ortíz	Médica especializada del sureste	15-06-2008

La Tabla 4.7 muestra información sobre la cita de un paciente; en esta tabla se aprecia el campo CURP que permite identificar a cada paciente dentro de la organización.

Tabla 4.7: Descripciones de Citas

Paciente	Médico	Fecha	CURP
Julanito	Rafael Zamora	28-09-2008	JUPE700115HCRY12
Menganito	Miguel Nieto	12-10-2008	MEPO601506HCRZ98

4.2 Herramientas utilizadas en el desarrollo de SIGCo

Las herramientas utilizadas para la implementación del sistema SIGCo son la **Base de datos** y el **Lenguaje de programación**, los cuales son parte fundamental en el desarrollo del sistema SIGCo. Las bases de datos son herramientas importante para almacenar información, además de que facilita la búsqueda de dicha información. El lenguaje utilizado es otra parte fundamental del desarrollo de cualquier aplicación y en nuestro caso del desarrollo del sistema SIGCo.

4.2.1 Base de datos

Hoy en día han surgido muchos manejadores de base de datos, tal como IBM DB2, Oracle, Sybase, SQL Server ente otros, pero que están restringidos por su capacidad y consumo de recursos. Por lo tanto, no pueden sastifacer las necesidades de los usuarios, especialmente cuando el número de usuarios se vuelve muy grande, mientras que las bases de datos empotradas tales como Mysql, Berkeley DB y SQLite muestran un excelente desempeño [Bi, 2009]. SQLite es una pequeña librería programada en lenguaje C que implementa un motor completo de base de datos multiplataforma, el cual no requiere configuración y se distribuye bajo licencia de dominio público. La ventaja fundamental de SQLite, es que permite utilizar el lenguaje estándar SQL en un amplio subconjunto de operaciones (e.g., inserción, borrado, actualización). SQLite destaca también por su versatilidad y combina el motor y la interfaz de usuario de la base de datos en una única biblioteca, y almacena los datos en un único archivo de texto plano. Esto hace que cada usuario pueda crear tantas bases de datos como desee, sin la necesidad de intervención de un administrador de bases de datos que gestione espacios de trabajo, usuarios y permisos de acceso. El hecho de almacenar toda la base de datos en un único archivo, facilita la portabilidad de los datos, y solamente tiene la restricción del espacio de disco asignado al usuario por parte del servidor.

En la implementación del sistema SIGCo se utilizó SQLite¹ debido a que no necesita ninguna configuración, ya que viene empotrada dentro de algún lenguaje de programación como Qt C++, Python y Ruby. En este caso se seleccionó debido a que soporta Qt C++ y el consumo de memoria es bajo. Además, su alta eficiencia en la búsqueda y el almacenamiento de la información la hacen ideal para nuestro desarrollo [Bi, 2009]. Todas las funcionalidades que provee una base de datos empotrada (e.g., inserción, borrado y modificación) son proporcionada por el propio lenguaje.

A continuación, se muestra un ejemplo de cómo crear una base de datos en SQLite:

1. Primero, se abre una ventana de línea de comandos en Windows.
2. Dirigirse al lugar donde está instalado SQLite.
3. Se escribe la siguiente línea `sqlite3 nombre_de_la_base_de_datosdb`, la cual indica la ejecución de SQLite y el nombre de la base de datos a crear. Para el ejemplo se

¹www.sqlite.org

escribe `sqlite conciencia.db` para crear la base de datos. Por lo tanto SQLite genera un archivo de texto con las estructuras correspondiente para cada tabla creada.

4. Después de haber creado la base de datos, se procede a crear una tabla. Para la creación se escribe en la línea de comandos de SQLite lo siguiente:

```
sqlite>
sqlite> CREATE TABLE Datos (...> id int(3) not null,
...> Apellido char(30) not null,
...> Nombre char(30) not null,
...> Documentos char(30) not null,
...> Observaciones chat(60) not null,
...> primary key (id));
sqlite>
```

SQLite genera un archivo de texto, el cual contiene las estructuras de las tablas creadas. El archivo generado se exporta al directorio donde estará la aplicación que lo utilizará.

4.2.2 Lenguaje de programación

La elección de un lenguaje para el desarrollo de aplicaciones es una tarea difícil y que en algunos casos esta elección se ve influenciada por la experiencia del programador en el manejo de un lenguaje específico mas no por su funcionalidades ni mucho menos por su rendimiento. Existe una gran variedad de lenguajes de programación tales como: Python, Java, Smalltalk, C y C++. Algunos de estos lenguajes proveen soporte para el desarrollo de aplicaciones en dispositivos móviles.

Para la implementación del sistema SIGCo se utilizó Qt C++ de Nokia, tanto para el sistema central SIGCo de la PC como para el sistema cliente de la PDA. Una de las razones principales por lo que se eligió Qt es por su interfaz de usuario *Drag y Drog*, ya que permite diseñar interfaces de manera fácil y rápida, así como por el modelo vista (MVC). Además, las aplicaciones pueden ser exportadas a otras plataformas (Linux y Mac) con sólo compilar el código.

Qt es un lenguaje que soporta las siguientes características:

- sistemas operativos como Windows, Mac y Linux;
- dispositivos móviles tales como Windows Mobiles, Symbian y Maemo;
- lenguajes como Java, Ruby, Visual Estudio, Python y C++.

El desarrollo del sistema SIGCo consiste de dos aplicaciones:

- La primera aplicación corresponde al sistema central SIGCo, la cual provee la información de conciencia a todos los miembros de una organización. Esta aplicación se desarrolló en una PC con sistema operativo *Windows Vista*, utilizando el lenguaje Qt C++ *Creator*.

- La segunda aplicación se refiere a la aplicación **cliente SIGCo** para una PDA (modelo HP iPAQ 610/610c). Esta PDA utiliza el sistema operativo *Windows Mobile* 6. Por lo tanto, para llevar a cabo la implementación de la aplicación cliente, se utilizó Qt C++ *Embedded* con Visual Studio Profesional 2007 y *Windows Mobile* 5.0 SDK (ver Apéndice A).

Qt *Creator* bajo la plataforma *Windows* puede ser descargado del sitio Web² de Qt de manera gratuita (GPL) y su instalación es de manera guiada paso a paso, así que no presenta mayores problemas en su instalación. Por otro lado, Qt *Embedded* presenta algunos detalles de instalación, para ello referirse al apéndice A para más detalle.

A continuación, se muestra un ejemplo de Qt, el cual puede ser probado en *Windows Vista* y *Windows Mobile*. Este código imprime un mensaje en una interfaz gráfica; el mensaje se presenta en la pantalla mediante un *widget QLabel*, el cual imprime la leyenda ¡Hola Mundo!; notese que en unas pocas líneas se puede desarrollar una interfaz gráfica.

```
#include <QtGui/QApplication>
#include <QtGui/QLabel>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QLabel label("¡Hola Mundo!");
    label.show();
    return app.exec();
}
```

4.2.3 Hardware

Para el desarrollo del sistema SIGCo se utilizó la siguiente infraestructura física:

1. PDA modelo HP, iPAQ 610/610c,
2. Computadora portátil y
3. PC de escritorio.

4.3 Implementación de las clases del sistema SIGCo

En secciones anteriores se mencionaron los diferentes tipos de información de conciencia de grupo que pueden ser administrados por el sistema SIGCo, así como la base de datos utilizada para almacenar cada tipo de recurso. En la implementación del sistema SIGCo se utilizó Qt, debido a que provee facilidad de diseñar interfaces de usuarios para PCs y para dispositivos móviles. Además, Qt implementa el modelo Vista-Controlador (MVC)

²qt.nokia.com

y diferentes vistas de gráficos. Además, no necesita ninguna máquina virtual para ejecutarse, debido a que genera código nativo para la plataforma que se desea utilizar (*Mac, Windows, Windows Mobile y Linux*).

En esta sección, se describen los algoritmos más importantes utilizados en la implementación del sistema SIGCo.

4.3.1 Núcleo funcional

En la sección 3.2.3 del capítulo 3, se describió la función principal de esta clase. El algoritmo implementado para esta clase es el algoritmo 1 de la clase **Núcleo Funcional**.

Este algoritmo realiza la interpretación de comandos que el objeto de la clase **Capa Servidor** envía al objeto de la clase **Capa Cliente**, que son interpretados por el objeto de la clase **Capa de Comunicación**.

En la línea 1 se muestra la variable i , la cual se utiliza como un conteo para el número máximo de roles que puede existir en una organización y k representa el número máximo de peticiones que un colaborador puede realizar. Las líneas 2 a 5 se encargan de verificar los mensajes remotos enviados por el objeto de la clase **Capa de Comunicación**, de segmentar el mensaje para poder eliminar los encabezados del protocolo tales como los comandos *Paciente, Rol, Información* y de enviar la información clara para un usuario al objeto de la clase **Información** para que presente la información en la **interfaz de Usuario**. La instrucción de la línea 6 significa que un colaborador solicita información al objeto de la capa **servidor**, pero para que se puedan enviar las peticiones, es necesario pasar por el proceso de conversión de comandos.

Primeramente, la línea 7 indica el proceso de verificación de los roles de una organización; la instrucción de la línea 8 realiza una comparación del rol del usuario con la lista de roles de la organización para verificar la autenticidad del usuario. Si el rol existe entonces se realiza una búsqueda para conocer qué tipo de consulta realiza el usuario.

Segundo, la línea 9 es el proceso para verificar tipos de consultas realizadas por un colaborador. Por consiguiente, las instrucciones de las líneas 10 a 18 se encargan de realizar una comparación entre la consulta del usuario local con los diferente tipos de consultas asignadas al rol del usuario. Si la consulta existe se agregan los encabezados correspondientes a la consulta para que el objeto de la clase **Capa de Comunicación** entienda el tipo consulta realizada por el usuario. Finalmente, las instrucciones de las líneas 19 a 26, se encargan de enviar los comandos al objeto de la clase **Capa de Comunicación**, cuando la consulta sea verdadera.

4.3.2 Interfaces

En esta sección se describen los algoritmos utilizados en las diferentes interfaces del usuario cliente mediante dos clases **Interfaz de Autenticación** e **Interfaz de Información de Conciencia**.

Algoritmo 1 Núcleo funcional

Entrada: *Nombre_Paciente*: nombre del paciente, *Rol_Usua*: función que desempeña el usuario en la organización

Consulta(alérgias o citas): tipo de información a consultar, *Tipos_Informacion*: 0..*Max*, donde *Max* representa el nivel máximo de conciencia que un usuario puede alcanzar, *Tipos_Informacion* es un conjunto de información de conciencia,

Lista: 1..*Max_Rol*, donde *Max_Rol* es el número máximo de roles que existe en una organización y *Lista* es un conjunto de roles

Salida: *Comandos = Paciente.Nom_pac, Rol.Rol_Usua, Infor.Consulta*)

```

1:  $i = 0, k = 1, coincide = falso$ 
2: si Mensaje_remoto == verdadero entonces
3:     segmenta mensaje
4:     elimina encabezado de protocolo (Paciente, Rol)
5:     envía información al componente de interfaz de usuario
6: si no
7:     mientras  $i < Max\_rols$  hacer
8:         si Rol_Usua == Lista[ $i$ ] entonces
9:             mientras  $k < Max$  hacer
10:                si consulta == Tipos_Informacion[ $k$ ] entonces
11:                     $k = Max$ 
12:                    comandos = concatena(Paciente.Nom_pac,
13:                    Rol.Rol_Usua, Infor.Consul)
14:                    coincide = verdadero
15:                si no
16:                     $k = k + 1$ 
17:                fin si
18:            fin mientras
19:        si no
20:             $i = i + 1$ 
21:        fin si
22:    si coincide == verdadero entonces
23:        envía comandos a la capa de comunicación
24:    si no
25:        /* envía un mensaje de error al usuario*/
26:    fin si
27: fin mientras
28: fin si

```

Comunicación entre cliente y servidor

Esta clase se encarga de establecer la comunicación inicial entre cliente y servidor. A continuación, se muestra el algoritmo de Comunicación entre Cliente y Servidor, que se encarga de realizar el proceso de comunicación.

La línea 1 muestra la instrucción *Server_envia_datagramas(IPServer, PortServer)*, la cual se encarga de enviar un mensaje de difusión (*broadcast*) a toda la red. Este mensaje, se utiliza para que todos los clientes conozcan la dirección del servidor central. La instrucción de la línea 2 es el proceso de enviar los datagramas cada 2000ms. Cuando un *cliente* se conecta, recibe el datagrama y contesta con otro datagrama (IP y puerto). El objeto de la clase **Servidor** captura los datagramas de los diferentes clientes; la instrucción de la línea 4 a la 12 verifican que la dirección IP y el puerto sean de un destinatario y no local. Si la dirección es de un remitente, se crea un objeto para un *cliente* y se guarda su información (*IP* y el *puerto*), para establecer la comunicación cliente/servidor. Después, el objeto de la clase **Servidor** envía el comando **Ping** al *cliente* para establecer la comunicación entre ellos. El objeto de la capa **Cliente** verifica el comando **Ping**; si el comando llegó, entonces dicho objeto envía el comando **Pong** al objeto de la capa **Servidor** para establecer la comunicación. Las instrucciones de las líneas 13 a la 19 se encargan de comparar los diferentes mensajes de los clientes. Si el objeto de la clase **Servidor** recibe el comando **Pong**, se establece una conexión TCP (cliente/servidor), la cual permite que los usuarios puedan consultar la información de conciencia de grupo almacenada en el servidor central.

Algoritmo 2 Comunicación entre Cliente y Servidor

Entrada: *IPServer*: Dirección del servidor, *PortServer*: Puerto del servidor

Salida: se establece la comunicación Cliente/Servidor

```
1: mientras verdadero hacer
2:   Server_envia_datagramas(IPServer, PortServer)
3:   Servidor_LeeDatagrama(IPRemitente, PortRemitente)
4:   si IPRemitente! = IPServer && PortRemitene! = PortServer entonces
5:     Crea_objeto_conexion()
6:     Guarda_direcciones(IPRemitente, PortRemitente)
7:     Servidor_envia(Ping)
8:     si Lee_cliente_Mensaje == Ping entonces
9:       Cliente_envia(Pong)
10:    si no
11:      /* es otro tipo mensaje*/
12:    fin si
13:    si Crea_objeto_conexion_lee_mensaje == Pong entonces
14:      /* Servidor TCP (petición/respuesta) está listo para recibir al cliente
15:      */
16:    si no
17:      /*otros comandos*/
18:    fin si
19:    esperar 2000ms
20: fin mientras
```

Autenticación del usuario

Esta clase se encarga de realizar el proceso de autenticación del usuario. El algoritmo 3 muestra una serie de instrucciones que indican las diferentes operaciones que se llevan a cabo durante este proceso.

Las instrucciones de las líneas 1 a 3 se refieren al objeto de la clase **Cliente**. Este objeto envía un mensaje con el nombre del *Usuario* y *Contraseña* y espera contestación del objeto de la clase **Capa Servidor**. Las instrucciones de las líneas 4 a la 13 pertenecen al objeto de la clase **Capa Servidor**, que se encarga de validar el nombre de *Usuario* y la *Contraseña* en la base de datos. Si el *Usuario* y *Contraseña* existen, entonces el objeto de la clase **Capa Servidor** envía el nombre del *Usuario*, *Rol*, *Lista de pacientes* y un valor *booleano* (falso o verdadero). Por otro lado, si el *Usuario* no existe en la base de datos, se envía un mensaje de falso al objeto de la clase **Capa cliente**, la cual indica que el *Usuario* no se encontró en el sistema SIGCo. Las instrucciones de las líneas 12 a la 16 se encargan verificar los diferentes mensajes enviado por el objeto de la clase **Capa Servidor**. Si el mensaje indica que el usuario existe, entonces se destruye el objeto de la clase **Interfaz de Autenticación de usuario** y se crea un objeto de la clase **Interfaz de Información de Conciencia**. Por último, si el mensaje indica *falso* significa que el usuario no es válido. Por lo tanto, se envía un mensaje de error a la interfaz de usuario.

Algoritmo 3 Autenticación

Entrada: *Usuario*: nombre del usuario, *Contrasena*: contraseña del usuario, *Login*: usuario remoto, *Password*: password remoto

Salida: *Autenticado = verdadero*

```
1: Autenticado = falso
2: Cliente_envia_mensaje(Usuario, Contrasena)
3: Cliente_espera_respuesta
4: ServidorTCP_recibe_mensaje(Login, Password)
5: si Login == usuario_base_datos && Password == password_base_datos entonces
6:     /*lado servidor TCP*/
7:     ServidorTCP_envia_mensaje(usuario, Rol, Lista_pacientes, Autenticado)
8: si no
9:     ServidorTCP_envia_mensaje(false, false, false)
10: fin si
11: si Autenticado! = false entonces
12:     /*lado cliente*/
13:     Cierra_interfaz_autenticacion
14:     Llama_interfaz_informacion
15: si no
16:     /*envía un mensaje de error*/
17: fin si
```

Interfaz de información de conciencia

Esta clase se encarga de presentar la información gradual de conciencia de grupo, de acuerdo a las peticiones realizadas por un usuario. El algoritmo de información de conciencia 4 indica el proceso de presentar la información en la interfaz del usuario, así como los diferentes métodos usados durante el proceso de presentación de la información.

La primera línea contiene tres variables usadas para el conteo de los tipos de información de conciencia permitidos a un usuario, así como para saber si la consulta es válida o no para un usuario. La instrucción de la línea 2 indica que un usuario ha sido autenticado (ver algoritmo 3), por lo cual el objeto de la clase **Interfaz de Información de Conciencia** recibe del objeto de la clase **Núcleo Funcional** la lista de pacientes y el rol del usuario, esta información son presentadas en la interfaz de información de conciencia de usuario. Las instrucciones de las líneas 3 a 23 realizan dos procesos. El primer proceso aparece en las líneas 5 a 17, el cual consiste en buscar el tipo de consulta que el usuario ha realizado. Si la consulta existe, entonces se envía al objeto de la clase **Núcleo Funcional** dicha consulta. Este objeto realiza las operaciones de conversión de comandos y los envía al objeto de la clase **Capa de Comunicación**. Después, este objeto obtiene la información y la envía al objeto de la clase **Núcleo Funcional** y posteriormente la presenta en la interfaz de usuario. El segundo proceso consiste en seguir solicitando información de conciencia repetidamente hasta que el usuario interrumpa las solicitudes. Esta solicitudes abarcan las líneas 3 a 23. En caso de no existir la consulta se envía un mensaje de error al usuario.

Grupos *ad-hoc*

Esta clase se encarga de crear grupos y agregar usuarios a dichos grupos, así como de difundir mensajes a los grupos de colaboradores. A continuación, se describe el algoritmo 5 que lleva a cabo este proceso de colaboración.

La primera instrucción de la línea 1 indica una variable *ConGrupo* que lleva el conteo de grupos creados, *ConCola* lleva el conteo del número de colaboradores que puede haber en un grupo dado. La instrucción de la línea 2 es un método **Colaborador_creaGrupo** de la clase **Grupos Ad-hoc** utilizado para crear un grupo. Por lo tanto, para llevar a cabo la creación de un grupo, es necesario pasar por dos procesos. El primer proceso abarca las líneas 4 a 11 y consiste en buscar la existencia del grupo que se desea crear en el arreglo *Lista_Grupos*, para verificar que no exista un grupo con el mismo nombre. Si el grupo no existe, entonces se guarda el nombre del grupo en el arreglo *Lista_Grupo*, pero en caso de existir envía un mensaje de grupo existente. El segundo proceso aparece en las líneas 12 a 20, el cual consiste en agregar usuarios al grupo creado en un arreglo *Lista_Colaboradores*, el cual es necesario que el usuario seleccione a los colaboradores de la *Lista_Colaboradores*, que él considere que pueden colaborar sobre algún diagnóstico. Las instrucciones de las líneas 21 a 28 se refieren a la difusión de mensajes a todos los usuarios que han sido seleccionados o agregados al grupo creado. Este mensaje, pregunta al usuario si desea aceptar o rechazar la invitación a formar grupo. Después, los colaboradores notifican su aceptación o rechazo al grupo; si el mensaje es de aceptación, entonces el objeto de la clase **Capa Servidor** envía un mensaje a los colaboradores para notificar que están listos para

Algoritmo 4 Información de conciencia

Entrada: *Tipos_informacion* : 1..*Max*, donde *Max* es el número máximo de consultas que un usuario puede realizar y *Tipos_informacion* representa un conjunto de información de conciencia

Salida: /*visualización de la información consultada*/

```

1: i = 0, consulta = falso, Informacion_conciencia = 0
2: recibe del objeto de la clase Núcleo funcional Lista_paciente y Rol
3: mientras verdadero hacer
4:   Consulta_usuario /*consulta realizada por el usuario*/
5:   mientras i < Max hacer
6:     si Tipos_informacion[i] == Consulta_usuario entonces
7:       Envia_consulta_nucleo_funcional(Consulta_usuarios)
8:       Nucleo_funcional_envia_capa_comunicacion(comandos)
9:       espera_respuesta
10:      Nucleo_funcional_recibe_consulta(Informacion_conciencia)
11:      Envia_consulta_interfaz_informacion
12:      i = Max
13:      Consulta = verdadero
14:     si no
15:       i = i + 1
16:     fin si
17:   fin mientras
18:   si Consulta == verdadero entonces
19:     i = 0, Consulta = falso, Consulta_usuario = 0
20:   si no
21:     /*Envía un mensaje de error al usuario*/
22:   fin si
23: fin mientras

```

colaborar. Por otra parte, si el colaborador no decidió participar se elimina del arreglo *Lista_nueva*. Finalmente, si los colaboradores han llegado a algún objetivo, se borra el grupo del arreglo *Lista_Grupos*.

4.4 Pruebas y Resultados del sistema SIGCo

Como ya se mencionó previamente, el sistema SIGCo consiste de dos aplicaciones para: PC y PDA.

La aplicación para PDA consiste en una aplicación cliente, que permite a los usuarios consultar y visualizar información de conciencia de grupo de acuerdo a su interés. La aplicación para PCs consiste en un servidor central, el cual se encarga de proveer la información necesaria a los clientes y administrar los diferentes grupos *ad-hoc* para que puedan establecer la colaboración entre los miembros de una organización.

Algoritmo 5 Grupos *ad-hoc*

Entrada: *Lista_Colaboradores* : 1..*NumMax*, donde *Lista_Colaboradores* es el conjunto de usuarios conectados a SIGCo y *NumMax* es el número de usuarios conectados, *Lista_Grupos* : 1..*NumGrupo*, donde *Lista_Grupo* es el conjunto de grupos creados y *NumGrupo* es el número de grupos creados

Salida: /*grupos creados y listos para colaborar*/

```
1: ConGrupo = 0, ConCola = 0, grupo = falso
2: Colaborador_crea_grupo(Nombre_grupo)
3: mientras ConCola < NumMax hacer
4:     mientras ConGrupo < NumGrupo hacer
5:         si Nombre_Grupo! = Lista_Grupos[ConGrupo] entonces
6:             Lista_Grupos[ConGrupo] = Nombre_Grupo /*guarda grupo creado*/
7:             grupo = verdadero
8:         si no
9:             ConGrupo = ConGrupo + 1
10:        fin si
11:    fin mientras
12:    ConGrupo = 0
13:    si grupo == verdadero entonces
14:        colaborador selecciona de Lista_Colaboradores[ConCola] y guarda en una
        Lista_nueva a colegas para participar en el grupo
15:    si no
16:        /*el grupo ya existe*/
17:    fin si
18:    grupo = falso
19:    ConCola = ConCola + 1
20: fin mientras
21: ConCola = 0
22: Envia_mensaje_Colaboradores(Lista_nueva) /* envía mensaje a todos los colabo-
        radores mediante agregados al grupo*/
23: Colaboradores_envian_mensaje_alCoordinador(Mensaje)
24: si Mensaje! = verdadero entonces
25:     se elimina al usuario de la Lista_nueva
26: si no
27:     /*usuarios listos para colaborar*/
28: fin si
    /* si todos los colaboradores han logrado sus objetivos se borra el grupo creado*/
```

4.4.1 Interfaz de usuario para la administración del sistema SIGCo

El sistema SIGCo central está encargado de administrar toda la información de la organización (e.g., agregar empleados o algún recurso). Dicho sistema cuenta con cinco

pestañas. Cada pestaña corresponde a una interfaz distinta para cada recurso y una pestaña para administrar los grupos *ad-hoc* (ver Figura 4.1).

A continuación, se describen cada una de las pestañas previamente mencionadas.

Pestaña de recursos humanos

En esta pestaña se permite agregar información de los miembros de una organización tal como su nombre, rol, ubicación dentro de la organización, password y su horario de trabajo, así como listar a los usuarios (empleados) que han sido agregados previamente. Cada miembro de una organización es agregados al sistema SIGCo para que pueda consultar la información de conciencia de grupo mediante el sistema cliente de la PDA.

La Figura (ver Figura 4.1) muestra una caja con la leyenda “Altas de usuarios” que se utiliza para visualizar los campos que van a ser llenados por un miembro de la organización, así también aparece otra caja con la leyenda “Agregar usuario”. Por otra parte, aparece otra caja titulada “Lista usuarios”, la cual se utiliza para mostrar a los miembros de una organización que han sido agregados al sistema.

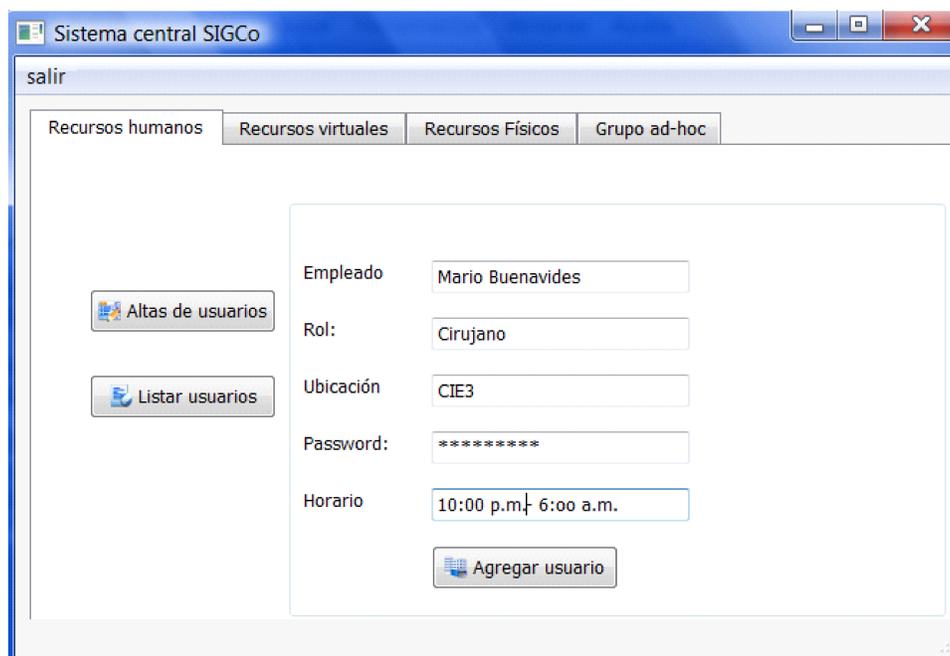


Figura 4.1: Pestaña para escribir y visualizar los recursos humanos en SIGCo

Pestaña de recursos virtuales en SIGCo

Esta pestaña permite agregar información sobre los pacientes que ingresan por primera vez al hospital. En esta misma pestaña existe seis *widgets* y cada uno consisten de una interfaz de usuario (ver Figura 4.2).

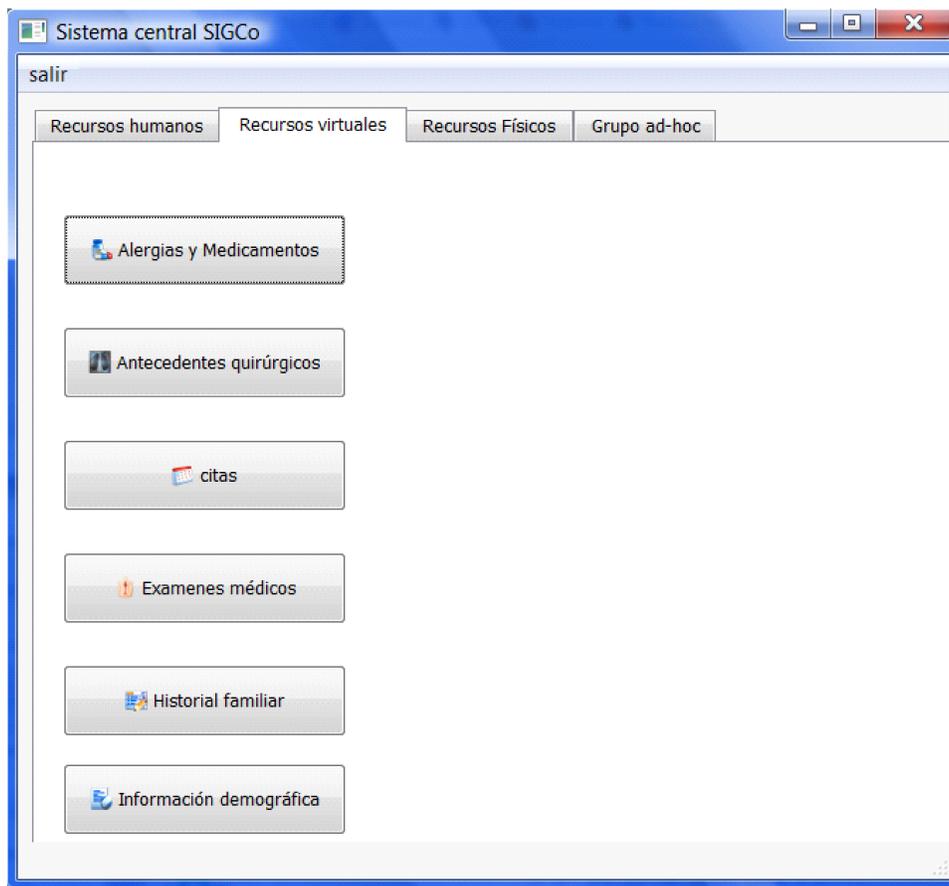


Figura 4.2: Pestaña de recursos virtuales para administrar la información de un paciente

A continuación, se describen cada una de las interfaces de usuario que permiten agregar información de los pacientes de un hospital.

1. **Alérgias y Medicamentos:** permite agregar información sobre aquellos pacientes que son sensibles a ciertos tipos de alimentos (e.g., los mariscos) y a medicamentos (e.g., antibióticos).
2. **Antecedentes quirúrgicos:** permite agregar información sobre intervenciones quirúrgicas de un paciente tales como las cirugías.
3. **Exámenes médicos:** permite agregar información sobre estudios clínicos tal como examen de glucosa y biometría hemática.
4. **Historial familiar:** permite agregar información sobre los antecedentes de enfermedades (e.g., tifoidea y sarampión) de los familiares más cercanos de un paciente (e.g., padre o primos).
5. **Información demográfica:** permite agregar información general de un paciente (e.g., números de hijos, conyugue y vivienda propia).

6. **Citas:** permite agregar información sobre el día, la hora y el nombre del médico que atenderá un paciente.

La Figura 4.3 muestra la interfaz de usuario que permite agregar información sobre “alérgias y medicamentos”. En dicha interfaz de usuario se ingresan los diferentes medicamentos que pueden ocasionar algunas alérgias al paciente, e.g., yodo/tinte/mariscos que ocasionan algún tipo de alérgias al paciente.

En la interfaz “Alérgias y Medicamentos” se muestra tres cuadros de texto que indican, primeramente el medicamento actual de un paciente, así como el motivo por el cual prescribieron el medicamento y la fecha, duración y última toma de este medicamento. Después, aparece una caja con una leyenda “Guardar” que se utiliza para almacenar dicha información en la base de datos.

Alérgias y Medicamentos

Id del paciente:

¿Tienes alguna alergia o sensibilidad?

Medicamentos: Menciona:

Yodo/tintes/mariscos: Menciona:

Latex: Menciona:

Anota los medicamentos que actualmente estás tomando. Escribe el nombre, el motivo por el cual lo estás tomando y desde hace cuanto tiempo:

Medicamento actual Motivo Fecha/Duración/última toma

Guardar

Figura 4.3: Interfaz de usuario para describir las alérgias y medicamentos de un paciente

La Figura 4.4 muestra la interfaz de usuario “Antecedentes Quirúrgicos”. En esta interfaz se ingresa el nombre del paciente, la fecha de la intervención quirúrgica, el tipo de cirugía, el nombre de los cirujanos y el nombre de la organización donde se hizo la cirugía. Cada información proporcionada en esta interfaz puede ser conocida por los miembros de una organización.

Figura 4.4: Interfaz de usuario para ingresar los antecedentes quirúrgicos de un paciente

Pestaña de recursos físicos

Esta pestaña permite agregar los diferentes equipos que tiene una organización, así como sus características. La Figura 4.5 muestra la interfaz de usuario correspondiente, la cual permite que se pueda ingresar el modelo del equipo, el tipo de aplicación (e.g., abdomen o urología), si es a color o no, los tipos de imágenes que procesa (2d, 3d y 4d) y la ubicación dentro de la organización.

Figura 4.5: Interfaz para agregar información sobre recursos físicos

Pestaña de grupos *ad-hoc*

Esta pestaña permite crear grupos *ad-hoc*, agregar usuarios a la lista de grupos, ver la lista de los miembros activos y los empleados de la organización. Además, permite enviar un mensaje a todos los miembros que pueden ser parte de un grupo, ya sea para su aceptación o rechazo para colaborar dentro del grupo. La Figura 4.6 muestra la interfaz de usuario llamada “Grupo *ad-hoc*”, el cual muestra los diferentes elementos que proveen información de conciencia tal como el nombre del grupo y la lista de los usuarios, así como de los elementos que permiten crear grupos y agregar miembros a la lista de grupos. Cabe mencionar que en un grupo se agregan los usuarios que son miembros de la organización, estos usuarios no necesitan estar activos para ser agregados. Además, dicha interfaz de usuario provee una caja de texto para enviar un mensaje a todos los miembros activos, así también se muestra una caja de texto donde se encuentran los diferentes mensajes que envían los colaboradores cuando hay alguna conversación.

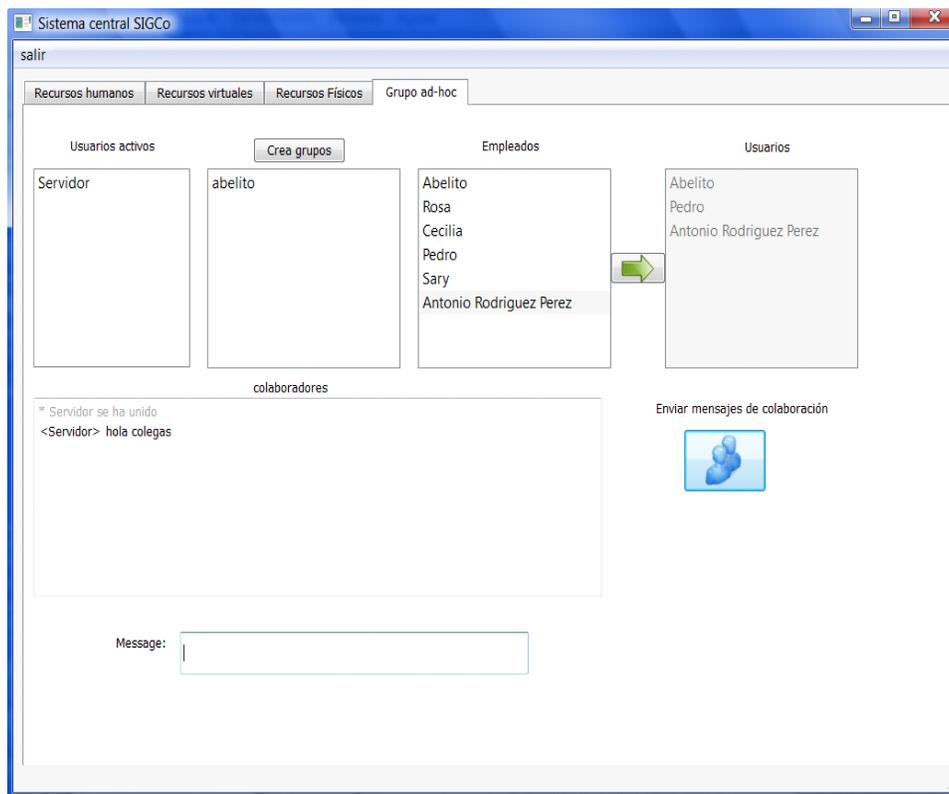


Figura 4.6: Pestaña para administrar los grupos *ad-hoc*

4.4.2 Interfaces de usuario del cliente SIGCo

Las interfaces clientes en el sistema SIGCo proporcionan información de conciencia a los miembros de una organización. Este sistema cuenta cinco interfaces, las cuales se describen en esta sección.

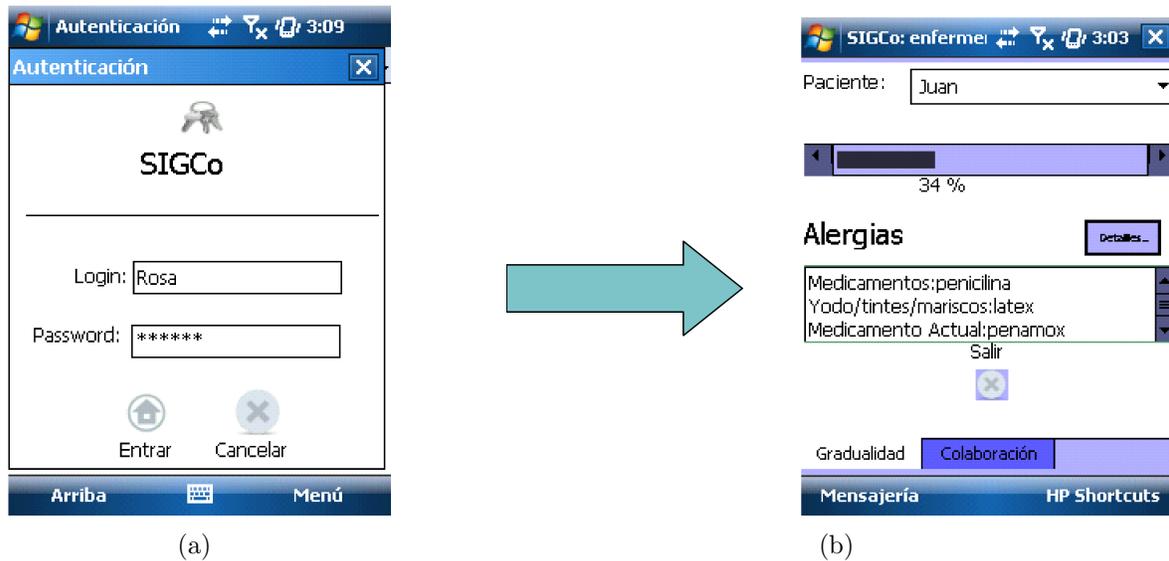


Figura 4.7: Interfaz de usuario de autenticación a interfaz de usuario de información de conciencia de grupo (PDA)

La primera interfaz del usuario es la “interfaz de usuario de autenticación”, la cual permite a un miembro de la organización ingresar sus datos correspondiente para acceder al sistema SIGCo. Esta información corresponde al nombre de usuario y a su contraseña de un miembro de la organización, que permite validar a dicho usuario al sistema SIGCo.

La “interfaz de usuario de autenticación” del usuario fue desarrollada en una PDA de tal manera que un colaborador puede autenticarse en cualquier punto de un edificio y solicitar información gradual de conciencia de grupo en cualquier momento que sea necesario.

Para hacer la prueba de la interfaz de usuario, un miembro de la organización ingresa sus datos personales (*Login y Password*) para entrar al sistema SIGCo. Estos datos son enviados al sistema central SIGCo para validar al usuario. En respuesta a la validación del usuario, el sistema decide mostrar la “interfaz de información de conciencia” o mantener la misma “interfaz de usuario de autenticación”, de acuerdo a la validez de la información del usuario. Si los datos son correctos, entonces se muestra la “interfaz de usuario de información de conciencia”, la cual indica la validez de un miembro de la organización (ver Figura 4.7(b)). En caso contrario el sistema envía un mensaje de error al usuario, indicando que los datos proporcionados son inválidos.

La “interfaz de usuario de información de conciencia” consta de dos pestañas, la primera pestaña se titula “Gradualidad” y la segunda “Colaboración”. La primera pestaña muestra una lista de pacientes mediante un *widget* (*ComboBox*). Este *widget* permite que un miembro de la organización pueda seleccionar un paciente para solicitar más información al sistema SIGCo, de acuerdo a su interés (ver Figura 4.8(a)). También, muestra un control Nimbus/Focus mediante un *widget* (*ProgressBar*) para que un miembro de la organización pueda gradualmente solicitar y ver los datos sobre un paciente (ver Figura 4.8(a)).

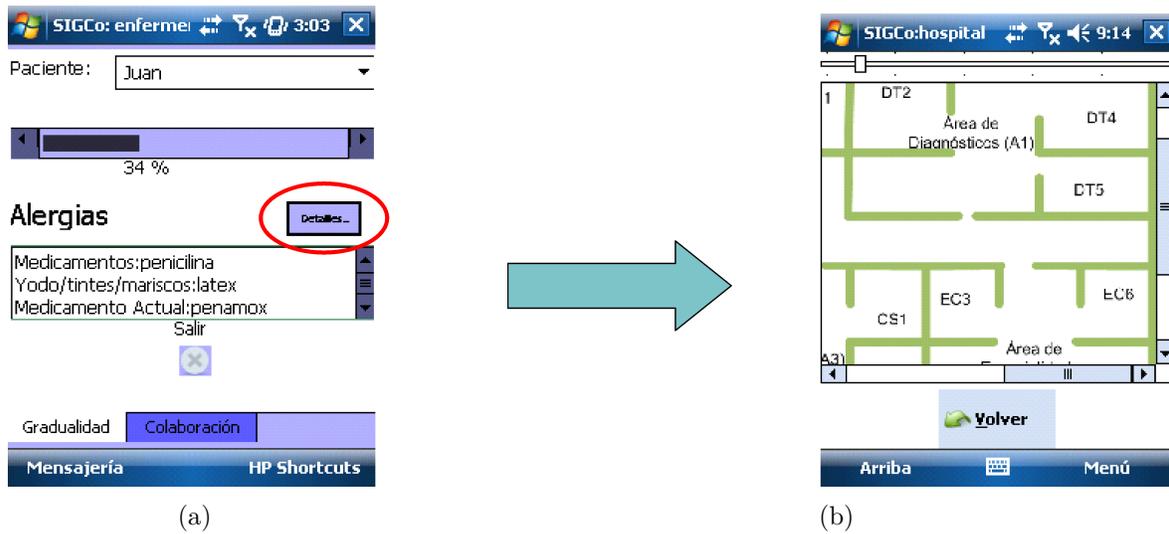


Figura 4.8: Interfaz de usuario de información de conciencia de grupo a interfaz de usuario de detalles en PDA

La Figura 4.8(a) muestra un círculo gris que permite a un miembro de la organización ver información sobre algún estudio realizado al paciente, ver ubicaciones de los diferentes miembros de dicha organización, así como de sus recursos físicos.

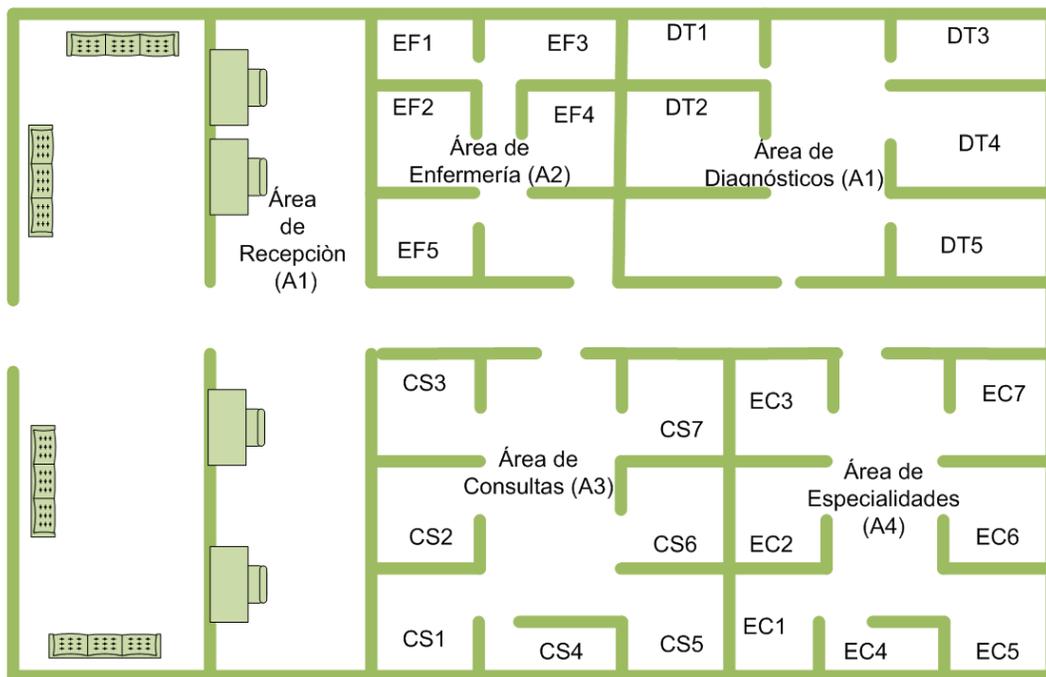


Figura 4.9: Ubicaciones de las áreas de un hospital en SIGCo

La “interfaz de información de conciencia” tiene una opción para mostrar las áreas de un hospital, entonces se pasa a la “interfaz de usuario de visualización de mapa”. Por ejemplo, la Figura 4.8(b) muestra la interfaz de visualización, en la cual se puede observar algunas áreas del hospital, pero debido a que la pantalla del dispositivo es pequeña no se puede observar todo el mapa completo.

La Figura 4.9 muestra el mapa de manera completa para ilustrar todas las áreas existentes, así como las ubicaciones de los equipos en las diferentes áreas de dicho hospital (ver Figura 4.8(b)).

La segunda pestaña titulada “Colaboración”, permite a un miembro de la organización colaborar sobre un diagnóstico de un paciente con sus colegas. Para establecer la cooperación es necesario crear un grupo, después seleccionar a los participantes uno a uno de una lista de miembros dada por el propio sistema SIGCo y finalmente el administrador del sistema activa al grupo enviando un mensaje de colaboración a todos los miembros de dicho grupo (ver Figura 4.10).

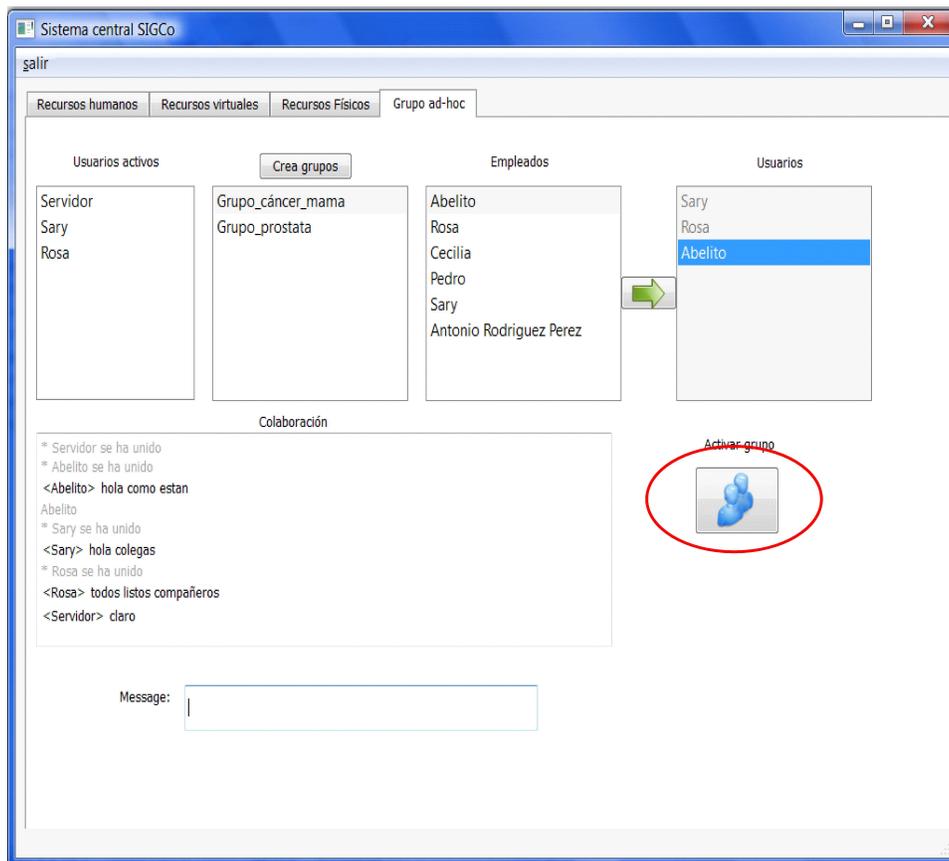


Figura 4.10: Pestaña para administrar los grupos *ad-hoc*

La Figura 4.10 muestra un círculo gris para indicar en qué parte de la interfaz del usuario del sistema SIGCo se activa un grupo. Por ejemplo, se crea un grupo llamado **Grupo_cáncer_mama** y se agregan los participantes (Sary, Rosa y Abelito) a dicho

grupo. Después, se procede a activar el grupo de colaboradores, el cual tiene por nombre **Grupo_cáncer_mama** enviando un mensaje a todos los miembros del grupo (Sary, Rosa y Abelito). Dicho mensaje, indica la aceptación o rechazo a la colaboración. Si los miembros aceptan, entonces el sistema SIGCo inmediatamente envía el nombre del grupo y la lista de usuarios a todos los participantes (ver Figura 4.11). Después, de recibir la información correspondiente al grupo creado, el sistema SIGCo está listo para llevar a cabo las interacciones entre los diferentes colegas del grupo (ver Figura 4.12).

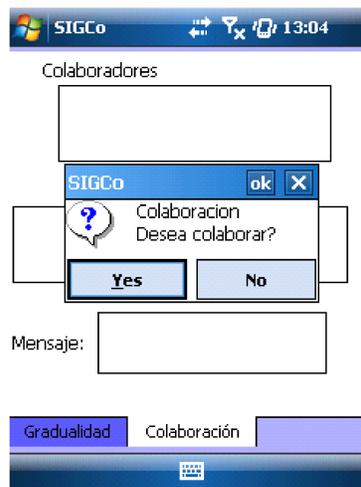


Figura 4.11: Mensaje de colaboración en la PDA



Figura 4.12: Interfaz de colaboración en la PDA

4.4.3 Resultados

Cada una de las interfaces descritas previamente permiten al colaborador solicitar y visualizar información con un mínimo esfuerzo de manera práctica, de tal forma que los miembros de la organización puedan adquirir información de conciencia de grupo de acuerdo a sus propios intereses.

Con base a las pruebas realizadas al sistema SIGCo, se llevo a cabo un análisis de las características de los sistemas de conciencia periférica y conciencia enfocada frente al sistema SIGCo (ver capítulo 2). Esta comparación nos lleva a decir que muchos de los sistemas colaborativos no satisfacen las exigencias de los colaboradores. En la tabla 4.8 se muestran las principales características del sistema SIGCo.

Tabla 4.8: Características del sistema SIGCo

Características	Descripción
Conciencia de grupo	SIGCo provee de información de recursos humanos, físicos y virtuales.
Heterogeneidad	Dispositivos PC, Laptop, PDA y Smartphone.
Movilidad	Los miembros de una organización pueden hacer uso del sistema SIGCo sin necesidad de cable.
Flexibilidad	SIGCo ofrece información gradual de conciencia a los miembros de una organización.
Portabilidad	El sistema SIGCo fue pensado para que pueda ser utilizado en otras plataforma (Linux, symbian y Maemo).
Tipos de sesión	Los usuarios pueden iniciar sesión para solicitar información de conciencia y a su vez colaborar con otros colegas.

La tabla 4.9 muestra una comparación a nivel de granuralidad de los sistemas de información de conciencia periférica y enfocada, así como del sistema SIGCo. En esta tabla se observa las ventajas del sistema SIGCo a nivel de granularidad, pero cabe mencionar que dicho sistema tiene la desventaja, que únicamente provee información a nivel textual e imágenes pero no provee información sobre video, e.g. de un ultrasonido, aunque está contemplado a nivel de diseño.

De acuerdo al análisis de los sistemas de información de conciencia y al sistema SIGCo, concluimos que ninguno de los sistemas analizados provee información gradual de conciencia. La mayoría la mayoría de los sistemas se enfocan en grupos de objetos que proveen

conjuntos de información de conciencia de grupo (vistas radar o barras multiusuario). La principal característica que ofrece el sistema SIGCo se refiere a la **información gradual de conciencia**, ya que permite ver la información de un paciente (e.g., sus citas o un estudio clínico), además de establecer colaboración con colegas de la organización.

En el siguiente capítulo se describen las limitaciones del sistema SIGCo, así como las posibles extensiones que podrían hacerse para mejorar dicho sistema.

Tabla 4.9: Granularidad de los sistemas de información del conciencia

Características	Sistemas de conciencia periférica	Sistemas de conciencia enfocada	Sistema SIGCo
Entorno físico	✗	✗	✓
Entorno virtual	✓	✓	✓
Grano grueso	✓	✗	Provisión de información gradual
Grano fino	✗	✓	

Capítulo 5

Conclusiones y trabajo futuro

Este trabajo de tesis tuvo como objetivo el diseño y el desarrollo de un sistema de información para proveer conciencia de grupo de manera gradual a los miembros de una organización. En este capítulo se retoma el planteamiento del problema, el cual se refiere al vacío de información que existe entre los sistemas de conciencia periférica y enfocada. Después, ofrece una conclusión referente al sistema desarrollado en la presente tesis para resolver el problema planteado. Por último, se proponen algunas mejoras y extensiones futuras de dicho sistema.

5.1 Síntesis de la problemática

Esta tesis se inscribe en el campo de investigación denominado Trabajo Cooperativo Asistido por Computadora (TCAC). El objetivo de este campo de investigación es el estudio de los aspectos sociales de las actividades individuales y colectivas, así como de los aspectos tecnológicos de la información y de las comunicaciones, con el fin de soportar la colaboración entre personas potencialmente distribuidas. Esta colaboración se refiere a un grupo de personas involucradas en una tarea u objetivo común. Por lo tanto, esta tesis se centra en los aspectos tecnológicos de este campo de investigación, con el fin de diseñar e implementar un sistema colaborativo que presente gradualmente a los colaboradores información de conciencia de grupo proveniente de entornos físicos y virtuales.

El concepto de “información gradual” se refiere al tipo y al volumen de información que los colaboradores pueden visualizar desde sus dispositivos de cómputo.

El concepto de “conciencia de grupo” se define como el conocimiento de las actividades de otros que provee un contexto para las actividades propias [Dourish and Bly, 1992]. De acuerdo con Dourish, la conciencia de grupo pretende asegurar que las actividades personales sean relevantes a las actividades colectivas [Dourish, 1997].

La mayoría de los sistemas colaborativos proveen información de conciencia de grupo en dos vertientes. La primera vertiente se refiere a la información de conciencia periférica o de nivel macro, que se define como el conocimiento de las actividades en un contexto global y por otro lado la información de conciencia enfocada o de nivel micro atañe el conocimiento preciso de las actividades del grupo. Sin embargo, estos tipos de información de conciencia previamente mencionados presentan problemas tales como: a) información

escasa que causan problemas de contactos inapropiados o pérdidas de oportunidades e b) información excesiva que ocasionan conflictos de privacidad, intrusión, alto consumo de ancho de banda y de escalabilidad.

Si bien, hoy en día, estos problemas de carácter tecnológico están relativamente resueltos, las violaciones de privacidad e intrusión son temas de investigación abiertos. Estos problemas de carácter social se deben principalmente a que los diseñadores de sistemas colaborativos carecen de una metodología para determinar con precisión qué información requieren las personas para facilitar el contacto entre ellas. En consecuencia, estos problemas han llevado a los diseñadores de sistemas colaborativos a confiar en su perspicacia y creatividad.

Además, el control del uso compartido de los recursos y la privacidad de la información asociada pueden cambiar, debido a que las necesidades de un grupo pueden evolucionar con el tiempo. Los recursos físicos y virtuales de una organización facilitan la realización de sus actividades diarias, pero no son aprovechados de manera adecuada. Por lo tanto, se necesita una herramienta que promueva el uso compartido de recursos y que agilice algunos procesos organizacionales. En esta tesis de maestría, se propuso un sistema que presenta de manera gradual la información de conciencia de grupo proveniente de espacios virtuales y físicos.

5.2 Conclusión

El desarrollo de esta tesis partió del análisis de los sistemas colaborativos que ofrecen información de conciencia de grupo periférica y enfocada. De hecho, existía un vacío entre estos dos tipos de conciencia que no había sido llenado por ninguno de los sistemas analizados. En consecuencia, se propuso el desarrollo del sistema SIGCo (Sistema de Información Gradual de Conciencia) para proveer información gradual de conciencia de grupo y así llenar el espacio vacío entre los dos tipos de información de conciencia previamente mencionados. El sistema SIGCo permite a los miembros de una organización obtener información gradual de conciencia de grupo sobre recursos humanos (e.g., un paciente de un hospital), recursos físicos (e.g., equipo de ultrasonido o de Rayos X) y recursos virtuales (e.g., historial de citas o una imagen de Rayos X), de manera que sean aprovechados por los miembros de una organización.

El sistema SIGCo cuenta con una arquitectura de tres capas: 1) capa de servidor, 2) capa de comunicación, 3) capa de cliente. Cada una de las capas tiene sus propios componentes que le permiten interactuar con las demás. La capa de servidor almacena la información de los recursos humanos, físicos y virtuales, provee información de estos recursos a los miembros de la organización, coordina las diferentes actividades colaborativas y permite la creación de grupos *ad-hoc* para que los usuarios puedan interactuar entre ellos. El sistema SIGCo utiliza un esquema de distribución centralizado para coordinar las actividades de los diferentes grupos *ad-hoc*. La capa de comunicación establece el enlace entre los diferentes clientes que se conectan al servidor, mediante la creación de sockets e interpreta los diferentes comandos enviados por los clientes que consultan la información compartida. Por último, la capa de cliente permite presentar la información de conciencia

de grupo en las diferentes interfaces de usuario, así como tratar las peticiones realizadas por los colaboradores.

La información gradual de conciencia de grupo permite satisfacer los requerimientos de los miembros de una organización debido a que ofrece información de conciencia de grupo a los colaboradores de acuerdo a sus necesidades y además permite establecer grupos *ad-hoc* entre colegas. Por lo tanto, el vacío entre los dos tipos de sistemas de información de conciencia (periférica y enfocada) de grupo ha sido llenado mediante el diseño y la implementación del sistema SIGCo. Además, este sistema provee información de los recursos humanos, físicos y virtuales de una organización.

5.3 Contribuciones

La principal contribución realizada en el presente trabajo, es la provisión de información gradual de conciencia de grupo en el campo de investigación denominado Trabajo Cooperativo Asistido por Computadora (TCAC).

La mayoría de los sistemas colaborativos proveen solo dos tipos información de conciencia: periférica y enfocada. En nuestro análisis de estudio sobre los sistemas colaborativos ninguno proveía el concepto de información gradual de conciencia.

A continuación se mencionan las siguientes contribuciones:

- Diseño de una arquitectura de tres capas para proveer información gradual de conciencia de grupo en una organización.
- Ofrece seguridad en la provisión de la información, ya que la información es provista de acuerdo a los roles que puedan existir en una organización (e.g., enfermera o médico general).
- Provee la funcionalidad de crear grupos *ad-hoc* para que múltiples usuarios puedan colaborar, sobre algún estudio clínico o diagnóstico de un paciente.
- Ofrece un control *Focus/Nimbus* para solicitar y visualizar la información de conciencia de grupo.

El concepto de información gradual de conciencia, puede ser aplicado en cualquier organización, e.g., puede ser aplicado en instituciones educativas o entornos hospitalarios.

5.4 Implicaciones

La mayoría de los sistemas que ofrecen conciencia de grupo se ha limitado a la administración de información proveniente de ambientes virtuales. Por lo general, la información producida en cada sitio participante es duplicada en todos los demás sitios, con el fin de que los colaboradores puedan acceder de una sola vez a toda la información, aun si no están interesados. En consecuencia, ni el productor ni el consumidor de información de conciencia de grupo pueden decidir qué tipo de información y volumen desean respectivamente enviar y recibir. Por lo tanto, estos problemas previamente mencionados se pueden

resolver mediante la arquitectura de software propuesta en el presente trabajo. En dicha arquitectura se aplica el concepto de información gradual de conciencia de grupo para proveer información gradual de conciencia de grupo a las organizaciones. Dicho concepto permite que los colaboradores tengan un control sobre la información que deseen consultar, además dichos colaboradores pueden solicitar información de conciencia gradual de grupo en cualquier parte de un edificio (movilidad).

Esta arquitectura de software, puede ser aplicado en entorno de una organización como pueden ser hospitales o universidades, donde sus recursos humanos están distribuidos en diferentes lugares públicos o privados como: salas de reunión, pasillos u oficinas. Además, en dicha arquitectura se preserva la privacidad de los colaboradores, debido a que la provisión de la información gradual de conciencia de grupo está provista de acuerdo a roles otorgados por una organización (e.g., una enfermera no puede ver la misma información que un médico general).

5.5 Limitaciones

- Las consultas efectuadas por los miembros de una organización, sobre información gradual de conciencia de grupo, están basadas sólo en roles.
- En la arquitectura de *software* de SIGCo únicamente se contempla información a nivel textual.
- La arquitectura de *software* de SIGCo no contempla las ubicaciones de los colaboradores en una organización (i.e., se desconoce su identidad o su foto).
- El diseño de la arquitectura de SIGCo, solamente provee soporte para grupos de participación conversacional.

5.6 Trabajo futuro

En seguida se presentan algunas mejoras y/o extensiones que pueden ser realizadas a la arquitectura de *software* del sistema SIGCo:

- Una mejora importante es que las consultas no sólo estén basadas en roles, sino que dependan de más criterios, tales como la localización del colaborador, su historia en el sistema, sus actividades desarrolladas durante un periodo de tiempo, así como sus preferencias. Por lo tanto, la información adquirida tendría un grano más fino de acuerdo a las necesidades del usuario.
- Una posible mejora consiste en implementar una base de conocimientos de acuerdo a los criterios especificados por los colaboradores, de manera que las consultas sean más complejas, e.g., si el doctor Juárez solicita los análisis de sangre de un paciente, el sistema SIGCo también deberá dar la dieta y los medicamentos prescritos a dicho paciente. De esta manera, el sistema SIGCo tendría la capacidad de dividir la

consulta en tres subconsultas (análisis sanguíneo, dieta y medicamentos prescritos). Dicha división se basa en el conocimiento almacenado en la base de conocimientos.

- Una extensión importante concierne en la localización de los colaboradores dentro de un edificio, e.g., en un hospital es importante saber donde se encuentra un médico. Por lo tanto, el sistema SIGCo podría hacer uso de las funciones de localización que ofrece el sistema RAMS, a fin de obtener las ubicaciones de los colaboradores dentro de un edificio mediante tomas instantáneas de video [García, 2009].
- Una mejora factible consiste en implementar una pizarra para mostrar una imagen o un video, donde cada colaborador pueda exponer su diagnóstico de un paciente a sus colegas.

Apéndice A

Instalaciones y configuraciones

A.1 Instalación de QT C++ *Embedded open source*

Los pasos de instalación son los siguientes:

1. Tener instalado *Microsoft Visual Studio 2005*¹ e instalar *Windows Mobile SDK*. Estas dos aplicaciones se utilizan para compilar el código de QT C++ y para emular el dispositivo móvil. La instalación de *Visual Studio* no requiere explicación ya que su instalación es de manera guiada (paso a paso);
2. bajar QT C++ *Embedded*²;
3. descomprimir el archivo en el directorio que se desea instalar QT, e.g., C:\qt-embedded-wince-opensource-src-4.4.3\4.4.3;
4. crear una variable de ambiente QT, especificando la ruta donde se encuentra las herramientas qmake y moc. Para agregar la ruta ir *Control Panel* → *System* → *Advanced* → *Environment variables* y agregar *c:\qt-embedded-wince-opensource-src-4.4.3\4.4.3\bin*. En esta ruta se encuentra el compilador qmake y moc. Asegurese de que las variables de ambiente de *Visual Studio* incluya *vcvars32.bat*;
5. después, abrir una ventana de MsDos y posicionarse en el directorio *c:\qt-embedded-wince-opensource-src-4.4.3\4.4.3*;
6. escribir *configure platform win32msvc2005 xplatform wincewm50pocketmsvc2005* para indicar que se utilizará *Visual Studio 2005* con plataforma *Pocket PC*. Si se desea configurar Qt para otra plataforma escribir *configure help*, lo cual lista todos los modos de instalación disponible;
7. actualizar PATH, INCLUDE y LIB donde esta instalado *Windows Mobile 5.0 Pocket PC SDK* de la siguiente manera:

¹http://download.microsoft.com/download/3/0/8/308c4508-30c4-4387-9440-eba2e63d6693/es_vs_2005_pro_trial.iso

²<http://qt.nokia.com/downloads>

```
set INCLUDE=C:\Program Files\Microsoft Visual Studio 8\VC\ce\include;C:\Program Files\Windows CE Tools\wce500\Windows Mobile 5.0 Pocket PC SDK\Include\Armv4i
set LIB=C:\Program Files\Microsoft Visual Studio 8\VC\ce\lib\armv4i;C:\Program Files\Windows CE Tools\wce500\Windows Mobile 5.0 Pocket PC SDK\Lib\ARMV4I
set PATH=C:\Program Files\Microsoft Visual Studio 8\VC\ce\bin\x86_arm;%PATH%;
```

8. después de realizar la actualización, ahora ir al directorio `c:\qt-embedded-wince-opensource-src-4.4.3\4.4.3\bin`, y escribir `setcepaths wincewm50pocket-msvc2005`;
9. ir al directorio `c:\qt-embedded-wince-opensource-src-4.4.3\4.4.3` y escribir `nmake` para compilar el código fuente de QT (esperar unos minutos) y
10. al finalizar la instalación aparece un mensaje diciendo “QT ha sido instalado”, para verificar su instalación buscar la carpeta `examples\doc\html` en donde se encontrará los ejemplos que han sido instalados, esto indica que la instalación ha sido exitosa.

A.2 Instalación del sistema SIGCo

Instalación para PC:

1. crear una carpeta con el nombre de su preferencia, e.g., Gradualidad;
2. copiar SIGCo.exe y Conciencia.db en el directorio previamente creado.

Instalación para la PDA:

1. crear una carpeta con el nombre y la ruta de su preferencia;
2. copiar SIGCocliente.exe, msvcr80d.dll, phonond4.dll, QtCored4.dll, QtGuid4.dll y QtNetwork4.dll en el directorio creado previamente;
3. ejecutar el archivo SIGCocliente.exe.

Referencias

- [Abel, 1990] Abel, M. J. (1990). *Experiences in an Exploratory Distributed Organization*. Intellectual Teamwork: Social and Technological Foundations of Cooperative Work, L. Erlbaum Associates Inc., pp. 489-510.
- [Benford and Fahlén, 1993] Benford, S. and Fahlén, L. (1993). *A Spatial Model of Interaction in Large Virtual Environments*. Proceedings of the Third Conference on European Conference on Computer-Supported Cooperative Work, Kluwer Academic Publishers, pp. 109-124, Milan, Italy.
- [Bi, 2009] Bi, C. (2009). *Research and Application of SQLite Embedded Database Technology*. W. Trans. on Comp., 8(1), World Scientific and Engineering Academy and Society (WSEAS), pp. 83-92.
- [Bérard and Coutaz, 1997] Bérard, F. and Coutaz, J. (1997). *Awareness As an Extended Notion of Observability*. CHI'97 Workshop on "Awareness in Collaborative Systems", Atlanta Georgia, United States.
- [Calvary et al., 1997] Calvary, G., Coutaz, J., and Nigay, L. (1997). *From Single-User Architectural Design to PAC*: A Generic Software Architecture Model for CSCW*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press, pp. 242-249, Atlanta Georgia, United States.
- [Coutaz et al., 1999] Coutaz, J., Bérard, F., Carraux, E., Astier, W., and Crowley, J. L. (1999). *CoMedi: Using Computer Vision to Support Awareness and Privacy in Mediaspaces*. Extended abstracts on Human factors in Computing Systems (CHI'99), pp. 13-14, Pittsburgh Pennsylvania, United States.
- [De Michelis and Grasso, 1994] De Michelis, G. and Grasso, M. A. (1994). *Situating Conversations Within the Language/Action Perspective: the Milan Conversation Model*. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, ACM Press, pp. 89-100, Chapel Hill North Carolina, United States.
- [Decouchant et al., 2001] Decouchant, D., Favela, J., and Martinez-Enriquez, A. M. (2001). *PINAS: A Middleware for Web Distributed Cooperative Authoring*. Proceedings of the 2001 Symposium on Applications and the Internet (SAINT 2001), IEEE Computer Society, pp. 187, San Diego California, United States.

- [Decouchant et al., 2008] Decouchant, D., Mendoza, S., and Rodríguez, J. (2008). *A Realistic and Efficient Distributed Infrastructure for Nomadic Web Cooperative Work*. Proceedings of the 2008 Mexican International Conference on Computer Science, IEEE Computer Society, pp. 151-162, Mexicali Baja California, México.
- [Decouchant et al., 2010] Decouchant, D., Mendoza, S., and Rodríguez, J. (2010). *Chapter 7: Suited Support for Distributed Web Intelligence Cooperative Work*. In Emergent Web Intelligence Book: Advanced Semantic Technologies, Springer Verlag.
- [Domingos et al., 1998] Domingos, H. J. L., Pregoça, N. M., and Martins, J. L. (1998). *Coordination and Awareness Support for Adaptive CSCW Sessions*. pp. 21-37.
- [Dourish, 1997] Dourish, P. (1997). *Extending Awareness Beyond Synchronous Collaboration*. Proceedings of the CHI'97 Workshop on Awareness in Collaboration Systems, ACM Press, Atlanta Georgia, United States.
- [Dourish and Bly, 1992] Dourish, P. and Bly, S. (1992). *Portholes: Supporting Awareness in a Distributed Work Group*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press, pp. 541-547, Monterey California, United States.
- [Ellis and Wainer, 1994] Ellis, C. A. and Wainer, J. (1994). *Goal Based Models of Collaboration*. Collaborative Computing, pp. 61-86.
- [Ender, 2009] Ender, G. (2009). *Real-Time*. Open Space-Online. Open space Online[®].
- [García, 2009] García, E. K. G. (2009). *Disponibilidad de recursos compartidos en un ambiente colaborativo con carácter ubicuo*. Tesis de Maestría, Departamento de Computación del Centro de Investigación y de Estudios Avanzados CINVESTAV-IPN.
- [García et al., 2008a] García, K., Mendoza, S., Olague, G., Decouchant, D., and Rodríguez, J. (2008a). *Shared Resource Availability within Ubiquitous Collaboration Environments*. Groupware: Design, Implementation, and Use: 14th International Workshop, CRIWG 2008, Omaha, Nbraska, United State, September 14-18, 2008, Revised Selected Papers, Springer-Verlag, LNCS 5411, pp. 25-40, Berlin, Heidelberg.
- [García et al., 2008b] García, K., Mendoza, S., Olague, G., Decouchant, D., and Rodríguez, J. (2008b). *Shared Resource Availability within Ubiquitous Collaboration Environments*. Groupware: Design, Implementation, and Use: 14th International Workshop, Springer-Verlag, pp. 25-40, Berlin, Heidelberg.
- [Granville and Hickey, 2005] Granville, K. and Hickey, T. J. (2005). *The Design, Implementation, and Application of the GrewpEdit Tool*. Proceedings of the 2005 Conference on Diversity in Computing, ACM Press, pp. 14-16, Albuquerque New Mexico, United States.
- [Greenberg, 1996] Greenberg, S. (1996). *Peepholes: Low Cost Awareness of One's Community*. Conference Companion on Human Factors in Computing Systems, ACM Press, pp. 206-207, Vancouver British Columbia, Canada.

- [Greenberg and Johnson, 1997] Greenberg, S. and Johnson, B. (1997). *Studying Awareness in Contact Facilitation*. Workshop on Awareness in Collaborative Systems of Computer Human Interaction, ACM Press, pp. 22-27, Atlanta Georgia, United States.
- [Gurevich, 2006] Gurevich, M. (2006). *JamSpace: Designing a Collaborative Networked Music Space for Novices*. Proceedings of the 2006 Conference on New Interfaces for Musical Expression, IRCAM - Centre Pompidou, pp. 118-123, Paris, France.
- [Gutwin and Greenberg, 1998] Gutwin, C. and Greenberg, S. (1998). *Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness*. Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, ACM Press, pp. 207-216, Seattle Washington, United States.
- [Haake and Wilson, 1992] Haake, J. M. and Wilson, B. (1992). *Supporting Collaborative Writing of Hyperdocuments in SEPIA*. Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work, ACM Press, pp. 138-146, Toronto Ontario, Canada.
- [James and Keith, 2004] James, F. K. and Keith, W. R. (2004). *Redes de Computadores: Un Enfoque Descendente Basado en Internet, Segunda Edición*. Pearson Addison Wesley.
- [Johnson, 1992] Johnson, P. (1992). *Supporting Exploratory CSCW with the EGRET Framework*. Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work, ACM Press, pp. 298-305, Toronto Ontario, Canada.
- [Kay, 2007] Kay, T. (2007). *AlphaMail is Scalable and Accessible Web Mail*. Specialized Systems Consultants, Inc., 2007(160), Linux Journal, pp. 11, Seattle WA, United States.
- [Lerman, 2004] Lerman, C. (2004). *UML y Patronos: Una introducción al análisis orientado a objetos y al proceso unificado, Segunda Edición*. Pearson Prentice Hall.
- [Mandal et al., 2007] Mandal, N., Deelman, E., Mehta, G., Su, M. H., and Vahi, K. (2007). *Integrating Existing Scientific Workflow Systems: the Kepler/Pegasus example*. Proceedings of the 2nd Workshop on Workflows in Support of Large-Scale Science, ACM Press, pp. 21-28, Monterey California, United States.
- [Maqsood and Javed, 2007] Maqsood, M. E. and Javed, T. (2007). *Practicum in software project management: an endeavor to effective and pragmatic software project management education*. Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, ACM Press, pp. 471-480, Dubrovnik, Croatia.
- [Mitchell, 1996] Mitchell, A. (1996). *Communication and Shared Understanding in Collaborative Writing*. Master's Thesis, University of Toronto, Department of Computer Science.

- [Moran et al., 2004] Moran, A. L., Favela, J., Enriquez, A. M. M., and Decouchant, D. (2004). *On the Design of Potential Collaboration Spaces*. Int. J. Comput. Appl. Technol., 19(3)(4), Inderscience Publishers, pp. 184-194.
- [Romero et al., 2007] Romero, N., McEwan, G., and Greenberg, S. (2007). *A Field Study of Community Bar: (mis)-matches Between Theory and Practice*. Proceedings of the 2007 international ACM Conference on Supporting Group Work, ACM Press, pp. 89-98, Sanibel Island Florida, United States.
- [Roth and Unger, 2000] Roth, J. and Unger, C. (2000). *An Extensible Classification Model for Distribution Architectures of Synchronous Groupware*. Proceedings of the Fourth International Conference on the Design of Cooperative Systems (COOP2000), IOS Press, Sophia Antipolis, France.
- [Salber et al., 1995] Salber, D., Coutaz, J., Decouchant, D., and Riveill, M. (1995). *De l'observabilité et l'honnêteté: le cas du contrôle d'accès dans la Communication Homme- Homme Médiatisée*. Proceedings of the Conference on Interaction Human Machine, pp. 27-34, Toulouse, France.
- [Sohlenkamp et al., 2000] Sohlenkamp, M., Mambrey, P., Prinz, W., Fuchs, L., Syri, A., Pankoke-Babatz, U., Klöckner, K., and Kolvenbach, S. (2000). *Supporting the Distributed German Government with POLITeam*. Multimedia Tools Appl., 12(1), Kluwer Academic Publishers, pp. 39-58.
- [Stallings, 2004] Stallings, W. (2004). *Comunicaciones y Redes de Computadores*. Pearson Education, septima edición edition.
- [Tee et al., 2009] Tee, K., Greenberg, S., and Gutwin, C. (2009). *Artifact Awareness Through Screen Sharing for Distributed Groups*. Int. J. Hum.-Comput. Stud., 67(9), Academic Press, Inc., pp. 677-702.
- [Vertegaal, 1997] Vertegaal, R. (1997). *Conversational awareness in multiparty VMC*. Extended Abstracts on Human Factors in Computing Systems, ACM Press, pp. 6-7, Atlanta, Georgia.
- [Xia et al., 2004] Xia, S., Sun, D., Sun, C., Chen, D., and Shen, H. (2004). *Leveraging Single-User Applications for Multi-user Collaboration: the CoWord Approach*. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, ACM Press, pp. 162-171, Chicago Illinois, United States.