

# **Propuesta Arquitectónica para el Soporte de Servicios de Awareness en Entornos de Aprendizaje Colaborativo Asistido por Computadora (CSCL)**



Tesis de Maestría

**Ing. Iván Darío Claros Gómez**

Director

Mag. Andrés Lara Silva

Codirector

PhD. Cesar A. Collazos

Asesora

PhD. Ruth Cobos Pérez  
(Universidad Autónoma de Madrid)

*Universidad del Cauca*

Instituto de Postgrados de Ingeniería Electrónica y Telecomunicaciones IPET

**Maestría en Ingeniería, Área Telemática**

**Línea de Investigación, Ingeniería de la Colaboración**

Popayán, mayo de 2010

## Publicaciones

Algunas ideas, materiales y figuras de esta tesis aparecen previamente en las siguientes publicaciones, las cuales se encuentran en el anexo de esta monografía:

Claros, I., Collazos, C., Lara, A., Ochoa, S., Guerrero, L.: Arquitectura Multiagente para la Gestión de Información de *Awareness* en Entornos CSCL, Jornadas Chilenas de Computación (Chile), Noviembre de 2009.

Ovalle, D., Jiménez, J., Collazos, C., Claros, I., Pantoja, W., Cobos, R., Moreno-Llorena, J., Pifarré, M., Argelagos, E.: "Guía metodológica para el seguimiento y evaluación de aprendizaje colaborativo asistido por el sistema KnowCat". Reunión Nacional y Expoingeniería ACOFI 2009, septiembre de 2009.

Cobos, R., Claros, I., Moreno-Llorena, J.. A proposal of *Awareness* Services for the Construction of Quality Community Knowledge supported by the Knowledge Management system KnowCat. 13th International Conference on Human-Computer Interaction. San Diego (Estados Unidos), Julio de 2009.

Claros, I., Collazos, C., Cobos, R.: Servicios *Awareness* para la Mejora del Escenario Educativo Soportado sobre KnowCat Revista Avances en Sistema e Informática, Vol. 6 No. 1, Junio de 2009, Medellín, ISSN 1657-7663.

Claros, I., Collazos, C., Cobos, R., Evaluación de Servicios *Awareness* para Sistemas de Gestión de Conocimiento, Experiencia KnowCat, IV Congreso Colombiano de Computación, Abril de 2009.

Collazos, C., Cobos, R., Jiménez, J., Moreno, J., Pifarré, M., Ovalle, D., Claros, I., Pantoja, L., Propuesta par ala monitorización de procesos colaborativos de un sistema de gestión del conocimiento, Interaccion 2008, Albacete (España), Junio 2008, pp. 95-98.

## Tabla de Contenido

Publicaciones .....	i
Tabla de Contenido .....	ii
Lista de figuras .....	iii
Lista de tablas.....	iv
Capítulo 1, Introducción .....	1
1.1 Motivación y Problemática .....	1
1.2 Proyectos Relacionados .....	3
1.2.1 Arquitecturas que soportan <i>Awareness</i> .....	3
1.2.2 Uso del Paradigma de Agentes dentro de los entornos CSCL.....	5
1.3 Objetivos .....	6
1.4 Metodología.....	7
1.5 Contribuciones.....	7
1.6 Estructura del Documento .....	8
Capítulo 2, Marco Conceptual.....	9
2.1 Aprendizaje colaborativo asistido por computadora CSCL.....	9
2.2 La Ingeniería de la Colaboración .....	11
2.3 <i>Awareness</i> .....	12
2.4 Sistemas Multi-Agente .....	13
2.4.1 Agente .....	14
2.4.2 Metodologías para Sistemas Multi-Agente.....	15
2.4.3 GAIA.....	15
Capítulo 3, Una Arquitectura para la Gestión del Awareness.....	18
3.1 Modelo Conceptual para la Gestión del Awareness.....	18
3.1.1 Modelo de Información .....	19
3.1.2 Agente .....	23
3.1.3 Gestor.....	23
3.2 Un Sistema Multi-Agente para la gestión del Awareness .....	24
3.2.1 Análisis .....	24
3.2.2 Diseño .....	34
3.2.3 Arquitectura Interna del Agente de Interfaz.....	38
3.2.4 Arquitectura Interna del Agente de Servicio .....	38
Capítulo 4, Escenarios de validación .....	40
4.1 Caso de estudio I, Módulo de <i>Awareness</i> para KnowCat.....	40
4.1.1 Modelo de Información .....	42
4.1.2 Módulo Agente .....	43
4.1.3 Módulo Gestor .....	45
4.1.4 Servicios <i>Awareness</i> implementados para KnowCat .....	45
4.2 Caso de estudio II, Libro Digital de Trabajo y Red Multi-Agente .....	50
4.2.1 Libro Digital de Trabajo y CETProfesor.....	50
4.2.2 AgNet, framework para el desarrollo de Sistema Multi-Agente .....	62
4.2.3 Sistema Multi-Agente para la Gestión del Awareness.....	70
4.2.4 Integración del Libro digital de trabajo con el Sistema Multi-Agente .....	75
Capítulo 5, Experimentación .....	76
5.1 Caso de Estudio I, Módulo de <i>Awareness</i> para KnowCat .....	76
5.1.1 Metodología.....	76
5.1.2 Resultados obtenidos .....	77
5.2 Caso de Estudio II, Libro Digital y Red Multi-Agente.....	79
5.2.1 Metodología.....	80

5.2.2 Resultados obtenidos .....	83
Capítulo 6, Conclusiones y Trabajos Futuros.....	87
6.1 Conclusiones .....	87
6.1 Trabajos Futuros .....	88
Referencias .....	90

## Lista de figuras

Fig. 1. Escenario de distribución característico del conocimiento en un entorno CSCL.....	1
Fig. 2. Modelo Conceptual para la Gestión del Awareness .....	18
Fig. 3. Los distintos enfoques de Awareness y sus relaciones de cooperación.....	20
Fig. 4. Modelo de Referencia para la Estructuración del Awareness.....	21
Fig. 5. Modelo de Clases Asociadas con el Rol Gestor Awareness Social.....	27
Fig. 6. Diagrama de Clases Esquema de Rol Gestor de Awareness de Tarea.....	28
Fig. 7. Diagrama de Clases Esquema Gestor de Awareness de Espacio Compartido .....	29
Fig. 8. Diagrama de Clases relacionadas con el esquema del Rol Gestor de Awareness de Conocimiento Compartido .....	31
Fig. 9. Modelo de Familiaridad.....	37
Fig. 10. Arquitectura interna para los Agentes de Interfaz.....	38
Fig. 11. Arquitectura interna para los Agentes de Servicio .....	39
Fig. 12. Diagrama de Clases simplificado de la implementación para KnowCat.....	42
Fig. 13. Modelo de información definido para el Sistema KnowCat.....	43
Fig. 14. Vista de la página principal del sistema, Consola de Awareness .....	44
Fig. 15. Estructura de un mensaje desde el Gestor para la actualización de la barra de tareas de la Consola de Awareness.....	44
Fig. 16. Vista de la Consola Resumen del Servicio Usuarios Registrados .....	45
Fig. 17. Resumen del Servicio Usuarios en Línea.....	46
Fig. 18. Resumen del Servicio Vista Histórica.....	47
Fig. 19. Resumen del Servicio Vista de Radar .....	47
Fig. 20. Resumen del Servicio Participómetro.....	48
Fig. 21. Resumen del Servicio Mapa de Interacción .....	49
Fig. 22. Diagrama de Casos relacionados con la herramienta Libro Digital de Trabajo....	51
Fig. 23. Caso de uso Abrir Hoja .....	52
Fig. 24. Caso de uso Editar Hoja .....	52
Fig. 25. Caso de uso Hacer Anotaciones .....	53
Fig. 26. Caso de uso Responder Pregunta .....	53
Fig. 27. Caso de uso Subir Respuesta.....	54
Fig. 28. Caso de uso Anotar respuesta.....	54
Fig. 29. Caso de uso Acceder a Hoja compartida .....	55
Fig. 30. Caso de uso Ver Compañeros de grupo .....	55
Fig. 31. Diagrama de casos de uso relacionados con la herramienta CETProfesor .....	56
Fig. 32. Caso de uso Gestionar Preguntas .....	56
Fig. 33. Caso de uso Gestionar grupos de trabajo .....	57
Fig. 34. Caso de uso Gestionar grupos de trabajo .....	57
Fig. 35. Caso de uso Ver Sociograma .....	58
Fig. 36. Diagramas de clase de análisis para CETProfesor .....	59
Fig. 37. Diagrama de clases de análisis para el Libro Digital de Trabajo.....	60
Fig. 38. Jerarquía de Clases de entidad asociadas con Shape.....	60
Fig. 39. Clase Anotación .....	61

Fig. 40. Vista de la arquitectura por capas .....	61
Fig. 41. Diagrama de implantación del Sistema .....	62
Fig. 42. Diagrama de Paquetes de AgNet.....	63
Fig. 43. Diagrama de clases del Framework AgNet .....	64
Fig. 44. Diagrama de secuencia para la interacción entre los agentes <i>Ping</i> y <i>Pong</i> .....	67
Fig. 45. Ejecución del Sistema Multi-Agente de ejemplo con los Agentes <i>Ping</i> y <i>Pong</i> ....	69
Fig. 46. Interfaz grafica del Agente de Persistencia .....	72
Fig. 47. Interfaz Gráfica de Agente Guardián.....	72
Fig. 48. Interfaz grafica del Agente Coordinador .....	73
Fig. 49. Interfaz gráfica del Agente SKA .....	74
Fig. 50. Interfaces gráficas del Agente de Interfaz .....	74
Fig. 51. Barra de Mensajes de Awareness, mecanismo de interacción entre el Agente de Intrefaz y el Libro Digital de Trabajo.....	75
Fig. 52. Fotografía tomada durante la reunión de experto (a la izquierda ULD, derecha UT) .....	81
Fig. 53. Reunión de los grupos de trabajo mientras discutían sus respuestas .....	82
Fig. 54. Fotografía tomada durante la aplicación del examen .....	82
Fig. 55. Comparación entre los resultados de CET y el informe final .....	84
Fig. 56. pantalla de respuestas típicas para los Usuarios de Libro Digital .....	85

## Lista de tablas

Tabla 1. Conceptos abstractos y concretos en GAIA .....	15
Tabla 2. Plantilla para construir esquemas de roles.....	16
Tabla 3. Convención de anotaciones para lista de Responsabilidad de un Rol.....	24
Tabla 4. Esquema del Rol Cliente.....	24
Tabla 5. Esquema del Rol Sensor.....	25
Tabla 6. Esquema del Rol Consola de Presentación .....	25
Tabla 7. Esquema del Rol Evaluador de Aportes.....	25
Tabla 8. Esquema del Rol Registrador .....	26
Tabla 9. Esquema del Rol Configurador .....	26
Tabla 10. Esquema del Rol Gestor de Acceso.....	26
Tabla 11. Esquema del Rol Gestor de Awareness Social .....	27
Tabla 12. Esquema del Rol Gestor de Awareness de Tarea.....	28
Tabla 13. Esquema del Rol Gestor de Awareness de Espacio Compartido .....	29
Tabla 14. Esquema del Rol Gestor de Awareness de Conocimiento Compartido .....	30
Tabla 15. Esquema del Rol Gestor de Awareness de Concepto.....	32
Tabla 16, Plantilla general para la descripción del Modelo de Interacción.....	32
Tabla 17. Protocolo Enviar Awareness .....	32
Tabla 18. Protocolo Recibir Awareness .....	32
Tabla 19. Protocolo Recibir Artefacto.....	33
Tabla 20. Protocolo Entregar Información.....	33
Tabla 21. Protocolo Notificar Cambios.....	33
Tabla 22. Protocolo Recibir Solicitud de Acceso .....	33
Tabla 23. Protocolo Notificar Estado.....	33
Tabla 24. Protocolo Generar Awareness .....	33
Tabla 25. Servicio Pasarela de Comunicación Awareness/Groupware .....	34
Tabla 26. Servicio de Captura de Paquetes.....	35
Tabla 27. Servicio Buscar Expertos .....	35

Tabla 28. Servicio Buscar Referencias .....	35
Tabla 29. Servicio Evaluar Aporte.....	35
Tabla 30. Servicio Awareness Social .....	36
Tabla 31. Servicio Awareness de Espacio de Trabajo .....	36
Tabla 32. Servicio Awareness de Tarea.....	36
Tabla 33. Servicio Awareness de Concepto.....	36
Tabla 34. Resumen del Servicio Usuarios Registrados.....	45
Tabla 35. Resumen del Servicio Usuarios en Línea.....	45
Tabla 36. Resumen del Servicio Vista Histórica.....	46
Tabla 37. Resumen del Servicio Vista de Radar .....	47
Tabla 38. Resumen del Servicio Participómetro.....	48
Tabla 39. Resumen del Servicio Mapa de Interacción .....	48
Tabla 40. Nivel de satisfacción de los usuarios con respecto al aprendizaje (escala de 0 a 5) .....	78
Tabla 41. Promedios de valoración obtenidos .....	83

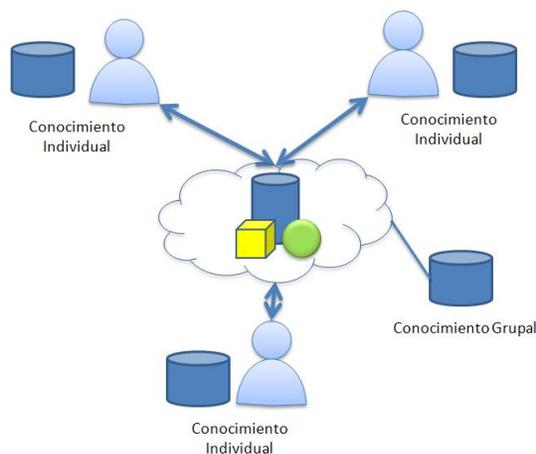
# Capítulo 1. Introducción

Este capítulo ofrece una visión general de la problemática que motivó el desarrollo de este trabajo. Para ello se han propuesto las secciones: *Motivación y Problemática*, con la cual se contextualiza el escenario problemático que aborda esta investigación; *Antecedentes*, análisis de algunas propuestas de investigación relacionadas con el objeto de estudio; *Objetivos*, donde se condensan las metas que este trabajo de maestría propuso alcanzar; *Metodología*, descripción de los pasos que orientaron el proceso de investigación; *Contribuciones*, lista de los resultados más relevantes alcanzados, y finalmente, *Estructura del Documento*, en la cual se sintetiza el contenido de cada uno de los capítulos que constituyen esta monografía.

## 1.1 Motivación y Problemática

El Aprendizaje Colaborativo Asistido por Computadora, de las siglas en inglés CSCL (*Computer Supported Collaborative Learning*), es un paradigma multidisciplinar dentro del área de innovación pedagógica y tecnológica que se construye a partir de un modelo educativo que busca aumentar la interacción entre los miembros de un grupo cuyo interés común es el aprendizaje, para lo cual cuenta con las Tecnologías de la Información y las Comunicaciones (TIC) como soporte a la interacción y comunicación entre actores [1]. Este paradigma se sustenta en la importancia de las interacciones sociales como elemento esencial del aprendizaje [2] [3] [4]. Para Johnson y Johnson [5] los ambientes de aprendizaje colaborativo favorecen el desarrollo de habilidades intelectuales de orden superior, tales como: la resolución de problemas, pensamiento crítico, creatividad, aprendizaje metacognitivo y retención de información.

En un entorno de aprendizaje colaborativo, cada participante posee un conocimiento, saberes y creencias que se necesitan compartir con los demás para lograr un entendimiento común del problema; entre más y de mejor calidad sea el conocimiento que se comparte, mejores posiblemente serán los resultados de los procesos de enseñanza/aprendizaje. En este sentido, como se ilustra en la figura 1, tomada de [6], cada individuo puede ser modelado como una fuente potencial de información, distribuida y heterogénea, en donde cada aporte individual, contribuye a la construcción de una memoria de trabajo de grupo con la cual se resuelven los problemas propuestos para el grupo.



**Fig. 1. Escenario de distribución característico del conocimiento en un entorno CSCL**

El objetivo entonces de los entornos CSCL es facilitar la apropiación del conocimiento grupal por parte de cada individuo, de tal forma que durante la interacción no solo se desarrollen con éxito las tareas propuestas, sino que además se incremente el conocimiento de cada persona. Para lograrlo, se requiere de una participación activa de todos los integrantes, lo cual demanda el establecimiento de mecanismos de interacción idóneos y una colaboración efectiva, sin embargo, no hay una garantía total de que ello ocurra [7], por lo cual es necesario identificar y abordar otras características de estos escenarios.

Entre otros requerimientos para la implantación de un entorno de aprendizaje colaborativo se tienen [8]: (1) la comunicación entre los miembros debe ser frecuente, fluida y rápida, (2) la exposición de las ideas, principios y acciones debe realizarse de forma clara y concisa, (3) todos los aportes deben ser tratadas de forma crítica y constructiva, (4) todos los miembros deben aportar ideas o argumentaciones, (5) la información debe estar disponible para todos los miembros y (6) no deben existir aportes ocultos. Además, es fundamental que se estructuren de forma adecuada las actividades que encausan la dinámica del grupo, de tal forma que se cumplan con los principios de la colaboración: *Condiciones de Igual Participación, Interdependencia Positiva, Capacidad de Respuesta Individual y Conciencia Compartida o Awareness*. Para esta investigación, el *Awareness* ha sido apropiado como toda información perceptible durante la interacción de un usuario con el entorno, que permite contextualizar sus acciones desde una perspectiva de grupo, lo cual ha sido desarrollado desde distintos enfoques que buscan filtrar aspectos o áreas de interés específicos de la dinámica de grupo, por ejemplo la interacción social, el entorno de trabajo, el conocimiento de grupo, entre otros [9] [6]. Pero en definitiva, todos los elementos mencionados contribuyen a la fluidez y efectividad de la información desde y hacia cada individuo, lo cual favorece la generación de nuevo conocimiento y la apropiación del ya existente [10].

Por otra parte, los entornos CSCL se pueden tomar como Sistemas de Gestión de Conocimiento (GC) por cuanto dan soporte a dos procesos fundamentales [11]: el primero, la conversión del conocimiento individual en colectivo; y el segundo, la conexión de las personas con ese conocimiento, basado en la interacción. En el marco de estos procesos, se han definido tareas como el monitoreo y coordinación de las actividades, la evaluación del desempeño individual y grupal, la intervención oportuna, la identificación de fuentes de conocimiento y el descubrimiento de las necesidades de conocimiento, entre otras. Estas circunstancias han llevado a que en la mayoría de las implementaciones de sistemas CSCL, tengan un soporte limitado o nulo a las tareas asociadas con gestión de conocimiento [12].

El contexto planteado, permite identificar entonces la necesidad por desarrollar propuestas que favorezcan este tipo de entornos al reducir los esfuerzos para lograr una mayor interacción de los alumnos con su entorno y brinden soporte a los procesos y tareas necesarios para lograr una colaboración efectiva.

Esta investigación, identifica el *Awareness* como elemento esencial de la dinámica colaborativa y propone mejorar los procesos de GC en los entornos CSCL mediante la implantación de servicios para la gestión de información de grupo, con lo cual se busca reducir el esfuerzo necesario para lograr una comunicación fluida, natural y eficiente, propiciando así una mayor interacción entre las personas, favoreciendo el establecimiento de un ambiente propicio al aprendizaje, intentando mitigar muchos de los problemas

señalados para estos entornos y brinda soporte a buena parte de las tareas identificadas como esenciales para el establecimiento un entorno de colaboración real.

Los desafíos para dar soporte a Servicios de *Awareness* en los entornos CSCL, establecen criterios de diseño para la solución como [13]:

- a. En un escenario de colaboración real, es común que los usuarios deban acceder a distintas herramientas para desarrollar tareas complejas y por tanto la arquitectura debe actuar como una entidad de mediación para la transmisión, registro y procesamiento de la información de *Awareness* independiente del servicio.
- b. La complejidad en los procesos de visualización de la información.
- c. Teniendo en cuenta que las discusiones en escenarios CSCL pueden llevarse a cabo de forma síncrona o asíncrona se debe garantizar el soporte a estos dos esquemas de comunicación.
- d. La heterogeneidad y distribución de los recursos involucrados durante la discusión no debe ser impedimento para la colaboración.
- e. Supervisar las actividades de los usuarios provee las evidencias necesarias para la intervención y evaluación del desempeño de los participantes.
- f. Es pertinente el aprovisionamiento de mecanismos de coordinación como una medida para mantener el control sobre tiempos de desarrollo de las actividades.
- g. Brindar soporte a los diversos enfoques de *Awareness* resulta lo más adecuado para la mayoría de los escenarios colaborativos.

Teniendo en cuenta estos factores y la demanda de servicios asociados con procesos de gestión de conocimiento de grupo tales como definición de criterios para la intervención oportuna, el aprovisionamiento de mecanismos que faciliten la evaluación del desempeño individual, la identificación de experto, entre otros, esta investigación, propuso desarrollar una propuesta conceptual y tecnológica a través de una propuesta Arquitectónica consistente en un Sistema Multi-Agente para el soporte de servicios de *Awareness* en entornos CSCL como estrategia para la mejora de los procesos de gestión de conocimiento en grupos de Trabajo. Para lo cual se desarrolló un modelo conceptual y posteriormente un Sistema Multi-Agente que cubre las consideraciones propuestas, pero primordialmente, establecer un entorno para el diseño y desarrollo de servicios orientados al mejoramiento de los procesos de enseñanza aprendizaje basados en el *Awareness*.

A continuación se presentan algunos trabajos relacionados con esta propuesta, lo cual permite identificar algunos de los aportes y diferencias de esta investigación.

## **1.2 Proyectos Relacionados**

En esta sección se presentan algunos proyectos relacionados tanto con arquitecturas para el soporte de *Awareness* como del uso de Agentes en entornos Colaborativos por estos los ejes temáticos de esta investigación.

### **1.2.1 Arquitecturas que soportan *Awareness***

Existen buen número de trabajos en cuando a arquitecturas o *frameworks* para el soporte de *Awareness* [14] [15] [16] [17] [18] [19] [20], donde la heterogeneidad y la sobrecarga de información son en su mayoría los problemas claves que se han intentado afrontar. A continuación se presenta distintas propuestas que son referentes para esta investigación.

En [14], se propone utilizar los DAG (*Directed Acyclic Graph*), como modelos para la representación de cualquier dominio mediante un gráfico de estados; tal que en el lado servidor se disponga de un gráfico global que contenga todas las posibles visiones parciales de los distintos clientes. A este gráfico se le conoce como Gs, mientras que las vistas parciales del lado clientes Gc. Además se propone diseñar reglas de transformación para gestionar una vista de datos a nivel de cliente (Gc) y componer una vista de datos global (Gs). De igual forma, una interfaz de usuario basada en widgets que comparten un bloque funcional común que da soporte a elementos básicos de comunicación.

Por otro lado en [15] divide su propuesta en tres elementos: sensor de actividades, el cual captura los eventos en el espacio de actividad; *Awareness* servidor y agentes, donde se interpretan las actividades y mantiene el estado del modelo, lo distintos agente gestionan unidades de información puntuales, tales como contadores, popularidad, etc.; y monitor de *Awareness*, que crea modelos de representación y notifica el cambio de eventos.

Otra propuesta de similar naturaleza está expuesta en [16] donde se presenta un *framework* llamado *Mainfold* y la arquitectura DISCIPLE [17]. DISCIPLE trabaja desde la propuesta de Arquitectura replicada, lo cual implica que distintos componentes que definen la funcionalidad, viajas hasta el lado cliente bajo demanda. Esta filosofía promete llevar la personalización de la aplicación a otro nivel dando al usuario la capacidad de seleccionar en tiempo de ejecución que componentes funcionales o de presentación quiere ejecutar en su entorno.

La arquitectura de DISCIPLE se divide en dos capas independientes, una de comunicaciones, denominada bus de colaborativo, que constituye canales de información en tiempo real de eventos que se intercambia, además de manejar la concurrencia a recursos, registro y remoción del registro al canal, su implementación refleja el seguimiento del patrón bróker. La segunda capa se denomina: capa de interfaces gráficas de usuario, y comprende una serie de *Widgets* que pueden ser utilizados para el despliegue de la información, por ejemplo los *telepunteros*, *Small Notes*, e intercambio de mensajes entre usuarios, entre otros.

Los *Widgets* son un conjunto de aplicaciones cliente que comparten un espacio de ejecución provisto por un entorno común, denominado *WidgetEngine*. El *WidgetEngine*, representa un conjunto de servicios y reglas que permite la coexistencia de distintas vistas de información compartiendo recursos de comunicación o de espacio, etc.

DISCIPLE puede ser vista como una aplicación semi-completa que puede ser personalizada para generar una aplicación a la medida. Es el usuario final quien selecciona los elementos que de tarea que finalmente utiliza. Finalmente es un contenedor compartido para JavaBeans, con la capacidad de cargar y manipular con operaciones *drag and drop* las clases.

Dentro de las propuestas para la representación del dominio de este tipo de aplicaciones [18] expone el uso de *Coordination Diagram based synchronous Session Model* (CDSM): un modelo para la representación de usuarios, herramientas y datos, que facilita la coordinación enriquecida de las actividades de una sesión de colaboración sincrónica. Pretendiendo modelar flujos de datos, herramientas, usuarios y relación entre usuarios. En forma complementaria, estos autores presentan una *Session Management*

*Architecture*, basada en el CDSM, un enfoque en capas donde se mantiene separación entre coordinación y aplicación.

En [19] se recuerda que al integrar el contexto del usuario como otro criterio en la evaluación de las peticiones de usuario puede reducir el riesgo de sobrecarga de información de *Awareness*, proponiendo perfiles predefinidos que deben ser aplicados bajo ciertas condiciones del contexto, todo esto configurado en un entorno denominado BW-M. BW-M define el procesamiento de la información en tres fases: registro, captura de eventos; monitoreo, una vez capturada la información esta es reportada al *framework* en forma de objetos; y notificación, fase en la cual el sistema responde a las peticiones de información de un usuario.

A continuación se listan algunas características compartidas por estos referentes y se comparan a los alcances de nuestra propuesta:

- Las arquitecturas propuestas en los referentes expuestos, no analizan distintos enfoques de *Awareness*, lo cual sugiere un manejo general de las distintas fuentes de información y su procesamiento. Nuestra propuesta, busca establecer pautas coherentes con el manejo diferenciado que debe darse a fuente y tipos de información que conforman el servicio de *Awareness* desde una perspectiva integral.
- Los aspectos centrales considerados son la heterogeneidad y sobrecarga de información, dedicando poco esfuerzo en la consolidación de una base conceptual o tecnológica para la distribución o análisis de los datos
- Estas propuestas se construyen dentro de la misma herramienta de soporte a la colaboración, lo cual no facilita su reutilización o aplicación en los casos donde se busca mejorar una plataforma existente. Además, implica la duplicidad de esfuerzos en la captura y análisis de información. Mientras tanto, la arquitectura que se propone busca el desarrollo una plataforma integral del *Awareness* y la integración de distintas fuente que representan un desarrollo transversal a los entornos colaborativos.
- En ninguno de los casos analizados, se sugiere un modelo de integración a herramientas ya desarrolladas, lo cual hace ambigua su aplicación en casos donde lo que se busca no es desarrollar una herramienta para el soporte a la colaboración sino mejorarla a partir de la integración de servicios de *Awareness*. Nuestra arquitectura propone el desarrollo de una red de gestión paralela a los servicios de soporte a la colaboración con una integración modular con otras herramientas.

### **1.2.2 Uso del Paradigma de Agentes dentro de los entornos CSCL**

El segundo tipo de referentes, involucra el uso del Paradigma de Agentes dentro de los entornos CSCL. En este caso, sus aportes más relevantes se relacionan con la adaptación y personalización del contenido educativo según las preferencias del usuario y al desarrollo de servicios auxiliares como buscadores especializados o sistemas tutor inteligentes encargados de ofrecer ayuda en ciertos procesos de coordinación o de asistencia técnica [21] [22] [23]. Sin embargo, los aportes relacionados con facilitar la difusión o manejo de información de *Awareness* son muy pocos [21]; estas propuestas se

han limitado a soportar herramientas como Chat, Foros, Correo Electrónico y Repositorios Compartidos de Documentos, sin plantear estructuras encaminadas a la interoperabilidad entre aplicaciones que garanticen el seguimiento continuo de las actividades de un usuario.

Este trabajo se propone una arquitectura escalable de servicios, primero desde una perspectiva conceptual y luego especializada a través de la tecnología de Agentes: una red cooperativa de agentes cuya responsabilidad es brindar seguimiento continuo a las actividades de usuario mientras provee facilidades para crear un conjunto de servicios que utilicen la información de *Awareness* para promover la colaboración entre las personas.

Los agentes de esta propuesta buscan favorecer los entornos CSCL capturando y procesando la información de *Awareness*, pero además, cooperando para lograr una dinámica superior a la que pudiera alcanzar servicios aislados. De forma simplificada, un Agente es entonces un productor y consumidor de *Awareness* donde sus comportamientos implementan servicios que buscan favorecer la dinámica de grupo y facilitar el desarrollo de sus procesos.

En resumen los elementos característicos de esta propuesta son:

1. El uso de Agentes como elemento activo en la captura, transmisión, análisis, despliegue y aprovechamiento de la información de *Awareness*, de tal forma que cada usuario es representado por una entidad software.
2. La integración de una red de agentes de servicio que cooperan en el tratamiento de la información y posibilitan procesos de personalización, adaptación, seguimiento, intervención y asistencia inteligente sobre los escenarios de enseñanza aprendizaje.
3. El manejo independiente de la herramienta colaborativa y la gestión de la información de *Awareness*.

A continuación se presentan los alcances propuestos para esta investigación en términos de objetivos generales y específicos.

### 1.3 Objetivos

El objetivo principal de este trabajo de maestría es **Definir una propuesta Arquitectónica para el soporte de servicios de *Awareness* en Entornos CSCL como Estrategia para la Mejora de los Procesos de Gestión de Conocimiento en Grupos de Trabajo**. La consecución de este objetivo principal implicó conseguir los siguientes objetivos específicos.

- a) Establecer un contexto técnico y tecnológico de las arquitecturas de sistemas para servicios del *Awareness* en el marco de la gestión de conocimiento.
- b) Proponer una arquitectura de referencia para el soporte del *Awareness* en el marco de sistemas de gestión de conocimiento.

- c) Validar a través de un prototipo la arquitectura propuesta en el escenario de un proyecto que sintetice y comparta conocimiento desde fuentes externas (KnowCat).
- d) Validar a través de un prototipo la arquitectura propuesta en el escenario de un proyecto que identifique fuentes y redes de experiencia (Digital Workbook).

#### **1.4 Metodología**

La forma de abordar el escenario problemático del cual parte esta investigación fue el siguiente: en primer lugar, se realizó una indagación de referentes tecnológicos, marcos de trabajo y arquitecturas, que pudiera dar soporte a los procesos de gestión de información de grupo en entornos CSCL. Luego de analizar sus principales características, se realizó el proceso de determinar los elementos más pertinentes que permitieran dar soporte a la gestión del *Awareness* y con ello mejorar los escenarios de enseñanza/aprendizaje, de lo cual se derivaron los principios de diseño para la propuesta arquitectónica planteada.

Una segunda fase en la evolución de esta investigación la constituyó la selección y construcción de casos de estudio: el primero, el desarrollo de un módulo de *Awareness* para monitorizar y analizar los procesos colaborativos que se producen durante la interacción de los usuarios con el sistema de gestión de conocimiento *KnowCat* [24] [25] [26] [27] [28], el segundo, el desarrollo de una herramienta de soporte a la construcción colaborativas de discusiones en el marco del proyecto "*A Digital Workbook Tool to Support Asynchronous Collaboration*" y la implementación de referencia de un Sistema Multi-Agente [29].

Mientras que el primer escenario permitió la validación del Modelo Conceptual para la Gestión del *Awareness*, el segundo escenario ofrece una visión más completa y compleja del entorno tecnológico que propone esta tesis basada en el paradigma de Sistemas Multi-Agente, además de configurar un entorno para la integración de nuevos servicios, herramientas y tecnologías, orientadas a la mejora de los procesos de gestión de conocimiento en el marco del CSCL.

Cada una de los casos de estudio fueron evaluados con usuarios reales, estudiantes de distintas universidades, gracias a convenios de trabajo establecidos entre la Universidad del Cauca con la Universidad Autónoma de Madrid en España y la Universidad de Chile. Estas evaluaciones fueron orientadas siguiendo los principios de Diseño Centrado en el Usuario [30] los cuales brinda un enfoque más cercano a la percepción de los usuarios.

Finalmente, estos resultados fueron consolidados y divulgados a través de distintas publicaciones para que la comunidad científica pudiera conocer y realimentar esta propuesta.

#### **1.5 Contribuciones**

A continuación se describen los principales aportes obtenidos durante el desarrollo de este trabajo:

- Definición de una Arquitectura Multi-Agente para el soporte a la gestión de información de *Awareness*, validada en entornos CSCL, como estrategia para la

mejora de los procesos de enseñanza/aprendizaje consecuencia de una interacción mayor y más cercana al enfoque colaborativo.

- Desarrollo de un Framework de referencia, basado en la arquitectura propuesta, que permite la captura, distribución, procesamiento e interpretación de información de *Awareness* entre Agentes Software que constituyen un escenario para la construcción de servicios orientados a la mejora de los entornos de aprendizaje colaborativo.
- Framework de desarrollo para la construcción de Sistemas Multi-Agente utilizando tecnología .NET, denominado AgNet.
- Construcción de herramientas CSCL, utilizadas como casos de estudio para la validación de la arquitectura propuesta.
- Desde el aspecto social, el mejoramiento de los entornos de enseñanza/aprendizaje contribuye a la masificación y apropiación de nuevo conocimiento lo cual favorece el desarrollo de una Sociedad de Conocimiento y todos los beneficios que esto conlleva [31].

## 1.6 Estructura del Documento

Este trabajo de maestría se ha dividido en seis capítulos. A continuación se comentan los cinco restantes.

**Capítulo II, Marco Conceptual:** describe los principales conceptos que fundamentan esta investigación. En primer lugar se aborda la definición de los entornos CSCL, luego se profundiza sobre el *Awareness* y finalmente el paradigma de Agentes desde la perspectiva de la metodología GAIA.

**Capítulo III, Una Arquitectura para la gestión del Awareness:** presenta en forma detallada la Arquitectura propuesta, primero desde un modelo conceptual y luego su especialización a través de un Sistema Multi-Agente.

**Capítulo IV, Escenarios de Validación:** describe el diseño y detalles de construcción de los dos escenarios de validación propuestos para la arquitectura, el modulo de *Awareness* para KnowCat y el Libro Digital de Trabajo junto con el Sistema Multi-Agente.

**Capítulo V, Experimentación:** contiene una descripción general de las diferentes pruebas realizadas y resultados obtenidos en la fase de validación de los artefactos diseñados.

**Capítulo VI, Conclusiones y Trabajos futuros:** presenta un resumen de los resultados alcanzados, recomendaciones y líneas de trabajo futuro resultado de esta investigación.

Finalmente, **Referencias y Anexos.**

## Capítulo 2. Marco Conceptual

En este capítulo se presenta una descripción general de los conceptos relacionados con la definición de una propuesta arquitectónica para el soporte los servicios *Awareness* en los entornos CSCL. En primer lugar se hace mención a las características generales de este tipo de entornos, luego se define el concepto *Awareness* desde la perspectiva de la Ingeniería de la Colaboración y finalmente se listan los aspectos más relevantes del paradigma de Agentes y con ello se introduce la metodología GAIA y el desarrollo de Sistemas Multi-Agente.

### 2.1 Aprendizaje colaborativo asistido por computadora CSCL.

Los ambientes de enseñanza/aprendizaje se han caracterizado dentro del aula de acuerdo a los siguientes términos [32]: *Competitivos* donde cada estudiante compite con sus compañeros y lucha por recompensas; *Individualistas* el estudiante trabaja de forma independiente, tiene metas individuales y existe poca interdependencia con sus compañeros, *Cooperativos* supone que los objetivos de cada miembro del grupo están interrelacionados y *Colaborativos*, donde cada integrante del grupo logra sus objetivos si el resto logra los suyos. En donde los ambientes de aprendizaje colaborativo son aquellos que favorecen de mejor forma el desarrollo de habilidades intelectuales de orden superior, tales como: resolución de problemas, pensamiento crítico, creativo, metacognitivo y retención de información [33].

Dado que en muchos referentes se ha encontrado que los términos aprendizaje cooperativo y aprendizaje colaborativo son utilizados indistintamente como sinónimos, es pertinente hacer una diferenciación: en el aprendizaje cooperativo el responsable del proceso es el profesor en un ambiente controlado y cerrado, donde el desarrollo personal está supeditado a los objetivos organizacionales, los pasos del proceso grupal se definen claramente y previamente mediante reglas rígidas y las tareas se dividen para llegar a una solución [5]. Por el contrario, en el aprendizaje colaborativo el responsable del proceso es el estudiante en un ambiente abierto y libre que estimula la creatividad, su objetivo es soportar el desarrollo individual y grupal, la motivación está supeditada al compromiso personal pero para encontrar una solución se requiere de la colaboración grupal.

La *colaboración* es una apuesta académica y empresarial para mejorar la productividad y el aprendizaje de los grupos de trabajo. Diferentes experimentos ha comprobado la efectividad de esta forma de trabajo, lo cual atrae el interés de muchas investigaciones por identificar y catalizar los elementos que rodean al fenómeno de la colaboración.

Tanto así, que se utiliza el término Ingeniería de la Colaboración, para denotar la disciplina encargada de diseñar escenarios y herramientas idóneas para el desarrollo de actividades colaborativas, identificando a las Tecnologías de Información y las Comunicaciones como aliados estratégicos [33].

El Aprendizaje es inherentemente un proceso individual, no colectivo, pero está influenciado por una variedad de factores externos, incluyendo interacciones grupales e interpersonales. Estas interacciones están relacionadas al uso de un lenguaje en la reorganización y modificación de entendimientos comunes. Por lo tanto, el aprendizaje es un fenómeno social y privado al mismo tiempo [34]. No obstante, el aprendizaje colaborativo no necesariamente significa aprender dentro de un grupo, implica la

posibilidad de contar con otras personas, y dar retroalimentación en un ambiente no competitivo, así aprender colaborativamente involucra compartir un conocimiento, interactuar e intercambiar roles. La colaboración envuelve sinergia y asume que de alguna forma “el todo es mayor que la suma de sus partes individuales”, esto significa, que el aprendizaje colaborativo puede producir beneficios mayores que el aprendizaje individual. No todos los intentos en aprender colaborativamente tendrán éxito. Bajo ciertas circunstancias este significa pérdida de un proceso, falta de iniciativa, mal entendimiento [36].

Por lo cual, el aprendizaje colaborativo puede ser definido entonces como la adquisición de conocimientos, habilidades o actitudes por parte de un individuo a partir de su interacción con otros [10]. En este contexto, se requiere un cambio de actitud de los actores que hacen parte del escenario educativo, principalmente estudiantes y profesores. Los estudiantes deben ser ahora protagonistas de su proceso formativo, construyendo activamente su propio conocimiento. Mientras tanto, el esfuerzo del profesor debe centrarse en desarrollar en los alumnos, talentos y competencias utilizando nuevos esquemas de enseñanza, asumiendo el rol de *facilitador* durante el proceso de enseñanza/aprendizaje [7] [35].

El uso de la tecnología juega un rol preponderante en los escenarios educativos actuales, en el contexto del aprendizaje colaborativo, surge como línea de investigación en las ciencias computacionales el Aprendizaje Colaborativo Apoyado por Computadora CSCL, el cual representa un paradigma multidisciplinar dentro del campo de las Tecnologías aplicadas para Mejorar la Educación (*Technology Enhanced Learning*, TEL), donde las Tecnologías de la Información y las Comunicaciones (TIC) se emplean en el fin de mejorar el aprendizaje y la enseñanza en diversos ámbitos de la educación. Una de las principales características de los entornos CSCL, es considerar la interacción social como elemento esencial del aprendizaje [37], por tal razón se requiere que haya interacción entre los participantes aun cuando no se les pida que colaboren.

Entre los retos que un entorno de aprendizaje colaborativo debe afrontar se encuentran: abandonar de la idea (por parte de los estudiantes) de que el profesor lo sabe todo; el profesor es un facilitador que da a los alumnos la oportunidad de desarrollar su propio aprendizaje; lo más importante es la adquisición de meta-conocimientos, donde el aprendizaje y la solución de problemas se hace en grupo; lograr el aprendizaje del estudiante a través del constructivismo, la colaboración y la enseñanza a los otros; motivación de los estudiantes por situaciones del mundo real.

Por lo tanto, las soluciones CSCL requieren de un conocimiento interdisciplinar, puesto que es necesario saber qué factores influyen en el aprendizaje, y que factores influyen en la dinámica de trabajo en grupo. Por otra parte se necesita saber cómo comunicarle esta información pedagógica y psicológica a las plataformas computacionales de soporte.

Entre las dificultades que determinan resultados no positivos en algunos ambientes de CSCL se encuentran: (1) las personas no han usado el computador para trabajar en grupo; (2) ofrecen exclusivamente herramientas para soportar el trabajo colaborativo, olvidándose de propiciar actividades cognitivas, y de actividades que propicien realmente el trabajo en grupo; (3) usualmente tienen objetivos educativos específicos, pero estos no están integrados dentro de otras actividades en las cuales los estudiantes puedan participar y (4) no hay una integración entre las herramientas con el entorno en general.

Las propuestas CSCL argumentan que bajo la práctica de actividades en grupo, los estudiantes, tienen la oportunidad de aprender en el proceso grupal, múltiples perspectivas del mismo tópico, liderazgo, manejo de tareas en grupo y comunicación. El computador soporta comunicación y colaboración, permitiendo la creación de situaciones imposibles en el mundo real.

Las ventajas tangibles del uso de la TIC en el proceso de aprendizaje colaborativo son: (1) estimulan la comunicación interpersonal; (2) posibilitan compartir la información más fácilmente, y la toma de decisiones conjunta; (3) permiten hacer seguimiento del grupo, a nivel individual y colectivo; (4) brindan acceso a información y contenidos de aprendizaje; (4) facilitan la gestión y administración de los alumnos; (5) proporcionan herramientas para la creación de ejercicios de evaluación y autoevaluación y (6) las TIC permiten un diálogo real asíncrono muchos a muchos. En resumen, la interacción social se ha convertido en el principal foco de CSCL y la adaptación de los individuos se ha conceptualizado en torno a su papel en el grupo.

El gran objetivo de las aplicaciones de CSCL es permitir que un conjunto de aprendices trabajen de forma remota con la misma simplicidad y naturalidad como si lo hicieran cara a cara [13] lo cual convierte al aprovisionamiento de servicios de Awareness en un requerimiento clave para lograr satisfacer este objetivo.

Los sistemas CSCL son entornos que maximizan la información relevante involucrada sobre una discusión de grupo, lo cual permite mejorar la calidad de las decisiones que tome [36]; por cuanto el máximo potencial de un grupo se presenta cuando todos sus miembros participan en la discusión [37]. Un entorno de soporte al aprendizaje colaborativo posee tres elementos básicos [38]:

- **Comunicación:** enfatizando en el intercambio efectivo y eficiente de información entre agentes remotos por uso de un medio habilitado (texto, gráficos, voz, etc.).
- **Colaboración:** entendido como el aporte derivado de las actividades en grupo. Una colaboración efectiva demanda que las personas compartan información en el contexto del grupo.
- **Coordinación:** la relación entre comunicación y colaboración está basada en la coordinación. Está relacionada con la integración y armonización de los esfuerzos individuales para alcanzar un objetivo superior, en los cual hay una fuerte incidencia de la tecnología y los protocolos sociales.

La siguiente sección expone algunos de los fundamentos la ingeniería de la colaboración y más aun del *Awareness* como la herramienta a utilizar para mitigar algunos de los problemas identificados en esta propuesta y procurar un entorno apropiado para los ambientes CSCL.

## 2.2 La Ingeniería de la Colaboración

La *colaboración* es una apuesta académica y empresarial para mejorar la productividad y el aprendizaje de los grupos de trabajo. Diferentes experimentos ha comprobado la efectividad de esta forma de trabajo, lo cual atrae el interés de muchas investigaciones por identificar y catalizar los elementos que rodean al fenómeno de la colaboración. Tanto así, que se utiliza el término Ingeniería de la Colaboración, para denotar la disciplina

encargada de diseñar escenarios y herramientas idóneas para el desarrollo de actividades colaborativas, identificando a las Tecnologías de Información y las Comunicaciones como aliados estratégicos [33].

Como se mencionó en el capítulo I, un escenario óptimo para la colaboración establece condiciones de *igual participación, interdependencia positiva, capacidad de respuesta individual y conciencia compartida del problema* o *Awareness* [9]. A continuación se define el concepto de *Awareness* por ser este el que se busca aprovisionar en esta investigación.

### 2.3 *Awareness*

El concepto de *Awareness* en el contexto del CSCW (Trabajo Colaborativo Asistido por Computador), busca reducir el esfuerzo necesario para lograr una comunicación natural y eficiente durante la ejecución de un proceso colaborativo [39]. Se definen distintos mecanismos de *Awareness* que buscan establecer procesos de comunicación implícita (o meta-comunicación) entre los participantes para mostrar la mayor cantidad de información relevante en la toma de decisiones, la cual puede ser presentada de distintas formas (textos, sonidos, imágenes, animaciones) y son el reflejo de lo que los usuarios hacen, saben, dudan, ven, etc. La importancia de soportar este concepto radica en que esta información transmitida ayuda a los actores a cambiar de un escenario individual a uno grupal [40]. No obstante, hay que tener cuidado como se aprovisiona el *Awareness* dado que en ningún caso se prenda saturar las interfaces graficas con información no relevante que reduzca la eficiencia en el desempeño de las tareas realizadas por el usuario.

Por otra parte, existen distintos aspectos del contexto sobre el cual se ejecuta una tarea, que pueden ser de interés para procurar una mejor colaboración del equipo de trabajo, lo cual origina distintos enfoques del concepto de *Awareness* [41]. A continuación se listan los encontrados en la literatura:

- ***Awareness Social***: se centra sobre las expectativas del grupo y cada uno de sus integrantes, además de la forma en la cual interactúan y el rol que desempeñan.
- ***Awareness de Tarea***: información relacionada con la estructura y conceptos de la tarea. Información de evaluación de resultados y herramientas o materiales necesarios para completar la tarea.
- ***Awareness de Concepto***: expone el conocimiento presente en los integrantes, disponible como inexistente, lo cual incentiva a cuestionar las ideas actuales y enriquecerlas con nueva información. Brinda un espacio para la definición de hipótesis sobre el trabajo.
- ***Awareness de Espacio de Trabajo***: trata aspectos del entorno de la tarea tales como la ubicación, actividades y acciones presentes y/o futuras [9].
- ***Awareness de Conocimiento Compartido SKA*** [6]: centra su atención en el proceso mismo de construcción de conocimiento por parte del grupo. Va mas allá de la comprensión compartida del problema y se compone de diversos aspectos tales como la coordinación, la estrategia de comunicación y el monitoreo. Es una propuesta integral de información.

Cada enfoque, sugiere además la existencia y manipulación de diversas fuentes de información de las cuales, luego de un proceso de análisis, se puede generar *Awareness*, algunos ejemplos de dichas fuentes son los usuarios, las herramientas, el entorno de ejecución, las condiciones de red, periféricos, etc.

Desde una perspectiva general, los sistemas de soporte para la colaboración no demandan el soporte de un solo tipo de *Awareness* ni su satisfacción rigurosa. Existen distintas condiciones visibles en tiempo de ejecución que exigen un soporte flexible y coordinado de distintos proveedores de *Awareness* para satisfacer las necesidades de información basadas en el contexto puntual del usuario.

## 2.4 Sistemas Multi-Agente

Los Sistemas Multi-Agente (SMA) es una rama de la Inteligencia Artificial Distribuida (IAD) y están constituidos por un conjunto de tareas a realizar, recursos y entidades inteligentes llamadas agentes que coordinan sus habilidades para la resolución de problemas individuales o globales [42]. Estos sistemas considerados como un todo, exhiben características particulares, tales como [43]:

- **Organización Social:** es la manera como el grupo de agentes está constituido en un instante dado puede ser estática o dinámica, dependiendo de las funciones o tareas de cada agente, está relacionada con la estructura de los componentes funcionales del sistema, sus características, sus responsabilidades, sus necesidades y la manera como realizan sus comunicaciones. La organización en los SMA depende del tipo de comunicación y el modo de cooperación entre agentes, así como del tipo de agentes que conforman el grupo. En general se pueden distinguir cuatro tipos de configuraciones organizacionales: (1) centralizada, donde existe un agente que controla la interacción de los demás agentes del sistema, (2) horizontal, en la cual todos los agentes que integran un sistema están al mismo nivel, es decir, no hay ningún agente que haga las veces de maestro o supervisor, ni tampoco agentes esclavos, (3) jerárquica, que se da cuando los agentes trabajan en diferentes niveles de abstracción de un problema y (4) *ad hoc*, mezcla de las tres anteriores, se caracteriza porque la dinamicidad de la estructura está regida por el ajuste mutuo entre los pequeños grupos de agentes en el sistema.
- **Cooperación:** permite a los agentes compartir resultados intermedios que lleven al progreso en la resolución de las tareas de otros agentes y al progreso de la solución global que debe alcanzar el sistema. Existen varios modelos de cooperación, dentro de los cuales se pueden mencionar: (1) *Cooperación compartiendo tareas y resultados*, los agentes tienen en cuenta las tareas y los resultados intermedios de los demás para realizar las tareas propias; (2) *Cooperación por delegación* donde un agente maestro descompone una tarea en subtareas y las distribuye entre los agentes esclavos, para que sean resueltas, después, el supervisor integra las soluciones para hallar la solución al problema inicial; (3) *Cooperación por ofrecimiento* un agente maestro descompone una tarea en subtareas y las difunde en una lista a la que tienen acceso los agentes que integran el sistema, esperando que ellos ofrezcan su colaboración de acuerdo a sus habilidades, el supervisor escoge entre los ofrecimientos y distribuye las subtareas [7].

- **Coordinación:** permite a un grupo de agentes considerar todas las tareas a realizar y coordinarlas para no ejecutar acciones no deseables, como evitar acciones redundantes. Hay varios modelos de coordinación de acciones entre agentes, entre los cuales se encuentra: *Coordinación Global*, cuando el SMA determina y planifica globalmente las acciones de los diferentes agentes; *Coordinación Individual* cuando el SMA le da completa autonomía a los agentes, es decir, cada agente decide qué hacer y resuelve localmente los conflictos que detecte con otros agentes. Además de los modelos existen dos tipos de coordinación: la primera orientada por los problemas donde los agentes deben coordinar los planes de realización de acciones para prevenir interbloqueos, repetición de acciones y creación de inconsistencias y la segunda la orientada por la cooperación donde los agentes no se coordinan a nivel de planes, sino a nivel de acciones, esto significa que los agentes se coordinan en el momento de ejecutar acciones.
- **Negociación:** proceso de toma de decisiones en el cual dos o más entidades, representando sus propios intereses, se ponen de acuerdo acerca de cómo resolver un conflicto de preferencias, de tal forma que beneficie a todos [44]. Los procesos de negociación tienen como resultado la modificación o confirmación de las creencias de cada agente involucrado, en lo relacionado con los demás agentes y con el mundo en el que se desenvuelve. Los mecanismos de negociación utilizan diferentes reglas en lo relacionado con la distribución de recursos del sistema [45] [46].
- **Control:** es el mecanismo básico que provee apoyo para la implementación de mecanismos de coordinación en un SMA. El control se relaciona directamente con: determinar cuáles son las subtareas más importantes a realizar en un momento dado, definir el contexto (resultados intermedios de otros agentes) que deben ser usados en la solución de una subtarea, estimar el tiempo de generación de la solución a una subtarea y evaluar si la solución de un problema ha sido generada.

Debido que un SMA es un conjunto de agentes a continuación se define este.

#### 2.4.1 Agente

Un agente es una entidad física o abstracta que puede percibir su ambiente a través de sensores, es capaz de evaluar tales percepciones y tomar decisiones por medio de mecanismos de razonamiento sencillos o complejos, comunicarse con otros agentes para obtener información y actuar sobre el medio en el que se desenvuelve a través de ejecutores [47].

Un agente puede ser clasificado por su rol o papel en una sociedad de agentes (SMA al que pertenece), por su especialidad o actividad en la que es experto, por sus objetivos o metas en el SMA, por su funcionalidad o lo que es capaz de hacer, por su capacidad de comunicación y por su capacidad de aprendizaje [48].

Un Agente Cognitivo es aquel que es capaz de efectuar operaciones complejas, es individualmente inteligente (posee criterios de evaluación de acciones y de selección, además es capaz de justificar sus decisiones), puede comunicarse con los demás agentes y llegar a un acuerdo con todos o algunos de ellos, sobre alguna decisión. Un sistema cognitivo está compuesto por un pequeño número de agentes cognitivos.

Un Agente Reactivo es un agente de bajo nivel, que no dispone de un protocolo ni de un lenguaje de comunicación y cuya única capacidad es responder a estímulos, estos no son individualmente inteligentes, sino globalmente inteligentes. Los sistemas reactivos por lo general están compuestos por un gran número de agentes reactivos.

Es posible tener sistemas heterogéneos cuyo comportamiento se derive de los dos tipos de agentes y tenga características de ambos. Es decir, es posible dotar a los agentes cognitivos de capacidades de reacción a los eventos, a tales agentes se les puede llamar Agentes Híbridos [7].

### 2.4.2 Metodologías para Sistemas Multi-Agente

Las metodologías tradicionales para diseñar sistemas *software* (Unified Process, CommonKADS, etc.) no tienen en cuenta las necesidades de diseño de las arquitecturas orientadas a agentes, como pueden ser la especificación de planificación de tareas, la comunicación mediante lenguajes específicos, la movilidad del código o la motivación de los componentes del sistema (los agentes). Por lo cual existen metodologías de diseño específicas para SMA, entre las cuales se encuentran [49] [50]: Vowel Engineering, GAIA [51], MaSE [52], MAS-CommonKADS [53] y MESSAGE/UML [54], además de la arquitectura BDI [55], de las cuales se eligió a GAIA dado que la especificación de los requerimientos es completamente independiente del proceso de análisis y diseño del sistema. A continuación se describe con más detalle la metodología GAIA.

### 2.4.3 GAIA

GAIA es una metodología para el desarrollo de aplicaciones bajo el paradigma de Agentes [49]. Las aplicaciones para las que se considera esta metodología son grandes sistemas del mundo real con las siguientes características:

- Los agentes son sistemas de cómputo de grano grueso, de modo que cada uno de ellos realiza un uso importante de recursos.
- Estos agentes son heterogéneos, en el sentido de que pueden ser implementados en diferentes lenguajes, tecnologías o arquitecturas.
- La organización de la estructura del sistema (las relaciones entre los agentes), así como las habilidades y servicios proveídos por los agentes son estáticos, en el sentido de que no varían en el tiempo.
- El sistema al completo suele contener un número relativamente bajo de tipos de agentes (normalmente menos de 100).

GAIA separa los conceptos que son entidades puramente abstractas (que, por lo tanto, no tienen porque tener una imagen directa de sí mismos sobre la implementación), de lo que son conceptos concretos, o lo que es lo mismo, entidades que tienen un reflejo directo de sí mismas en la implementación del sistema. Todos estos conceptos, pertenecientes a las fases de análisis y diseño, respectivamente, pueden verse reflejados en la Tabla 1.

**Tabla 1. Conceptos abstractos y concretos en GAIA**

<b>Conceptos abstractos (análisis )</b>	<b>Conceptos Concretos (diseño)</b>
<b>Roles</b>	Tipos de agentes (clases)
<b>Permisos</b>	Servicios
<b>Responsabilidades</b>	Conocimientos
<b>Protocolos</b>	
<b>Actividades</b>	

Propiedades dinámicas	
Propiedades de seguridad	

### 2.4.3.1 Fase de análisis

El objetivo de la fase de análisis es el de desarrollar y comprender el sistema y su estructura sin hacer referencia a ningún detalle de implementación. En este caso, se trata de desarrollar la organización del sistema de agentes, entendiendo el término como un conjunto de roles organizados que mantienen ciertas relaciones unos con otros, entendiendo rol como una descripción abstracta de la función esperada de una determinada entidad.

Un rol queda definido mediante 4 atributos: responsabilidades, permisos, actividades y protocolos. Las responsabilidades definen las funcionalidades del rol, estas se dividen en 2 grupos: las propiedades dinámicas (cosas buenas que pueden ocurrir) y las propiedades de seguridad (las cuales hay que respetar para que nada malo ocurra) [56]. Para llevar a cabo las tareas que especifican las responsabilidades que un rol tiene, se establecen una serie de permisos asociados a dicho rol, estos son los que determinan los recursos que el rol puede disponer y cuál es la cantidad máxima que tienen disponible de estos recursos. GAIA utiliza una notación formal basada en la notación FUSION [57] para expresar los permisos.

Cada uno de los roles tendrá que realizar una serie de cálculos internos sin necesidad de interactuar con otros roles, los cuales sin discutir su finalidad se les denomina actividades. Por otra parte, cada rol puede tener asociados una serie de protocolos que le sirven para establecer comunicación con otros roles [58].

La representación del modelo organizacional en GAIA se encuentra subdividida en dos modelos: el modelo de roles y el modelo de interacción. El modelo de roles identifica los roles principales que aparecen en el sistema y los describe a través de un conjunto de esquemas, uno para cada rol del sistema; dichos esquemas se muestra en la tabla 2. El modelo de interacción representa las dependencias y relaciones entre los distintos roles y la organización Multiagente; estas interacciones son el punto central del funcionamiento del sistema y consiste en un conjunto de definiciones de protocolos, uno para cada tipo de interacción entre roles. Una definición de protocolo consta de un propósito, el rol iniciador, el rol receptor, las entradas (información usada por el iniciador durante la comunicación), las salidas (Información provista por el receptor durante la interacción) y el procesado (breve descripción de todo cálculo que el iniciador realice durante la comunicación).

**Tabla 2. Plantilla para construir esquemas de roles**

<i>Esquema del rol</i>	<i>Nombre del rol</i>
<b>Descripción</b>	Descripción corta del rol en lenguaje natural
<b>Protocolos y Actividades</b>	Protocolos y actividades en los que el rol participa.
<b>Permisos</b>	“Derechos” asociados a este rol.
<b>Responsabilidades Dinámicas</b>	Responsabilidades dinámicas
<b>Seguridad</b>	Responsabilidades Seguridad

En resumen, el proceso a seguir en la fase de análisis en GAIA es:

1. Identificar los roles del sistema (individuos, departamentos, organizaciones, etc.), y crear un prototipo de modelo de roles.
2. Para cada rol, identificar y documentar los protocolos asociados, produciendo un modelo de interacción.
3. Usando el modelo de interacción como base, refinar y elaborar el modelo de roles.
4. Repetir los pasos 2 y 3 hasta obtener el nivel de detalle deseado.

#### **2.4.3.2 Fase de diseño**

El objetivo principal de todo proceso de diseño es la transformación de los modelos abstractos creados durante el análisis, en modelos con suficiente nivel de detalle como para poder ser implementados. Sin embargo, este no es el caso del diseño orientado a agentes. En GAIA se pretende transformar el resultado del análisis en un modelo con el suficiente detalle, como para poder aplicarle técnicas clásicas de diseño orientado a objeto, que nos lleven a una posterior implementación del sistema y para ello se generan 3 modelos: el modelo de agentes, el modelo de servicios y el modelo de conocidos.

**El modelo de agentes:** documenta los distintos tipos de agentes que serán utilizados en el sistema y las instancias que se crearán en tiempo de ejecución. Un tipo de agente no es más que un conjunto de roles que guardan una cierta relación entre sí. Este modelo se define mediante un árbol con nodos de tipos de agente y cuyas hojas corresponden con los roles del modelo de roles.

**El modelo de servicios:** identifica los servicios asociados con cada rol-agente y especifica sus principales propiedades; cada servicio corresponde a una actividad identificada en la fase de análisis, aunque no todos los servicios provienen necesariamente de actividades. Para cada servicio que pueda proporcionar un agente es necesario documentar sus propiedades: entradas, salidas, precondiciones y post-condiciones. Las entradas y salidas derivarán de forma obvia del modelo de interacción, mientras que las precondiciones y post-condiciones representan restricciones sobre estos servicios, y son derivadas de las propiedades de seguridad del rol. La metodología GAIA no incluye una forma de implementar los servicios que documenta.

**El modelo de conocidos:** define los enlaces de comunicación que existen entre los tipos de agentes, sin entrar en el detalle de los mensajes que son enviados, solamente indicando que existe la vía de comunicación. El objetivo concreto de este modelo es la identificación de cuellos de botella en las comunicaciones. La representación del modelo es tan sencilla como un grafo dirigido, con los nodos correspondiéndose a los tipos de agente y los arcos a caminos de comunicación. Este modelo puede ser derivado apropiadamente de los modelos de roles, interacciones y agentes.

En resumen, el proceso de diseño a partir de los documentos del análisis es como sigue:

- a. Crear el modelo de agentes juntando roles en tipos de agente y formando una jerarquía.
- b. Desarrollar un modelo de servicios examinando las actividades, protocolos y propiedades dinámicas y de seguridad de los roles.
- c. Desarrollar un modelo de conocidos a partir del modelo de interacción y el de agentes.

## Capítulo 3. Una Arquitectura para la Gestión del Awareness

Este capítulo presenta los detalles de la propuesta arquitectónica para el soporte de servicios de *Awareness* en entornos CSCL. Su presentación inicia con la descripción del Modelo Conceptual para la gestión del *Awareness* que se sustenta en tres conceptos: Modelo de Información, Agente y Gestor; luego se expone su especialización a través de un Sistema Multi-Agente, siguiendo las recomendaciones de la metodología GAIA.

### 3.1 Modelo Conceptual para la Gestión del Awareness

El *Modelo Conceptual* de esta propuesta busca afrontar los problemas esenciales de un escenario con fuentes de información distribuidas mediante la definición de tres conceptos: *Modelo de Información*, *Agente* y *Gestor*. En conjunto, estos elementos resumen una serie de estrategias enfocadas a resolver problemas tales como la heterogeneidad, duplicidad de la información, ambigüedad de conceptos y falta de entendimiento a nivel de datos y funciones, mientras sugiere una arquitectura escalable al desarrollo de servicios.

La figura 2, presenta la forma general de un escenario para la gestión del Awareness e identifica los elementos esenciales del modelo. El sistema de soporte a la colaboración ha sido estructurado en forma de red, donde cada nodo representa una fuente potencial de información de Awareness, en este caso, un usuario que interactúa a través de un conjunto de herramientas que le dan soporte a una actividad de grupo. Al suponer distintas herramientas, se presume heterogeneidad y posible falta de entendimiento entre nodos por la carencia de un modelo de comunicación estándar.

Para afrontar estos inconvenientes, la entidad *Agente* envuelve la particularidad del nodo y ofrece un punto de acceso unificado y claro a los datos y funciones que se quiera compartir.

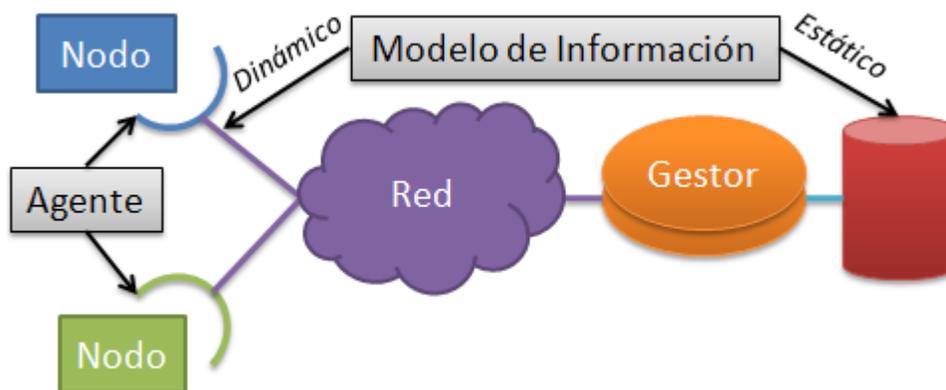


Fig. 2. Modelo Conceptual para la Gestión del Awareness

Por otra parte, para llegar a un mejor entendimiento de los mensajes y artefactos que circulan por la red, es necesario introducir un *Modelo de Información*, entendido como un conjunto de especificaciones que ordenan tanto a nivel de comunicaciones como de persistencia la información a gestionar, lo cual permite llevar registro permanente de la actividad de grupo y proponiendo resolver problemas asociados con la sintaxis y semántica del *Awareness*.

Para complementar las capacidades del Agente y el Modelo de Información, se propone un tipo de entidad de servicio, denominada *Gestor*, que se encargue, por ejemplo, de mantener el registro de la información que luego puede ser analizada para generar *Awareness* o para dar soporte a tareas como el monitoreo de usuario o la evaluación de su desempeño. Uno o más gestores, deben ser especializados en enfoques particulares de *Awareness*, con el fin de dividir la complejidad de los procesos de análisis de datos y de esta forma mejorar la capacidad de respuesta y flexibilidad ante condiciones particulares del contexto, por ejemplo, a través de una composición de servicios que provean los distintos gestores.

A continuación se profundiza sobre la descripción y estrategias de implementación para cada uno de los elementos del modelo.

### **3.1.1 Modelo de Información**

El Modelo de Información aborda aspectos dinámicos y estáticos relacionados con la información del sistema: el primero, mediante la especificación de un *Modelo de Comunicaciones* que definen la sintaxis y semántica de los mensajes que se intercambian entre entidades; y el segundo, a través de un *Modelo de Datos* que estructura la información de *Awareness* para su captura, almacenamiento y análisis.

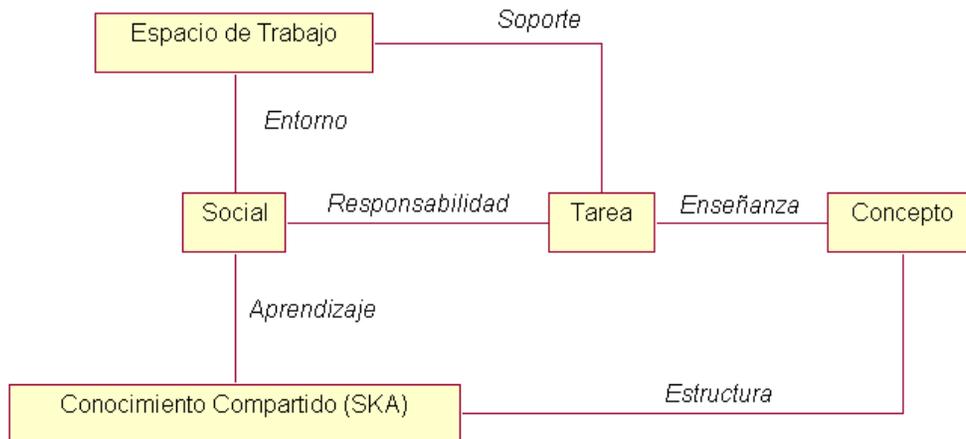
#### **3.1.1.1 Modelo de Comunicación**

El *Modelo de Comunicación* corresponde a un conjunto de clases agrupadas en distintos dominios o paquetes donde cada dominio se relaciona con un enfoque particular de *Awareness* y cada clase representa un tipo de mensaje que circula por la red. Estos mensajes permiten la implementación de los distintos protocolos que toman lugar en los procesos de comunicación.

Si bien, cada enfoque de *Awareness* propone analizar la información desde una perspectiva aparentemente delimitada, en necesario ser flexible y plantear esquemas de información compuestos, para maximizar las oportunidades de interacción basados en lo que hemos definido como *Relaciones de Cooperación* entre enfoques. En cada *Relación de Cooperación* se pueden presentar distintos escenarios que combinan enfoques de *Awareness* para propiciar la colaboración. La identificación de dichos escenarios constituye un criterio de diseño en la implementación de la arquitectura. La figura 3 presenta las seis Relaciones de Cooperación propuestas en este trabajo: *Responsabilidad, Entorno, Soporte, Enseñanza, Estructura y Aprendizaje*. A continuación se describe cada una de ellas y se citan algunos de los escenarios de colaboración a los se pueden llegar:

**Responsabilidad:** vínculo entre los enfoques sociales y de tarea, que representa la planeación de la actividad. Propone escenarios de interacción basados en la asignación y seguimiento de responsabilidades. Algunos escenarios pertinentes para la colaboración según esta relación serían:

- Personas que asumen un mismo rol o tarea.
- Compañeros de grupo que han asumido un rol o tarea.
- Personas que han asumido el rol o tarea con anterioridad.
- Roles complementarios.
- Roles de coordinación.
- Roles de trabajo subordinado.



**Fig. 3. Los distintos enfoques de Awareness y sus relaciones de cooperación**

**Entorno:** relación entre el enfoque Social y de Espacio de Trabajo. Propone escenarios basados en las limitaciones y capacidades del entorno de los usuarios.

- Cercanía de las personas.
- Cercanía a puntos geográficos específicos.
- Entornos de trabajo compatibles, por ejemplo en plataforma de características especiales que requieran un conocimiento o entrenamiento particular.
- Disponibilidad de recursos específicos.

**Soporte:** relación entre el enfoque de Espacio de Trabajo y Tarea. Propone escenarios derivados del soporte de las tareas sobre entornos tecnológicos específicos. Su intención es procurar una interacción basada en la manipulación de artefactos específicos y tratar de establecer formas de comunicación, incluso si estos no están siendo manipulados a través del mismo tipo de herramienta.

- Entornos especiales para el soporte de tareas.
- Tareas que requieran entornos de trabajo específicos.
- Identificación de recursos necesarios para resolver una tarea.
- Manipulación de los mismos artefactos: interacción por concurrencia de objetos.

**Enseñanza:** relación entre el enfoque de Tarea y Concepto. Propone escenarios derivados de la conceptualización pedagógica detrás de las tareas. Utilización de los objetivos de enseñanza como criterios para propiciar la interacción.

- Tareas que se sustentan sobre conceptos similares.
- Conceptos que pueden ser asimilados a través de distintas tareas.
- Identificación de expertos en conceptos que pueden brindar soporte a tareas.

**Estructura:** relación entre el enfoque de Concepto y de Conocimiento Compartido. Definición de escenarios donde se propicie la conversión de conocimiento implícito no estructurado en conocimiento explícito estructurado.

- Transformación de aportes en experiencia de grupo e incluso conocimiento.

- Estructuración de aportes.
- Gestión de experiencias.
- Aportes de constituyen la fundamentación de un Concepto.

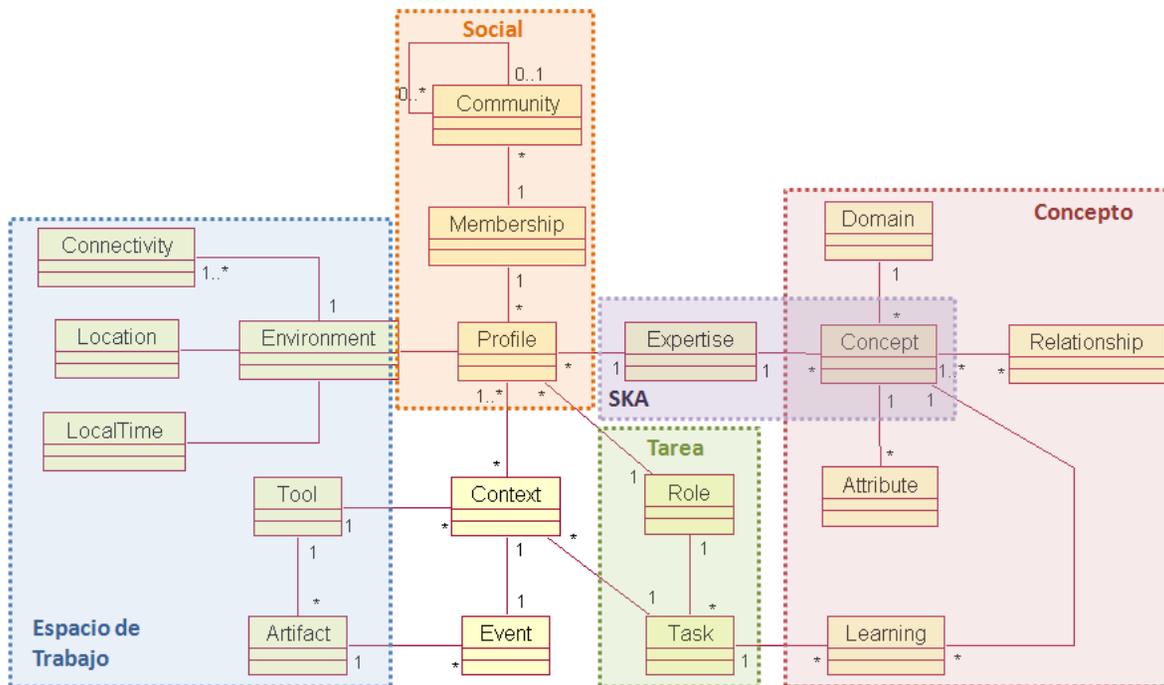
**Aprendizaje:** relación entre el enfoque Social y de Conocimiento Compartido. Propone escenarios de socialización del conocimiento.

- Realización de aporte en el contexto de una tarea.
- Evaluación de aportes.
- Formulación y resolución de inquietudes formuladas por otros usuarios.
- Construcción compartida de aportes.
- Manifestación de dudas o comentarios a los aportes de otro usuario.
- Búsqueda de autores.

A continuación se presente el enfoque de estructura a la información de Awareness.

### 3.1.1.2 Modelo de Datos del Awareness

La figura 4, muestra el conjunto de clases propuesto como referencia para la definición del modelo de datos para la gestión del *Awareness*, además, permite ver la relación entre estas entidades y los distintos enfoques. A continuación se presenta una descripción del diagrama propuesto.



**Fig. 4. Modelo de Referencia para la Estructuración del Awareness**

**Awareness Social:** representados por las clases *Profile*, *Membership* y *Community*. La clase *Profile* representa la información del perfil del usuario en donde se propone mantener información personal, por ejemplo su nombre completo, correo electrónico, palabra clave para el acceso al sistema, *Avatar* (representación gráfica asociada con el usuario, preferiblemente una fotografía de la persona para facilitar su identificación), un

mensaje personalizable que permita a los usuarios definir una frase o pensamiento que exprese su estado de ánimo u ocupación actual, básicamente es la respuesta a la pregunta ¿Qué estás haciendo o en que piensas?, similar como lo hacen la herramienta social *Twitter* [59]. Adicionalmente, la clase Perfil almacena la información del último acceso y su estado actual de conexión. Las clases *Membership* y *Community* se utilizan para soportar la formación de grupos o comunidades. Una comunidad, puede estar definida dentro de otra y un usuario puede pertenecer a muchas comunidades.

**Awareness de Espacio de trabajo:** para soportar este enfoque, la clase *Environment* agrupa distintas características del entorno tales como: la ubicación del usuario (*Location*), el tiempo local (*LocalTime*) y las condiciones de conectividad (*Connectivity*). En determinados escenarios, la cercanía o la ubicación específica de un usuario, puede ser el criterio que permita procurar la interacción entre dos o más usuarios. Por otra parte, el tiempo local puede ser un factor de interés sobre todo para usuarios cuya ubicación geográfica se vea claramente reflejada en una variación horaria. Finalmente, características de conectividad que permiten un manejo más adecuado de la información y servicios que pueden ser desplegados durante la interacción, permitiendo definir criterios para la negociación de capacidades entre las terminales de usuario antes de intentar lanzar un servicio de interacción.

Por otra parte, la clases *Tool* y *Artifact* establecen un contexto más cercano de las acciones del usuario permitiendo identificar qué herramientas y artefactos están siendo manipulados en un momento dado.

**Awareness de Tarea:** establece el contexto de las acciones de los usuarios mediante un modelado básico de Roles (*Role*) y Tareas (*Task*). Cada tarea establece su secuencialidad o paralelismo con otras, una prioridad (que tiene sentido si es de ejecución paralela) y un tiempo máximo de ejecución, entre otras cualidades que permiten describir su comportamiento. Un Rol, por su parte puede heredar las responsabilidades de otro y de esta forma generar una estructura jerárquica, pero por otro parte puede establecer tipos de relación con otros roles por ejemplo de subordinación/coordinación, complemento y similitud.

**Awareness de Conceptos:** el desarrollo de este tipo de *Awareness* involucra dos aspectos: el primero, la representación del dominio de conocimiento con el cual se diseña la actividad colaborativa, y para ellos se proponen las clases: Dominio (*Domain*), Concepto (*Concept*), Atributo (*Attribute*) y relación (*Relationship*), una versión simplificada de un *metamodelo* para la descripción de conocimiento, interpretado como la definición de dominios que permiten la agrupación de conceptos, los cuales son estructurados a partir de atributos y que además permiten establecer relaciones con otros conceptos. El segundo, la clase Aprendizaje (*Learning*) la cual relaciona los conceptos de conocimiento con las tareas propuestas; lo cual facilita los procesos de análisis de la información de grupo desde el contexto del conocimiento.

**Awareness de Conocimiento Compartido (SKA):** para el soporte a este enfoque se propone una entidad que resume el conocimiento del usuario sobre un concepto del dominio específico como *Experticia* (*Expertise*). La *experticia* ha sido modelada a través de una expresión matemática que basada en la evaluación de aportes, reúne la experticia previa, experticia del evaluador y cuantificación de calidad del aporte para formar un curva de conocimiento. Dicho modelo es tratado en detalle en una sección posterior de este

capítulo. La experticia es el criterio fundamental con el cual se organizan servicios tales como la búsqueda de expertos o el seguimiento del desempeño individual. Las clases *Context* y *Event* permite el almacenamiento persistente de los eventos que han sido generados por los usuarios, lo cual constituye una base de información de bajo nivel.

En la práctica, para la identificación de las entidades que finamente hacen forman parte del *Modelo de Información* se propone una exploración a dos niveles: *Nivel Individual* y *Nivel Grupal*. El *Nivel Individual* representa toda aquella información generada directamente por el usuario a través de su interacción con el sistema; son datos de bajo nivel capturados desde las interfaces de usuario o de eventos de sistema, por ejemplo: eventos de ratón, barras de desplazamiento, teclado, dimensión de ventana, características de la conectividad, hora local, etc. La información a este nivel es abundante pero carece de sentido de grupo, por lo cual debe ser complementada y abstraída en un segundo nivel, el *Nivel de Grupo*, en el cual se propone la conversión de los datos los datos del *Nivel Individual*, en objetos del espacio virtual de los usuarios desde los servicios y enfoques de *Awareness* que se quieren soportar.

A continuación se presenta la descripción y responsabilidades de la entidad Agente.

### 3.1.2 Agente

E Agente, es una entidad de mediación entre el cliente y el sistema de gestión. Su comportamiento se define a través de dos roles: el primero denominado *Sensor*, capturando los eventos que produce el usuario mientras interacción con su entorno y luego replicándolos hacia el sistema de gestión; el segundo, como *Consola de Interacción*, ofreciendo distintas vistas a la información, a través de mecanismos de interacción como los *Widgets*.

El termino *Widget* ha sido utilizado por otros autores [60] para referirse a componentes gráficos de aplicación cuyo objetivo es proveer acceso a información o servicios; para ello, comparte un espacio de ejecución denominado *WidgetEngine*; este concepto a tenido tal aceptación y desarrollo que para los entornos Web la W3C ha propuesto una especificación para su empaquetado y configuración [61]. A continuación se presentan algunos lineamientos para la implementación de los roles propuestos para la entidad Agente.

**Rol Sensor:** la implementación de canales de comunicación con las aplicaciones de usuario locales a través de los cuales facilita la divulgación de eventos de interacción y acceso a servicios. Para logra un entendimiento adecuado entre el entorno y el Agente, se propone incluir entre el Modelo de Comunicación del sistema, una colección de clases que structure su comunicación, lo cual se ha denominado (Tool).

**Rol Consola de Interacción:** los *Widgets* ofrecen acceso con cierto nivel de autonomía a los distintos servicios de *Awareness* que se provean. Las cuales La información de *Awareness* presentada al usuario está contenida en el *GroupDataModel* y la clase *AwarenessSession* se encarga de gestionar la conexión entre Agente y Gestor.

### 3.1.3 Gestor

Entidad encargada del procesamiento de la información y despliegue de servicios. Corresponde al centro de gestión del *Awareness*, determina el tipo de información que se ofrece a los distintos agentes en función de un enfoque particular de *Awareness* para el cual se puede especializar. El despliegue de un servicio podría involucrar la participación

de más de un gestor, lo cual reduce la complejidad en los procesos de análisis de datos, pero aumenta los relacionados con comunicación y coordinación.

Un gestor provee mecanismos de persistencia de datos y gestiona los canales de comunicación. La clase *AwarenessSessionManager* es responsable de combinar la información parcial que viene de los Agentes y gestionar el *GroupDataModel*.

Por otra parte, el Gestor debe dar soporte a distintos roles (Role) dentro del proceso colaborativo que unido con información del contexto (Context) y preferencias de usuario (Preference), implementa filtros (Filter) sobre el *GroupDataModel* para mitigar el problema de sobrecarga de información.

### 3.2 Un Sistema Multi-Agente para la gestión del Awareness

La arquitectura propuesta adopta el paradigma de agentes por ser una metáfora natural para los sistemas cuyo propósito es la interacción entre entidades complejas, autónomas, sociales, dispersas dentro de escenarios heterogéneos, lo cual resulta una abstracción cercana a la forma como las personas piensan y realizan sus actividades cuando trabajan en grupo [62], [63], [64]. Este criterio de diseño fue validados durante la construcción de dos caso de estudio descritos en [24] [29] y descritos en el capítulo 4, su respectiva experimentación se encuentra en [25] [26].

A continuación se presenta el modelado del Sistema Multi-Agente siguiendo las recomendaciones de la metodología GAIA [49], por lo cual dicho modelado se ha dividido en las etapas de Análisis y Diseño. Posteriormente, esta sección presenta las arquitecturas internas para los Agentes.

#### 3.2.1 Análisis

Los artefactos de salida de esta fase son: el Modelo de Roles y el Modelo de Interacción. A continuación se presenta cada uno de ellos.

##### 3.2.1.1 Modelo de roles

Este modelo contiene los distintos *Esquemas de Rol* que caracterizan el Sistema. Cada esquema está definido por una descripción del rol, un conjunto de actividades y protocolos que representa los servicios y actividades internas generales que realiza el rol, lista de privilegios o permisos a nivel de acceso de datos y la lista de responsabilidades o acciones que debe desempeñar el rol; para esta última lista, se ha seguido la convención de anotaciones recomendadas en GAIA, la cuales se resumen en la tabla 3.

**Tabla 3. Convención de anotaciones para lista de Responsabilidad de un Rol**

$x.y$	$x$ seguido de $y$	$x   y$	Ocurrencia de $x$ o $y$
$x^*$	Ocurrencia de 0 o más veces de $x$	$x^+$	Ocurrencia de 1 o más veces de $x$
$x^w$	Ocurrencia infinitamente frecuente	$[x]$	Ocurrencia opcional de $x$
$x//y$	Ocurrencia intercalada de $x$ y $y$		

**Tabla 4. Esquema del Rol Cliente**

Esquema del rol	Cliente
<b>Descripción</b>	Entidad que genera y percibe la información de Awareness mientras participa en actividades de grupo, interactuando con la red de Agentes a través de las interfaces del Sistema.

<b>Protocolos/Actividades</b>	Generar Awareness, Recibir Awareness, Colaborar.
<b>Permisos</b> <b>Lectura</b> <b>Cambio</b> <b>Genera</b>	Perfil (Otros usuarios), Estado (Otros usuarios) Perfil (Propio), Estado (Propio), Configuración de Privacidad Eventos, Mensajes (Awareness)
<b>Responsabilidades</b> <b>Dinámicas</b>  <b>Seguridad</b>	(Iniciar Sesión)+, (Terminar Sesión)+, (Asumir Rol)+, (Asumir Tarea)+, (Proponer Respuesta)*, (Buscar Respuestas)*, [(Buscar Respuestas).(Evaluar Respuestas)] <sup>w</sup> , (Buscar Expertos)*, (Enviar Mensajes)* Existe Registro de Usuario

**Tabla 5. Esquema del Rol Sensor**

<b>Esquema del rol</b>	<b>Sensor</b>
<b>Descripción</b>	Captura y transfiere toda la información de interacción hacia el sistema de gestión.
<b>Protocolos/Actividades</b>	Percibir Awareness, Enviar Awareness.
<b>Permisos</b> <b>Lectura</b> <b>Cambio</b> <b>Genera</b>	Reglas de privacidad Percepción del Entorno Paquetes de Información
<b>Responsabilidades</b> <b>Dinámicas</b> <b>Seguridad</b>	(Capturar Eventos)*, (Enviar Paquetes)* En Sesión

**Tabla 6. Esquema del Rol Consola de Presentación**

<b>Esquema del rol</b>	<b>Consola de Presentación</b>
<b>Descripción</b>	Ofrece soporte a distintos mecanismos de presentación del Awareness.
<b>Protocolos/Actividades</b>	Recibir Awareness, Presentar Awareness, Interactuar con el Rol Cliente.
<b>Permisos</b> <b>Lectura</b> <b>Cambio</b> <b>Genera</b>	Perfil (Otros usuarios), Estado (Otros usuarios) Perfil (Propio), Estado (Propio) Mensajes (Awareness)
<b>Responsabilidades</b> <b>Dinámicas</b>  <b>Seguridad</b>	(Asumir Rol)+, (Asumir Tarea)+, (Proponer Respuesta)*, (Buscar Respuestas)*, [(Buscar Respuestas).(Evaluar Respuestas)] <sup>w</sup> , (Buscar Expertos)*, (Enviar Mensajes)* En Sesión

**Tabla 7. Esquema del Rol Evaluador de Aportes**

<b>Esquema del rol</b>	<b>Evaluador de Aportes</b>
<b>Descripción</b>	Permite cuantificar el nivel de satisfacción o calidad que un Cliente le da a un artefacto relacionado con la solución de una Tarea. En la dinámica colaborativa, la definición de Tareas es un mecanismo eficaz de coordinación y sus resultados se materializan a través de Artefactos. Este rol representa la evaluación de la calidad de un artefacto en el contexto de una Tarea, como estrategia para estimar el nivel de progreso sobre la actividad y del conocimiento de quien realiza el aporte.
<b>Protocolos/Actividades</b>	Recibir Artefacto, Presentar Artefacto, Evaluar Aporte.

<b>Permisos</b> Lectura Cambio Genera	Perfil Conocimiento Compartido Evaluación
<b>Responsabilidades</b> Dinámicas Seguridad	(Buscar Aportes)*, [(Visualizar Aporte).(Evaluar Respuestas)] <sup>w</sup> En Sesión

**Tabla 8. Esquema del Rol Registrador**

<b>Esquema del rol</b>	<b>Registrador</b>
<b>Descripción</b>	Escucha y almacena todos los mensajes que se producen en la red. Ofrece los servicios de persistencia a la información que puede ser consultada por demanda o por notificación, previo a una suscripción.
<b>Protocolos/Actividades</b>	Recibir Paquete, Actualizar Registro, Procesar Consultas, Entregar Información, Evaluar Condiciones de Suscripción, Notificar Cambios.
<b>Permisos</b> Lectura Cambio Genera	Perfiles, Estados, Registro de Acciones/Mensajes, Parámetros de Configuración. Notificación de Cambio
<b>Responsabilidades</b> Dinámicas Seguridad	(Capturar Paquete)*, (Actualizar Registros)*, [(Suscribir cambio).(Enviar Notificación)] <sup>w</sup> , [(Procesar Consulta).(Entregar Información)] <sup>w</sup>

**Tabla 9. Esquema del Rol Configurador**

<b>Esquema del rol</b>	<b>Configurador</b>
<b>Descripción</b>	Ofrece las interfaces gráficas necesarias para consultar y modificar los parámetros de la Red.
<b>Protocolos/Actividades</b>	Gestionar Configuración.
<b>Permisos</b> Lectura Cambio Genera	Perfiles, Parámetros de Configuración. Notificación de Cambio
<b>Responsabilidades</b> Dinámicas Seguridad	((Configurar Agentes).(Enviar Notificación de Cambio))*

**Tabla 10. Esquema del Rol Gestor de Acceso**

<b>Esquema del rol</b>	<b>Gestor de Acceso</b>
<b>Descripción</b>	Valida el inicio de sesión de un Cliente con la Red de Gestión.
<b>Protocolos/Actividades</b>	Recibir Solicitud de Acceso, Validar Parámetros de Seguridad, Registrar Evento de Acceso, Notificar Estado.
<b>Permisos</b> Lectura Cambio Genera	Perfil (Parámetros de Acceso) Estado Notificación de Cambio de Estado
<b>Responsabilidades</b> Dinámicas Seguridad	((Recibir Solicitud). (Validar Parámetros). (Registrar Evento). (Notificar Estado))* Reintentos Fallidos seguidos < 10

Tabla 11. Esquema del Rol Gestor de Awareness Social

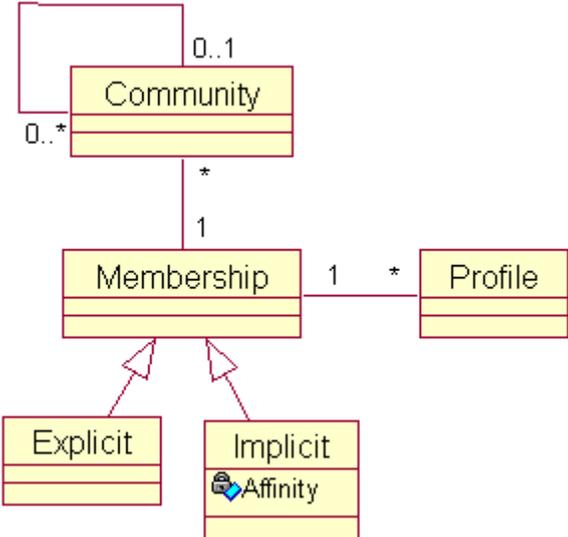
Esquema del rol	Gestor de Awareness Social
<p><b>Descripción</b></p>	<p>Procesa los mensajes y paquetes recibidos en la red para aplicar reglas de inferencia para generar Awareness Social, es decir, conciencia del grupo humano que le rodea. Para ello, se ha utilizado los conceptos de Comunidad, Membrecía y Perfil. Cada Perfil representa a un único Cliente que puede estar asociado explícitamente a una comunidad por una suscripción directa (Membrecía Explícita) o por presentar una afinidad que lo relaciona, por ejemplo su estado, conocimiento, acción o espacio de trabajo; a este segundo tipo de membrecía indirecta se ha denominado simplemente como asociación o Membrecía Implícita. La figura 5, presenta el segmento de clases asociadas con estos conceptos.</p>  <pre> classDiagram     class Community     class Membership     class Profile     class Explicit     class Implicit     class Affinity      Community "0..1" -- "0..*" Community     Community "*" -- "1" Membership     Membership "1" -- "*" Profile     Membership &lt; -- Explicit     Membership &lt; -- Implicit     Implicit -- Affinity     </pre> <p><b>Fig. 5. Modelo de Clases Asociadas con el Rol Gestor Awareness Social</b></p> <p>La clase Perfil agrupa los elementos esenciales para propiciar una interacción persona a persona, por ejemplo información de contacto, estados de conectividad e información emocional reflejadas en un avatar y mensajes de estado personalizables. Es importante destacar que para lograr una interacción satisfactoria, el Perfil de contener parámetros relacionados con la privacidad de la información o niveles de confianza a la hora de manipular información personal.</p> <p>Las Membrecías Implícitas se caracterizan por su atributo de Afinidad (Affinity), a través del cual dos o más Perfiles encuentran un punto de unión, por ejemplo un amigo común, estar cerca, compartir una tarea dentro de la actividad, comentar el mismo recurso, etc.</p>
<p><b>Protocolos/Actividades</b></p>	<p>Capturar Paquetes, Generar Awareness.</p>
<p><b>Permisos</b> Lectura Cambio Genera</p>	<p>Perfiles, Parámetros de Configuración. Awareness Social</p>
<p><b>Responsabilidades</b> Dinámicas Seguridad</p>	<p>((Analizar Paquetes).(Generar Awareness Social))* Niveles de Confianza Adecuados.</p>

Tabla 12. Esquema del Rol Gestor de Awareness de Tarea

Esquema del rol	Gestor de Awareness de Tarea
<p><b>Descripción</b></p>	<p>Procesa los mensajes y paquetes recibidos en la red para inferir Awareness de Tarea, es decir, conciencia de las Actividades, Metas y Objetivos propuestos por el grupo. Permite desarrollar una visión táctica de la realidad del grupo mediante el establecimiento de un estado de progreso, que combinado con responsabilidades y responsables, permita ejercer procesos de coordinación adecuados y eficientes. Para el desarrollo de este esquema se han definido los conceptos Tarea (<i>Task</i>) y Rol (<i>Role</i>), que combinados con conceptos como Perfil, Contexto, Evento y Artefacto, constituyen los elementos básicos para la generación de información de Awareness desde este enfoque; ver figura 6.</p> <div data-bbox="755 653 1190 1108" data-label="Diagram"> <pre> classDiagram     class Profile     class Role     class Context     class Task     class Event     class Artifact      Profile "1..*" -- "1" Role     Profile "1..*" -- "*" Context     Role "1" -- "*" Task     Context "*" -- "1" Task     Context "*" -- "*" Event     Task "1" -- "1" Artifact     </pre> </div> <p><b>Fig. 6. Diagrama de Clases Esquema de Rol Gestor de Awareness de Tarea</b></p> <p>Al definir y estructurar la actividad de grupo en forma de tareas, estas pueden ser combinadas con herramientas de flujo de trabajo para ofrecer métodos de coordinación efectivos; por esta razón, se propone enriquecer el concepto de Tarea con atributos tales como la secuencialidad, prioridad y resumen de metas u objetivos.</p> <p>El concepto Rol, permite identificar responsables. La asignación de un rol puede realizarse por asignación a través de configuración o un Cliente puede decidir asumir dinámicamente un Rol o Tarea, en la medida que el avance sobre la actividad. El cambio de Rol o Tarea es evidencia directa de Awareness, permitiendo crear enlaces entre las personas que los comparten, o que presentan dificultades que otros usuarios ya superaron, lo cual se plasma como una afinidad.</p> <p>Las Clases Contexto, Evento y Artefacto se utilizan para registrar los aportes que un Cliente realiza a una Tarea, lo cual constituyen evidencia del progreso y la participación. Estos aportes, combinados con procesos de verificación y cuantificación de su calidad, son la base para la gestión del conocimiento compartido.</p>
<b>Protocolos/Actividades</b>	Capturar Paquetes, Generar Awareness.
<b>Permisos</b>	

<b>Lectura Cambio Genera</b>	Perfiles (Rol y Tarea) Awareness de Tarea
<b>Responsabilidades Dinámicas Seguridad</b>	((Analizar Paquetes).(Generar Awareness de Tarea))*

**Tabla 13. Esquema del Rol Gestor de Awareness de Espacio Compartido**

Esquema del rol	Gestor de Awareness de Espacio Compartido
<b>Descripción</b>	<p>Procesa los mensajes y paquetes recibidos en la red para inferir Awareness de Espacio Compartido, es decir, conciencia sobre el entorno en el cual se está desarrollando la actividad. Los conceptos desarrollados alrededor de este esquema son: Entorno (Environment), Conectividad (Connectivity), Localización (Location), Tiempo (LocalTime), Herramientas (Tool) y Artefacto (Artefact), que combinados resumen las condiciones de trabajo de un Cliente. La figura 7, muestra la relación de estos conceptos. La clase Entorno, reúne las condiciones que son perceptibles del trabajo de un Cliente, por ejemplo el nivel de Conectividad con el cual cuenta, o su ubicación geográfica y virtual (ubicación dentro de las estructuras lógicas de una aplicación, por ejemplo en una aplicación Web, sería la página en la cual esta “ubicado”). El tiempo local cobra relevancia para Clientes cuyo uso horario sea distinto.</p> <div style="text-align: center;"> <pre> classDiagram     class Connectivity     class Location     class LocalTime     class Environment     class Profile     class Tool     class Context     class Artifact      Environment --- Connectivity     Environment --- Location     Environment --- LocalTime     Environment --- Profile     Environment --- Tool     Profile "1..*" --- "*" Context     Tool "1" --- "*" Context     Tool "1" --- "*" Artifact     </pre> </div> <p><b>Fig. 7. Diagrama de Clases Esquema Gestor de Awareness de Espacio Compartido</b></p> <p>Las clases Herramienta y Artefacto, son también elementos importantes del entorno de trabajo, por cuanto permiten conocer cuáles son los objetos de trabajo que un Cliente está manipulando y con qué aplicación se realiza. Al contar con una descripción suficientemente</p>

	<p>detallada del entorno y los objetos manipulados, se puede propiciar cierto nivel de interacción incluso con herramientas distintas, definiendo reglas de interoperabilidad adecuadas.</p> <p>El Awareness de Espacio Compartido pretende crear relación entre personas que comparten un entorno o manipulan el mismo objeto o su complemento. Con una estructuración adecuada, incluso se podría llegar a la identificación de espacios de trabajo de complemento necesario que permitan un mejor aprovechamiento de los recursos disponibles.</p>
<b>Protocolos/Actividades</b>	Capturar Paquetes, Generar Awareness.
<b>Permisos</b> <b>Lectura</b> <b>Cambio</b>  <b>Genera</b>	Entorno, Conectividad, Localización, Tiempo Local, Herramienta y Artefacto Registros de Contexto
<b>Responsabilidades</b> <b>Dinámicas</b> <b>Seguridad</b>	((Analizar Paquetes).(Generar Awareness de Espacio Compartido))* Niveles de Confianza Adecuado.

**Tabla 14. Esquema del Rol Gestor de Awareness de Conocimiento Compartido**

<b>Esquema del rol</b>	<b>Gestor de Awareness de Conocimiento Compartido</b>
<b>Descripción</b>	<p>Procesa los paquetes de Evaluaciones de Aportes para inferir Awareness de Conocimiento Compartido. Este tipo de enfoque de Awareness pretende identificar las fuentes y unidades de conocimiento involucradas en la solución de los problemas, con el fin facilitar la implantación de estrategias de gestión para dicho conocimiento.</p> <p>Identificar las fuentes y niveles de conocimiento, permite el desarrollo de servicios tales como la búsqueda de expertos o la cuantificación del aprendizaje. Además de propiciar la conversión de conocimiento implícito a explícito, pieza clave para una buena gestión de conocimiento.</p> <p>Desde nuestra perspectiva, el conocimiento está implícito en los Mensajes y Aportes de un Usuario. Extraerlo de los Mensajes es un trabajo complejo y requiere de herramientas como el análisis de lenguaje natural y la correlación de mensajes, lo cual es ajeno a esta investigación y por tanto se propone como línea de trabajo futura y se relega por ahora como un trabajo manual destinado a expertos.</p> <p>Por su parte, los Aportes se materializan a través de Artefactos que pueden ser evaluados sin mayor dificultad por la mayoría de usuario. La evaluación justa de un Aporte debe combinar la experticia de quien aporta, con la de quien realiza la evaluación, estableciendo una Curva de Evaluación y un nivel de Experticia del Usuario. Al estructurar el conocimiento relacionado con la Tarea, la evaluación de un Artefacto permite identificar el nivel de conocimiento de un Usuario por Concepto, permitiendo construir un Mapa de Experticia.</p> <p>Esta investigación plantea la forma general de la ecuación que podría modelar el conocimiento compartido:</p> <p style="text-align: center;"><b><math>Ea = \text{Función}(Ee, Ea, Q) (1)</math></b></p>

	<p style="text-align: center;">Donde:</p> <p style="text-align: center;">Ea: Experticia de quien Aporta  Ee: Experticia de quien evalúa  Q: Valoración dada al Aporte</p> <p>Al generalizar, todos los conceptos no se tratan al mismo nivel de profundidad en una Tarea, por lo cual es necesario introducir un valor de ponderación o factor de impacto de la evaluación sobre el nivel de experticia para el concepto específico.</p> <p>La figura 8 presenta el diagrama de clases con los conceptos propuestos para la gestión del conocimiento compartido. El dominio de conocimiento (Domine) es modelado a través de conceptos (Concept) y cada Tarea de la actividad propone un nivel de aprendizaje (Learning) para distintos conceptos (asignando un factor de impacto). Cuando el evento de la evaluación es detectado, se calcula el factor de Experticia y se almacena en el registro (Expertise).</p> <div style="text-align: center;"> <pre> classDiagram     Domine "1" -- "*" Concept     Expertise "1" -- "*" Concept     Task "1" -- "*" Learning     Concept --- Learning </pre> </div> <p><b>Fig. 8. Diagrama de Clases relacionadas con el esquema del Rol Gestor de Awareness de Conocimiento Compartido</b></p> <p>El registro de Experticia es utilizado para desarrollar servicios tales como la búsqueda de expertos, con base en la definición de un nivel mínimo de Experticia sobre un concepto particular y solicitar una exploración de la lista de expertos. Se pueden crear combinaciones de dos o más conceptos.</p> <p>Una valoración promedio alta de un Artefacto, es indicio de un despliegue de conocimiento adecuado en los conceptos con los cuales se relaciona, por lo tanto, puede ser utilizado como criterio de referencia en caso de solicitar una búsqueda de fuentes de información relevante.</p>
<b>Protocolos/Actividades</b>	Capturar Evaluaciones, Actualizar Registros de Experticia, Generar Awareness.
<b>Permisos</b> Lectura Cambio	Aprendizaje, Perfil, Conceptos Experticia

<b>Genera</b>	Lista de Expertos, Niveles de Experticia Promedio, Lista de Artefactos (Fuentes de Información Relevantes)
<b>Responsabilidades Dinámicas</b>	((Analizar Paquetes).(Generar Awareness de Conocimiento Compartido))* , (Buscar Expertos)*, (Buscar Fuentes de Información Relevantes)*
<b>Seguridad</b>	

**Tabla 15. Esquema del Rol Gestor de Awareness de Concepto**

<b>Esquema del rol Gestor de Awareness de Concepto</b>	
<b>Descripción</b>	El Awareness de Concepto permite situar el aprendizaje dentro de un contexto de conocimiento particular. Constituye una fuente de información de referencia para conocer acerca de los Conceptos que estructuran la Tarea y por tanto se pueden considerar como elementos de información estáticos.  En trabajos futuros, a partir de estos dominios y conceptos se podrían implementar servicios de sistemas expertos o sistemas tutores basados en fuentes e información estructuradas.
<b>Protocolos/Actividades</b>	Generar Awareness.
<b>Permisos</b> <b>Lectura</b> <b>Cambio</b> <b>Genera</b>	Dominio, Concepto  Awareness de Concepto
<b>Responsabilidades Dinámicas</b> <b>Seguridad</b>	((Recibir Solicitud de Información).(Generar Awareness de Concepto))*

### 3.2.1.2 Modelo de Interacción

El modelo de interacción presenta un resumen más específico sobre los distintos protocolos que relacionan dos Roles, además, señala entradas salidas y una breve descripción del protocolo. La tabla 16 presenta la forma general de la plantilla utilizada para el Modelo de Interacción. Posteriormente se listan las tablas que forman el modelo.

**Tabla 16, Plantilla general para la descripción del Modelo de Interacción**

<b>Nombre del Protocolo</b>		
Rol	Rol	Entradas
Descripción del protocolo.		Salidas

**Tabla 17. Protocolo Enviar Awareness**

<b>Enviar Awareness</b>		
Sensor	Gestores de Awareness	Eventos / Mensajes
El sensor envía constantemente la información las acciones y mensajes que el Cliente produce mientras interactúa		Awareness

**Tabla 18. Protocolo Recibir Awareness**

<b>Recibir Awareness</b>		
Gestores de Awareness	Consola	Awareness

Los gestores, como resultado de sus procesos de análisis, generan paquetes de Awareness para la Consola	
---	--

**Tabla 19. Protocolo Recibir Artefacto**

Recibir Artefacto		
Gestor	Consola / Herramienta	Artefacto
Por demanda de algunos servicios de Awareness, el Gestor envía hacia la Consola algunos Artefactos, los cuales son desplegados sobre las herramientas compatibles		

**Tabla 20. Protocolo Entregar Información**

Entregar Información		
Consola/Sensor	Herramienta	Artefacto
A través de un servicio de comunicación local, la Consola entrega información hacia las herramientas compatibles. De igual forma las herramientas puede general información de Awareness que recibe el Rol Sensor para ser enviada a la Red.		Awareness

**Tabla 21. Protocolo Notificar Cambios**

Notificar Cambios		
Registrador	Suscriptores	Notificación
El Registrador analiza las suscripciones de notificación y envía un mensaje cuando ocurre un cambio en la estructura de datos que se ajusta a lo requerido.		

**Tabla 22. Protocolo Recibir Solicitud de Acceso**

Recibir Solicitud de Acceso		
Gestor de Acceso	Consola	Datos de Acceso
Luego de una validación exitosa por parte de una Consola, se envía su Perfil.		Perfil

**Tabla 23. Protocolo Notificar Estado**

Recibir Solicitud de Acceso		
Gestor de Acceso	Gestores de Awareness	
Luego de un acceso exitoso, se notifica a los distintos Gestores de Awareness para que envíen toda aquella información que consideren pertinente.		Perfil

**Tabla 24. Protocolo Generar Awareness**

Recibir Solicitud de Acceso		
Gestores de Awareness	Gestores / Consola	
De forma autónoma, los Gestores de Awareness estructuran la información a través del Modelo de Datos y los distribuyen a través de la Red, por lo general estos van dirigidos hacia la Consola, pero en ocasiones pueden enviarse con copia a otros Gestores para propiciar la comunicación por Cooperación entre Gestores.		Awareness

### 3.2.2 Diseño

Esta fase del proceso de modelado se caracteriza por tres modelos: un Modelo de Agentes, Modelo de Servicios y el Modelo de Conocidos o Familiaridad. A continuación se presentan cada uno de ellos.

#### 3.2.2.1 El modelo de agentes

El Modelo de Agentes presenta en forma jerárquica, los distintos tipos de Agentes y Roles definidos para la red; además, se señala la cardinalidad de instancias esperadas por tipo de Agente.

Solamente el Ag. Interfaz presenta múltiples instancias dentro de la Red, para los demás Agentes se plantea una única instancia que sea la que desarrolle el servicio. Este esquema no implica que una implementación posterior no pueda distribuir la responsabilidad en la implementación de un enfoque de Awareness a través de varios Agentes, lo que sugiere es que dada la complejidad de los proceso de análisis de información se recomienda el desarrollo de entidades especializadas para la gestión de la información, además de brindar flexibilidad en la composición de servicios.

##### **Ag. Interfaz (2...\*)**

- Sensor
- Consola de Presentación
- Evaluador de Aportes

##### **Ag. Persistencia (1)**

- Registrador
- Configurador
- Gestor de acceso

##### **Ag. Aw. Social (1)**

- Gestor de Awareness Social

##### **Ag. Aw. Tarea (1)**

- Gestor de Awareness de Tarea

##### **Ag. Aw. Espacio de Trabajo (1)**

- Gestor de Awareness de Espacio de Trabajo

##### **Ag. Aw. Conocimiento Compartido (1)**

- Gestor de Awareness de Conocimiento compartido

##### **Ag. Aw Concepto (1)**

- Gestor de Awareness de Concepto

#### 3.2.2.2 El modelo de servicios

Este modelo describe con más detalle los servicios activos en la Red, señalando cual es el Agente que lo implementa. A la plantilla general sugerida por GAIA se adicionó un campo de descripción para mejorar su detalle.

**Tabla 25. Servicio Pasarela de Comunicación Awareness/Groupware**

<b>Servicio Pasarela de Comunicación Awareness / Groupware</b>	
<b>Descripción</b>	Crea un servicio local que facilita la comunicación entre aplicaciones Groupware que se ejecutan sobre la estación cliente y la red de Gestión. Facilita básicamente el intercambio de paquetes de Awareness, Artefactos y la invocación de servicios tales como la búsqueda de expertos o valoración de aportes.
<b>Agente</b>	Interfaz

<b>Entrada</b>	Paquetes Awareness
<b>Salidas</b>	Paquetes Awareness
<b>Precondición</b>	Inicio de sesión
<b>Postcondición</b>	

**Tabla 26. Servicio de Captura de Paquetes**

<b>Servicio</b>		<b>Captura de Paquetes</b>	
<b>Descripción</b>	Servicio implementado por todos Agente en la red. Habilita los canales de comunicación para la captura de los distintos paquetes que circulan por la Red. El filtrado de los mensajes es un proceso interno que descarta información que no es pertinente para el Agente. Cada paquete es marcado las direcciones de uno o más destinos o un encabezado broadcast que permite difundir datos a toda la red.		
<b>Agente</b>	Agente		
<b>Entrada</b>	Paquete de Información		
<b>Salidas</b>			
<b>Precondición</b>			
<b>Postcondición</b>	Captura de Paquete		

**Tabla 27. Servicio Buscar Expertos**

<b>Servicio</b>		<b>Buscar Expertos</b>	
<b>Descripción</b>	Este servicio permite la búsqueda de Perfiles cuyo registro de experticia muestre un cierto nivel mínimo sobre un conjunto dado de conceptos.		
<b>Agente</b>	Conocimiento Compartido		
<b>Entrada</b>	Conceptos, Nivel de Experticia		
<b>Salidas</b>	Lista de Perfiles		
<b>Precondición</b>	Contar con registros de experticia sobre los conceptos a consultar		
<b>Postcondición</b>			

**Tabla 28. Servicio Buscar Referencias**

<b>Servicio</b>		<b>Buscar Referencias</b>	
<b>Descripción</b>	Permite la búsqueda de aportes que se relacionen con los conceptos de una Tarea específica. Dichos aportes puede luego ser valorados para mejorar el criterio del Agente de Conocimiento Compartido o divulgados para su manipulación.		
<b>Agente</b>	Conocimiento Compartido		
<b>Entrada</b>	Tarea		
<b>Salidas</b>	Lista de Artefactos de referencia		
<b>Precondición</b>	Contar con registros de Artefactos registrados para la Tarea específica.		
<b>Postcondición</b>			

**Tabla 29. Servicio Evaluar Aporte**

<b>Servicio</b>		<b>Evaluar Aporte</b>	
<b>Descripción</b>	Una vez se disponga de un artefacto como material de referencia para una tarea, el cliente puede dar una valoración cuantitativa al aporte, la cual se convierte en experticia tanto para el evaluado como para el evaluador, desde la perspectiva del agente de conocimiento compartido.		

<b>Agente</b>	Conocimiento Compartido
<b>Entrada</b>	Artefacto
<b>Salidas</b>	Evaluación y Experticia.
<b>Precondición</b>	
<b>Postcondición</b>	

**Tabla 30. Servicio Awareness Social**

<b>Servicio</b>	<b>Awareness Social</b>
<b>Descripción</b>	Analiza la información capturada e infiere posible Awareness de enfoque social que distribuye entre los interesados. Por ejemplo, la conexión o desconexión de un compañero de trabajo.
<b>Agente</b>	Social
<b>Entrada</b>	Paquetes de Información
<b>Salidas</b>	Awareness Social
<b>Precondición</b>	
<b>Postcondición</b>	

**Tabla 31. Servicio Awareness de Espacio de Trabajo**

<b>Servicio</b>	<b>Awareness de Espacio de Trabajo</b>
<b>Descripción</b>	Analiza los paquetes asociados con posibles cambios en el contexto de trabajo de un usuario y lo retransmite en forma de Awareness para un grupo de usuarios de interés. Por ejemplo, el acceso recurrente a un artefacto, la cercanía a una ubicación específica, etc.
<b>Agente</b>	Espacio de Trabajo
<b>Entrada</b>	Paquetes de Información
<b>Salidas</b>	Awareness de Espacio de Trabajo
<b>Precondición</b>	
<b>Postcondición</b>	

**Tabla 32. Servicio Awareness de Tarea**

<b>Servicio</b>	<b>Awareness de Tarea</b>
<b>Descripción</b>	Registra y entrega información asociada con los Roles y Tareas que se asocian a la actividad de grupo. Utiliza esta información para procurar interacción entre personas que, por ejemplo, comparten un rol o Tarea, o presentan dificultad sobre una actividad que otra persona realizó con éxito.
<b>Agente</b>	Tarea
<b>Entrada</b>	Paquetes de Información
<b>Salidas</b>	Awareness de Tarea
<b>Precondición</b>	
<b>Postcondición</b>	

**Tabla 33. Servicio Awareness de Concepto**

<b>Servicio</b>	<b>Awareness Concepto</b>
<b>Descripción</b>	Almacena y permite la consulta de la lista de conceptos asociados con una Tarea específica en el dominio de conocimiento para el cual se diseñó la actividad, lo cual permite estructurar, en términos de conocimiento, los aportes se realicen sobre la tarea.
<b>Agente</b>	Concepto
<b>Entrada</b>	Tarea
<b>Salidas</b>	Awareness de Concepto

Precondición	
Postcondición	

### 3.2.2.3 El modelo de Conocidos o Familiaridad

Este modelo presenta los distintos Agentes y las relaciones de comunicación que existen entre ellos. La figura 9 muestra el modelo de Conocidos propuesto, en él, se han realizado dos divisiones, la primera, señalado como Unidad, representa la estación de trabajo del Usuario. Cada Agente de Interfaz puede conectarse con otro Agente de Interfaz durante una interacción. La segunda división hace referencia a la Red de Gestión, en esta red existen los distintos Agentes que implementan la gestión del Awareness. El Agente de Persistencia se conecta con la Red para señalar que en un momento específico tiene comunicación con cada uno de los Agentes de la Red, de igual forma sucede con el Agente de Interfaz.

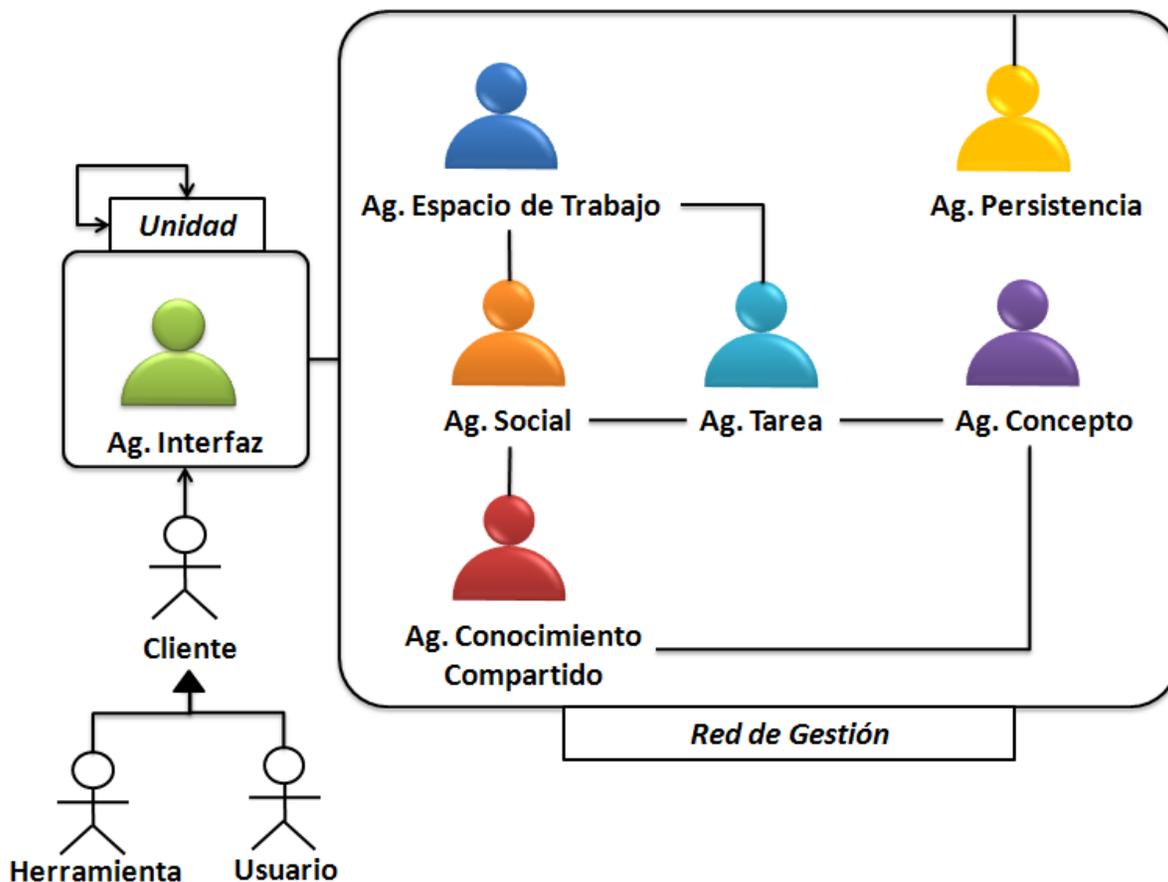


Fig. 9. Modelo de Familiaridad

La representación de Actores en este diagrama, ilustra dos conceptos: primero, desde la perspectiva de la gestión del Awareness, los actores externos a la red solo acceden a los servicios a través del Agente de Interfaz; segundo, el Actor Usuario representa a la persona que utiliza la interfaz gráfica para interactuar con la red, mientras que el Actor Herramienta utiliza interfaces de servicio para utilizar al Ag. Interfaz como pasarela de comunicación con la Red de Agentes.

Este diagrama concluye la especificación del sistema según la metodología GAIA. A continuación se presenta una descripción de las arquitecturas internas de los Agentes, para ellos, se ha identificado dos tipos generales de Agentes: los Agentes de Interfaz y los Agentes de Servicio, en donde los agentes de Servicio contiene los bloques funcionales necesarios para el desarrollo de un Agente de Gestión de Awareness para cualquiera de los enfoques estudiados.

### 3.2.3 Arquitectura Interna del Agente de Interfaz

La figura 10 presenta el diagrama en bloques de la estructura interna de este tipo de agentes: la capa de comunicación, la cual permite el acceso a la información de la red y gestiona todos los procesos de en colado de mensajes necesarios para los Agentes; procesamiento de datos y filtrado, permiten analizar las reglas que determinan la pertinencia de la información percibida; seguridad, que implementa los esquemas apropiados para lograr una comunicación eficiente y acorde con políticas de privacidad; sensado, la cual habilita las rutinas necesarias para capturar los eventos de interacción del usuario; interfaz de servicios, que corresponde a los servicios locales de comunicación con otras aplicaciones que se ejecutan sobre la estación del cliente y el WidgetEngine, el cual corresponde a un entorno de despliegue para los Widgets, aplicaciones semiindependientes que definen la interacción entre el Usuario y el Agente.



Fig. 10. Arquitectura interna para los Agentes de Interfaz

WidgetEngine provee acceso a los Widgets sobre los servicios básicos de comunicación y sensado. Los Widgets toman la información de Awareness y la presentan de distintas formas, combinando las características del Perfil y las Políticas de presentación y privacidad.

### 3.2.4 Arquitectura Interna de un Agente de Servicio

Las arquitecturas específicas de Agentes de Awareness Social, de Tarea, Conocimiento Compartido, Concepto y Espacio de Trabajo, contienen un conjunto común de capas que describen su funcionalidad general. A este conjunto de capas se ha denominado la Arquitectura Interna de un Agente de Servicio.

Los Agentes de servicio capturan, infieren y transforman la información emitida por los Agentes de Interfaz para generar información de Awareness. Desde esta perspectiva, la figura 11 presenta el diagrama en bloques de la arquitectura general para este tipo de Agentes; su principal diferencia con los Agentes de Interfaz, está en la ausencia de las

capas de sensado, interfaz y WidgetEngine, las cuales son remplazadas por bloques de procesamiento más específicos donde se desarrolla la lógica del servicio y la lógica de cooperación del agente.

La capa de Filtrado es más específica y contiene la implementación de distintas reglas que reducen la carga de procesamiento de mensajes para el Agente. Además, la lógica de Cooperación detecta cuando un determinado Evento que se supone afecta un enfoque específico, puede derivar en la generación de Awareness desde otros enfoques, lo cual busca enriquecer el escenario de interacción de los Clientes.



**Fig. 11. Arquitectura interna para los Agentes de Servicio**

El acceso a la información en los entornos de aprendizaje grupal implica consideraciones importantes en aspectos de seguridad [65], como por ejemplo garantizar a los usuarios distintos niveles de confianza para la manipulación de los datos. Para este trabajo se ha propuesto una capa transversal a la arquitectura de servicio que garantice la identidad y acceso selectivo y limitado a la información compartida de acuerdo a las políticas que se propongan. De forma parecida a otros sistemas, los requerimientos de seguridad de información en sistemas groupware están descritos en el estándar ISO-7498-22. Este estándar prevé los siguientes servicios de seguridad de la información: identificación y autenticación, autorización (control de acceso), confidencialidad, integridad, y no-repudio. Los mecanismos para cada uno de estos servicios deben ser implementados para asegurar la confianza en el entorno de soporte a la colaboración.

Por otra parte, los elementos particulares que diferencian a cada uno de los Agentes de Servicios se encuentran en la lógica de filtrado y la lógica del servicio, en la cual se genera la información de Awareness. Cada agente implementa distintas reglas que permiten transformar eventos y mensajes de la red en información de Awareness desde un enfoque particular. Dichas reglas podrían ser implementadas a través de sistemas expertos u otras técnicas de inteligencia artificial. El desarrollo de la lógica de los servicios basados en reglas, se propone como línea de trabajo futuro.

En el siguiente capítulo se presenta los prototipos desarrollados que validan la propuesta realizada: primero, desde el modelo conceptual y luego de la Red Multi-Agente para la Gestión del Awareness.

## Capítulo 4. Escenarios de validación

En este capítulo se describe el diseño y detalles de construcción de los dos escenarios de validación propuestos para la arquitectura: el primero, el desarrollo de un módulo de *Awareness* para monitorizar y analizar los procesos colaborativos que se producen durante la interacción de los usuarios con el sistema de gestión de conocimiento *KnowCat* [66], una herramienta Web que se relaciona con los procesos de síntesis y distribución conocimiento desde fuentes externas. El desarrollo de este primer escenario se realizó en el marco del proyecto titulado: “Entorno basado en Tecnologías de la Información y las Comunicaciones (TICs) para monitorizar y analizar los procesos colaborativos en actividades con contenidos educativos”, financiado por la Agencia Española de Cooperación Internacional para el Desarrollo (AECID), identificado con el código A/7954/07, que contó con la participación de las universidades: Autónoma de Madrid (España), Nacional de Medellín (Colombia) y Universidad del Cauca.

El segundo escenario de validación, partió del diseño y desarrollo de una herramienta de soporte a la construcción colaborativas de conocimiento basado en la discusión, utilizando plataformas TabletPC. Dicha herramienta se enmarcó dentro de los alcances propuestos para el acuerdo de colaboración para el desarrollo de proyectos de investigación "*A Digital Workbook Tool to Support Asynchronous Collaboration*" entre la Universidad de Chile, Microsoft Corporation Collaborating Research Institution Research Project Agreement y la Universidad del Cauca financiado por LACCIR (Latin American and Caribbean Collaborative ICT Research Federation) código Grant R0308LAC001. Además, fue necesario desarrollar un prototipo funcional del Sistema Multi-Agente que abordó los procesos de identificación de fuentes y redes de experiencia.

Mientras que el primer escenario ha permitido la validación del Modelo Conceptual para la Gestión del *Awareness*, el segundo escenario ofrece una visión más completa y compleja del entorno tecnológico que propone esta investigación basada en el paradigma de Sistemas Multi-Agente, además de configurar un entorno para la integración de nuevos servicios, herramientas y tecnologías orientadas a la mejora de los procesos de gestión de conocimiento en el marco del CSCL.

### 4.1 Caso de estudio I, Módulo de *Awareness* para *KnowCat*

En el contexto del proyecto: “Entorno basado en Tecnologías de la Información y las Comunicaciones (TICs) para monitorizar y analizar los procesos colaborativos en actividades con contenidos educativos”, se planteó la necesidad de desarrollar propuestas conceptuales y técnicas que dieran soporte a los procesos de monitorización y análisis de las actividades de los usuarios, como herramienta para la implementación de estrategias más efectivas de seguimiento y apoyo a los estudiantes, seleccionando como caso de estudio el sistema *KnowCat*.

*KnowCat* es el acrónimo de "Knowledge Catalyser" o "catalizador de conocimiento", es un sistema distribuido para GC, creado en 1998 por la Universidad Autónoma de Madrid (España), que permite la creación incremental de conocimiento estructurado [63]. Su motivación subyacente es permitir la generación de materiales educativos de alta calidad como resultado de la interacción de los usuarios con los propios materiales. Es un sistema groupware de trabajo asíncrono que proporciona un entorno de colaboración sobre la Web, basado en el concepto de "cristalización de conocimiento" y soportado por comunidades virtuales de expertos. Permite la construcción de sitios Web denominados

nodos KnowCat los cuales estructuran en forma de árbol el conocimiento sobre un área o tema específico, en donde cada rama contiene un conjunto de documentos que describen un tópico particular. Además, a cada documento se asocia un grupo de anotaciones que expresan dudas, comentarios, apoyo o rechazo a su contenido. Con base en los mecanismos de anotaciones, votos y “calidad de usuario”, este último permite dar mayor credibilidad a las opiniones de expertos sobre las de usuarios ocasionales, los documentos ubicados en una misma rama compiten entre sí por una ponderación de calidad y capacidad descriptiva del tópico, lo que resume la esencia el proceso de cristalización.

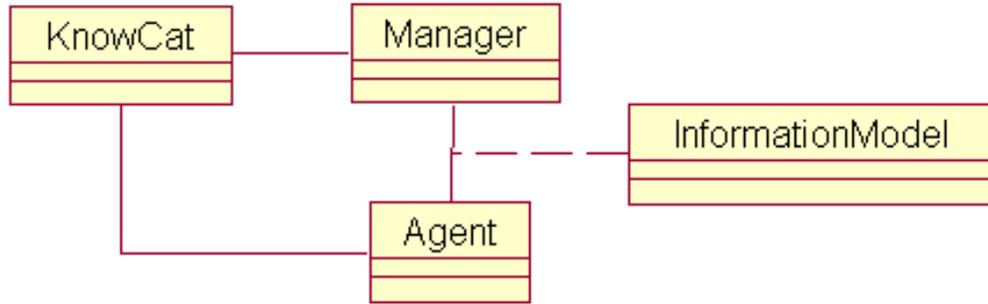
Por otra parte, el entorno de desarrollo de servicios para KnowCat presentó las siguientes características:

- Desarrollo basado en tecnologías CGI (Perl) y Javascript, sin el soporte a muchas librerías utilizadas en el desarrollo Web actual, por ejemplo librerías para el manejo de sesiones.
- Plataforma experimental de diversas investigaciones, por lo cual se encontraron distintos estilos de programación y una estructuración limitada de la Arquitectura sobre todo en las capas de presentación y control.
- Apremiante necesidad de mantener separación entre la extensión a desarrollar y el núcleo funcional del sistema, tal que cualquier mejora que se relacionara con el KnowCat pudiera ser replicada en otras investigaciones sin la dependencia de la extensión.

Estas características impusieron retos importantes en el desarrollo los cuales fueron asimilados por el Modelo Conceptual propuesto a través de los tres elementos fundamentales: un Modelo de Información, las entidades Agente y Gestor. Por otra parte, según el contexto del proyecto, para dar soporte a las tareas de monitoreo de usuario se propuso implementar los servicios de *Awareness*: Usuarios Registrados, Usuarios en Línea, Vista Histórica, Vista de Radar, Participómetro y Mapa de Interacción, los cuales se describen en la última de las secciones dedicada a este escenario.

Es importante aclarar que para el desarrollo de los servicios *Awareness* propuestos para el sistema KnowCat no se involucró el uso del paradigma de Agentes Inteligentes ni por tanto el despliegue o desarrollo de una red de gestión Multi-Agente. En este primer caso de estudio, el procesamiento y generación de la información de *Awareness* es responsabilidad de un conjunto de servicios desarrollados con tecnología compatibles con la plataforma KnowCat (CGIs, HTML y javascript) que en trabajos futuros podría migrar a una plataforma Multi-Agente como Jade [67]. Los potenciales beneficios de esta migración serían: mayor flexibilidad en la evolución de los servicios, interoperabilidad con entornos de escritorio u otras plataformas Web que le permite ofrecer servicios general de gestión del *Awareness*, flexibilidad en la definición y creación de nuevos servicios para la mejora de los ambientes de enseñanza/aprendizaje tales como sistemas tutor inteligente, con compatibilidad con entornos Web.

En la figura 12 se puede ver un diagrama general de clases de diseño simplificado que describe la arquitectura desarrollada para KnowCat. En dicho diagrama se identifican los tres elementos básicos del modelo conceptual: las entidades Agente, Gestor y el Modelo de Información. Las siguientes secciones describen los detalles más relevantes en la implementación de los elementos mencionados.



**Fig. 12. Diagrama de Clases simplificado de la implementación para KnowCat**

#### 4.1.1 Modelo de Información

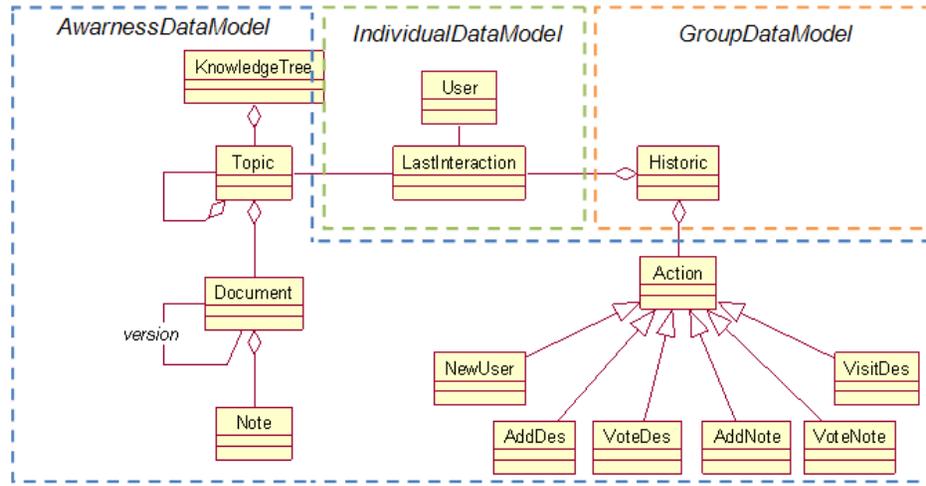
El primer paso para la implementación del Modelo de Información en KnowCat, consistió en analizar e identificar los conceptos relevantes dentro del modelo de datos del sistema relacionados con los procesos de monitoreo de las actividades de usuario y aprovisionamiento de *Awareness*. Luego de ello fue necesario proponer nuevas entidades para el soporte al registro de eventos tanto en tiempo real como desde una perspectiva histórica. Todas las entidades fueron gestionadas a través de un motor de bases de datos relacional.

En el figura 13, se puede ver el conjunto de entidades identificadas como parte del Modelo de Información del sistema. Las entidades *User*, *KnowledgeTree*, *Topic*, *Document* y *Note* fueron retomadas del modelo de datos del sistema, mientras que *LastInteraction*, *Historic* y *Action*, fueron introducidas para dar soporte a los nuevos servicios. Las entidades que hicieron parte de la extensión implementadas son:

- *User*, almacena datos relacionados con el perfil del usuario tales como nombre, ocupación y correo electrónico.
- *LastInteraction*, mantiene un registro temporal de la última acción relacionada con un usuario. Permite deducir el estado o “ubicación” del usuario en el árbol de conocimiento.
- *KnowledgeTree*, objeto raíz para la estructura de conocimiento.
- *Topic*, rama del árbol de conocimiento. Se puede asociar con un tópico padre para dar forma arborescente al conocimiento.
- *Document*, registro del documento asociado con un tópico. Puede ser asociado con otro documento a través de versiones, con lo cual se da soporte al proceso incremental de mejora de aportes.
- *Note*, registro de anotaciones. Adicionalmente a un texto asociado como comentario, las anotaciones poseen un tipo, por ejemplo: apoyo, aclaración o rechazo que les permiten una mayor estructuración. Además pueden relacionarse con otra anotación basadas en el concepto de votación *a favor* o *en contra*.
- *Historic*, mantiene el registro de todas las acciones realizadas por un usuario a lo largo de su interacción con el sistema. La diferencia entre los registros de

*LastInteraction* e *Historic*, es básicamente el nivel de impacto de la acción sobre el sistema, mientras que *LastInteraction* puede almacenar registro temporales y de bajo impacto como visitar un tópico, las acciones que se almacenan en *Historic* representan cambios significativos sobre el estado de un usuario, por ejemplo aportar un documento, anotarlo o votarlo.

- *Action*, estructura los eventos de usuario significativos para el sistema tales como: *NewUser*, cuando se registra el usuario ante el sistema; *AddDes*, cuando se aporta un documento a un tópico; *VoteDes*, cuando se vota por un documento; *AddNote*, cuando se hace una anotación; *VoteNote*, al votar una Anotación, sea a favor o en contra; *VisitDes*, al visitar un documento.



**Fig. 13. Modelo de información definido para el Sistema KnowCat**

Adicionalmente, la figura 13 muestra una separación a tres niveles de las distintas entidades que conforman el modelo de información: *IndividualDataModel*, agrupa aquellas entidades que contiene información personal del usuario y cuya gestión se mantiene de forma independiente a la de los demás usuarios. Es una información altamente cambiante en el tiempo y como se había mencionado antes, no implica un registro persistente o cambio significativo del estado del usuario para el sistema. *GroupDataModel*, son aquellas entidades relacionadas con las acciones persistentes de los usuarios, condensan múltiples acciones de usuario en pasos significativos de avance sobre las tareas; son fuente para analizar el desempeño de un usuario en el contexto de grupo. Finalmente el *AwarenessDataModel*, son las entidades que transforman y dan contexto a las acciones individuales de un usuario para convertirse en acciones de grupo, significativas para el sistema o para otros usuario. En el caso particular de KnowCat, las entidades *KnowledgeTree*, *Document* y *Note*, son utilizadas para construir el concepto de *localización del usuario dentro del sistema*, concepto utilizado por distintos servicios de *Awareness* para contextualizar al usuario en función del árbol de conocimiento.

#### 4.1.2 Módulo Agente

El módulo Agente desarrollado interpreta los dos roles propuestos para esta entidad en el modelo Conceptual: el rol Sensor y el rol Interfaz. Desde su rol Sensor, la implementación del Agente requirió el desarrollo tanto de una rutina CGI del lado servidor, como de funciones Javascript del lado del navegador. En el servidor, su responsabilidad es

detectar los eventos de disparos a rutinas específicas del sistema y registrarlos sobre el Modelo de Información. En el navegador, el Agente captura eventos de interfaz tales como movimientos de ratón, pulsaciones y eventos de teclado que periódicamente se resumen y se enviaban hacia el servidor para su procesamiento haciendo uso de tecnología AJAX.

Por otra parte, el espacio para el despliegue de los servicios *Awareness* dentro KnowCat es la Consola de Awareness, figura 14, un componente gráfico embebido en la interfaz principal que suministra acceso a las distintas vistas de los servicios. Esta consola cuenta con una barra de herramientas que permite, por demanda, acceder a los distintos servicios.



**Fig. 14. Vista de la página principal del sistema, Consola de Awareness**

Desde la perspectiva del rol Interfaz, el Agente despliega la consola que funcionalmente combina código HTML dinámico y rutinas Javascript para implementar las funciones de un *WidgetEngine*, un entorno para el despliegue de Widgets: componentes gráficos avanzados para el despliegue de la información suministrada para los servicios. Haciendo uso de la tecnología AJAX, cada Agente implementa un canal de comunicación con el Gestor utilizando el protocolo HTTP y empaquetando los datos sobre estructuras XML. Dichas estructuras permiten el intercambio de información entre los Widgets de cada servicio y los servicios *Awareness* soportados por el Gestor. En la figura 15 se puede ver la estructura de un mensaje de actualización de la consola, intercambiado entre Gestor y Agente.

```

Fuente de: http://knowcat.iluam.es/6.0/awc/xml/awConsoleRsm.pl - Mozilla Firefox
Archivo  Editar  Ver  Ayuda
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE awConsoleRsm SYSTEM 'dto/awConsole.dtd'>
<CtgInf id='RemAwr' desc='Información de grupo'>
  <BlqInf id='RemAwrCnc' desc='Información de grupo'>
    <AprInf id='UserRgs' desc='Usuarios Registrados'>
      <Inf id='NmrUserRgs' desc='Usuarios Registrados'>
        <Dto id='NmrUser' desc='Users Number'>12</Dto>
      </Inf>
    </AprInf>
    <AprInf id='UserCnc' desc='Usuarios en línea'>
      <Inf id='NmrUserCnc' desc='Usuarios en línea'>
        <Dto id='NmrUser' desc='Users Number'>1</Dto>
      </Inf>
    </AprInf>
    <AprInf id='RdView' desc='Vista de radar'>
      <Inf id='NmrUserCnc' desc='Vista de radar'>
        <Dto id='NmrUser' desc='Users Number'>-1</Dto>
      </Inf>
    </AprInf>
    <AprInf id='GlbPr' desc='Participación'>
      <Inf id='NmrUserCnc' desc='Participación'>
        <Dto id='NmrUser' desc='Users Number'>-1</Dto>
      </Inf>
    </AprInf>
  </BlqInf>
</CtgInf>
Línea 5, Columna 39
  
```

**Fig. 15. Estructura de un mensaje desde el Gestor para la actualización de la barra de tareas de la Consola de Awareness**

### 4.1.3 Módulo Gestor

El módulo Gestor fue desarrollado por completo en el lado servidor a través de rutinas CGI y su responsabilidad fue la de proveer un entorno para el despliegue de los servicios Awareness, para lo cual requirió soportar dos grandes tareas: la primera, la gestión del Modelo de Información a través de librerías Perl para la manipulación de bases de datos MySQL; la segunda, la habilitación de un protocolo de comunicación con el Agente basado en estructuras XML que flexibilizaba su interpretación en el lado cliente.

Los servicios provistos por el Gestor no solo buscaron favorecer las tareas de monitoreo del docente, sino además a todos los usuarios, por lo cual se hicieron disponibles como herramientas para el descubrimiento y mantenimiento del contexto grupal. A continuación se presenta un resumen de los distintos servicios implementados para KnowCat y un resumen general de su impacto sobre los usuarios.

### 4.1.4 Servicios Awareness implementados para KnowCat

Las siguientes tablas buscan resumir los servicios Awareness implementados a través de los campos: *Nombre*, nombre del servicio; *Propósito*, una descripción general del servicio; *Impacto*, un resumen desde la perspectiva colaborativa del tipo de Awareness que provee y una vista de ejemplo del servicio.

Tabla 34. Resumen del Servicio Usuarios Registrados

<b>Nombre</b>	<i>Usuarios registrados</i>
<b>Propósito</b>	Permite conocer la lista de usuarios registrados en el sistema junto con sus datos personales, información adicional (ocupación y observaciones hechas en el perfil) y una lista parcial de las últimas acciones que ha realizado el usuario sobre la plataforma, figura 16.
<b>Impacto</b>	Promueve la interacción entre usuarios basado en su perfil y últimas acciones, provee Awareness Social y de Tarea.

Fig. 16. Vista de la Consola Resumen del Servicio Usuarios Registrados

Tabla 35. Resumen del Servicio Usuarios en Línea

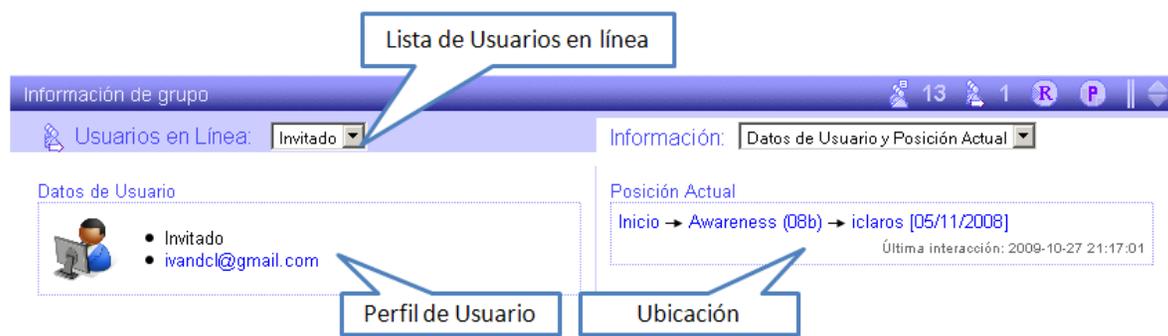
<b>Nombre</b>	<i>Usuarios en línea</i>
---------------	--------------------------

**Propósito**

Además de presentar los datos de contacto del usuario (nombre y correo electrónico), muestra la ubicación actual del usuario dentro del árbol de conocimiento, permitiendo saber si el usuario seleccionado está en una rama del árbol, un documento e incluso una anotación específica. La información de ubicación sirve de enlace directo al tópico o documento que el usuario este viendo.

**Impacto**

Promueve la interacción entre usuarios basado en la ubicación actual del usuario en el árbol de conocimiento y provee *Awareness* Social y de Espacio de Trabajo, figura 17.



**Fig. 17. Resumen del Servicio Usuarios en Línea**

**Tabla 36. Resumen del Servicio Vista Histórica**

**Nombre**

*Vista Histórica*

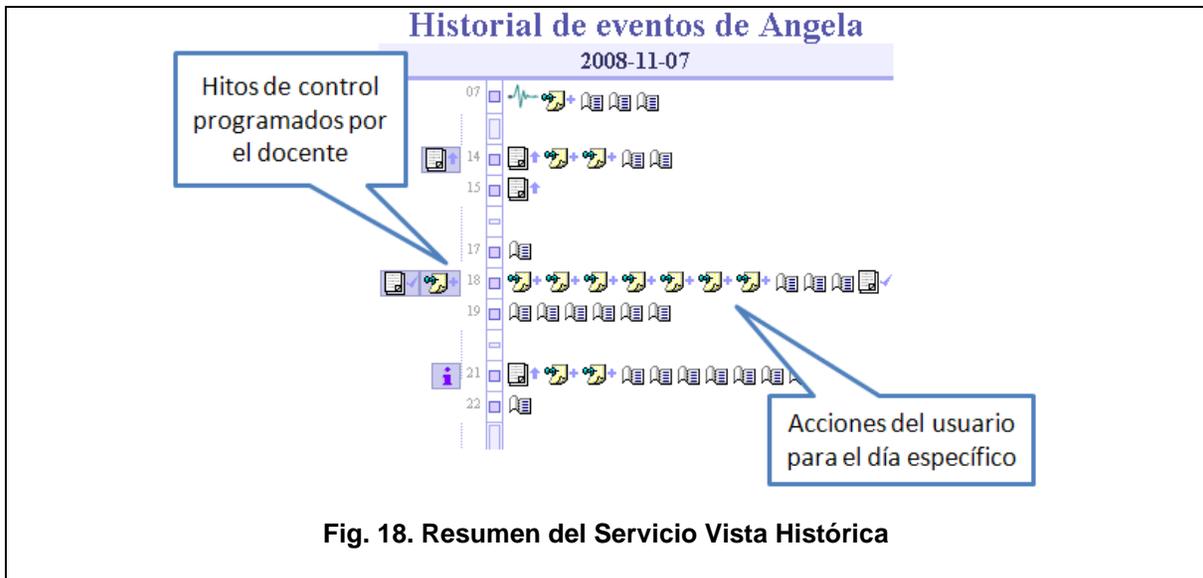
**Propósito**

Este *Widget* se presenta a través de una ventana externa a la consola y su propósito es desplegar el historial de acciones que un usuario ha realizado durante su interacción con el sistema. Para evitar la sobrecarga de texto en la interfaz, se utilizó una nomenclatura gráfica basada en íconos para señalar a través de una línea de tiempo todas las acciones que se registran de un usuario.

Adicionalmente, en la parte izquierda de la línea de tiempo se puede ver el registro de eventos programados por el docente para efectos de coordinación de la actividad, señalando hitos en el tiempo para entregas o estados en que debería un usuario encontrarse en un momento dado, figura 18.

**Impacto**

Es una herramienta que permite conocer el desempeño de los usuarios a lo largo del tiempo y contrastarlo con hitos programados por el docente. Promueve el cumplimiento de los objetivos propuestos y la interacción entre usuarios en función de su desempeño. Provee *Awareness* de Tarea.



**Fig. 18. Resumen del Servicio Vista Histórica**

**Tabla 37. Resumen del Servicio Vista de Radar**

<b>Nombre</b>	<i>Vista de Radar</i>
<b>Propósito</b>	Permite conocer la distribución de los usuarios en línea sobre el árbol de conocimiento en el momento específico. Al navegar por las distintas ramas del Radar se puede ver la ubicación exacta de los usuarios en términos del documento o anotación que está viendo, figura 19.
<b>Impacto</b>	Este mecanismo permite descubrir el área de interés de un grupo de usuarios en un momento dado, motivándolos a interactuar. Provee <i>Awareness Social</i> , de <i>Espacio de Trabajo</i> y de <i>Conocimiento Compartido</i> , este último al enfatizar sobre el área de conocimiento de interés de los usuarios.

Número de Usuarios que comparten Área

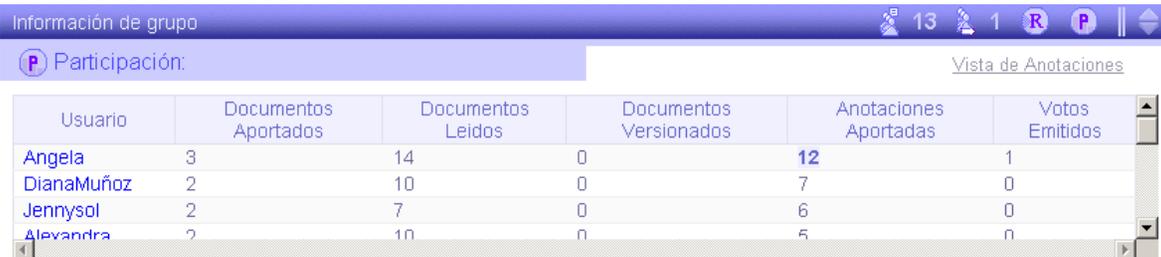
Árbol de Conocimiento

Ubicación

**Fig. 19. Resumen del Servicio Vista de Radar**

**Tabla 38. Resumen del Servicio Participómetro**

<b>Nombre</b>	<i>Participómetro</i>
<b>Propósito</b>	<p>Ofrece una vista estadística de la participación de los usuarios en las distintas tareas que soporta el sistema. Utiliza una tabla donde se ordenan los usuarios del sistema en función de: documentos aportados, documentos leídos, documentos versionados, anotaciones aportadas y votos emitidos. Cada nombre de usuario es a su vez un enlace al servicio de vista histórica con el cual se detalla la participación en una línea de tiempo.</p> <p>En la parte superior de esta interfaz se ha ubicado el enlace Vista de Anotaciones, el cual es el acceso al servicio <i>Mapa de Interacción</i>, el cual se describe posteriormente, figura 20.</p>
<b>Impacto</b>	<p>Es una herramienta de control y seguimiento de las actividades de los usuarios eficaz puesto que resume su desempeño y permite contrastarlo con el resto del grupo. Provee <i>Awareness Social</i> y de Tarea, permite identificar usuarios rezagados, lo cual suministra al docente un criterio para la intervención.</p>

The screenshot shows a web interface titled 'Información de grupo' with a navigation bar containing icons for 13 users, 1 document, and a 'P' icon. Below the bar, there is a 'Participación:' section with a 'Vista de Anotaciones' link. A table displays participation data for four users: Angela, DianaMuñoz, Jennysol, and Alejandra. The table columns are: Usuario, Documentos Aportados, Documentos Leídos, Documentos Versionados, Anotaciones Aportadas, and Votos Emitidos.

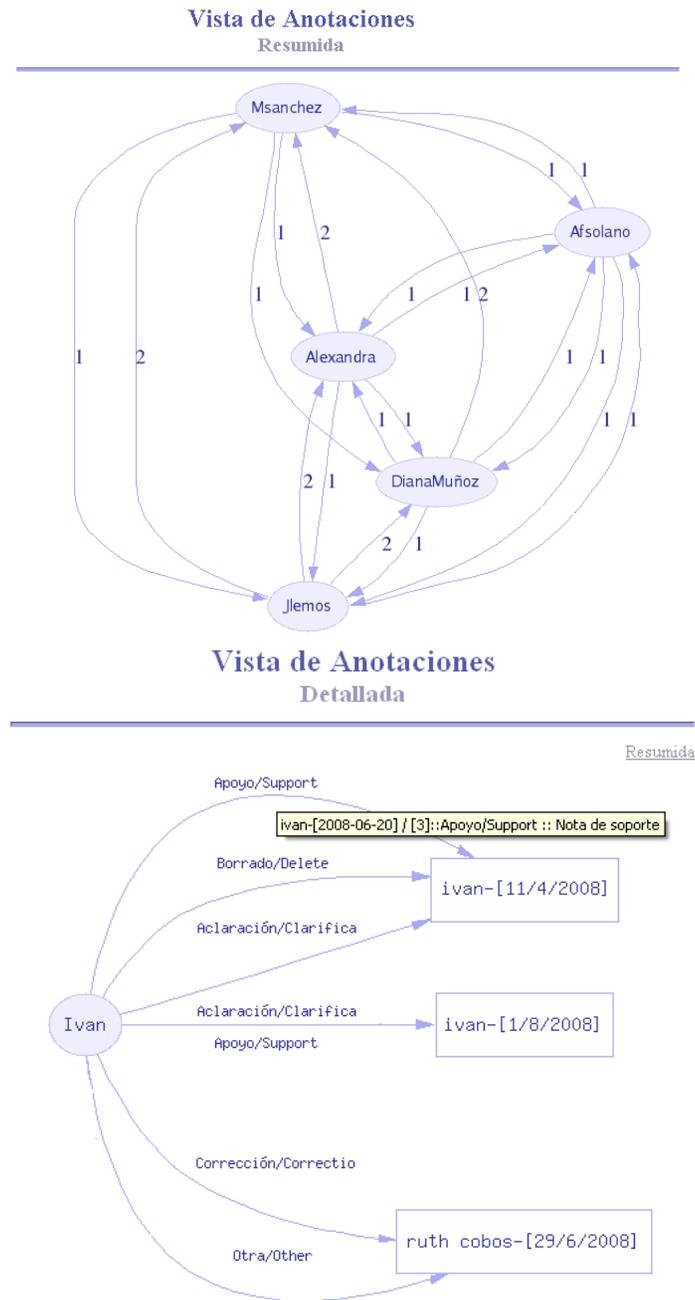
Usuario	Documentos Aportados	Documentos Leídos	Documentos Versionados	Anotaciones Aportadas	Votos Emitidos
Angela	3	14	0	12	1
DianaMuñoz	2	10	0	7	0
Jennysol	2	7	0	6	0
Alejandra	2	10	0	5	0

**Fig. 20. Resumen del Servicio Participómetro**

**Tabla 39. Resumen del Servicio Mapa de Interacción**

<b>Nombre</b>	<i>Mapa de interacción</i>
<b>Propósito</b>	<p>Este servicio despliega en forma de grafo como se ha llevado a cabo el proceso de anotaciones entre usuarios, desde una perspectiva resumida o detallada. La vista resumida solo muestra a los distintos autores, y como éstos se han hecho anotaciones unos a otros. El arco que acompaña al grafo indica cuantas anotaciones tienen un autor a otro. La vista detallada, combina autores y documentos en un mismo grafo, mostrando explícitamente cada anotación, figura 21.</p>
<b>Impacto</b>	<p>Constituye una herramienta de análisis social. Permite identificar el grado de actividad o pasividad de los usuarios y cuál ha sido el comportamiento de estos en funciones de las anotaciones realizadas. Provee <i>Awareness Social</i> centrándose en la conducta</p>

del grupo en función de las anotaciones.



**Fig. 21. Resumen del Servicio Mapa de Interacción**

Un ejemplo del prototipo desarrollado se encuentra disponible en <http://knowcat.ii.uam.es/aw08b/indexAw.pl> el cual fue utilizado para la fase de evaluación de la propuesta. Se ha dispuesto temporalmente un usuario de prueba con nombre: invitado y contraseña: invitado. Los datos que aparecen en ese nodo pertenecen a usuarios reales. En la sección de pruebas de esta monografía se detalla el proceso experimental del mismo.

## 4.2 Caso de estudio II, Libro Digital de Trabajo y Red Multi-Agente

El segundo escenario de validación desarrollado se dividió en tres fases: en la primera, se construyeron dos aplicaciones: un Libro Digital de Trabajo y una herramienta de configuración para el docente (CETProfesor), como soporte a la colaboración en el contexto de la Técnica de Evaluación Colaborativa, CET [68], partiendo del requisito de utilizar un entorno TabletPC. La segunda fase, consistió en la implementación de un Sistema Multi-Agente que proporciona elementos de soporte para la gestión de información de *Awareness* y el despliegue de servicios orientados a mejorar el escenario educativo promoviendo la interacción entre usuarios e implantando estrategias de gestión de conocimiento tales como la búsqueda de expertos. Finalmente, la tercera fase, consistió en la integración de las herramientas desarrolladas en la primera fase con el Sistema Multi-Agente.

Por otra parte, durante la segunda fase de este desarrollo se afrontó el inconveniente de no contar con entornos de trabajo adecuados para el desarrollo de sistemas Multi-Agente sobre la plataforma .NET. Si bien en la tecnología Java se encuentran diversas plataformas comerciales y libres para el soporte de agentes, entre las que se conoce: *Jade* [67], *Jack*[69], *jadex* [70], *Tryllian* [71], *AgentBuilder* [72], *3APL* [73] y *AgentFactory* [74], entre muchas otras, solo se encontró una única aproximación para el uso de agentes sobre plataformas .NET llamada *JADELEAP.NET* [75], sin embargo esta no ofrece un soporte adecuado a los requerimientos del sistema ni resultó compatible con el entorno tecnológico para el cual se debía diseñar la solución; por lo tanto, para complementar el desarrollo de este segundo caso de estudio se implementó un conjunto de librerías para la creación de Sistemas Multi-Agente basado en tecnología .NET *Windows Communication Foundation (WCF)* [76] al cual se denominó **AgNet** (Agentes para .NET). La descripción de AgNet como framework para el desarrollo de un Sistema Multi-Agente se retoma en la sección 4.2.2 de ese capítulo.

A continuación se presentan las aplicaciones Libro Digital de Trabajo y CETProfesor; luego se describe el framework para el desarrollo de Sistemas Multi-Agente *AgNet* y posteriormente el Sistema Multi-Agente implementado. Finalmente se desarrolla la integración entre el Sistema Multi-Agente y las herramientas Libro Digital de Trabajo y CETProfesor.

### 4.2.1 Libro Digital de Trabajo y CETProfesor

La Técnica de Evaluación Colaborativa CET (Collaborative Evaluation Technique), estructura en forma colaborativa las evaluaciones para convertirlas en mecanismos de aprendizaje [68]. Propone dividir el proceso de evaluación en 3 etapas, las cuales se desarrollan en forma consecutiva: *pre-test*, *test* y *post-test*, cada una de estas etapas ocurre en un espacio de tiempo diferente.

El *pre-test* posee una estructura similar a las actividades llevadas a cabo en la técnica *Jigsaw* [77], busca disminuir la ansiedad de los estudiantes observando un cuestionario similar al del examen, teniendo la posibilidad de construir y validar las respuestas en forma grupal. Por su parte, el *test* consiste en presentar el examen de forma individual siguiendo los procedimientos tradicionales. Finalmente el *post-test* tiene como objetivo que los estudiantes aprendan de sus errores y puedan establecer guías para el proceso de corrección; para ello reciben una copia del examen realizado y en grupos de 4 o 5

personas intentan resolver el examen, construyendo así lo que ellos consideran que sería la pauta de solución.

Al concluir el *post-test*, los estudiantes se autoevalúan asignándole un puntaje a su examen en función de las respuestas que ellos consideran correctas. La autoevaluación debe argumentar el puntaje que el alumno le asignó a cada pregunta. Aquellos estudiantes cuya autoevaluación está muy cerca de la evaluación de los profesores, tienen un bono adicional en cada pregunta el cual retribuye el hecho de que el estudiante para llegar a una autoevaluarse acorde a la realidad, hayan encontrado las respuestas correctas a las preguntas, aunque fuera del tiempo para resolver el examen.

En este contexto, se propuso el desarrollo de la herramienta denominada Libro Digital de Trabajo la cual implementa la metáfora de un libro de apuntes y permite a los estudiantes construir de forma colaborativa las respuestas a los problemas planteados en las fases de pre-test y post-test. El Libro Digital de Trabajo faculta a los estudiantes para crear hojas de trabajo que pueden compartir, anotar y publicar como solución a preguntas.

Por otra parte, la herramienta que complementa las funciones del Libro Digital de Trabajo desde el contexto de la evaluación colaborativa es *CETProfesor*: una aplicación con la cual el docente registra las preguntas del examen, configura el acceso a los grupos de usuario que van a aplicar a la evaluación e inicia los servicios de comunicación que permite al Libro Digital de Trabajo conocer las preguntas y publicar las respuestas del examen. Además, provee algunos servicios de monitoreo tales como Sociograma, Participómetro y Usuarios en Línea.

A continuación se presentan algunos de los diagramas UML utilizados en el análisis y diseño de estas herramientas.

#### 4.2.1.1 Diagramas de Casos de Uso

La figura 22 presenta el diagrama de casos de uso relacionados con la aplicación Libro Digital de Trabajo, los cuales se describen a continuación:

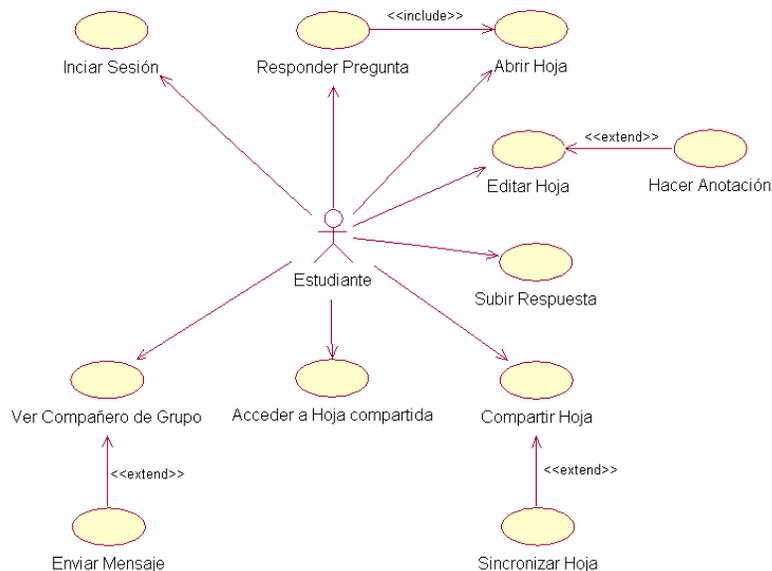
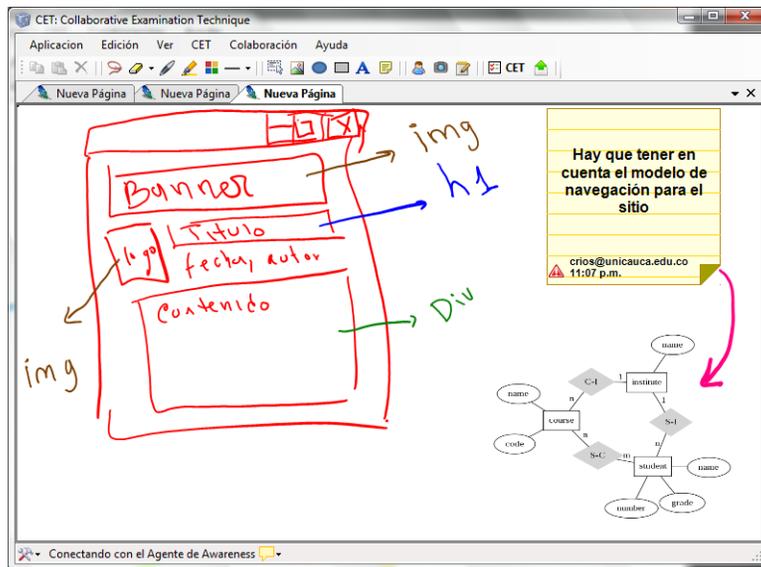


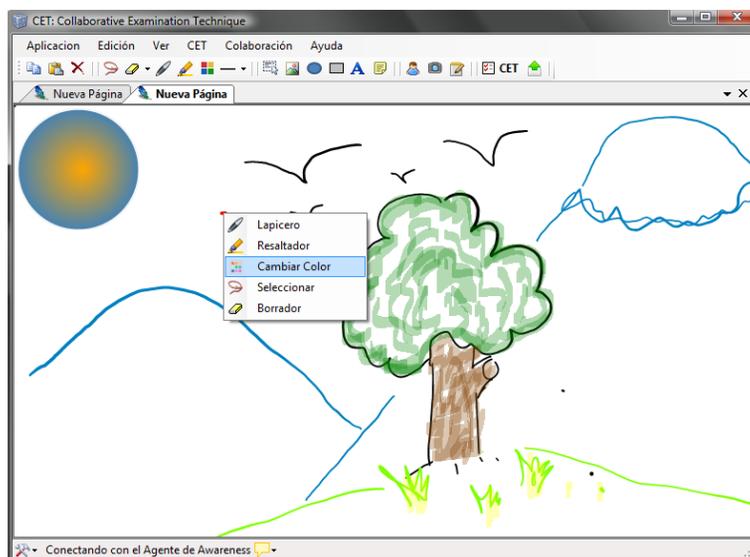
Fig. 22. Diagrama de Casos relacionados con la herramienta Libro Digital de Trabajo

- **Iniciar Sesión:** permite que el Estudiante se registre ante el servicio de examen para luego recibir la lista de compañeros de grupo y de preguntas del examen. Es precondition a los casos de uso Responder Pregunta y Subir Respuesta.
- **Abrir Hoja:** permite agregar una nueva hoja al libro de trabajo, sobre la cual se puede realizar anotaciones, ingresar imágenes y dibujar trazos libres, como se presenta en la figura 23. Cuando es creada desde el contexto de una pregunta, es asociada como la respuesta a dicha pregunta.



**Fig. 23. Caso de uso Abrir Hoja**

- **Editar Hoja:** permite la edición del contenido de una hoja de trabajo teniendo en cuenta las capacidades de los entornos TabletPC, tales como lapicero, resaltador, cambiar color de tinta, seleccionar y borrar trazos, como se visualiza en la figura 24.



**Fig. 24. Caso de uso Editar Hoja**

- **Hacer Anotaciones:** permite agregar Anotaciones al contenido de una página, la cual relaciona el texto con el correo del autor y la fecha en la cual se hizo la anotación, además se tienen tres tipos de anotaciones: de aceptación, rechazo y duda como se muestra en la figura 25.

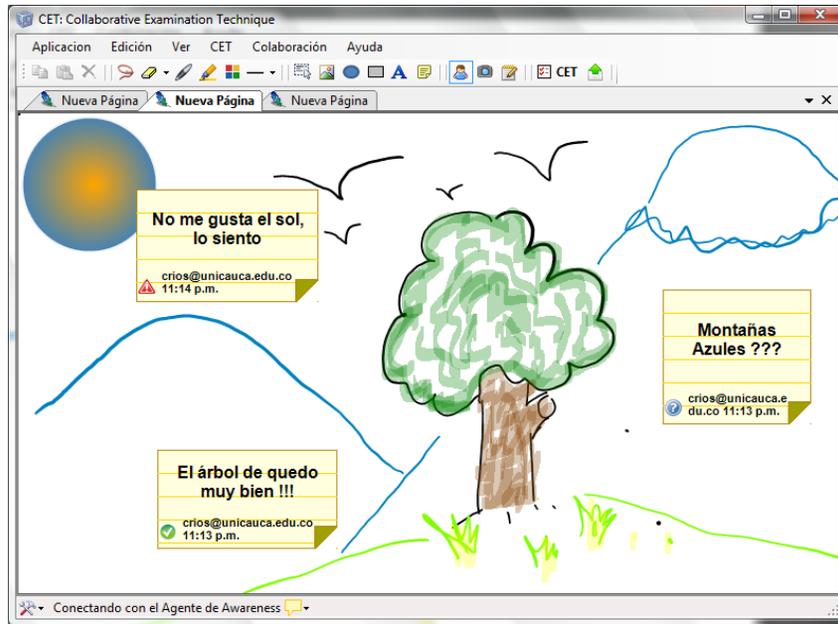


Fig. 25. Caso de uso Hacer Anotaciones

- **Responder Pregunta:** a partir de una pregunta asignada, crea una hoja de trabajo asociada como respuesta, como se presenta en la figura 26.

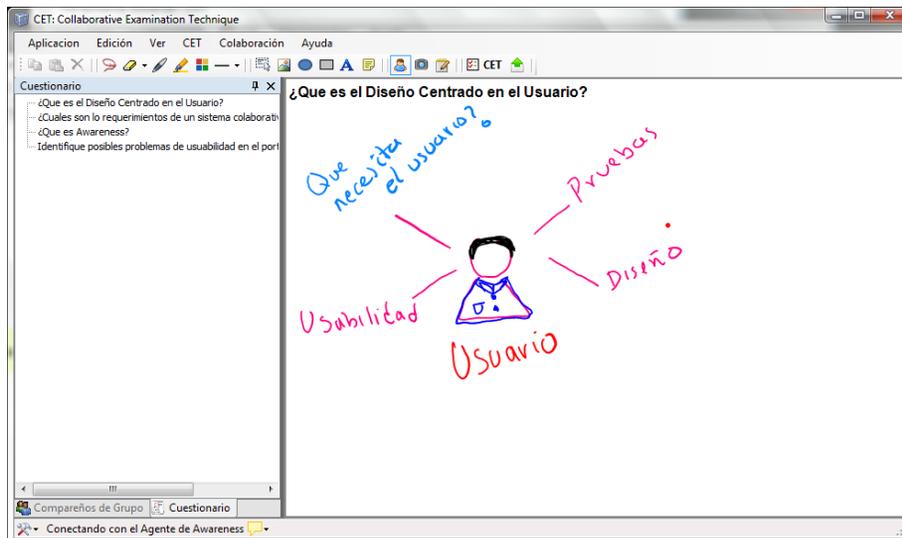


Fig. 26. Caso de uso Responder Pregunta

- **Subir Respuesta:** Publica ante el docente, el contenido de la hoja como respuesta a la pregunta asociada, en la figura 27 se presenta como el docente puede acceder a la respuesta del estudiante presentado en la anterior figura.

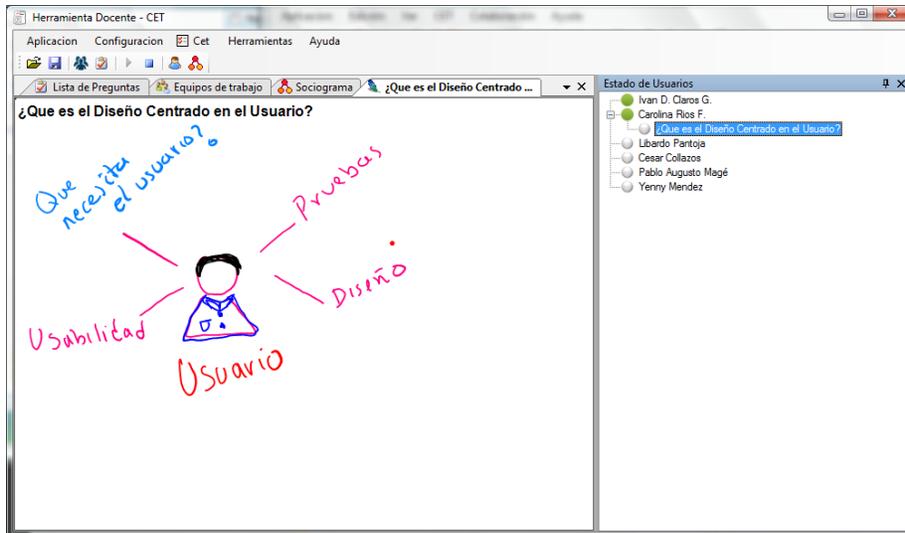


Fig. 27. Caso de uso Subir Respuesta

- **Anotar respuesta:** permite acceder a la hoja de respuesta de otro estudiante para cambios o anotaciones, lo cual se registra como una nueva solución a la pregunta planteada como se visualiza en el lado izquierdo de la figura 28, similar a los hilos de discusión que se presentan en los foros virtuales.

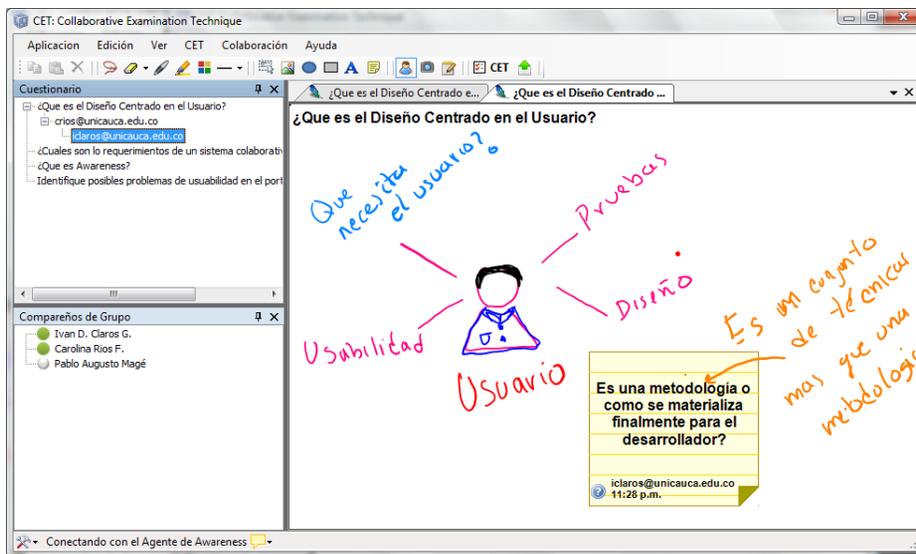
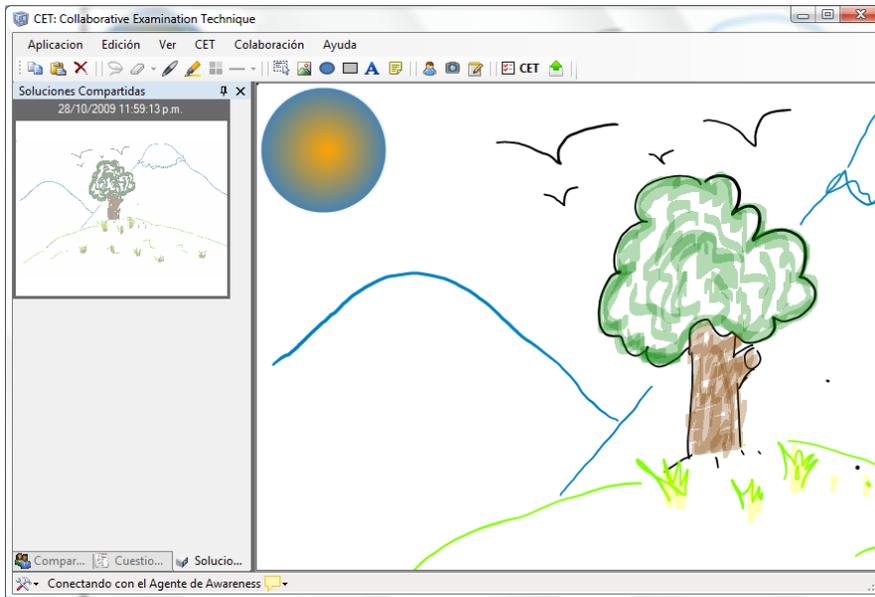


Fig. 28. Caso de uso Anotar respuesta

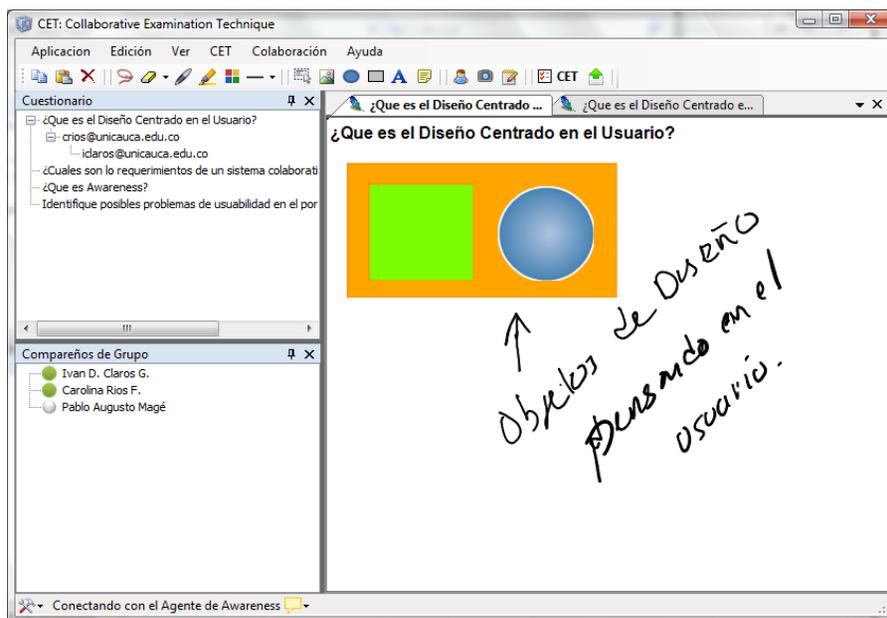
- **Compartir Hoja:** habilita mecanismos que hacen disponible compartir distintas hojas de trabajo de forma independiente al servicio de CET, permitiendo a los estudiantes interactuar incluso con usuarios que no estén registrados en su grupo de trabajo inicial.

- **Acceder a Hoja compartida:** permite acceder a las hojas compartidas por otro usuario fuera del contexto de CET, figura 29.



**Fig. 29. Caso de uso Acceder a Hoja compartida**

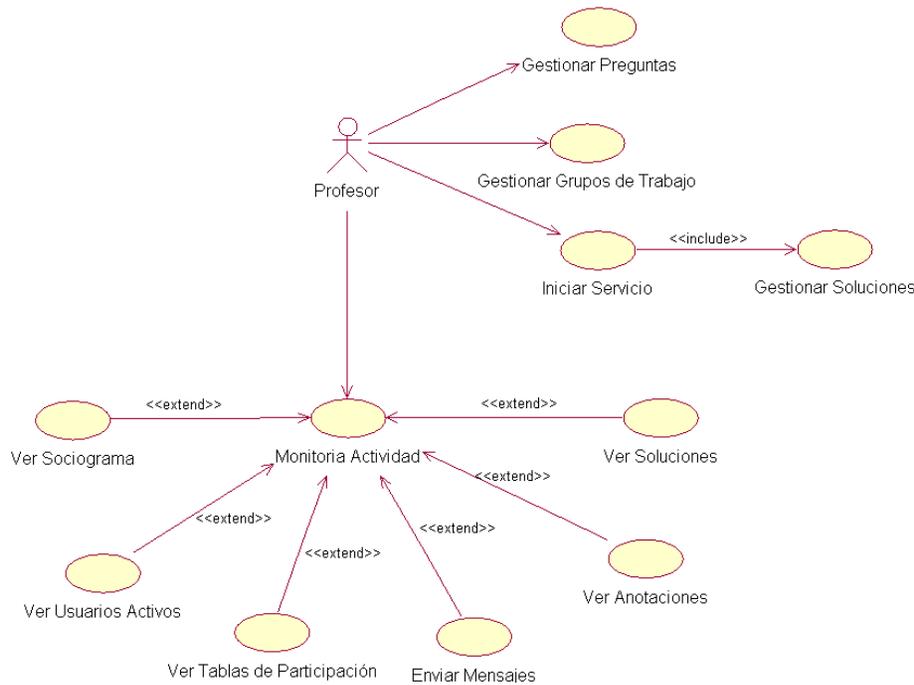
- **Sincronizar Hoja:** actualiza el contenido de una hoja compartida a través de eventos manuales o automáticos.
- **Ver Compañeros de grupo:** permite acceder a la lista de estudiantes asociados en su grupo de trabajo visualizando su estado en el sistema (en línea o desconectado), como se muestra en la parte de abajo con lado izquierdo de la figura 30.



**Fig. 30. Caso de uso Ver Compañeros de grupo**

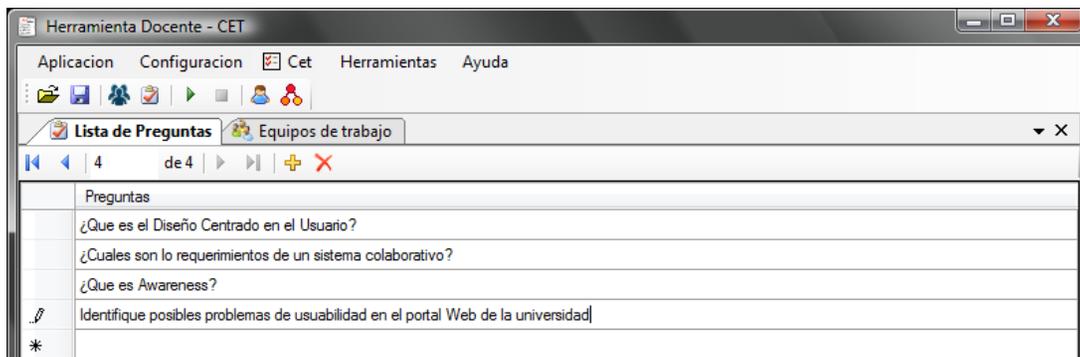
- **Enviar Mensaje:** permite enviar mensajes a sus compañeros de grupo.

Por su parte, en la figura 31 se presenta el diagrama de casos de uso para CETProfesor, los cuales se describen a continuación:



**Fig. 31. Diagrama de casos de uso relacionados con la herramienta CETProfesor**

- **Iniciar Servicio:** inicia los mecanismos de comunicación que permite a los estudiantes acceder a la información de grupo y a la lista de preguntas del examen.
- **Gestionar Preguntas:** brinda al docente toda la funcionalidad relacionada con la creación, edición y eliminación de las preguntas utilizadas para el examen, figura 32.



**Fig. 32. Caso de uso Gestionar Preguntas**

- **Gestionar grupos de trabajo:** provee al docente toda la funcionalidad relacionada con la creación, edición y eliminación de estudiantes y grupos de trabajo, figura 33.

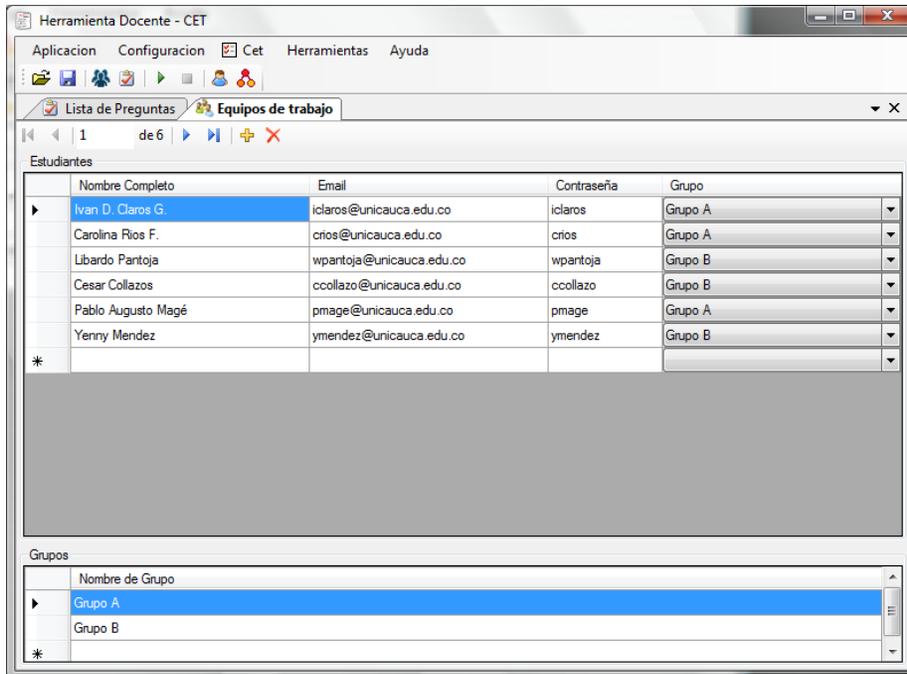


Fig. 33. Caso de uso Gestionar grupos de trabajo

- **Gestionar Soluciones:** permite llevar un control de las soluciones publicadas por los usuarios. como se muestra en la parte de arriba con lado derecho de la figura 34.

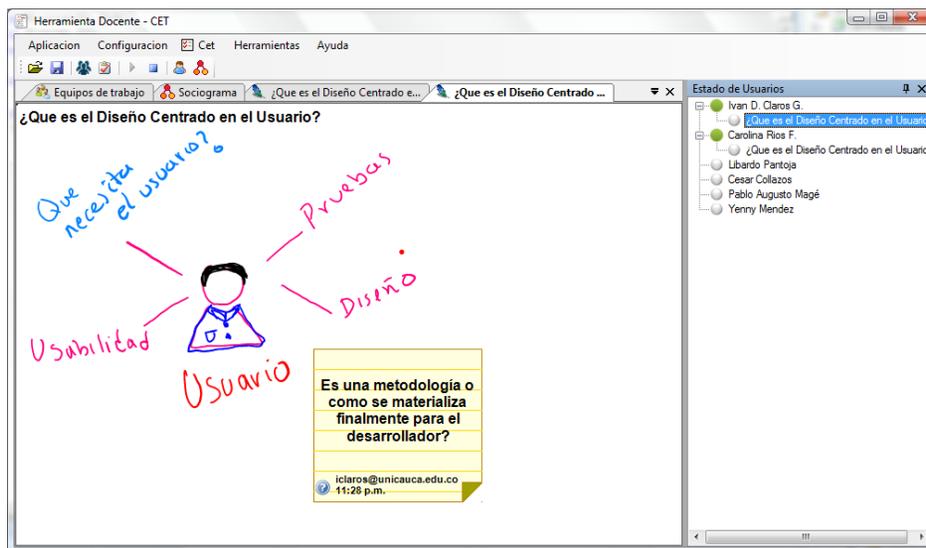
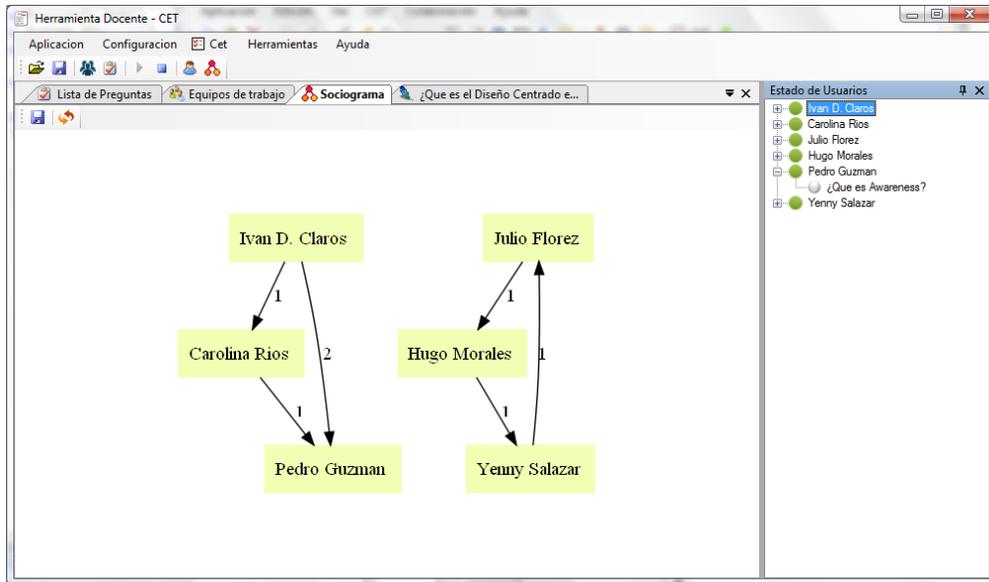


Fig. 34. Caso de uso Gestionar grupos de trabajo

- **Monitorear Actividad:** inicia un conjunto de servicios relacionados con el monitoreo y control de la actividad colaborativa, tales como:
  - **Ver Sociograma:** despliega en forma de grafo la relación de los usuarios a partir de las anotaciones en sus hojas de trabajo, como se presenta en la figura 35.

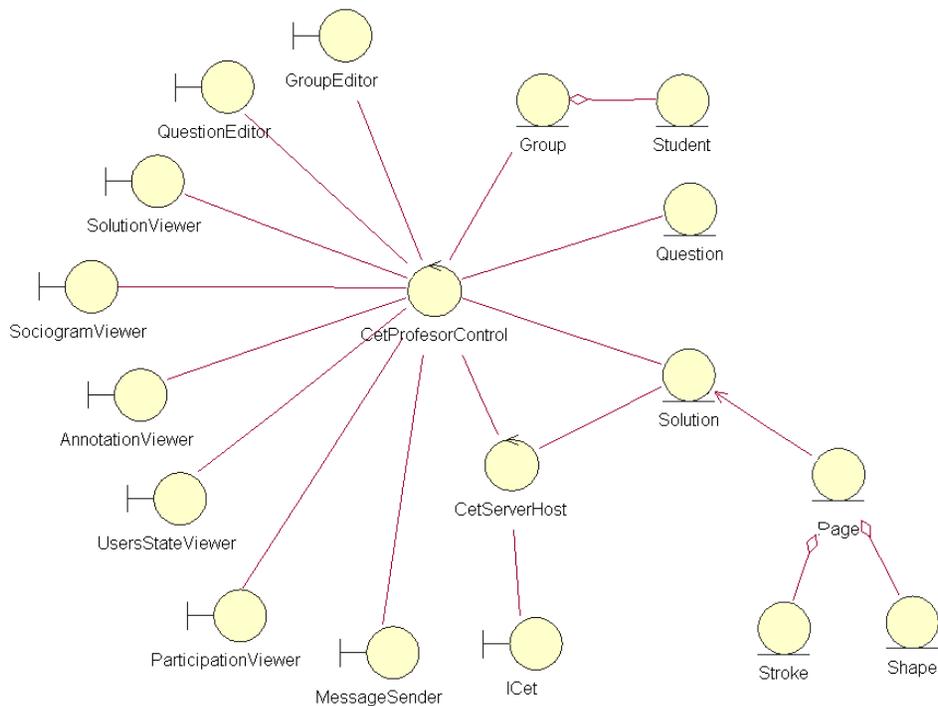


**Fig. 35. Caso de uso Ver Sociograma**

- **Ver Usuarios Activos:** presenta la lista de estudiantes registrados en el sistema y su estado de conexión (en línea / desconectado).
- **Ver Tablas de Participación:** muestra una vista estadística de los aportes de cada usuario y su interacción con el sistema.
- **Enviar Mensajes:** permite al docente enviar mensajes a los grupos de trabajo.
- **Ver Anotaciones:** permite al docente ver las anotaciones publicadas por un estudiante a una hoja de trabajo.
- **Ver Soluciones:** permite al docente ver las soluciones publicadas por un estudiante.

#### 4.2.1.2 Diagrama de Clases de análisis

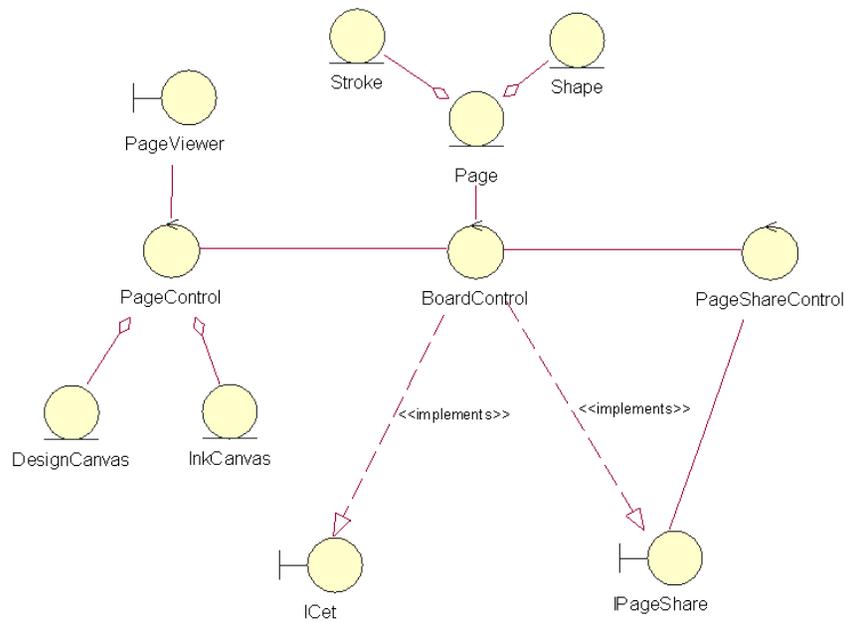
La figura 36 presenta el conjunto de clases de análisis asociadas con CETProfesor, donde se destacan las clases de control CetProfesorControl y CetServerHost. Mientras que la primera gestiona las interfaces gráficas utilizadas para la gestión de la información y monitoreo de la actividad, la segunda es utilizada para gestionar las interfaces de comunicación con el Libro Digital de Trabajo.



**Fig. 36. Diagramas de clase de análisis para CETProfesor**

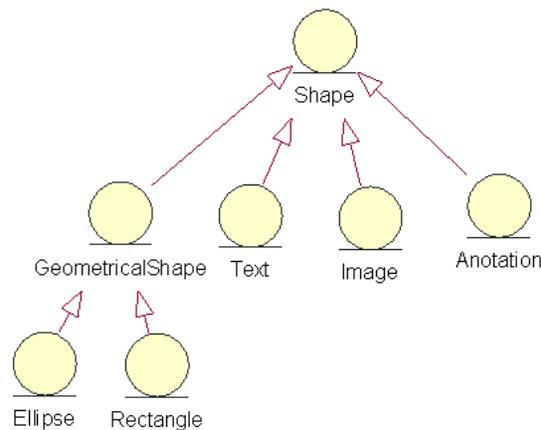
Las entidades *Group*, *Student* y *Question* son creadas en tiempo de diseño del examen, mientras que las instancias de la entidad *Solution* son creadas mientras se ejecutan los procesos asociados con la aplicación del examen. Cada *Solution* se asocia con un objeto de la entidad *Page* que corresponde al contenido de una hoja de trabajo el cual se expresa en términos de trazos (*Stroke*) y figuras (*Shape*). Un tipo particular de figura son la Anotaciones, las cuales son utilizadas para análisis social de la interacción de los estudiantes.

Por otro lado, la figura 37 presenta el diagrama de clases de análisis de la aplicación Libro Digital de Trabajo donde llama la atención las clases de control *PageControl*, *BoardControl* y *PageShareControl*. *PageControl* se encarga de gestionar los aspectos gráficos de la herramienta para lo cual utiliza dos clases de modelo: *DesignCanvas*, que ofrece funciones para la manipulación de figuras (*Shape*) e *InkCanvas* para la manipulación de trazos (*Stroke*). Por su parte *BoardControl* manipula las entidades *Page* e implementa las interfaces *ICet* e *IPageShare*, la primera corresponde al contrato de servicio con la aplicación CETProfesor y la segunda permite acceder a hojas compartidas por otros Libros Digitales. La clase de control *PageShareControl* es la que se encarga de proveer el servicio que comparte las hojas de trabajo entre Libros.



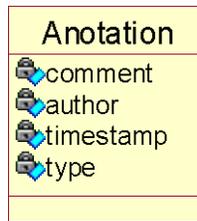
**Fig. 37. Diagrama de clases de análisis para el Libro Digital de Trabajo**

En la figura 38 se ve la jerarquía de clases de entidad asociadas con la clase *Shape* (Figura), una hoja de trabajo puede contener figuras geométricas básicas, tales como elipses y rectángulos, texto, imágenes pero en especial Anotaciones.



**Fig. 38. Jerarquía de Clases de entidad asociadas con Shape**

La estructura de la clase *Anotation* (Anotación) se puede ver en la figura 39, es un tipo especial de figura y se utilizada para plasmar un comentario sobre el contenido de la hoja de trabajo. Su atributos son: *comment*, que contiene el texto que forma la anotación; *author*, correo electrónico del usuario que hizo la anotación, *timestamp*, marca de tiempo del momento en que fue creado el comentario y *type*, como el tipo de anotación. Para este prototipo se han propuesto los siguientes tipos de anotaciones: *Support*, cuando el autor apoya el contenido; *Question*, cuando el autor quiere expresar una duda relacionada con el contenido, y *Reject*, cuando el autor rechaza o no está de acuerdo con elementos del contenido. Se espera que para futuras mejoras, esta clasificación sea ampliada y enriquecida con nuevos atributos.

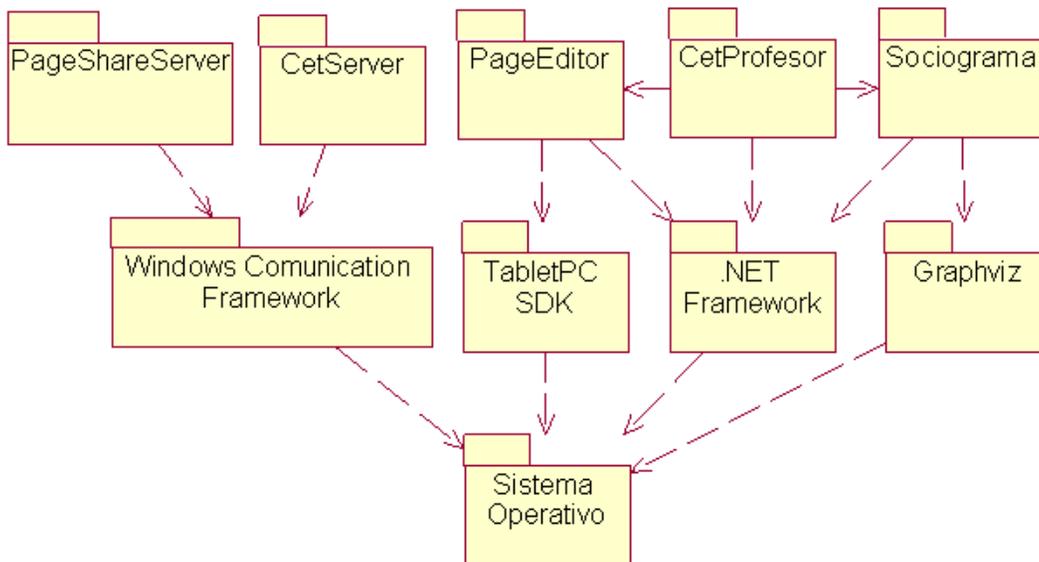


**Fig. 39. Clase Anotación**

El propósito de la definición de una clasificación para las anotaciones es simplificar su interpretación por parte del sistema, de tal forma que utilizando este atributo se pueda estimar el propósito de la anotación sin entrar a analizar el contenido de la misma.

#### 4.2.1.3 Diagrama de Paquetes

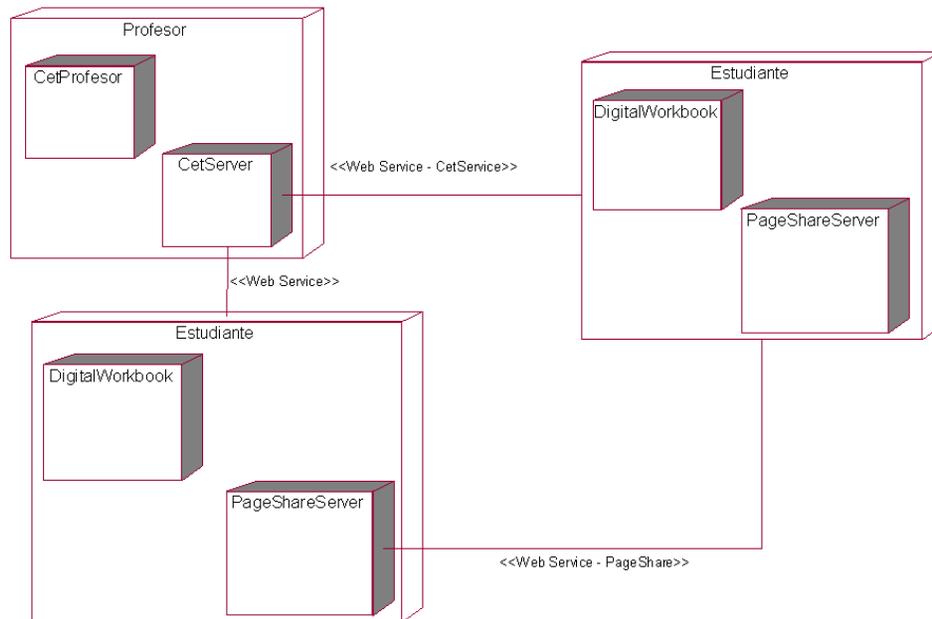
La figura 40 presenta el diagrama de paquetes relacionados con las aplicaciones Libro Digital de Trabajo y CETProfesor. La capa intermedia de paquetes muestra la dependencia que existe entre estas herramientas y librerías tales como el framework de comunicaciones de Windows, librerías para el manejo de las capacidades de los TabletPC, el framework básico .NET 3.5 y Graphviz, esta última es un conjunto de utilidades utilizadas para la construcción de los grafos que se despliegan en el Sociograma.



**Fig. 40. Vista de la arquitectura por capas**

#### 4.2.1.4 Diagrama de implantación

En la figura 41 se presenta el diagrama de implantación del sistema, en el cual se puede observar dos tipos de nodos, Profesor y Estudiante. Además, se han dispuesto dos tipos de canales de comunicación que utilizan *WebServices*: el primero se emplea para distribuir las preguntas, respuestas, mensajes y listas de usuario, desde el Profesor al Estudiante; el segundo, para compartir hojas de trabajo entre Estudiantes.



**Fig. 41. Diagrama de implantación del Sistema**

Sobre el nodo profesor corren los procesos *CETProfesor* y *CETServer*. el primero corresponde al manejo gráfico de la configuración del sistema, permitiendo al profesor gestionar el examen y grupos de trabajo; el segundo corresponde al servicio de comunicaciones que se inicia cuando el docente publica el examen.

Por su parte, sobre el nodo Estudiante corren los procesos Libro Digital de Trabajo y *PageShareServer*. el primero actúa como la aplicación cliente del libro de trabajo, mientras que el segundo es el servicio que se lanza cuando se comparte el contenido de una hoja. El proceso de *PageShareServer* es opcional y brinda al estudiante la capacidad de compartir sus hojas de trabajo con otros Estudiantes, antes de publicarlas como solución a las preguntas.

#### **4.2.2 AgNet, framework para el desarrollo de Sistema Multi-Agente**

*AgNet* es nuestra propuesta de un framework de desarrollo para Sistemas Multi-Agente compatible con tecnología .NET. Esta desarrollado en el lenguaje C# y soportado en la framework .NET 3.5. Implementa los conceptos básicos del paradigma de Agentes Inteligentes basado en los estándares propuestas por la FIPA (*Foundation for Intelligent Physical Agents*) [78], si embargo se encuentra en etapas tempranas de su desarrollo y no cuenta aun con la implementación de servicios como el AMS (Sistema Manejador de Agentes o Páginas Blancas) y el DF (Directorio Facilitador ó Páginas Amarillas) [77]. La implementación completa del estándar FIPA se propone como un trabajo futuro.

Las principales características de esta framework son:

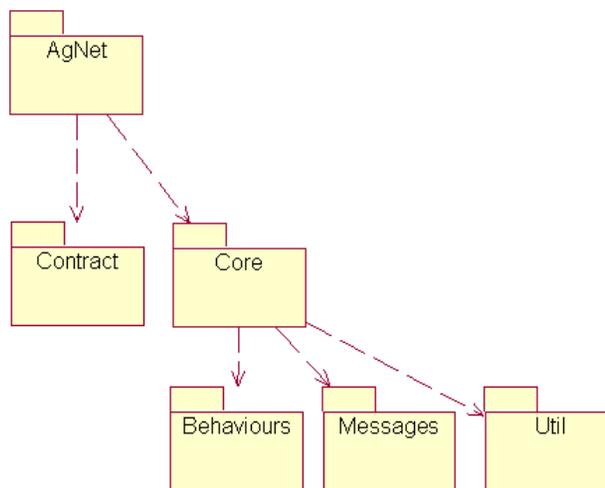
- Plataforma distribuida para agentes los cuales se interconectan a través de canales de comunicación *Peer To Peer* soportados por WCF de .NET.

- Cada Agente existe en el contexto del *RuntimeAgent*, una clase de control que le permite acceder a distintos servicios en tiempo de ejecución entre los que se cuentan la comunicación.
- Un agente es implementado como una clase de control para múltiples hilos derivados de los distintos comportamientos (*Behaviour*) con los cuales se asocia.
- La estructura de mensajes basada en la especificación *ACL Message Structure* de la FIPA [79].
- Los canales de comunicación implementan colas de mensajes para evitar la pérdida de datos.
- Soporta el transporte de cualquier objeto serializable definido en el sistema.

A continuación se presentan los detalles de la arquitectura del framework y un guía resumida de cómo es utilizado mediante el caso práctico de los agente *Ping* y *Pong*.

#### 4.2.2.1 La Arquitectura AgNet

En la figura 42 se presenta el diagrama general de paquetes de la plataforma AgNet, mientras que en la figura 43 su diagrama de Clases de diseño. A continuación se describen los paquetes y clases que definen la arquitectura de este framework.



**Fig. 42. Diagrama de Paquetes de AgNet**

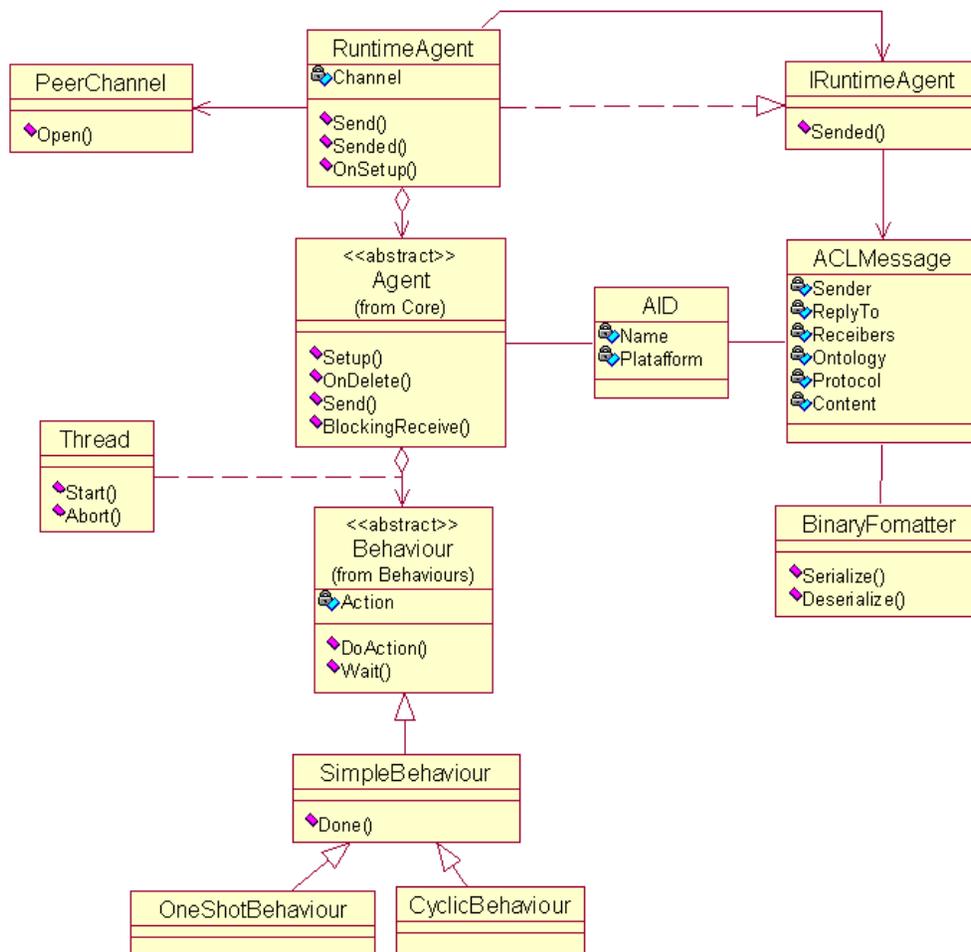


Fig. 43. Diagrama de clases del Framework AgNet

**Paquete Contract.** contiene la definición de la interfaz *IRuntimeAgent* la cual define la interfaz de servicio entre las instancias del *RuntimeAgent*. Su contenido se presenta a continuación:

```
[ServiceContract(CallbackContract = typeof(IRuntimeAgent))]
public interface IRuntimeAgent
{
    [OperationContract(IsOneWay = true)]
    void Sended(ACLMessage msg);
}
```

*IRuntimeAgent* es una interfaz con un único método llamado *Sended*, el cual es invocado sobre las instancias del canal de comunicaciones para enviar los paquetes de información a la red y de forma complementaria, esta interfaz es implementada para recibir los eventos de llegada de un mensaje.

Toda la información que se transmite sobre la red de agentes es encapsulada en instancias de la clase *ACLMessage*, del paquete *Messages*, aplicando algunas de las recomendaciones del estándar FIPA 061 [79]. Su definición es la siguiente:

```

[MessageContract]
public class ACLMessage
{
    [MessageHeader]
    public byte[] Content { get; set; }

    [MessageHeader]
    public string Protocol { get; set; }

    [MessageHeader]
    public Guid ConversationId { get; set; }

    [MessageHeader]
    public AID Sender { get; set; }

    [MessageHeader]
    public List<AID> Receivers { get; set; }

    [MessageHeader]
    public List<AID> ReplyTo { get; set; }

    [MessageHeader]
    public string Ontology { get; set; }
}

```

Los atributos *Ontology* y *Protocol* de la clase *ACLMessage* son la clave del entendimiento en la comunicación puesto que a través de estas dos cadenas de texto, el Agente expresa el tipo de contenido dispuesto en el mensaje y su propósito. Para simplificar el proceso de serialización de los objetos que hacen parte del contenido, se ha utilizado una instancia de la clase *BinaryFormatter* con la cual se transforma cualquier objeto serializable en un cadena de bytes.

Por su parte el **paquete Core** contiene la definición de las clases *AID*, *Agent* y *RuntimeAgent*, entre otras. La clase *AID* encapsula la identificación del Agente con dos atributos: *name*, el nombre del agente y *plataform*, el identificador el *RuntimeAgent* que lo contiene.

La clase *Agent* es la clase base de la cual se derivan todos los agentes de sistema. Actúa como entidad de control para los múltiples hilos que se van siendo creados en función de los comportamientos (*Behaviours*) que se programen para el Agente. Además, provee funcionalidades tales como *Send* o *BlockingReceive* para enviar y capturar información respectivamente.

La clase *RuntimeAgent* representa las funciones de un contenedor de agente, proporcionando al agente un conjunto de servicios y gestionando su ciclo de vida, este último se define a través de la invocación de los métodos *Setup* y *DoDelete*. El atributo principal del *RuntimeAgent* es el canal de comunicaciones el cual utiliza para enviar información a la red Peer. De igual forma, el *RuntimeAgent* implementa la interfaz *IRuntimeAgent* para poder recibir mensajes desde la red. Cada vez que se invoca el método *Sended* sobre el canal, este evento se refleja a todos las entidades *RuntimeAgent* que se encuentren conectadas a la red. El método *OnSetup* de esta clase se invoca luego que se ha inicializado de forma correcta el canal de comunicación, por lo tanto es dentro de este método o luego de su ejecución donde se agregan los Agentes al *RuntimeAgent*.

Cuando un mensaje es capturado por el *RuntimeAgent*, este analiza la cabecera de destino y lo compara con las direcciones de los agentes que tiene registrados, al encontrar coincidencia, lo coloca en la cola del buffer de comunicaciones del agente y procede a disparar señales de desbloqueo a través de un objeto *Mutex* el cual sirve como semáforo para los procesos de consumo de paquetes.

El **Paquete *Behaviours*** contiene la definición de las clases que definen el comportamiento de un Agente. La clase de la cual se derivan todos los comportamientos se llama *Behaviour*. Esta clase posee un atributo principal llamado *Action* de tipo Delegado. Un Delegado en la tecnología .NET, es un objeto que encapsula un método, de forma muy similar a los punteros a funciones de C/C++. Dependiendo del tipo de comportamiento que se programe, el delegado es invocado una o muchas veces.

En AgNet se ha dado soporte a tres tipos específicos de *Behaviours*: *SimpleBehaviour*, como comportamientos que repiten su delegado hasta que el método *Done* de esta clase retorna falso, *OneShotBehaviour*, para un único disparo del delegado y *CyclicBehaviour*, para repeticiones continuas e indeterminadas de comportamiento. Cada *Behaviour* es ejecutado en el contexto de un objeto de la clase *Thread* y gestionado por el agente, de esta forma se describen como procesos paralelos.

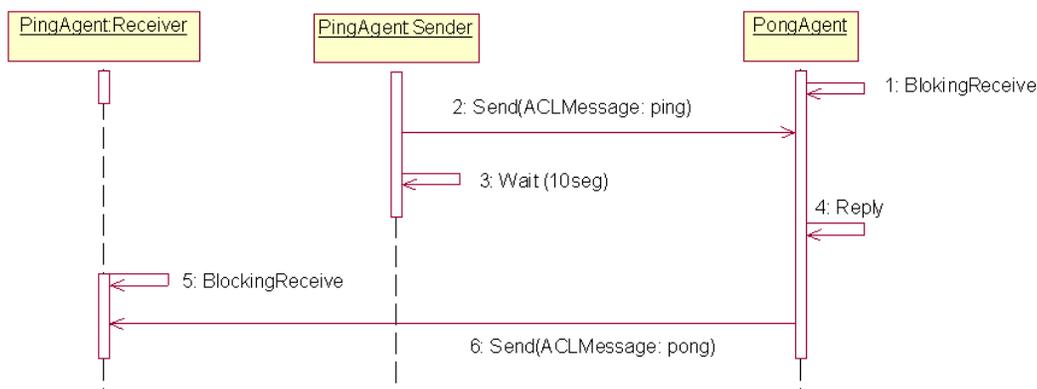
El **paquete *Messages***, además de contener la definición de la clase *ACLMessage*, contiene clase *MessageTemplate* la cual puede ser utilizada con argumento en la captura de mensajes por parte del agente para filtrar mensajes que se ajusten a un mensaje *ACLMessage* que es utilizado como plantilla.

El **paquete *Util***, por su parte, implementa algunas clases que facilitan el desarrollo de tareas comunes, por ejemplo la implementación del patrón decorador para retornar una clase para el manejo de flujo de datos *PrintWriter*, a partir de un Área de Texto, lo cual es empleado para redireccionar los mensajes de depuración del sistema hacia interfaces gráficas de usuario.

#### 4.2.2.2 El agente Ping y el Agente Pong

Para dar una visión pragmática de AgNet, a continuación se describe la implementación de un sistema que incluye el desarrollo de los agentes *Ping* y *Pong*. El Agente *Ping* exhibe dos comportamientos: el primero, es generar un mensaje *ping* y enviarlo al agente *Pong*, luego de lo cual detiene su ejecución por 10 segundos antes de repetirlo; el segundo comportamiento es capturar los mensajes *pong* de respuesta. Por su parte, el Agente *Pong* implementa un único comportamiento, el de escuchar los mensajes *ping* y replicarlos cambiando el contenido del mensaje por la palabra “pong”.

El diagrama de secuencias de la figura 44 muestra la dinámica entre los Agentes *Ping* y *Pong*. Para el agente *Ping* se han asociado las clases: *Sender* y *Receiver* que corresponden a los dos comportamientos esperados para este agente. Cabe recordar que dichos comportamientos ocurren en paralelo.



**Fig. 44. Diagrama de secuencia para la interacción entre los agentes *Ping* y *Pong***

Un paso importante en la construcción del sistema es la edición del archivo de configuración del proyecto (*App.Config*) por cuanto contienen las variables relacionadas con el canal de comunicación. Para este ejemplo dicho archivo contiene la siguiente estructura:

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <client>
      <endpoint name="AgentEndPoint"
        address="net.p2p://agnet"
        binding="netPeerTcpBinding"
        bindingConfiguration="PeerTcpConfig"
        contract="AagnetFramework.Contract.IRuntimeAgent" />
    </client>
    <bindings>
      <netPeerTcpBinding>
        <binding name="PeerTcpConfig" port="0"
          maxReceivedMessageSize="2147483647" maxBufferPoolSize="2147483647">
          <security mode="None" />
          <resolver mode="Auto" />
          <readerQuotas maxDepth="2147483647"
            maxStringContentLength="2147483647" maxArrayLength="2147483647"
            maxBytesPerRead="2147483647" maxNameTableCharCount="2147483647" />
        </binding>
      </netPeerTcpBinding>
    </bindings>
  </system.serviceModel>
</configuration>
  
```

Para la construcción del sistema, se crean los proyectos *Ping* y *Pong*, lo cuales deben incluir la referencia al framework y demás dependencias. La clase *RuntimeAgent*, hereda de la clase *Form* y está pensada para ser la interfaz principal de cada nodo que conforma la red de agentes, por lo tanto se reemplaza la clase *Form* por *RuntimeAgent* en las clases por defecto que se crean al iniciar un proyecto de tipo *Windows Form*. Se sobrescribe el método *OnSetup* para agregar los agentes *PingAgent* y *PongAgent* a sus respectivos contenedores. A continuación se ve el código necesario para esta fase para el proyecto *Ping*, el cual es similar a la definición utilizada para el proyecto *Pong*.

```

namespace Ping
{
    public partial class Form1 : RuntimeAgent
    {
        public Form1()
        {
            InitializeComponent();

            RichTextBox log = new RichTextBox();
            log.Dock = DockStyle.Fill;
            Controls.Add(log);

            Log.Out = new RichTextWrapper(log);
        }

        // Este método se dispara luego de inicializar canales
        protected override void OnSetup()
        {
            Agents.Add(new PingAgent());
        }
    }
}

```

La clase *PingAgent* se extiende de clase *Agent* y sobre el método *Setup* se programan los dos comportamiento propuestos. El código que se muestra a continuación presenta su implementación:

```

namespace Ping
{
    public class PingAgent:Agent
    {
        public override void Setup()
        {
            // Compartamiento de Envio de mensajes Ping
            Behaviours.Add(
                new CyclicBehaviour(
                    new Action(
                        delegate{
                            ACLMessage ping = new ACLMessage();
                            ping.ObjectContent = "ping";
                            ping.ConversationId = Guid.NewGuid();
                            //Configuramos su destino
                            ping.Receivers.Add(new AID()
                                { Name = "PongAgent" });
                            // Se envía
                            Send(ping);
                            // Espere 10 seg y repite
                            Behaviour.Wait(10000);
                        }));
            // Segundo compartamiento: Escucha de mensajes
            Behaviours.Add(
                new CyclicBehaviour(
                    new Action(
                        delegate {
                            ACLMessage pong = BlockingReceive();
                            Container.Log.Out.WriteLine("[Pong] "+pong);
                        }
                    )
                )
            );
        }
    }
}

```

```

        }));
    } // Fin del método Setup
} // Fin definición de clase PingAgent
} // Fin Namespace Ping

```

Por su la clase *PongAgent* se implementa de la siguiente forma:

```

namespace Pong
{
    public class PongAgent:Agent
    {
        public override void Setup()
        {
            // Adicionamos el compartamiento de Replica de mensajes
            Behaviours.Add(
                new CyclicBehaviour(// Ciclico
                    new Action(
                        delegate{
                            // Espera hasta recibir mensaje
                            ACLMessage ping = BlockingReceive();
                            // Crea un mensaje de respuesta
                            ACLMessage pong = ping.CreateReply();
                            pong.ObjectContent = "pong";
                            // Lo envia
                            Send(pong);
                        }
                    )
                )
            );
        }
    }
}

```

Una vez creados y configurados correctamente los proyecto Ping y Pong, al ejecutarlos se abren las interfaces correspondientes a los contenedores que, según lo programado, desplegarán la traza de los mensajes que reciben o envían a la red, ver figura 45.

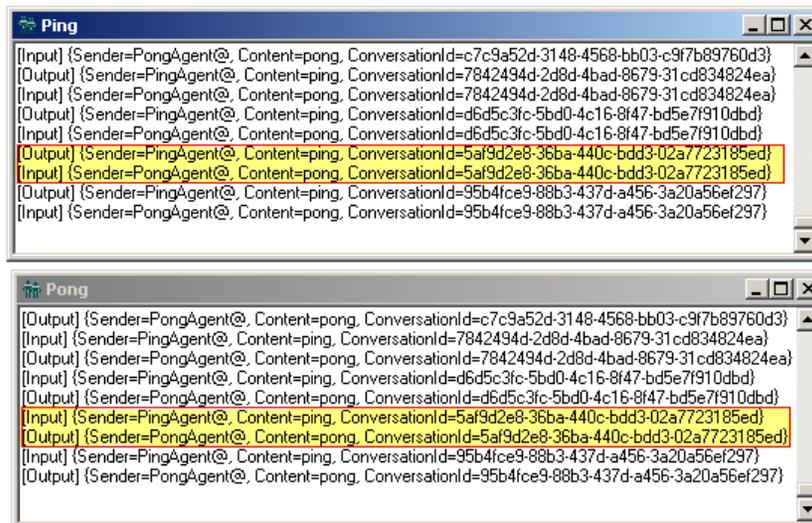


Fig. 45. Ejecución del Sistema Multi-Agente de ejemplo con los Agentes Ping y Pong

En la figura 45 se han resaltado los mensajes que se producen durante la invocación de un ciclo de envío y recepción: primero *PingAgent* envía su mensaje con *ConversationId* que se señala, el cual es recibido por *PongAgent* el cual lo replica y envía para finalmente llegar de nuevo al buffer de comunicaciones de *PingAgent*.

Con base en el procedimiento que se ha descrito en este apartado, a continuación se presenta el resumen de la implementación del Sistema Multi-Agente para la Gestión del *Awareness* el cual implementa la arquitectura propuesta en esta investigación.

### **4.2.3 Sistema Multi-Agente para la Gestión del Awareness**

Para la validación del Sistema se implementaron cinco tipos de agentes: *Persistencia*, *Guardián*, *Coordinador*, *SKA* e *Interfaz*. Ellos configuran una red cooperativa con la que se procesa y distribuyen distintos tipos de *Awareness* tales como: Social, Tarea, Espacio de Trabajo y Conocimiento Compartido. Para esta implementación solo el agente de Interfaz posee múltiples instancias, puesto que es el canal a través del cual se comunica toda la funcionalidad de la red hacia el usuario, mientras captura y difunde eventos relacionados con su interacción. Además, este agente se integra con otras herramientas de trabajo para ofrecer características de mediación e interoperabilidad de la información.

A continuación se presentan los detalles de implementación más relevantes de estos agentes y de la dinámica cooperativa que lo gobierna.

#### **4.2.3.1 Modelo de Información**

El *Modelo de Información* está definido por dos grandes estructuras: el *Modelo de Información de Awareness* y el *Modelo de Comunicación*. Mientras que la primera da forma y sentido a la información del sistema desde la perspectiva del *Awareness* según el modelo que plantea esta tesis, la segunda se utilizó para lograr un entendimiento de los agentes a nivel de comunicación definiendo la sintaxis de los paquetes de datos que se intercambian a través de clases y atributos.

*Modelo de Información de Awareness* fue implementado según la propuesta y para ello se utilizó la definición de esquemas XML soportados por los objetos *DataSet* del framework .NET. La manipulación de estos esquemas implementa la base de datos con la cual se almacena la memoria permanente de la red. Para lograr una gestión adecuada de este repositorio, se propuso un acceso centralizado gestionados por un *Agente de Persistencia* cuya responsabilidad es la de manipular el registro de datos, en función de las peticiones que generan otros agentes. Los detalles funcionales de la gestión que realiza el Agente de Persistencia al registro, se realiza más adelante durante su descripción.

El *Modelo de Comunicación* fue implementado a través de seis paquetes: *Coordination*, *Interaction*, *Knowledge*, *Security*, *Social* y *Tool*. Esta división obedece a una separación según el tipo de información de *Awareness* que manipulan, pero en la implementación, dicha separación no pudo darse de forma absoluta por cuanto existen conceptos de distintos enfoques que guardan una estrecha relación con otros por lo cual resultaba inapropiado forzar su aislamiento. Este fenómeno refuerza la idea de que el aprovisionamiento del *Awareness* en un sentido práctico, implica la combinación de distintas perspectivas y por tanto su desarrollo desde la visión integral, lo cual se puede lograr a través de una red cooperativa de servicios.

La principal característica de las clases que conforman el Modelo de Comunicación, es la capacidad que tienen sus instancias para poder ser serializadas como cadenas de bytes, lo cual permite su envío a través de redes de comunicación. A continuación se resume cada paquete del modelo y de las principales clases que lo conforman:

**Coordination:** agrupa clases relacionadas con el *Awareness* de Tarea entre las que se tienen: *Asignature*, relaciona perfil de usuario y rol; *ProfileTask*, conecta el perfil con las tareas asignadas según su rol; *Role*, descripción textual de los roles del sistema; *Task*, descripción textual de las tareas propuestas para el sistema.

**Interaction:** agrupar clases relacionadas con la interacción entre usuarios, por ejemplo la clase *Conversation* es la estructura que permite el envío de mensajes en un servicio de *Chat* implementados por el agente de Interfaz.

**Knowledge:** permiten gestionar el conocimiento de grupo desde la perspectiva de evaluación y experticia. La clase *Evaluation*, encapsula el concepto que un usuario tiene de los aportes de otro mediante la asignación de una nota cualitativa y comentario en el contexto de una tarea. Este mensaje es interpretado por el agente SKA el cual aplica lo utiliza como evidencia del conocimiento que el usuario evaluado posee desde la perspectiva del evaluador. Este razonamiento construye una tabla la tabla de experticia del sistema en la cual cada registro es mapeado a instancias de la clase *Expertise*.

**Security:** su propósito es agrupar clases relacionadas con aspectos de seguridad como por ejemplo el acceso a la red, por lo cual está definida la clase *Login*: mensaje utilizado por el Agente de interfaz para iniciar sesión en el sistema.

**Social:** define las clases relacionadas con el aspecto social del *Awareness*. En esencia propone estructuras para el perfil de usuario (*Profile*) y las comunidades (*Comunity*), además de otras estructuras que permiten consultar las comunidades a las cuales pertenece un usuario o una copia de su perfil.

Finalmente se encuentra el paquete **Tool**, cuyo propósito es definir las estructuras de comunicación entre el agente de Interfaz y otras herramientas que hagan parte del entorno de trabajo del usuario. En este paquete se encuentran las clases *ToolDescriptor* que contiene la descripción de una herramienta con lo cual se puede dar contexto a sus eventos y *ToolObjet* un contenedor genérico para objetos propios de la herramienta los cuales constituyen el formato de los aportes de un usuario a la solución de las tareas.

Con base en los paquetes descritos y el uso de los atributos *Ontology* y *Protocol* de la clase *ACLMessage*, los agentes logran un entendimiento y cooperación con los demás agente. A continuación se realiza la descripción de los cinco tipos de agentes antes mencionados.

#### 4.2.3.2 Agente de Persistencia

La responsabilidad de este agente es ofrecer al Sistema Multi-Agente servicios relacionados con la gestión de registro permanente de datos basado en el Modelo de Información. Su implementa se basa en el desarrollo de un único comportamiento en el cual escucha las peticiones de otro agentes y las transforma en operaciones de consulta o modificación de los datos.

Como se muestra en la figura 46, el agente de Persistencia despliega además un conjunto de interfaces gráficas que permiten a un usuario administrador acceder directamente sobre los registros del modelo de información y configurar por ejemplo los perfiles, comunidades, roles, tareas y conocimiento que modela la dinámica del grupo.

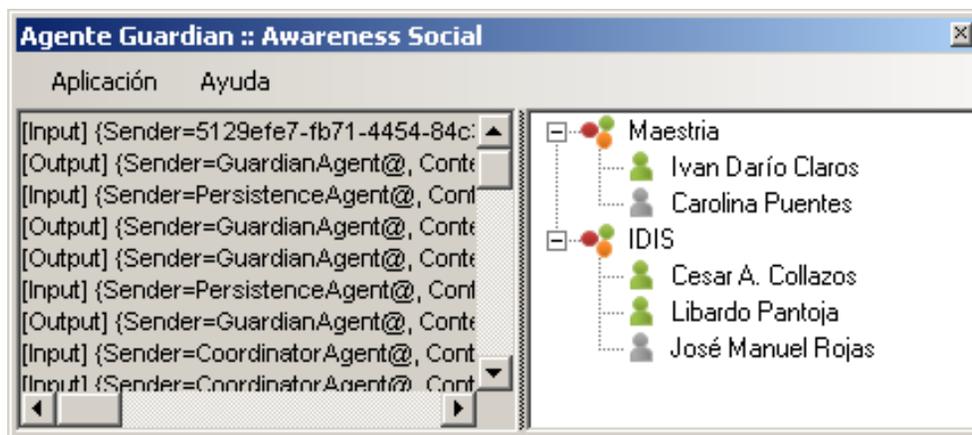
Nombre Completo	Correo	Contraseña	Avatar	Estado	Role	ThinkingAbout	Último Acceso
Ivan Darío Claros	iclaros@unicauca.edu.co	iclaros		Offline	Redactor	Terminando desarrollo para empezar con la ...	03/11/2009 09:0...
Carolina Puentes	cpuentes@unicauca.edu.co	cpuentes		Online	Editor	Preparandome para defender mi tesis :)	03/11/2009 09:2...
Cesar A. Collazos	ccollazo@unicauca.edu.co	ccollazo		Online	Coordinador		03/11/2009 09:4...
Libardo Pantoja	wpantoja@unicauca.edu.co	wpantoja		Offline	Coordinador	Preparando el anteproyecto	03/11/2009 09:5...
José Manuel Rojas	jrojas@unicauca.edu.co	jrojas		Offline	Coordinador	Alistando maleta para España	03/11/2009 09:5...
Innes Trujillo	itrujillo@unicauca.edu.co	itrujillo		Online	Editor	Y la verdad hoz hará libres...	03/11/2009 09:5...

**Fig. 46. Interfaz grafica del Agente de Persistencia**

Dentro de la dinámica implementada por el Agente de Persistencia se requirió que a nivel de comunicación se manipularán objetos del paquete *Security, Coordination, Social y Knowledge*. En aspectos de seguridad, este agente permite el procesamiento de todas las peticiones soportadas siempre y cuando no provengan directamente del Agente de Interfaz, es decir que la manipulación de los registros

#### 4.2.3.3 Agente Guardián

Implementa las restricciones de acceso a la red para los Agentes de Interfaz y provee y soporta el *Awareness Social* de la red. La responsabilidad de soportar este tipo de *Awareness* está justificada por ser este el punto de acceso a la Red, por lo cual conoce de primera mano los eventos de acceso y abandono de la red por parte de un usuario. En la figura 47 se puede ver la interfaz desarrollada para este agente. En la parte izquierda de esta figura se puede ver el registro de mensajes que llegan o salen del agente, mientras que en lado derecho una estructura en árbol que las comunidades y perfiles de los usuarios que han sido registrados dentro del modelo de información.



**Fig. 47. Interfaz Gráfica de Agente Guardián**

En el momento que el Agente Guardián es notificado con éxito de la validación del acceso de un usuario o la finalización de su sesión de trabajo, éste, basado en el registro de las comunidades a las cuales pertenece, genera un mensaje de *Awareness Social* con el cual notifica a sus compañeros, el nuevo estado de dicho perfil. Con esta dinámica y sumado a la información del perfil, los usuarios pueden saber que usuarios pertenecen a su comunidad y cuál es su estado dentro del sistema.

#### 4.2.3.4 Agente Coordinador

El Agente Coordinador gestiona la dinámica del grupo en términos de dos conceptos Roles y Tareas. Estos conceptos han sido identificados dentro del diseño del Modelo de Información como parte esencial del aprovisionamiento de *Awareness* de Tarea.

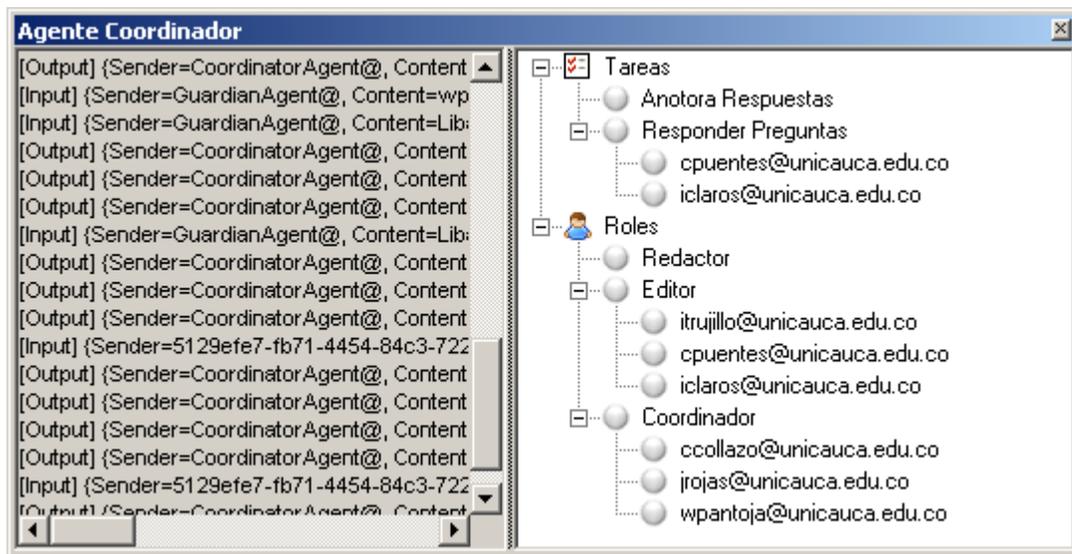


Fig. 48. Interfaz grafica del Agente Coordinador

La figura 48 presenta la interfaz diseñada para el agente Coordinador, en la parte izquierda se puede ver el registro de mensajes que entran y salen del Agente, mientras que en la parte derecha se despliega en forma de árbol la estructura de roles y tareas modeladas para el sistema. Sobre cada rama de un rol o tarea, cuelgan la identificación de los usuarios que según el registro asumen o asumieron el rol o tarea específico. Cada vez que un usuario asume un rol o tarea, produce un mensaje hacia el Agente Coordinador el cual lo traduce en un mensaje de *Awareness* que notica a otros usuarios bien sea con los que compartía o en adelante compartirá un rol o tarea. Esta lógica crea una oportunidad de interacción entre usuarios basados en la conciencia de qué rol esta o ha asumido un usuario, cuál es la tarea puntual que un usuario esta realizado.

#### 4.2.3.5 Agente de Conocimiento Compartido SKA

El Agente de Conocimiento Compartido o SKA, gestiona la lógica del modelo de experticia propuesto, basado en el procesamiento de mensajes de evaluación que emite un usuario sobre los aportes que ha realizado otro usuario. Este Agente no solo emite mensajes de *Awareness* sobre el nivel de experticia que un usuario va adquiriendo durante la dinámica del sistema, sino que además despliega el servicio de búsqueda de expertos, basado en identificación de perfiles con un nivel mínimo requerido en un concepto particular.



Fig. 49. Interfaz gráfica del Agente SKA

La figura 49 muestra la interfaz gráfica desarrollada para el Agente SKA, en la parte izquierda el registro de mensajes que entran y salen del agente y en la parte derecha la tabla de registros de experticia el agente ha ido registrando.

#### 4.2.3.6 Agente de Interfaz

La implementación de este agente involucró el desarrollo de distintos *Widgets* para el despliegue de la información de *Awareness*, entre los que se tienen las listas de usuarios, la vista detalle de un perfil y mecanismos de interacción basado en mensajería instantánea.

La funcionalidad de este agente permite al usuario no solo acceder a los servicios propuestos para la red sino que además es una entidad de mediación con otras aplicaciones que corren sobre la estación del cliente para generar y consumir información de *Awareness*.

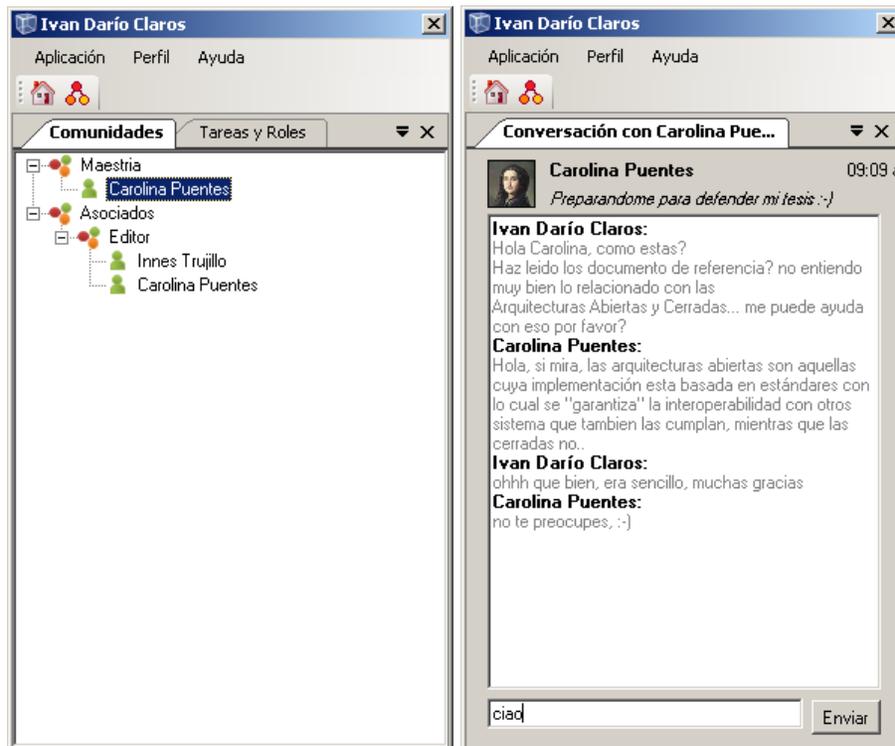


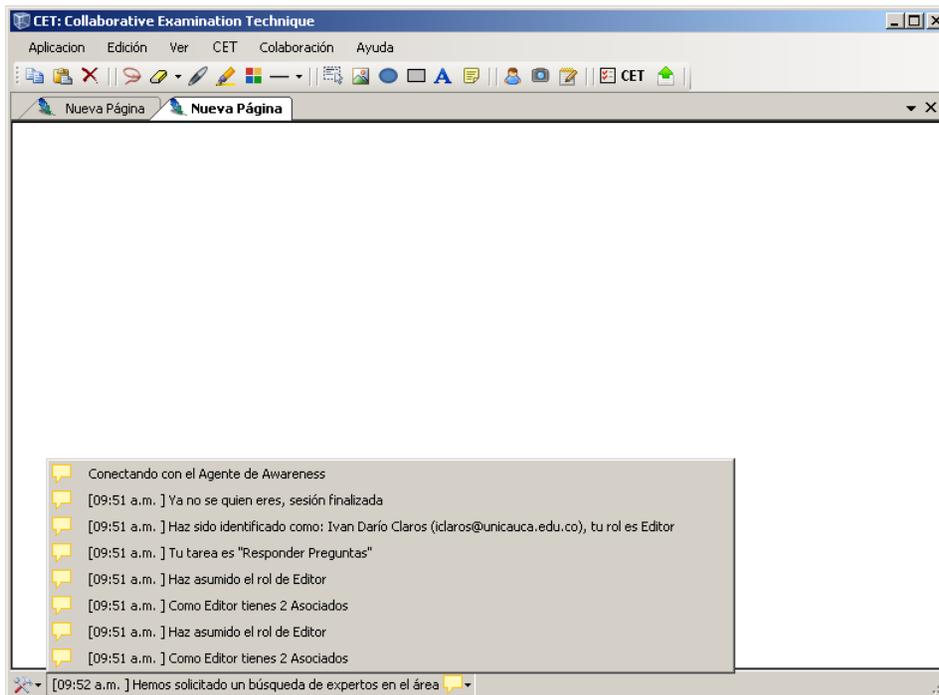
Fig. 50. Interfaces gráficas del Agente de Interfaz

En la figura 50 se presentan dos de las interfaces que despliega el agente de Interfaz, en la primera (lado izquierdo) se muestra la pestaña de comunidades, en ella se puede ver la lista de usuarios con los cuales un usuario puede interactuar, bien sea porque comparte un comunidad o porque la red se lo propone al compartir un rol, por ejemplo el rol Editor, como lo presenta esa figura. La segunda (lado derecho), presenta un servicio de interacción incluido dentro del Agente que permite establecer un dialogo entre dos usuarios.

#### 4.2.4 Integración del Libro digital de trabajo con el Sistema Multi-Agente

Para establecer una forma de interacción entre la red de agentes, particularmente con el Agente de Interfaz, y un conjunto de aplicaciones utilizadas como soporte a la colaboración, dentro del Agente de Interfaz se implementó un canal de comunicación interno al cual una aplicación pudiera conectarse y acceder a los servicios del agente. Para la aplicación Libro Digital, la conexión con el agente se traduce en un barra de mensajes en la cual se van reportando algunos mensajes que pueden resultar relevantes para las acciones que realiza el usuario, por ejemplo cuando alguien se ha conectado o cuando alguien ha asumido el mismo rol. Además, ofrece acceso a servicios tales como la búsqueda de expertos y la posibilidad de evaluar los aportes que un usuario ha realizado en el contexto de una tarea.

La figura 51 presenta un vistazo al mecanismo de interacción entre el Agente de Interfaz y el Libro Digital de trabajo. En esta figura se despliega la lista de los últimos 10 mensajes recibidos del agente y la lista de herramientas que permiten acceder a los otros servicios mencionados.



**Fig. 51. Barra de Mensajes de Awareness, mecanismo de interacción entre el Agente de Intrefaz y el Libro Digital de Trabajo**

## Capítulo 5. Experimentación

Este capítulo describe las diferentes pruebas realizadas en la fase de verificación de la propuesta arquitectónica definida en ese trabajo. En coherencia con los prototipos desarrollados, este capítulo se divide en dos secciones: la primera describe la experimentación con el módulo de *Awareness* para KnowCat y la segunda, con el Libro Digital de Trabajo y la Red Multi-Agente. Cada una de estas experiencias es descrita en términos de la metodología utilizada y los resultados obtenidos.

### 5.1 Caso de Estudio I, Módulo de *Awareness* para KnowCat

Este experimento permitió verificar no solo el comportamiento funcional del prototipo desarrollado, sino que además ofreció indicios sobre el efecto de los servicios de información de *Awareness* frente a los procesos de GC, en este caso, soportados a través del sistema KnowCat.

La experimentación involucró el uso de dos versiones del sistema, en la primera se habilitó el uso del módulo de *Awareness* (en adelante CA o Con *Awareness*), mientras que la segunda versión no dispuso de dicho módulo (en adelante SA o Sin *Awareness*). A continuación se describe la metodología con la cual se desarrolló la prueba.

#### 5.1.1 Metodología

En la prueba participaron 20 estudiantes del curso electivo Trabajo Colaborativo Asistido por Computadora del programa de formación en Ingeniería de Sistemas de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, segundo periodo de 2008. Se formaron 4 grupos: 2 que trabajaron sobre la versión CA y los otros con SA, estos últimos sirvieron como grupos de control. Se organizaron 3 tópicos dentro del árbol de conocimiento: *Awareness en la CSCW (Trabajo Colaborativo Asistido por Computadora)*, *Problemas del Awareness* y *Awareness en CSCL (Aprendizaje Colaborativo Asistido por Computadora)*; a cada grupo le fue asignado un subtema para su trabajo.

La dinámica propuesta consistió en los siguientes pasos:

1. Una vez configurados los nodos CA y SA, se creó el árbol de conocimiento con los tres tópicos mencionados.
2. Sobre la raíz del árbol de conocimiento se publicaron artículos de referencia que sirvieron como base de información inicial para hacer un análisis documental del tema. Dicho listado no presentó ningún tipo de clasificación previa con el fin de motivar a los participantes a establecer estrategias de grupo para asimilar un gran volumen de información de forma eficiente.
3. Se realizó una instrucción inicial a los participantes sobre el uso de KnowCat y sus funciones generales, posteriormente se amplió dicha explicación hacia el módulo de *Awareness* implementado, solo a los dos grupos que trabajarían sobre la versión CA.
4. Posterior a dicha reunión se envió por correo electrónico la distribución exacta de los grupos junto con el subtema específico a tratar. Además, se solicitó que internamente se seleccionara un miembro como coordinador, el cual debía asumir la responsabilidad de velar por el cumplimiento de los compromisos y tiempos proyectados para la experimentación.

5. Se propuso realizar una indagación documental sobre el tema asignado y consignarla en un informe individual que debía ser puesto a disposición de los demás miembros del grupo a través de la plataforma.
6. Por grupo, se debía realizar anotaciones o comentarios a cada documento publicado con el propósito de mejorar la calidad del mismo.
7. Adicionalmente, cada individuo tenía la posibilidad de dar su voto a los dos documentos que considerará como los de mejor calidad.
8. A partir de las anotaciones realizadas a sus informes, los usuarios debían construir una nueva versión y nuevamente hacerlo disponible sobre la plataforma.

El tiempo total para la realización de las tareas propuestas fue de dos semanas, en las cuales, la primera se dedicó a la indagación documental y la generación de la primera versión del informe. A mediados de la siguiente semana, los participantes ya debían haber anotado y dado sus votos a los documentos, disponiendo hasta el final de la semana para generar y hacer disponible la segunda versión del informe.

Para estimar el impacto de los servicios de *Awareness* sobre la plataforma se realizó seguimiento continuo a las actividades realizadas a los usuarios y finalmente se aplicó una encuesta como instrumento clave para capturar la valoración de los usuarios a los servicios desarrollados. Las preguntas propuestas, en su mayoría, incluían una valoración cuantificada y un espacio para complementar o justificar el valor dado a cada pregunta. En otros casos solo se preguntaba por la opinión del encuestado sobre temas específicos. Las preguntas fueron enfocadas para cubrir 4 aspectos: la herramienta, el trabajo individual, el trabajo grupal y los servicios de *Awareness*. Los resultados obtenidos de la aplicación de este instrumento fueron contrastados con el grupo de control.

### **5.1.2 Resultados obtenidos**

El primer bloque de preguntas de la encuesta tenía el propósito de medir el nivel de satisfacción del usuario con respecto a KnowCat en sus versiones CA y SA, como herramienta para el soporte a la colaboración. Como resultado directo de este bloque se identificaron y contrastaron tanto los aspectos positivos como negativos de cada versión, demostrando una mayor satisfacción por parte de los usuarios de la versión CA que SA, en particular por la inclusión del servicio de *Awareness* Participómetro. Por otra parte, los usuarios CA sugerían ampliar las capacidades de servicios de *Awareness* con mecanismos de comunicación sincrónica como un chat, mientras que los usuarios SA centraron sus opiniones en mejoras a la interfaz gráfica; este fenómeno es asociado con un creciente interés por parte de los usuarios CA en fortalecer los lazos de comunicación mientras que para los usuarios SA esta necesidad no fue evidente.

El segundo bloque de la encuesta se centró en medir el desempeño individual del usuario a través de preguntas relacionadas con los tiempos de interacción con la herramienta, nivel de satisfacción con respecto al aprendizaje y las estrategias utilizadas para la elección de los documentos a leer y sus aportes en forma de anotaciones. Para este caso se observó que:

- Los usuarios CA interactuaron durante más tiempo con la herramienta que usuarios SA;
- Solo en los usuarios CA hubo al menos dos personas que utilizaron una estrategia de grupo para seleccionar los documentos a leer y hacer sus anotaciones y

- El nivel de satisfacción con respecto al documento generado, conocimiento adquirido y trabajo en grupo fue más alto para los usuarios CA, pueden verse los resultados en la tabla 40.

**Tabla 40. Nivel de satisfacción de los usuarios con respecto al aprendizaje (escala de 0 a 5)**

Documento Generado		Conocimiento Adquirido		Trabajo en Grupo	
CA	SA	CA	SA	CA	SA
4,09	3,83	4,33	4,17	3,94	3,67

El tercer bloque de preguntas fue centrado en el trabajo en grupo, formulando preguntas relacionadas con la frecuencia de interacción entre usuarios, el establecimiento de una estrategia de grupo, la actividad como un proceso colaborativo y el aporte de trabajar en grupo respecto al trabajo individual. Los resultados obtenidos muestran que:

- La interacción directa con otros usuarios se mantuvo baja para ambas versiones, sin embargo los usuarios CA se sintieron satisfechos de poder visualizar la información de otros individuos a través de los servicios *Awareness*. La escasa interacción directa se asocia con la falta de experiencia en prácticas colaborativas, reflejando la necesidad de una apropiación mayor de estas técnicas en las aulas de clase.
- Para la mayoría de los usuarios SA, el sistema no brindó herramientas de soporte adecuados para definir una estrategia de grupo, mientras que para los usuarios CA este concepto fue aceptable. Solo un grupo de usuarios SA aseguraron haber establecido una estrategia de grupo en cuanto a la recopilación y lectura de documentos, con lo cual el establecimiento del rol de monitor cobro relevancia. En los demás casos el monitor estableció una posición pasiva y de baja interacción con sus compañeros.
- Al sugerir que la practica realizada había sido un proceso colaborativo, los usuarios CA tuvieron una mayor aceptación que para los usuarios SA. Por lo cual, al incorporar los servicios *Awareness* al sistema, se logró un entorno de colaboración más adecuado, según los encuestados.

El último bloque de preguntas formuladas se relacionó con la información de *Awareness* necesaria o expuesta y como fue utilizada. En este caso, los usuarios SA evidenciaron la necesidad de soportar información de *Awareness* relacionada con los usuarios en línea encontrando potenciales beneficios; mientras tanto, los usuarios CA prefirieron la información histórica brindada por los servicios de *Awareness*.

Al evaluar directamente los servicios de *Awareness* ofrecidos al grupo CA, se observa que el servicio de Participómetro fue el mejor valorado y de uso más frecuente, seguido de Usuarios Registrados y Gráfico de Anotaciones. Los servicios Usuarios en Línea y Vista Histórica aunque valorados como de buena presentación, usabilidad y utilidad, no tuvieron una frecuencia de uso alta, mientras que la Vista en Radar se ubicó en un nivel medio.

Se puede suponer que el comportamiento en el uso de los servicios refleja dos situaciones: 1) aunque la formación de los grupos fue aleatoria, estos venían de un grupo general conocido, lo cual pudo reducir el interés por visitar las fuentes de información personal; y 2), el alto uso de los servicios Participómetro y Gráfico de Anotaciones, refleja

una tendencia del grupo hacia una dinámica competitiva, por ser esta la que por tradición se ejecuta en las actividades de clase. El Participómetro motivó a los usuarios a mantener control de su desempeño, pero en el caso de los usuarios SA, la mayor dificultad por establecer un punto de comparación a su labor, redujo el interés y la frecuencia de hacerlo.

## 5.2 Caso de Estudio II, Libro Digital y Red Multi-Agente

Para este segundo caso de estudio participaron 21 estudiantes de la materia electiva *Interacción Humano Computador* del programa de Ingeniería de Sistemas de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, segundo periodo de 2009. Los participantes se dividieron en equipos de tres personas, conformados de manera aleatoria. Estos grupos se separaron en dos categorías: 1) *Usuarios Tradicionales (UT)*, estudiantes que participaron del caso sin el soporte de herramientas informáticas especializadas; y 2) *Usuarios con Libro de Digital (ULD)*, los cuales contaron con acceso al sistema software desarrollado. Por razones de disponibilidad en el número de TabletPCs, se trabajó con dos grupos ULD y cinco UT.

A nivel metodológico, esta experimentación fue estructurada siguiendo las recomendaciones descritas en la Técnica de Evaluación Colaborativa, CET (Collaborative Evaluation Technique) [64]. Para ello se seleccionó un tema central de evaluación el cual fue el de **Interfaces de Usuario en entornos Colaborativos** segmentado en tres ejes: **Mecanismos de Awareness, Usabilidad en entorno Colaborativos e Interfaces Tangibles**. Para cada eje se formularon un conjunto de preguntas y postulados que buscaban orientar al estudiante sobre una exploración documental. Los ejes y las preguntas formuladas fueron las siguientes:

- Mecanismos de Awareness
  - ¿Cuáles mecanismos de *Awareness* conoce? Descríbalos brevemente.
  - Describa distintas formas de *Awareness* y proponga una representación.
- Usabilidad
  - ¿Cuáles son los problemas más frecuentes de usabilidad en los sistemas colaborativos?
  - ¿Cuál es el impacto positivo y negativo del *Awareness* sobre la usabilidad de los sistemas colaborativos?
  - Describa algunas consideraciones a la hora de medir la usabilidad de un sistema colaborativo.
- Interfaces Tangibles
  - ¿Cómo las interfaces tangibles afectan la usabilidad de los sistemas?
  - ¿Qué formas de representación tangible encuentra para el *Awareness* Emocional?
  - Proponga formas de representación tangibles para el *Awareness*.

Con estos temas se ofreció al grupo una perspectiva general de distintos elementos relacionados con la interacción de usuario cuando se trabaja en entornos colaborativos, con una doble intención: primero, la de extender y afianzar los conocimientos relacionados con los entornos colaborativos y el HCI (núcleo temático central del curso) y segundo, dotar al estudiante de criterios para evaluar u opinar del desempeño de las herramientas desarrolladas.

A continuación se presenta una descripción más detallada de la metodología con la cual se desarrollo la experiencia.

### 5.2.1 Metodología

La experimentación dio inicio con una sesión de dos horas donde fueron expuestos los conceptos más relevantes relacionados con el Trabajo Colaborativo y Aprendizaje Colaborativo, con particular énfasis en la Conciencia de Grupo o *Awareness*, dada su importancia para la comprensión de los temas de trabajo. Esta charla, le ofreció al estudiante una base conceptual que le permitió iniciar una fase documental sobre los tres ejes propuestos. Finalizada la sesión, se suministró material de referencia bibliográfico para responder a las preguntas de orientación formuladas para cada eje, pero dado que se contaba con poco tiempo y gran número de material, cada individuo debió responsabilizarse de estudiar y preparar el resumen para un tema específico y hacerlo disponible la siguiente sesión.

El propósito de CET es estructurar en forma colaborativa el proceso de evaluación para convertirlo en un mecanismo de aprendizaje, y para ello, propone dividir dicho proceso en 3 etapas o fases consecutivas: *pre-test*, *test* y *post-test*.

El *pre-test* posee una estructura similar a las actividades llevadas a cabo en la técnica *Jigsaw* [74], buscando disminuir la ansiedad de los estudiantes observando un cuestionario similar al del examen, teniendo la posibilidad de construir y validar las respuestas en forma grupal. Por su parte, el *test* consiste en presentar el examen de forma individual siguiendo los procedimientos tradicionales. Finalmente, el *post-test* tiene como objetivo que los estudiantes aprendan de sus errores y puedan establecer guías para el proceso de corrección; para ello reciben una copia del examen realizado y por grupos intentan resolver el examen, construyendo así lo que ellos consideran que sería la pauta de solución.

Al concluir el *post-test*, los estudiantes se autoevalúan asignándole un puntaje a su examen en función de las respuestas que ellos consideran correctas. La autoevaluación debe argumentar el puntaje que el alumno le asignó a cada pregunta. Aquellos estudiantes cuya autoevaluación está muy cerca de la valoración real, tienen un bono adicional el cual retribuye el hecho de que el estudiante para llegar a autoevaluarse coherentemente haya encontrado las respuestas correctas a las preguntas, aunque esto suceda por fuera del examen.

A continuación se describen las distintas fases y actividades que formaron parte de la experiencia siguiendo los lineamientos de CET.

#### 5.2.1.1 Primera fase: Pre-Test

**Configuración del entorno tecnológico:** dado que para esta experiencia solo se contó con la disponibilidad de 3 estaciones de cómputo tipo *TabletPC*, la sesión de trabajo para los dos grupos ULD fue segmentada, llegando a durar un poco más de dos horas, como consecuencia de la introducción de pausas entre las distintas actividades para permitir el intercambio de los computadores con el segundo grupo. Una de las estaciones fue seleccionada como nodo de gestión, ejecutando allí tanto *CETProfesor* y los Agentes de Servicio como una instancia del Libro Digital y el Agente de Interfaz; mientras que sobre las dos estaciones restantes, solo se hicieron disponibles instancias de estas dos últimas aplicaciones.

En la actividad denominada “Reunión de expertos”, mientras dos estaciones eran utilizadas por los alumnos, el instructor accedía al nodo para hacer supervisión de la actividad a través de los Servicios *Awareness* implementados tanto por *CETProfesor* como por los distintos Agentes de la red.

**Definición de vocero (5 min):** luego de reunir a cada alumno con sus compañeros de equipo, se le pidió que seleccionen un vocero o coordinador, el cual, por su madurez y liderazgo, tenía la responsabilidad de mediar en las discusiones para evitar posiciones dominantes y procurar el desarrollo de la actividad según lo tiempos y metas propuestas.

**Resolución de preguntas de forma individual (15 min):** con base en su investigación, cada estudiante respondió a una pregunta asignada a su eje. Los grupos UT consignaron sus respuestas en hojas marcas con sus nombre e indicativos de grupo, mientras que los grupos ULD utilizaron el Libro Digital para conectarse a *CETProfesor* y descargar de ahí las preguntas formuladas.

**Reunión de expertos (15 min):** reunidos por pares, estudiantes de distintos grupos que compartían una pregunta se reunieron para discutir sus respectivas respuestas. Los cambios o aporte respecto a sus respuestas fueron guardados como una nueva versión de respuesta. La figura 52, presenta una fotografía tomada durante la experiencia mientras se realizaba la reunión de pares (expertos) tanto para ULD y UT.



**Fig. 52. Fotografía tomada durante la reunión de experto (a la izquierda ULD, derecha UT)**

**Reunión de grupo – Resolución grupal (15 min):** una vez finalizada la reunión de expertos, se reunieron los grupos y discutieron cada respuesta que les fue asignada. Nuevamente, cualquier cambio fue guardado como otra versión de la respuesta. El total de documentos generados fue entregado al vocero el cual a su vez los reenvió al instructor; en el caso de los grupos ULD esto no fue necesario dado que estos eran sincronizados directamente con la herramienta *CETProfesor*. La figura 53 presenta una fotografía tomada durante la discusión de grupo.



**Fig. 53. Reunión de los grupos de trabajo mientras discutían sus respuestas**

#### **5.2.1.2 Segunda fase: Test**

Esta fase consistió en la aplicación de un examen escrito en el cual se incluyó algunas de las preguntas formuladas en el pre-test, como compensación al trabajo previamente realizado, según las recomendaciones de CET. De forma adicional a la técnica, también fueron solicitados los resúmenes de investigación, lo cuales fueron evaluados en su calidad e incluidos como criterio para la valoración del desempeño de los estudiantes. El desarrollo de este fase no conto con el uso de software. La figura 54 presenta una fotografía de los estudiantes durante la fase de test.



**Fig. 54. Fotografía tomada durante la aplicación del examen**

#### **5.2.1.3 Tercera fase: Post-test**

Teniendo ya una valoración previa de los exámenes presentados, el post-test inició con una charla que resumía las principales conclusiones del tema y procuraba pautas para las respuestas del examen pero sin formular una respuesta definitiva al mismo. Luego de esta charla, se regreso a los estudiantes sus exámenes y se les pidió que formaran sus grupos

de trabajo y debatieran nuevamente sobre sus respuestas. En la hoja del examen podían incluir notas aclaratorias o correctivas según fuera el caso. Al finalizar la actividad, cada estudiante debió asignar una auto-valoración a su examen, junto con sus notas aclaratorias. La coherencia entre las aclaraciones y la valoración asignada, fue utilizada para cuantificar los posibles bonos post-test.

En esta última sesión se abrió una discusión sobre la experiencia y los temas tratados, la cual fue orientada a través de entrevistas tanto a ULD como UT. Adicionalmente a CET, se asignó un último trabajo de grupo el cual consistió en el diseño de un mecanismo de *Awareness* basado en interfaces tangibles, al cual debían medir su usabilidad y presentarlo a través de un informe escrito. Este último requerimiento de clase, tuvo como propósito afianzar los conocimientos adquiridos y obtener una base de comparación del desempeño del grupo, fuera de la experiencia.

### 5.2.2 Resultados obtenidos

La tabla 41 presenta el resumen de valoraciones cuantitativas obtenidas del desempeño de los participantes para cada una de las fases. Los valores dados en esta tabla esta en escala de 0 a 5 y los grupos señalados como G y H corresponden a ULD. La columna marcada como post-test corresponde al valor de bono promedio alcanzado por los estudiantes en sus respectivos grupos.

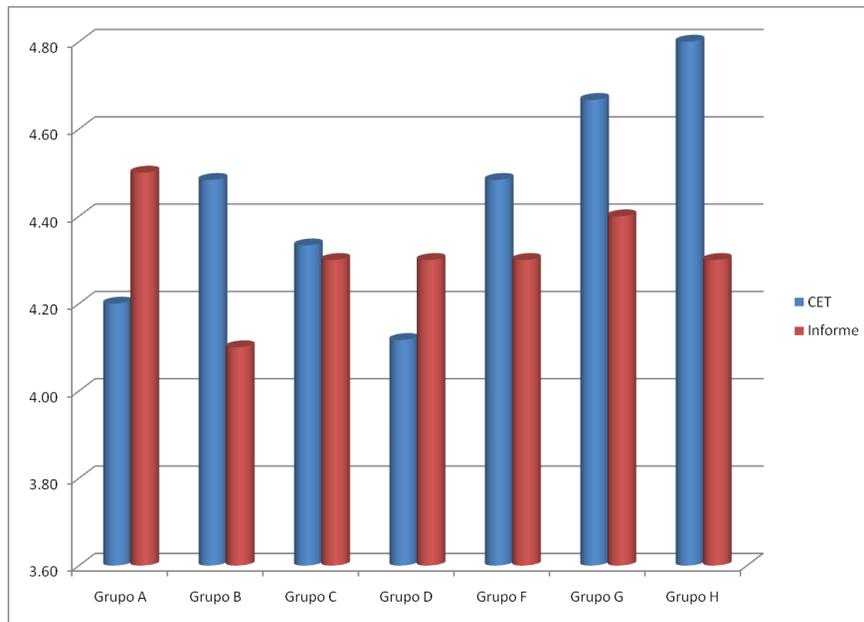
**Tabla 41. Promedios de valoración obtenidos**

	pre-test	test	post-test	CET	Informe	Final
<b>Grupo A</b>	3.50	4.10	0.10	4.20	4.5	4.35
<b>Grupo B</b>	4.27	4.43	0.05	4.48	4.1	4.29
<b>Grupo C</b>	4.50	4.23	0.10	4.33	4.3	4.32
<b>Grupo D</b>	3.50	4.07	0.05	4.12	4.3	4.21
<b>Grupo F</b>	4.33	4.33	0.15	4.48	4.3	4.39
<b>Grupo G</b>	4.50	4.57	0.10	4.67	4.4	4.53
<b>Grupo H</b>	4.40	4.70	0.10	4.80	4.3	4.55

Como primer resultado de esta experiencia se puede ver que los grupos ULD obtuvieron un mejor desempeño cuantitativo en las fases de CET que los UT. De hecho, el valor promedio de ULD es de 4,73 mientras que para UT fue de 4,32. En un análisis más detallado del comportamiento del desempeño de ULD se muestran valores de desviación bajos respecto al promedio del grupo, lo cual sugiere una distribución de conocimiento homogéneo y con ello una gestión de conocimiento adecuada, lo cual es favorable al aprendizaje.

En cuanto a CET, los usuarios ULD presentaron incremento en sus calificaciones de la fase pre-test al test, sin embargo, para los grupos C y D sus calificaciones decayeron, lo cual se interpreta como una aplicación inadecuada de la técnica por parte de estos grupos. Pero, al no contar con mecanismos de supervisión, no es posible ofrecer una explicación contundente, confirmando la necesidad de disponer de sistemas de registro y control de las actividades colaborativas. Cuando se preguntó a los involucrados sobre una posible justificación a este fenómeno, los estudiantes apelaron a tiempos de explicación muy cortos por parte de sus compañeros y fallas de coordinación internas.

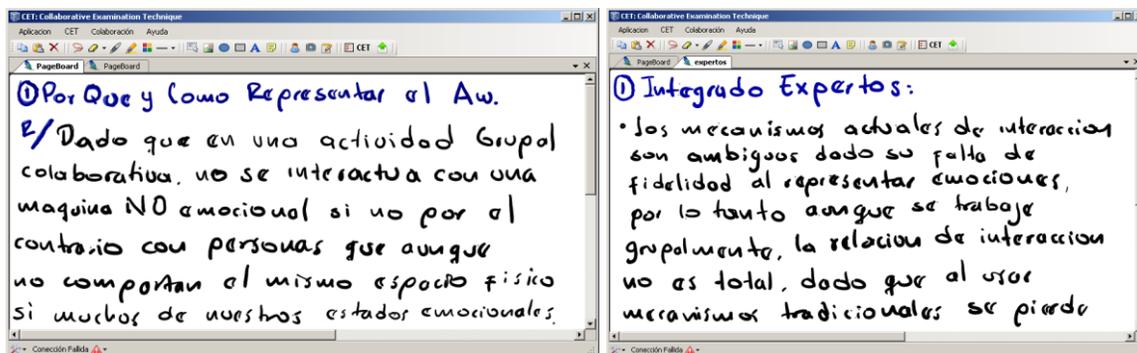
La figura 55 presenta un gráfico con los valores promedio tanto de las notas asignadas a CET, como a los informes finales. Durante CET, los grupos G y H presentaron valores superiores al resto de los grupos UT; sin embargo, cuando estos mismos trabajaron sin estructuración ni soporte de herramientas, su desempeño no fue superior. Este fenómeno refleja la importancia de una planeación adecuada de la actividad colaborativa y el uso de sistemas de soporte como elemento clave en el éxito de los procesos de aprendizaje.



**Fig. 55. Comparación entre los resultados de CET y el informe final**

Durante las entrevistas a ULD, se preguntó acerca de la usabilidad de la herramienta construida, su funcionalidad y el papel de los mecanismos de *Awareness* para el desarrollo de las actividades. Las observaciones más significativas fueron:

- La herramienta presenta un nivel de usabilidad adecuado, destacando la naturalidad del proceso de interacción que ofrece la tecnología TabletPC. Sin embargo, la velocidad con la cual manipulan el lápiz de la TabletPC no es óptimo, incurriendo en errores frecuentes donde se perdía continuidad de los trazos, llevándolos a sobrescribir palabras o hacer uso del borrador.
- La experiencia de una escritura prolongada sobre la pantalla de la TabletPC no fue muy satisfactoria. Se sugirió modelos de pantallas sin bordes u otro tipo de diseños que permitieran ofrecer un descanso al antebrazo y la mano.
- Una conducta reiterativa por parte de los usuarios fue el escaso uso de elementos gráficos para la expresión de sus ideas, de tal forma que se obtuvo respuestas con mucho texto. El texto rápidamente lleno las pantallas y llevo al uso de las barras de desplazamiento, las cuales incomodaron la interacción. Un ejemplo de este tipo de respuesta puede ser visto en la figura 56.



**Fig. 56. pantalla de respuestas típicas para los Usuarios de Libro Digital**

- En cuanto a la funcionalidad a nivel gráfico y de comunicación presentado en el Libro Digital, los niveles de satisfacción fueron buenos. Llegando a sugerir la inclusión de nuevos componentes o plantillas que permitieran agilizar los procesos de escritura, sobre todo.
- Cuando se iniciaron sesiones de trabajos concurrentes sobre preguntas por parte de dos o más usuarios, se presentaron algunos inconvenientes asociados con la latencia en la sincronización de los contenidos, lo cual llevó ocasionalmente a una pérdida parcial de trazos, resultando incómodo para los participantes. Sin embargo, una vez alcanzado un nivel de coordinación adecuado este problema era mitigado. No obstante, queda a revisión de futuras versiones del prototipo, diseñar e implementar nuevos esquemas de sincronización que resulten más confiables.
- Las anotaciones fueron un mecanismo de interacción muy eficiente a la hora de señalar las inquietudes de un usuario sobre los portes de otros, sin embargo, luego de ser atendidas debían ser eliminadas para liberar el espacio en pantalla lo cual hacía perder una posible línea de discusión documentada. Se sugirió algún tipo de control automático de versiones o comportamientos dinámicos por parte de estos componentes que permitieran minimizar su tamaño mientras no fueran el foco de atención del usuario.
- Las pausas e interrupciones producidas en la actividad por la rotación de las TabletPCs entre los grupos, resultó incómodo para algunos participantes debido a que se perdía la atención de la tarea, lo cual no favorece el aprendizaje. En futuras experiencias se espera contar con un número mayor de usuarios y dispositivos.
- En cuando al aprendizaje alcanzado, los estudiantes lograron un nivel de asimilación adecuado de los temas propuestos, lo cual es evidencia de la efectividad de CET como técnica para los entornos educativos.
- Para el instructor, contar con un registro de eventos y seguimiento en tiempo real de las actividades de los usuarios le sugiere una forma más adecuada de estar presente en el proceso educativo sin interrumpir o truncar los aportes que entre compañeros se pueda estar dando. Sin embargo, un registro de eventos no es suficiente, es necesario contar con distintas vistas de la actividad de grupo que faciliten la comprensión y evaluación de los aportes de un usuario. En esta

experiencia, dicha diversidad se logra dar mediante los distintos servicios de *Awareness* implementados tanto en el libro digital, CETProfesor y la red de Agentes, los cuales facilitaron la comprensión del escenario colaborativo.

- Tanto los Agentes de Interfaz como de Servicio obtuvieron comentarios favorables respecto a su usabilidad y funcionalidad en las actividades de grupo. La búsqueda de expertos ayudó a los usuarios a identificar sus pares de trabajo e incluso a estimar un nivel de confianza sobre sus aportes en función de su nivel de experticia.
- Durante su ejecución, los Agentes se comportaron de forma estable, manteniendo a las estaciones de cómputo en niveles de consumo de memoria y procesador normales, lo cual sugiere la posibilidad de escalar tanto en servicios como en complejidad de los mismos.

En resumen, esta experiencia presentó la verificación exitosa tanto del prototipo construido para CET, como de la Red de Agentes y Servicios *Awareness* soportados. Los valores de desempeño y satisfacción, muestran una mejora en el escenario educativo, demostrando la efectividad de la propuesta. Los prototipos desarrollados cumplieron las expectativas propuestas y permitieron verificar la viabilidad técnica de la propuesta Arquitectónica.

## Capítulo 6. Conclusiones y Trabajos Futuros

El presente capítulo expone las principales conclusiones y aportes de este trabajo de investigación y propone líneas de investigación y desarrollo futuras que expanden los alcances de esta tesis.

### 6.1 Conclusiones

Los resultados mas visibles de este trabajo de maestría se resumen en: primero, la definición de una arquitectura para el soporte de Servicios Awareness; y segundo, su verificación a través de dos casos de estudio que permitieron corroborar su viabilidad técnica y validez como estrategia para mejorar los procesos de enseñanza en escenarios de aprendizaje colaborativo. Algunos postulados que sustentan estas afirmaciones son:

Se llegó a la definición de una arquitectura escalable a nivel de servicios y de fácil implementación sustentada en un diseño modular y orientada a través de un esquema de cooperación que combina los distintos enfoques de Awareness, procurando una propuesta que se adapta a distintos contextos de trabajo. Es de recordar que los dos prototipos desarrollados utilizaron plataformas y tecnologías distintas: primero un entorno Web con lenguaje Perl y luego redes P2P, con el lenguaje C# de la plataformas .NET de Microsoft.

Esta arquitectura, tanto a nivel conceptual como a través de su especialización en un Sistema Multi-Agente, se caracteriza por una filosofía de integración con el entorno, lo cual configura un espacio de interoperabilidad y seguimiento continuo de las actividades de los usuarios.

Según los modelos propuestos, cada servicio de *Awareness* se desarrolla con un nivel de autonomía que le permite especializarse en aspectos de la interacción de grupo sin perder la posibilidad de cooperar con otros servicios. Esta estrategia permite realizar una gestión más efectiva de la información y posibilita la diversificación de las formas de representación del Awareness.

La base conceptual de la arquitectura, la separación a tres niveles de la gestión del *Awareness* a través de las entidades Agente, Gestor y Modelo de Información, resulto en una solución adecuada para los escenarios en los cuales fue evaluada. Esto y las observaciones realizadas durante el desarrollo de los prototipos, permiten suponer que se trata de un esquema de gestión válido en diversos contextos y que por tanto su especialización a través de una Red de Agentes, ofrece una extensión favorable a la personalización y desarrollo de soluciones de gestión complejas.

El paradigma software de Agentes Inteligentes, fue exitosamente utilizado para el desarrollo de Servicios de *Awareness* Cooperativos para ofrecer gestión de la información de grupo. Las capacidades de este paradigma amplifican los alcances de la gestión y configura un escenario para el desarrollo de nuevos servicios encaminados a mejorar no solo los entornos de aprendizaje sino también laborales.

A nivel conceptual se ha podido verificar que los servicios de *Awareness* no solo favorecen los procesos de monitoreo y evaluación del desempeño de los estudiantes, al brindar una herramienta de supervisión constante en tiempo real de las actividades del usuario, sino que además asisten a procesos de mejoramiento como la intervención

temprana y oportuna por parte del instructor o el desarrollo de servicios que se integren a la dinámica de grupo para mejorar su desempeño.

De acuerdo con las experiencias realizadas, los Servicios de *Awareness* propician un espacio de colaboración más satisfactorio para los participantes, mejorando la percepción de usabilidad de los sistemas groupware. Además, los usuarios de estos servicios mostraron un creciente interés por participar de formas más activas en procesos de comunicación con otros participantes, lo cual, desde la perspectiva de la ingeniería de la colaboración, favorece el intercambio de conocimiento y por tanto aumenta los niveles de aprendizaje.

La experimentación también mostró que disponer de herramientas de soporte a la colaboración, no garantiza por sí solo el trabajo en grupo. Es necesario contar con una estructuración de tareas que garantice espacios de interacción adecuados pero que a su vez, sea suficientemente flexible para permitir un desarrollo creativo de soluciones. Además, dada la arraigada formación en métodos tradicionales no colaborativos, es necesario contar con la figura de un facilitador o coordinador, que motive a los estudiantes hacia las prácticas colaborativas.

## **6.1 Trabajos Futuros**

Desde una interpretación más profunda, el *Awareness* supone la capacidad de recopilar el conocimiento de los individuos en la medida que estos lo comparten; si se desarrollaran las herramientas adecuadas, se podría llegar a gestionar el conocimiento de forma automática y transparente para las personas. Esta tesis ha planteado estrategias de GC tales como directorios de expertos y búsqueda de los mismos; se propone entonces desarrollar nuevos servicios encaminados a implementar otras estrategias de gestión de conocimiento basadas en información de *Awareness*.

Diversos factores del desarrollo de los prototipos han sido tratados débilmente, por ejemplo aspectos de seguridad dentro de los esquemas de comunicación de los servicios, el no repudio de mensajes o acciones y la privacidad de los datos. Es pertinente extender este estudio hacia la implantación de políticas de seguridad que ofrezca esquemas de comunicación más confiables y para ello, se sugiere un modelo de confianza por niveles que le permitan a un usuario establecer y aplicar reglas de privacidad en función del grado de cercanía o confianza con otro usuario o hacia una comunidad o grupo de trabajo.

Durante la experimentación se observó que cada tipo de servicio de *Awareness* produjo en los estudiantes ciertos estímulos o motivaciones que podrían ser útiles al momento de configurar un entorno colaborativo, por ejemplo: el uso del Participómetro, propició rivalidad y competencia por destacarse; los Usuarios en Línea y la Vista de Radar, una sensación de acompañamiento y necesidad por comunicarse; mientras que las Vistas Historias, la apropiación de los espacios y la sensación de control y trascendencia sobre las acciones. Al combinar adecuadamente estos estímulos y procurar diseños más intensos, se podría llegar a un diseño emocional de los entornos, cubriendo un aspecto de investigación de gran interés en la actualidad para el HCI.

Quedan abiertas las posibilidades de nuevas herramientas groupware que se integren con la red de gestión, modelos de interoperabilidad entre plataformas y el soporte de nuevos mecanismos de *Awareness*. El modelo de información debería extender sus definiciones a

dominios más específicos e incluso dejar abiertas las posibilidades a la inclusión de nuevos enfoque para el Awareness.

Para validar esta propuesta en el contexto CSCW, es necesario diseñar nuevas pruebas y evaluar nuevas posibles modificaciones o extensiones del modelo expuesto. Sería interesante poder contar con lineamientos para el soporte de estos contextos e integrarlos en el ámbito educativo, procurando un espacio de aprendizaje desde el terreno empresarial.

## Referencias

- [1] Hernández D.: "A pattern-based design process for the creation of CSCL macro-scripts computationally represented with IMS LD", Tesis Doctoral, Universidad De Valladolid, 2007.
- [2] Johnson, R., Johnson, D., Stane, M.: "Comparison of computer-assisted cooperative, competitive, and individualistic learning". American Educational Research Journal. Vol. 23, 1985, pp. 246-268.
- [3] Plantamura P., Roseli T., y Rossano V. "Can a CSCL environment promote effective interaction?" en Proc. 4th. IEEE International Conference on Advanced Learning Technology (ICALT04). 2004.
- [4] Ong S., y Hawryszkiewicz I. "Towards Personalised and Collaborative Learning Management Systems" en Proc. 3rd. IEEE International Conference on Advanced Learning Technology (ICALT03), 2003.
- [5] Johnson, D., Johnson, R., Stanne, M.: Cooperative Learning Methods: A Meta-Analysis. University of Minnesota, 2000.
- [6] Collazos, C., Guerrero, L., Pino, J., Ochoa, S.: "Introducing Knowledge-Shared Awareness". Proceedings. Of IASTED International Conference: Information and Knowledge Sharing (IKS 2002). St. Thomas, Virgin Islands, USA, November, pp. 13-18. 2002
- [7] Collazos, C., Guerrero, L., Vergara, A.: "Aprendizaje Colaborativo: un cambio en el rol del profesor", Gestión de Páginas Web Educativas, Universidad Autónoma Metropolitana Unidad Iztapalapa, México, 2008.
- [8] Salomon S.: "What does the design of effective CSCL require and how do we study its effects?" SIGCUE Outlook, Special Issue on CSCL, 21(3), 62-68, 1992.
- [9] Dourish, P. and Bellotti, V., *Awareness and Coordination in Shared Workspaces.*, Proceeding ACM Conference on Computer Supported Cooperative Work (CSCW 92), Toronto, ACM Press, 1992.
- [10] Gonzales, M., Zanfrillo, A.: "Entornos de aprendizaje colaborativos para el desarrollo de tesis de posgrado", Buenos Aires, Argentina. Julio de 2007.
- [11] Pavez, A.: "Modelo de implantación de Gestión del Conocimiento y Tecnologías de Información para la Generación de Ventajas Competitivas", Universidad Técnica Federico Santa María, 2000.
- [12] Robert E. Cole, "Special Issue on Knowledge and the Firm", California Management Review, EEUU, 1998
- [13] Hill, J.: "A Direct Manipulation Toolkit for *Awareness* Support in Groupware", Thesis of Master of Science, University of Saskatchewan, 2003.

- [14] Correa, C., Marsic, I.: A Flexible Architecture to Support *Awareness* in Heterogeneous Collaborative Environments, Center for Advanced Information Processing (CAIP), Rutgers - The State University of New Jersey Piscataway, 2002
- [15] Liechti, O.: *Awareness* and the WWW: an Overview, ATRMedia Integration and Communication, Research Laboratories, Seika-cho, Soraku-gun, Kyoto Japan, 2000.
- [16] Marsic, I.: An Architecture for Heterogeneous Groupware Applications, Rutgers University, 2000
- [17] Kirsch-Pinheiro, M., Villanova-Oliver, M., Gensel, J., Martin, H.: BW-M: A Framework for *Awareness* Support in Web-Based Groupware Systems, 2002.
- [18] Drury, J., Williams, M.: A Framework for Role-Based Specification and Evaluation of *Awareness* Support in Synchronous Collaborative Applications, University of Massachusetts, 2001.
- [19] Lacey, T., Mills, R., Raines, R., Williams, P.: "A qualia framework for *Awareness* in cyberspace", Air Force Research Laboratory, USA, 2007
- [20] Zarate, V., De la Cruz E., Ramos, F.: "AWISPA: an *Awareness* framework for collaborative spontaneous networks", Instituto Tecnológico de Estudios Superiores de Monterrey Campus Cuernavaca, Electronic & Mechatronic Department Cuernavaca, Morelos, Mexico, 2006
- [21] Costaguta, R.: "Una Revisión de Desarrollos Inteligentes para Aprendizaje Colaborativo Soportado por Computadora", Revista Ingeniería Informática, edición 13, Chile, 2006.
- [22] Ayala G. "Towards lifelong learning environments: Agents supporting the collaborative construction of knowledge in virtual communities". En Proc. International Conference on Computer Support for Collaborative Learning, pp. 141–149. 2003.
- [23] Taurison N, y Tchownikine P.: "Supporting a Learner Community with Software Agents". Educational Technology & Society. Vol. 7(2), pp. 82-91. 2004.
- [24] Cobos, R, Claros, I., Moreno-Llorena, J.. A proposal of *Awareness* Services for the Construction of Quality Community Knowledge supported by the Knowledge Management system KnowCat. 13th International Conference on Human-Computer Interaction. San Diego (Estados Unidos), Julio de 2009.
- [25] Claros, I., Collazos, C., Cobos, R.: "Servicios *Awareness* para la mejora del escenario educativo soportado sobre KnowCat", Revista Avances en Sistema e Información, Vol.6. No. 1, ISSN 1657-7663, pp. 127-133, 2009.
- [26] Ovalle, D., Jiménez, J., Collazos, C., Claros, I., Pantoja, W., Cobos, R., MorenoLlorena, J., Pifarré, M., Argelagos, E.: "Guía metodológica para el seguimiento y evaluación de aprendizaje colaborativo asistido por el sistema KnowCat". Reunión Nacional y Expoingeniería ACOFI 2009, septiembre de 2009.

- [27] Claros, I., Collazos, C., Cobos, R., Evaluación de Servicios *Awareness* para Sistemas de Gestión de Conocimiento, Experiencia KnowCat, IV Congreso Colombiano de Computación, Abril de 2009.
- [28] Collazos, C., Cobos, R., Jiménez, J., Moreno, J., Pifarré, M., Ovalle, D., Claros, I., Pantoja, L., Propuesta para la monitorización de procesos colaborativos de un sistema de gestión del conocimiento, Interaccion 2008, Albacete (España), Junio 2008, pp. 95-98.
- [29] Claros, I., Collazos, C., Lara, A., Ochoa, S., Guerrero, L.: Arquitectura Multiagente para la Gestión de Información de *Awareness* en Entornos CSCL, Jornadas Chilenas de Computación (Chile), Noviembre de 2009.
- [30] Shneiderman, B.: Designing the user interface. Strategies for effective human-computer interaction. 3d ed. Reading, MA: Addison-Wesley. 1988.
- [31] UNESCO: "Hacia las sociedades del conocimiento", Informe Mundial de La Unesco, Ediciones UNESCO, 2005
- [32] Jhonson and Jhonson. "Instructional Goal Structure: Cooperative, Competitive, or Individualistic". Review of Educational Research 1999.
- [33] Vreede, G. J., Briggs, R.: "Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks" 2004.
- [34] Kaye, A. "Learning Together Apart. In: Collaborative Learning Through Computer Conferencing". (Ed.) A Kaye, Springer-Verlag (Berlin), pg 1-24. 1991
- [35] Ovalle, D., Jiménez, J., Collazos, A.: "Model for supporting awareness in the CSCL ALLEGRO environment through a blackboard architecture", Revista Ingeniería e Investigación vol. 26 no.3, 2006, pp. 67-77.
- [36] Jarboe, S.: Procedure for enhancing group decision making. In B. Hirokawa and W. Poole (Eds), Communicacion and Group Decision Making, pp. 345-383. 1996
- [37] Collazos, C., Guerrero, L., Pino, J., Ochoa, S.: Improving the use of strategies in computer-supported Collaborative Processes, In J. Favela and D. Decouchant (Eds): Groupware: Design, Implementation and Use, Lecture Notes in Computer Science 2806, pp. 247-260. 2003.
- [38] Ellis, C.A., Gibbs, S.J., & Rein, G.L.: Groupware: Some issues and experiences. Communications of the ACM, 34(1), 38-58. 1991
- [39] Ogata H, Matsuura, K., Yano Y.: Combining knowledge Awareness and information filtering in an open-ended collaborative learning environment. Int. Journal of Artificial Intelligence in Education, Vol. 11, pp. 33-46, 2000.
- [40] Gaver, W. Sound support for collaboration. Procs. of the ESCW'91, 293-308, 1991.
- [41] Calvary, G., Coutaz, J, Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers (2003) 289-308, 2003.

- [42] Durfee E.E., Lesser V.R., and Corkill D.D.. "Trends in cooperative distributed problem solving". IEEE Transactions on Knowledge and Data Engineering, 1(1):63-83, 1989.
- [43] Durfee, E.; Rosenschein, J. "Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples". Proc. 13th Int. DAI Workshop, 1994, p. 94-104.
- [44] Hernández, M.: Modelaje de Procesos de Negociación como Sistemas de Comunicación, 1995.
- [45] Albin, C.: Fairness Issues in Negotiation: Structure, Procedures and Outcome. IIASA. 1992.
- [46] Bazerman, M., Neale, M.; La Negociación Racional. Paidós Empresa 20, 1993
- [47] Russell, S., Norvig, P.: Artificial Intelligence, A modern Approach. Prentice Hall International. 1995.
- [48] Labidi, S., Lejouad, W.: De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents, 1993
- [49] Mestras J. "Metodologías de construcción de sistemas Multi-Agente" Agentes Inteligentes, sistemas multiagente y aplicaciones. Editorial Club Universitario, 2002.
- [50] Gómez J. "Metodologías para el desarrollo de sistemas multi-agente". Sanz. Revista Iberoamericana de IA, nº 18. AEPIA 2003.
- [51] Wooldridge M., Jennings N. R., Kinny D. "The Gaia Methodology for Agent-Oriented Analysis and Design". Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers, 2000.
- [52] DeLoach S.: "Analysis and Design using MaSE and agentTool". Midwest Artificial Intelligence and Cognitive Science Conference, 2001.
- [53] Iglesias C., Garijo M., González J., Velasco, J.: "Analysis and Design of Multiagent Systems using MAS-CommonKADS", 1998.
- [54] Caire G., Leal F., Chainho P., Evans R., Garijo F., Gomez J., Pavón J., Kearney P., Stark J., Massonet P. "Agent oriented analysis using MESSAGE/UML". Proc. of 2nd International Workshop on Agent Oriented Software Engineering, Montreal Canada, August 2001.
- [55] Bratman, M. E. "Intentions, Plans, and Practical Reason" Harvard University Press. 1987.
- [56] Pnueli A. "Specification and development of reactive systems". Information Processing 86, Elsevier Science Publishers B.V.: Amsterdam, the Netherlands, 1986.
- [57] Coleman D., Arnold P., Bodoff S., Dollin C., Gilchrist H., Hayes F., Jeremaes P. "Object-Oriented Development: The fusion Method". Prentice Hall International: Hemel Hempstead, England, 1994.
- [58] Smith R.: "A Framework for Distributed Problem Solving". UMI Research Press, 1980.
- [59] Dorsey, J.: Twitter, <http://twitter.com/>, Consultada: Marzo 10 de 2010.
- [60] Gutwin, C., Greenberg, S.: "Workspace Awareness Support with Radar Views", 1996.

- [61] W3C: Widgets 1.0 Packaging and Configuration, <http://www.w3.org/TR/widgets/>, 2009. Consultada: Marzo 10 de 2010.
- [62] Wooldridge, M., Ciancarini, P.: Agent-oriented software engineering: The state of the art. In Wooldridge M., Ciancarini, P. (Eds.), Agent Oriented Software Engineering. Springer Verlag, LNAI 1975, 2001.
- [63] Hamill, L., Gilbert, N.: "Social Circles: A Simple Structure for Agent-Based Social Network Models". Journal of Artificial Societies and Social Simulation 12(2)3, 2009.
- [64] Soto, J., Vizcano, A., Portillo, J., Piattini, M., Kusche, O.: A Two-Layer Multi-agent Architecture to Facilitate Knowledge Sharing within Communities of Practice, Inteligencia Artificial 42 (2009), 46-54, doi: 10.4114/ia.v13i42.1045, 2009.
- [65] Carreto, C., Menchaca F.: odelo de Seguridad para Sistemas de Trabajo Colaborativo (Sec-CSCW), ISBN: 970-36-010409, Actas del Segundo Congreso Iberoamericano de Seguridad Informática CIBSI '03. Servicio de Publicaciones del Instituto Politécnico Nacional, México DF. 2003
- [66] Cobos, R., Alamán, X., Esquivel, J. A.: "KnowCat: Catalizador de conocimiento", Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, 2000.
- [67] Jade: Java Agent Develop environment, <http://jade.tilab.com/>, Consultada: Marzo 10 de 2010.
- [68] Ochoa, S., Guerrero, L., Pino, J. Collazos, C., Fuller, D.: "*Improving Learning by Collaborative Testing*". The Journal of Student Centered Learning (JSCL). Forum Press. USA. 1(3), pp. 123-135. 2003.
- [69] <http://www.agent-software.com/index.html>, Consultada: Marzo 10 de 2010.
- [70] Jadex, <http://jadex.informatik.uni-hamburg.de/> , Consultada: Marzo 10 de 2010.
- [71] Tryllian, <http://www.tryllian.org/>, Consultada: Marzo 10 de 2010.
- [72] AgentBuilder, <http://www.agentbuilder.com/>, Consultada: Marzo 10 de 2010.
- [73] APL 3, <http://www.cs.uu.nl/3apl/>, Consultada: Marzo 10 de 2010.
- [74] Agent Factory, <http://www.agentfactory.com/>, Consultada: Marzo 10 de 2010.
- [75] ERXA, <http://www.erca.it/prodotti.php?id=146>, Consultada: Marzo 10 de 2010.
- [76] Windows Communication Foundation, .NET Framework, <http://msdn.microsoft.com/en-us/netframework/aa663324.aspx>, Consultada: Marzo 10 de 2010.
- [77] Aronson, E., Patnoe, S. The Jigsaw Classroom: Building Cooperation in the Classroom, New York, Longman, 1997.
- [78] FIPA, Foundation for Intelligent Physical Agents, Membeship IEEE Computer Society, <http://www.fipa.org/>, Consultada: Marzo 10 de 2010.

[79] FIPA, ACL Message Structure Specification, Id. 61, <http://www.fipa.org/specs/fipa00061/> ,  
Consultada: Marzo 10 de 2010.