AYLLU. PLATAFORMA DE COOPERACIÓN MEDIADA POR AGENTES APLICADA EN UN CONTEXTO DE E-LEARNING COLABORATIVO

Aprender a cooperar y cooperar para aprender

AYLLU. PLATAFORMA DE COOPERACIÓN MEDIADA POR AGENTES APLICADA EN UN CONTEXTO DE E-LEARNING COLABORATIVO

Aprender a cooperar y cooperar para aprender

Enrique González
Demetrio Arturo Ovalle Carranza
Ángela Carrillo Ramos
Editores académicos

Pontificia Universidad JAVERIANA Bogotá | Vicerrectoría de Investigación

editorial Pontificia Universidad JAVERIANA

Colección Libros de Investigación
Vicerrectoría de Investigación

Reservados todos los derechos
© Pontificia Universidad Javeriana
© María Paula Arias Baez, Francisco Arias Sánchez, Mónica Brijaldo Rodríguez, Ángela Carrillo Ramos, Juan Pablo Garzón, Enrique González Guerrero, Hervé Martin, Julián Moreno Cadavid, Demetrio Arturo Ovalle Carranza, Alexandra Pomares Quimbaya, Martha Leonor Sabogal Modera, Luis Guillermo Torres Ribero, Rosa María Vicari

Primera edición: Bogotá, D. C., marzo del 2023
ISBN impreso: 978-958-781-820-8
ISBN digital: 978-958-781-821-5

Editorial Pontificia Universidad Javeriana Cra. 7.ª, núm. 37-25, oficina 1301 Teléfono: 3208320, ext. 4205
www.javeriana.edu.co/editorial
editorialpuj@javeriana.edu.co
Bogotá, D. C.

Corrección de estilo:
Julián Naranjo Guevara

Diagramación y montaje de cubierta:
Francisco Jiménez

Pontificia Universidad Javeriana | Vigilada Mineducación. Reconocimiento como Universidad: Decreto 1297 del 30 de mayo de 1964. Reconocimiento de personería jurídica: Resolución 73 del 12 de diciembre de 1933 del Ministerio de Gobierno.

Pontificia Universidad Javeriana. Biblioteca Alfonso Borrero Cabal, S. J.
Catalogación en la publicación

González Guerrero, Enrique, autor, editor académico
AYLLU : plataforma de cooperación mediada por agentes aplicada en un contexto de e-learning colaborativo. Aprender a cooperar y cooperar para aprender / autores, Enrique González Guerrero [y otros doce] ; editores académicos, Enrique González Guerrero, Demetrio Arturo Ovalle Carranza, Ángela Carrillo Ramos. -- Primera edición. -- Bogotá : Editorial Pontificia Universidad Javeriana, 2023. (Colección Libros de Investigación)

252 páginas ; 17 x 24 cm
ISBN: 978-958-781-820-8 (impreso)
ISBN: 978-958-781-821-5 (electrónico)

1. Aprendizaje colaborativo 2. Educación basada en web 3. Enseñanza con ayuda de computadores 4. Estrategias educativas I. Ovalle Carranza, Demetrio Arturo, autor, editor académico II. Carrillo Ramos, Ángela, autora, editora académica III. Pontificia Universidad Javeriana. Bogotá. Vicerrectoría de Investigación

CDD 371.148 edición 23

inp 09/03/2023

Prohibida la reproducción total o parcial de este material, sin autorización por escrito de la Pontificia Universidad Javeriana.

CONTENIDO

1
APRENDIZAJE COLABORATIVO A TRAVÉS DEL USO DE TIC: GENERALIDADES 11
Mónica Brijaldo, Martha Sabogal

2
ENSEÑANZA Y APRENDIZAJE COLABORATIVOS ASISTIDOS POR COMPUTADOR: ESTRATEGIAS, MÉTODOS Y HERRAMIENTAS 21
Demetrio A. Ovalle C., Julián Moreno C., Francisco Arias S.

3
SELECCIÓN DE ESTRATEGIAS EDUCATIVAS EN UN CONTEXTO DE APRENDIZAJE COLABORATIVO 41
Demetrio A. Ovalle C., Francisco Arias S., Julián Moreno C.

4
OBJETOS VIRTUALES DE APRENDIZAJE 63
Julián Moreno, Demetrio A. Ovalle, Rosa M. Vicari

5
SISTEMAS DE SOPORTE A LA COLABORACIÓN 75
Luis Guillermo Torres-Ribero, Enrique González

6
AYLLU: PLATAFORMA BASADA EN AGENTES PARA APOYO A LA COLABORACIÓN 95
Enrique González, Ángela Carrillo-Ramos, Alexandra Pomares Quimbaya, Luis Guillermo Torres, María Paula Arias

7
AES: *FRAMEWORK* BASADO EN AGENTES PARA ENRIQUECER SERVICIOS COMPUTACIONALES EN CONTEXTOS MULTIDIMENSIONALES 113
Ángela Carrillo-Ramos, María Paula Arias-Báez, Luis Guillermo Torres-Ribero, Juan Pablo Garzón, Enrique González, Hervé Martin

8
AES-EDU: ENRIQUECIMIENTO DE SERVICIOS EN UN ENTORNO EDUCATIVO 133
Ángela Carrillo-Ramos, Luis Guillermo Torres-Ribero, María Paula Arias-Báez, Enrique González

9
MATEO: SISTEMA ADAPTATIVO PARA LA CONFORMACIÓN DE EQUIPOS DE TRABAJO 151
María Paula Arias-Báez, Ángela Carrillo-Ramos

10
MATEO-EDU: UN CASO DE ESTUDIO SOBRE CONFORMACIÓN DE EQUIPOS DE TRABAJO EN UN CONTEXTO EDUCATIVO 169
María Paula Arias-Báez, Ángela Carrillo-Ramos

11
AYPUY: ADMINISTRADOR DE RECURSOS EN AMBIENTES COLABORATIVOS 193
Alexandra Pomares Quimbaya, Luis Guillermo Torres-Ribero, Enrique González

12
PLATAFORMA BESA 3 209
Juan Pablo Garzón, Andrea Barraza, Enrique González

13
CASO DE ESTUDIO DE AYLLU-EDU EN CONTEXTOS EDUCATIVOS 233
Ángela Carrillo-Ramos, Enrique González, Demetrio Ovalle

1
APRENDIZAJE COLABORATIVO A TRAVÉS DEL USO DE TIC: GENERALIDADES

Mónica Brijaldo, Martha Sabogal

Este capítulo realiza una revisión de los conceptos generales relacionados con el aprendizaje colaborativo. La perspectiva que se aplica es la de potenciar el desarrollo del conocimiento gracias a la mediación de la colaboración. En particular, se presenta la colaboración como un componente importante para lograr procesos de aprendizaje más efectivos. En este marco, se analizan algunos trabajos previos que permiten tener claridad sobre el rol del trabajo en equipo y se exponen las diferencias entre el aprendizaje cooperativo y el aprendizaje colaborativo. Finalmente, se introducen las estrategias didácticas, los ambientes colaborativos de aprendizaje asistidos por computador, el uso de Objetos Virtuales de Aprendizaje y el concepto de adaptabilidad en el aprendizaje colaborativo.

1. Introducción

El uso de plataformas adaptadas para lograr más y mejores aprendizajes es un área en la que principalmente los ingenieros de sistemas, así como los educadores, han estado colaborando con el fin de proponer ambientes de aprendizaje más acordes a las necesidades generacionales individuales y colectivas que la sociedad de la información ha hecho más visibles en los últimos veinte años. Es así que la necesidad de explorar cómo potenciar los logros educativos con el apoyo de plataformas es un tema vigente en investigación y desarrollo de tecnologías, y es una de las pretensiones del proyecto AYLLU, que está particularmente enfocado en analizar de qué manera el aprendizaje colaborativo puede ser una estrategia que potencia el conocimiento y cómo se mediría esta colaboración con apoyo de una plataforma tecnológica.

Por lo anterior, este capítulo tiene la siguiente estructura: presentar los conceptos básicos relacionados con el aprendizaje colaborativo, sus diferencias con el aprendizaje cooperativo, las estrategias didácticas utilizadas, sus ambientes de aprendizaje asistidos por computador y la relación entre la colaboración y la adaptación; este último proceso consiste en identificar las características del usuario y de su contexto con el objetivo de brindarle información en el momento y lugar en que la necesite.

2. Concepto base de aprendizaje colaborativo

El ser humano es eminentemente un ser social, esto indica que su desarrollo está mediado por la influencia de sus pares, y por ello el aprendizaje es una actividad de reconstrucción de cúmulo de la cultura. La posibilidad de enriquecernos cognitivamente, ampliar perspectivas y el propio desarrollo humano está mediada por la comunicación y la interacción en los diversos grupos humanos. Por lo anterior, la psicología cognitiva y las ciencias del lenguaje se han interesado por el estudio de las interacciones que ocurren entre profesores y estudiantes, y entre estudiantes. Desde allí se observa que la interacción educativa está constituida por situaciones intencionadas en las que especialmente los estudiantes desarrollan actividades simultáneas y recíprocas alrededor de unos contenidos específicos con objetivos previamente definidos por el profesor. Este es el principio básico del aprendizaje colaborativo, el cual, según Vygotsky, se fundamenta en comprender que el aprendizaje despierta una variedad de procesos que operan con la interacción y en colaboración con otras personas (Vygotsky, 1978). Este autor indica que la construcción del conocimiento es el resultado de interacciones sociales y del uso del lenguaje, por ello el aprendizaje es una experiencia compartida (social) en la que se intercambian ideas, se desarrollan habilidades comunicativas y sociales, y se logran metas en colaboración con otros. Este planteamiento le atribuye al desarrollo cognitivo una relación muy fuerte con la interacción social y es por esto que su potencia tiene grandes posibilidades didácticas.

Para Díaz Barriga (1999), el aprendizaje colaborativo se caracteriza por la igualdad que debe tener cada individuo en el proceso de aprendizaje y la mutualidad, entendida como la conexión, profundidad y bidireccionalidad que alcance la experiencia, siendo esta una variable en función del nivel de competitividad existente, la distribución de responsabilidades, la planificación conjunta y el intercambio de experiencias. Por su parte, Gros (2000) afirma que se trata de un proceso en el que las partes se comprometen a aprender algo juntas. Lo que debe ser aprendido solo puede conseguirse si el trabajo del grupo es realizado en colaboración, por ello el grupo decide cómo realizar la tarea, qué procedimientos adoptar, cómo dividir el

trabajo o tareas que se van a realizar bajo unos parámetros en los que la comunicación y la negociación son clave.

Es así como los actores tendrán bien definidos los roles y responsabilidades desde dos perspectivas: desde el rol del docente y desde los roles que asumen los estudiantes. Para el caso del profesor se plantea el rol de facilitador o tutor del trabajo en equipo y su responsabilidad está en direccionar el trabajo de cada grupo, resolver dudas e inquietudes, observar las interacciones de los estudiantes y hacer las sugerencias necesarias en caso de que así se requiera. Para que este rol de facilitador-docente se pueda llevar a cabo es clave, en el desarrollo del trabajo, tener plataformas tecnológicas con las cuales se pueda recoger, sistematizar y analizar el trabajo en equipo de los estudiantes, pues esta dimensión ayuda al cumplimiento de la responsabilidad del docente (Özyurt y Özyurt, 2015).

Para los estudiantes, el rol se plantea desde las acciones que se deben realizar para lograr el propósito común planteado en el trabajo de equipo. Con este fin, es necesario que haya conciencia en los miembros del equipo acerca de la manera en que van a asumir su rol en el trabajo que se va a desarrollar. Una de las acciones es la interdependencia positiva, entendida como la confianza en que todos los miembros cumplirán con su responsabilidad individual y todos asumirán las consecuencias del incumplimiento de alguno. De igual manera, es necesario que se comprometan a que algunas de las actividades tendrán que realizarse cara a cara para discutir e interactuar, promover el liderazgo de los miembros, las capacidades para la toma de decisiones y la resolución de conflictos, así como la aplicación de correctivos en la dinámica del equipo (Barrios y González-Bravo, 2011).

2.1. Cercanía con el concepto de aprendizaje cooperativo

El aprendizaje cooperativo es una estrategia que promueve la interdependencia de los miembros del grupo en todos los niveles y la posibilidad de adquirir herramientas que les permitan aprender de los otros efectivamente. El objetivo es que trabajen juntos para alcanzar metas sociales y académicas claramente establecidas que los beneficien a todos. Este tipo de aprendizaje puede ser entendido como un tipo de aprendizaje colaborativo en el cual los estudiantes cooperan para desarrollar una actividad estructurada en que se valora tanto el trabajo individual como el grupal. Según Johnson *et al*. (1991) la enseñanza y el aprendizaje colaborativos están determinados por las siguientes consideraciones desde las perspectivas individual y grupal:

1. *Interdependencia positiva*. Los estudiantes de un grupo están obligados a confiar en los otros para lograr un objetivo. Si algún estudiante falla en su función, todos

sufren las consecuencias. El aprendizaje colaborativo considera aspectos de interdependencia en el establecimiento de metas, tareas, recursos, roles y premios, entre otros [2].

2. *Responsabilidad individual*. Todos los estudiantes de un grupo tienen la responsabilidad de hacer su parte de trabajo para que todo el material sea aprendido. Además, deben poseer espacios para compartir el material con el grupo y recibir sus contribuciones.
3. *Promover la interacción cara a cara*. Aunque algunas partes del trabajo en grupo pueden realizarse de manera individual, otras deben hacerse interactuando con otros miembros del grupo que aportan, discuten conclusiones y razonamientos y, quizás la más importante, enseñando y apoyándose el uno al otro. En la medida en que se posean diferentes medios de interacción, el grupo podrá enriquecerse, aumentar sus refuerzos y retroalimentarse [2].
4. *Uso apropiado de habilidades colaborativas*. El estudiante es impulsado a desarrollar y practicar habilidades como el liderazgo, la toma de decisiones, la comunicación y la solución de conflictos.
5. *Mejoramiento continuo grupal*. Los grupos establecen objetivos comunes, periódicamente evalúan lo que han realizado bien como equipo e identifican cambios que pueden funcionar de una manera más eficiente en el futuro.

En el capítulo 5, Sistemas de Soporte a la Colaboración, se desarrolla la diferencia entre los conceptos de colaboración y cooperación y se aclara con cuál definición se entenderá la idea de colaboración y cooperación.

3. Estrategias didácticas en el aprendizaje colaborativo

En la formación educativa actual los procesos de enseñanza se han enfocado en la transmisión del conocimiento y en la transferencia de información, pero la construcción del conocimiento se manifiesta como un proceso de adecuación de las capacidades de los estudiantes. Esto nos lleva a reflexionar sobre los procesos de aprendizaje mediante los cuales los estudiantes apropian el conocimiento.

Uno de los elementos primordiales en el aprendizaje colaborativo es la comunicación, debido a que esta fomenta las habilidades de trabajo en grupo y la resolución de problemas de manera conjunta. Por ello, los procesos educativos deben evolucionar de un aprendizaje individual a uno grupal. Esto implica también el cambio en el proceso de la enseñanza y el uso de diferentes herramientas tecnológicas que

apoyen el trabajo colaborativo, pues estas se convierten en una mediación pedagógica pertinente para el contexto educativo.

En el capítulo 2, Enseñanza y Aprendizaje Colaborativo, se desarrollan los planteamientos sobre cómo desde un contexto colaborativo se producen resultados en el aprendizaje mucho más favorables para los estudiantes cuando se emplean espacios como plataformas que ayudan a plantear el trabajo en equipo y a asumir los roles asignados.

4. Ambientes colaborativos de aprendizaje asistidos por computador

Las interacciones colaborativas que se suceden en los procesos de enseñanza-aprendizaje mediante el empleo de plataformas se ven favorecidas desde dos aspectos: el espacio-temporal que permite la discusión y el trabajo, no solo en el contexto presencial de una clase, sino también a lo largo del proceso formativo; el segundo aspecto es el trabajo realizado por un propósito común y sobre el cual se puede hacer la construcción colectiva a partir de los diferentes aportes y avances en conocimiento del grupo. Como lo menciona Crook (1998), la tecnología puede "mediar unas formas de actividad que creen comunidades de conocimiento compartido [...]. Esto surge en situaciones en las que las personas se unen en comunidades que comparten un conjunto común de preocupaciones, como las que, a veces, aparecen en la educación institucionalizada" (p. 237).

Siguiendo al autor, dentro de los aspectos que el uso del computador debe fortalecer en el aprendizaje colaborativo se puede mencionar:

- Estructuras de red para la actividad conjunta: el diseño de la red se compone de espacios que permiten la discusión, el apoyo en la escritura e iniciar las prácticas colaborativas básicas.
- Estructuras de participación: la arquitectura de los sistemas permite las intervenciones amplias, horizontales y democráticas de todos los miembros de un equipo de trabajo, lo cual ayuda a configurar posibles comunidades de aprendizaje.

Para el proyecto AYLLU se empleó el concepto de sistemas colaborativos que se elabora en el capítulo 5, Sistemas de Soporte a la Colaboración, y las estrategias de aprendizaje colaborativo desarrolladas en el capítulo 2, Enseñanza y Aprendizaje Colaborativo.

En cuanto al manejo de la colaboración en AYLLU, podemos decir que se logra a través del diseño de estrategias de colaboración que se expresan por medio de diálogos bien estructurados que se llevan a cabo mediante mensajes (con intencionalidad

y semántica explícitas) entre roles que son asignados a los estudiantes y profesores. El correcto desarrollo de estos diálogos es soportado por un conjunto de agentes de *software* que, además de apoyar el proceso de comunicación, realizan la gestión de los Objetos Virtuales de Aprendizaje (OVA) que se encuentran ubicados en diferentes repositorios. La arquitectura multiagente que soporta la colaboración se explora con mayor profundidad en los capítulos 6 y 11, AYLLU: Plataforma basada en Agentes para Apoyo a la Colaboración y AYPUY: Administrador de Recursos en Ambientes Colaborativos.

5. El uso de Objeto Virtual de Aprendizaje

Para el Ministerio de Educación Nacional (2016), en Colombia un Objeto Virtual de Aprendizaje (OVA) es un

> conjunto de recursos digitales, autocontenible y reutilizable, con un propósito educativo y constituido por al menos tres componentes internos: contenidos, actividades de aprendizaje y elementos de contextualización. El objeto virtual de aprendizaje debe tener una estructura de información externa (metadatos) que facilite su almacenamiento, identificación y recuperación.

Para el uso de los OVA en este libro de investigación, además de estas características generales, se plantearon dos factores como ejes de reflexión pedagógica para la selección de los OVA que se van a emplear:

- El estilo de aprendizaje del estudiante.
- Las preferencias del estudiante en el manejo y presentación de la información.

La manera en la que se establecieron los estilos de aprendizaje, entre otras características de los estudiantes en los procesos de aprendizaje, se desarrolla en los capítulos 4 y 8, Objetos Virtuales de Aprendizaje; y AES-EDU: Enriquecimiento de Servicios en un Entorno Educativo.

6. El concepto de adaptabilidad en el aprendizaje colaborativo

La *adaptabilidad* se define como la capacidad de analizar las diferentes variables que componen un contexto de aprendizaje determinado y las posibilidades de modificación o cambios que sean necesarios para desarrollar un proceso de

enseñanza-aprendizaje efectivo, por ello, la pertinencia del currículum y de la oferta educativa deben responder a los usuarios de este contexto (Stewart, Niederee y Metha, 2004).

Los aspectos que se analizan en un proceso adaptativo tienen que ver con la cultura, los horarios, los contenidos, las mediaciones pedagógicas, las tecnologías, las metodologías, los instrumentos y los procedimientos de evaluación, y que estos se adapten a cada contexto específico; por ejemplo, zona geográfica, período del año, clima, edad, género, nivel educativo, motivaciones, ritmos y estilos de aprendizaje, necesidades especiales, entre otros. Esto supone conocimiento de las realidades locales, capacidad para anticipar y rectificar oportunamente y participación de los actores educativos involucrados (Chang *et al*., 2016).

En el campo educativo, la adaptabilidad a través de un sistema inteligente pretende mejorar la experiencia del usuario tomando como base el conocimiento sobre las características que son relevantes para la aplicación educativa, por ejemplo, estilo de aprendizaje, personalidad, capacidades, habilidades, competencias, así como lo relacionado con su contexto (equipo de trabajo, profesor, herramientas TIC). Dicho contexto puede afectar las necesidades de información del usuario, por lo cual es necesario que en una plataforma como la que se propuso en esta experiencia, se incorpore un módulo que adapte la información para los diversos escenarios (móviles o no), y que enriquezca los servicios educativos por medio del ajuste a las necesidades y preferencias del usuario, las características del contexto y las de su dispositivo de acceso (Niño, 2016). Por ello, la adaptabilidad tiene que ver con la diferencia en la manera en que se desarrollan los procesos de aprendizaje y debe generar una oferta flexible y diversificada, que responda a las diferencias individuales y colectivas.

Como se puede apreciar en el capítulo 10, MATEO-EDU: Un Caso de Estudio sobre Conformación de Equipos de Trabajo en un Contexto Educativo, en la presente investigación se realizó la adaptabilidad desde dos aspectos básicos: la selección de equipos de trabajo, según las necesidades de las actividades planteadas por el profesor, y la posibilidad de distribuir en equipos heterogéneos y en equipos homogéneos. Así mismo se podrá analizar en el caso de las actividades, las acciones denominadas como "mejor ajuste" o "peor ajuste".

7. Conclusiones

Los procesos de enseñanza-aprendizaje que se realizan aplicando el paradigma del aprendizaje colaborativo y cooperativo se desarrollan de mejor manera cuando están soportados en plataformas colaborativas que ayudan a hacer más eficientes las

actividades y los retos planteados a los estudiantes; pero también es necesaria la identificación de fortalezas y habilidades que tienen los sujetos participantes en un grupo.

Por esto, la propuesta de investigación que se formaliza y desarrolla en los capítulos siguientes muestra el análisis de los sistemas de aprendizaje colaborativos, el *awareness* o consciencia del proceso del aprendizaje de los integrantes de un grupo de trabajo, las arquitecturas necesarias y la ingeniería de colaboración, para lograr desarrollar el concepto de sistema de aprendizaje colaborativo apoyado por computador. Así mismo, se trabaja desde la idea de cómo deben conformarse los grupos de trabajo y los sistemas de CSCL, requeridos y necesarios para comprobar los aprendizajes de los estudiantes.

Finalmente, a partir de la revisión de los conceptos que fundamentaron el desarrollo de plataformas basadas en aprendizaje colaborativo, se puede concluir que la planeación de las diversas actividades depende inicialmente del profesor, quien tiene a cargo el diseño de la ruta de aprendizaje en la cual, tanto las tareas, así como la definición de roles que se requieren, son una de las primeras apuestas que se ponen al servicio de la colaboración entre los estudiantes o participantes. Una vez inicia la actividad, la manera como los estudiantes desarrollan la tarea y se apoyan en las herramientas tecnológicas toman un curso diferente, dependiendo de la manera como se integren los usuarios, las interdependencias positivas que se generen y las metas que compartan, con el apoyo permanente del profesor.

Referencias

Barrios, L. y González-Bravo, C. (2011): Aproximación a una Metodología de Aprendizaje Emocional (MAE) en las enseñanzas de Arquitectura. Actas del I Congreso Internacional son Aprendizaje, Innovación y Competitividad. Madrid: Universidad Politécnica de Madrid.

Chang, Y. H., Chen, Y. Y., Chen, N. S., Lu, Y. T. y Fang, R. J. (2016). Yet Another Adaptive Learning Management System Based on Felder and Silverman's Learning Sstyles and Mashup. *Eurasia Journal of Mathematics, Science & Technology Education*, *12*(5), 1273-1285.

Crook, Ch. (1998). *Ordenadores y aprendizaje colaborativo*. Madrid: Ediciones Morata.

Díaz Barriga, R. (1999). *Estrategias docentes para un aprendizaje significativo. Una interpretación constructivista*. Ciudad de México: Mcgraw-Hill.

Gros, B. (2000). *El ordenador invisible*. Barcelona: Gedisa.

Johnson, D. W., Johnson, R. T. y Smith, K. A. (1991). *Learning Together and Alone*. Englewood Cliffs, N. J.: Prentice Hall.

Ministerio de Educación Nacional de Colombia (2016). Portal Colombia Aprende. Disponible en http://www.colombiaaprende.edu.co/html/directivos/1598/article-172369.html. Consultado el 7 diciembre de 2016.

Niño, M. (2016). La personalización de ambientes educativos digitales basados en estilos de aprendizaje y estilos cognitivos. Una revisión sistemática sobre su eficacia y percepción. *RELATEC Revista Latinoamericana de Tecnología Educativa*, *15*(3).

Özyurt, Ö. y Özyurt, H. (2015). Learning Style Based Individualized Adaptive E-Learning Environments: Content Analysis of the Articles Published from 2005 to 2014. *Computers in Human Behavior*, *52*, 349-358.

Stewart, A., Niederee, C. y Metha, B (2004). *State of the Art in User Modelling for Personalization in Content, Service and Interaction*, NSF/DELOS Report on Personalisation.

Vygotsky, L. S. (1978). *Pensamiento y lenguaje*. Madrid: Paidós.

2
ENSEÑANZA Y APRENDIZAJE COLABORATIVOS ASISTIDOS POR COMPUTADOR: ESTRATEGIAS, MÉTODOS Y HERRAMIENTAS

Demetrio Arturo Ovalle Carranza, Julián Moreno Cadavid y Francisco Arias Sánchez

Las propuestas innovadoras sobre estrategias de enseñanza y aprendizaje han permitido el avance de la educación buscando cambiar la clásica concepción de aprendizaje individual y competitivo por nuevos modelos académicos como la enseñanza y aprendizaje colaborativos, los cuales parten de la premisa "dos cabezas piensan y aprenden mejor que una". En efecto, el aprendizaje colaborativo es un acto social natural en el cual los participantes interactúan entre sí, y es a través de esta interacción que el aprendizaje se obtiene. Dentro de este marco de referencia, los estudiantes tienen la oportunidad de conversar con sus compañeros, proponer y defender ideas, intercambiar diversas creencias y puntos de vista y estar involucrados activamente en este proceso educativo de construcción de conocimiento. El aprendizaje colaborativo surge mayormente de instancias de trabajo grupal en las que los participantes se organizan en equipos, ya sean heterogéneos, homogéneos o una mezcla de estos; las habilidades, destrezas y conocimientos de cada uno de los miembros del equipo se complementan para lograr una meta común. Los participantes de cada equipo asumen unos roles y responsabilidades que garantizan el éxito del aprendizaje colaborativo. En este capítulo se presentan las estrategias, métodos y herramientas del Aprendizaje Asistido por Computador (CSCL, por sus siglas en inglés), así como el monitoreo y evaluación de sistemas de CSCL.

1. Introducción

La adopción de estrategias de aprendizaje colaborativo ha sido una innovación importante en la educación actual. En la tabla 1 se presentan algunos de los conceptos que sobre aprendizaje colaborativo se encuentran en la literatura (Johnson *et al.*, 1991).

TABLA 1. CONCEPTUALIZACIONES SOBRE APRENDIZAJE COLABORATIVO

Autores	Conceptualizaciones
Panitz y Panitz (1998)	Proceso de interacción cuya premisa básica es la construcción del consenso. Se comparte la autoridad y entre todos se acepta la responsabilidad de las acciones del grupo.
Guitert y Simérez (2000)	Es un proceso en el que cada individuo aprende más de lo que aprendería por sí solo, y es fruto de la interacción de los integrantes del equipo. El trabajo colaborativo se da cuando existe una reciprocidad entre un conjunto de individuos que saben diferenciar y contrastar sus puntos de vista de tal manera que llegan a generar un proceso de construcción de conocimiento.
Gros (2000)	Es un proceso en el que las partes se comprometen a aprender algo juntas. Lo que debe ser aprendido solo puede conseguirse si el trabajo del grupo es realizado en colaboración. Es el grupo el que decide cómo realizar la tarea, que procedimientos adoptar, cómo dividir el trabajo o tareas realizadas. La comunicación y la negociación son claves en este proceso.
Salinas (2000)	Considera fundamental el análisis de la interacción profesor-estudiante y estudiante-estudiante, teniendo en cuenta que el trabajo colaborativo busca el logro de metas de tipo académico y también el mejoramiento de las propias relaciones sociales.
Lucero (2004)	Conjunto de métodos de instrucción y entrenamiento apoyados con estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social), en el que cada miembro del grupo es responsable tanto de su aprendizaje como del de los demás miembros del grupo.

Fuente: Maldonado, 2007.

La importancia del aprendizaje colaborativo se describe en Calzadilla (2000):

> Cada participante asume su propio ritmo y potencialidades, impregnando la actividad de autonomía, pero cada uno comprende la necesidad de aportar lo mejor de sí al grupo para lograr un resultado sinérgico al que ninguno accedería por sus propios

> medios; se logra así una relación de interdependencia que favorece los procesos individuales de crecimiento y desarrollo, las relaciones interpersonales y la productividad.

El aprendizaje colaborativo se soporta en la interacción social. Así, los individuos participantes, organizados en grupos, se complementan entre sí para lograr un objetivo común en un contexto de aprendizaje.

La expresión aprendizaje colaborativo también se refiere a metodologías de aprendizaje que incentivan la colaboración entre individuos. El aprendizaje colaborativo surge mayormente de instancias de trabajo en grupos o trabajo colaborativo. Los diferentes participantes se organizan en grupos, ya sea heterogéneos y homogéneos o una mezcla de estos, los cuales se complementan para lograr una meta común.

2. Estrategias de aprendizaje colaborativo

El principal objetivo de las estrategias de aprendizaje colaborativo es estimular la discusión, el análisis crítico sobre un tema específico, el desarrollo de habilidades individuales y sociales, el logro de metas de tipo académico, el mejoramiento de las relaciones interpersonales, la construcción de conocimiento, la comunicación y la negociación, etc., entre los estudiantes que pertenecen a un grupo-equipo. Hoy en día existen más de 60 tipos de estrategias de aprendizaje grupal (Domingo *et al*., 2004). Entre las más destacadas están las siguientes: (a) Resolución de problemas (*Problem Solving*); (b) Cuestionario recíproco guiado (*Guided Reciprocal Peer Questioning*); (c) Mesa redonda (*Roundtable/Brainstorming*); (d) Rompecabezas o *Puzle* (JIGSAW); (e) Grupos de investigación; (f) Torneos entre grupos (*Teams-Games-Tournaments* [TGT]; (g) Diagrama causa-efecto (*Cause and Effect Diagram*); (h) Diagrama de afinidad (*Affinity Diagram*); (i) Descubrimiento (*Finding Out*); (j) Multivotación (*Multi-voting*); (k) Coincidencias divergentes (*Uncommon Commonalities*), entre otros.

3. Roles y responsabilidades

Los participantes de cada equipo asumen roles y responsabilidades que garantizan el éxito del aprendizaje colaborativo. Algunos de los roles que pueden asumir los integrantes de un equipo que interactúan bajo una estrategia de aprendizaje colaborativo, de las mencionadas en la sección anterior, son los siguientes: (a) Supervisor: monitorea a los miembros del equipo en la comprensión del tema de discusión y detiene el trabajo cuando algún miembro del equipo requiere aclarar

dudas; (b) Abogado: cuestiona sobre ideas y conclusiones ofreciendo alternativas; (c) Motivador: se asegura de que todos tengan la oportunidad de participar en el trabajo en equipo y elogia a los miembros por sus contribuciones; (d) Administrador: provee y organiza el material necesario para las tareas y proyectos; (e) Observador: monitorea y registra el comportamiento del grupo con base en la lista de comportamientos acordada; (f) Secretario: toma notas durante las discusiones de grupo y prepara una presentación para toda la clase; (g) Reportero: resume la información y la presenta a toda la clase; (h) Controlador: monitorea el progreso y eficiencia del grupo; entre otros.

Los conceptos asociados con las responsabilidades que deben asumir cualquiera de los roles presentados anteriormente son los siguientes: (a) Motivar: despertar la atención e interés de los estudiantes antes de introducir un nuevo concepto o habilidad; (b) Proporcionar: presentar alguna experiencia concreta antes de iniciar la explicación de una idea abstracta o procedimiento; (c) Verificar: cerciorarse de que los estudiantes hayan entendido las explicaciones y demostraciones; (d) Ofrecer: documentos adicionales que permitan ampliar el conocimiento en el área o dominio del problema en cuestión; (e) Revisar: verifica que los estudiantes estén desarrollando el trabajo de forma correcta; (f) Cubrir: proporciona información textual adicional a los estudiantes para que puedan completar sus conocimientos; (g) Solicitar: recolecta informes finales sobre el trabajo llevado a cabo.

4. Ambientes colaborativos de aprendizaje asistidos por computador

El Aprendizaje Colaborativo Asistido por Computador (*Computer Supported Collaborative Learning* [CSCL], por sus siglas en inglés) es uno de los enfoques más promisorios para mejorar los procesos de enseñanza y aprendizaje en los estudiantes y está soportado por los avances que han alcanzado las TIC (Tecnologías de la Información y Comunicación) en las últimas décadas. De esta forma, se busca promover el aprendizaje a través del esfuerzo colaborativo entre aprendices en una determinada tarea de aprendizaje, suministrando un ambiente que aviva y enriquece el proceso, en el que el alumno interactúa con otros colaboradores para solucionar un problema (Jiménez, 2006).

Los ambientes de CSCL difieren de los ambientes de CSCW (siglas en inglés para *Computer Supported Collaborative Working*) en que los primeros buscan que los aprendices logren obtener un conocimiento en grupo, en vez de buscar la eficiencia en el trabajo. Cabe señalar que dependiendo del enfoque de trabajo la segunda "C" de la abreviatura puede entenderse como cooperativo o colaborativo (Collazos, 2003).

La palabra *cooperación* se refiere a la búsqueda de una meta común mediante la división de tareas mientras que la palabra *colaboración* hace referencia al logro de un objetivo compartiendo el proceso mismo de la resolución de tareas (Ellis *et al*., 1991).

Respecto a las arquitecturas de CSCL, puede decirse que no hay un consenso en la comunidad académica. Surthers (2001), por ejemplo, presenta una arquitectura la cual utiliza el patrón Controlador de la Vista del Modelo (*Model-View-Controller* [MVC]), la cual puede variar de acuerdo con las actividades educativas que los usuarios realizan y con el soporte de la tecnología, en: centralizada, replicada y distribuida. Dorneich (2002), por su parte, propone la arquitectura exhibida en la figura 1, la cual está conformada por cuatro módulos en los que indica que el principal propósito del ambiente debe ser la categorización del grupo de aprendices y la pedagogía del aprendizaje.

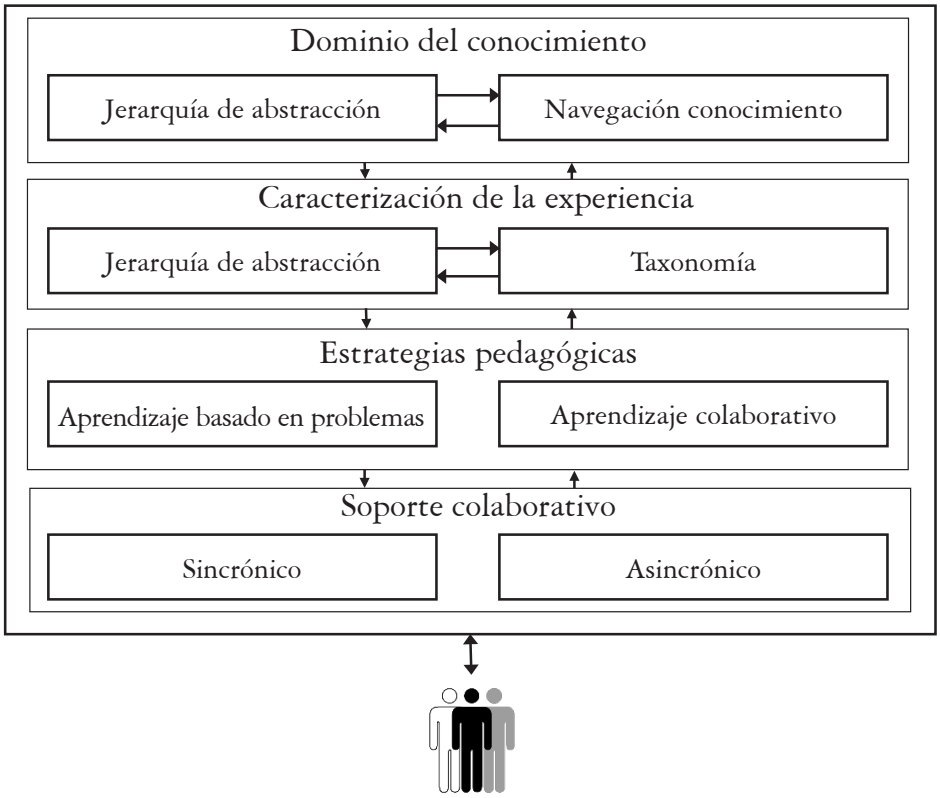


FIGURA 1. ARQUITECTURA GENÉRICA DE UN CSCL.

Fuente: Dorneich (2002).

Por otro lado, Núcleo es un sistema de CSCL que contiene un modelo de adaptación del aprendizaje (figura 2). El enfoque pedagógico de Núcleo se fundamenta en dos paradigmas constructivistas: el aprendizaje colaborativo y el basado en problemas. El funcionamiento se caracteriza en el juego de roles, donde cada uno de los estudiantes es un avatar que desempeña una tarea específica dentro de la misión conjunta.

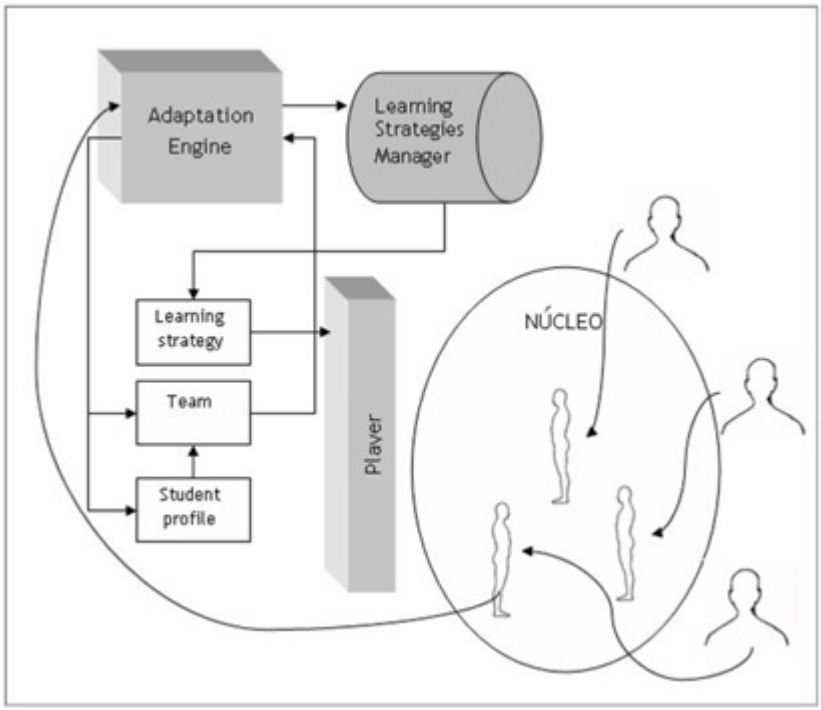


FIGURA 2. ARQUITECTURA DEL CSCL NÚCLEO.

Fuente: Surthers (2001).

La adaptación del usuario en el ambiente es caracterizada por la habilidad del sistema para personalizar la experiencia de aprendizaje para las diferentes condiciones individuales del juego (Sancho *et al*., 2008).

5. Herramientas generales

En la figura 3 se observan diferentes herramientas que pueden ser utilizadas en los ambientes de CSCL. Estas herramientas están divididas en dos grupos principales: herramientas sincrónicas y herramientas asincrónicas.

5.1. *Herramientas asincrónicas*

A continuación, se describirá brevemente cada una de las herramientas asincrónicas:

- Correo electrónico (en inglés *e-mail*) es un servicio de red para permitir a los usuarios enviar y recibir mensajes. Esta comunicación se realiza de forma asincrónica, es decir, que el destinatario no debe estar conectado para recibir el mensaje.
- Foros: los foros son también conocidos como foros de mensajes o foros de discusión y son una aplicación web que le da soporte a discusiones en línea, en la que los usuarios pueden enviar y contestar mensajes que pueden ser leídos por otros usuarios.

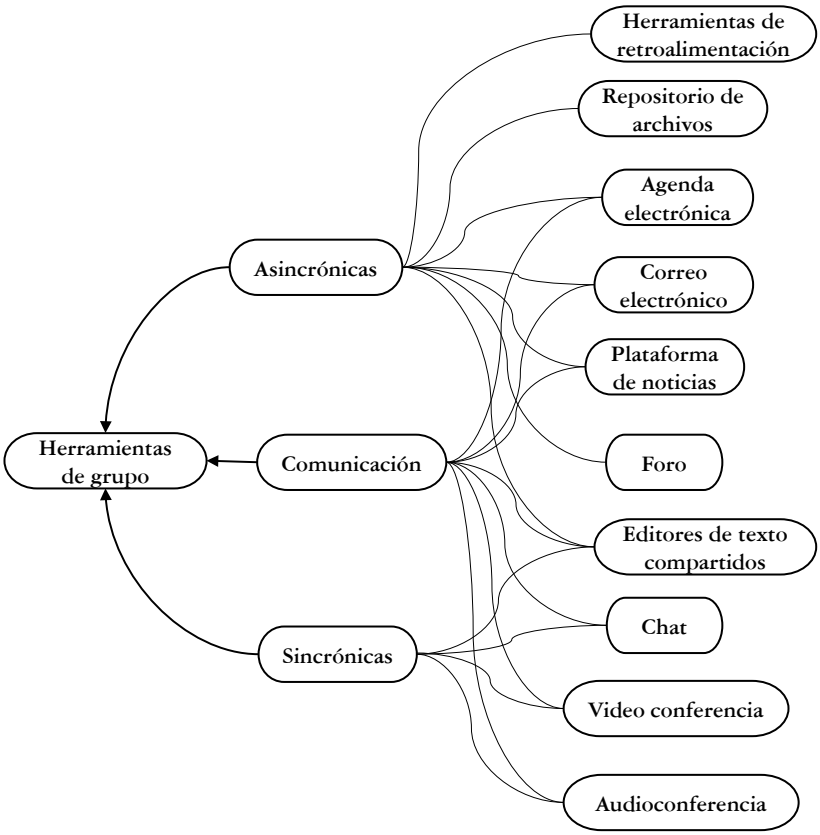


FIGURA 3. HERRAMIENTAS CSCL.

Fuente: Salinas (2000).

- Repositorio de archivos: es una herramienta con la que los usuarios pueden almacenar documentos, videos, imágenes, audio y todo tipo de archivos que puedan ser de utilidad a los compañeros de grupo y a futuros estudiantes.
- Editor de texto compartido: esta herramienta permite a estudiantes y al facilitador modificar un documento de texto: en él, los cambios que uno de los usuarios realice serán visibles a todos los usuarios.
- Plataforma de noticias: esta herramienta es un espacio en el sistema donde el facilitador publica noticias relevantes para el grupo de trabajo.
- Herramienta de retroalimentación: después de implementar la estrategia y haber realizado la actividad grupal se evaluará el resultado de esta; la herramienta de retroalimentación permitirá que realice esta evaluación.
- Agenda electrónica: en este espacio se encontrarán las fechas de las actividades programadas.

5.2 Herramientas sincrónicas

En esta sección se describirán brevemente cada una de las herramientas sincrónicas:

- Chat (charla en inglés): se usa para referirse a las distintas formas posibles de comunicarse en tiempo real con otras personas mediante mensajes escritos, bien sea empleando utilidades como los Internet Relay Chat (IRC), o bien mediante servicios de mensajería instantánea.
- Videoconferencia: la videoconferencia consiste en un servicio multimedia que permite a varios usuarios mantener una conversación a distancia en tiempo real con interacción visual, auditiva y verbal.
- Audioconferencia: la audioconferencia consiste en un servicio multimedia que permite a varios usuarios mantener una conversación de voz a distancia en tiempo real.

6. Métodos para la conformación de grupos-equipos

Una de las tareas más importantes, pero muchas veces subvaloradas dentro del aprendizaje colaborativo, es la conformación de los grupos-equipos de estudio. Básicamente las estrategias de conformación pueden agruparse en tres tipos: las basadas en escogencia aleatoria, las basadas en la autoconformación por parte de los estudiantes y las que emplean algún criterio de selección, sea para buscar grupos homogéneos o heterogéneos. Si bien las dos primeras son las más empleadas, dada su simplicidad y facilidad de implementación, es la tercera la que desde el

punto de vista pedagógico puede tener mayor interés. A continuación, se presentan algunos trabajos realizados por investigadores en el área.

En Ovalle (2009) se propone el uso del Test Revelador del Cociente Mental Triádico (TQDT, por sus siglas en inglés) como medio para caracterizar los estudiantes según sea su propensión a utilizar alguno de los hemisferios del cerebro, bajo la premisa de que dicha información es útil para la asignación de roles, como se muestra en la tabla 2. Una vez realizado el test los resultados son utilizados para conformar grupos de estudio; se emplea un algoritmo sencillo que busca que, en el mejor de los casos, se tenga por grupo cada uno de los tres cerebros dominantes, teniendo en cuenta la cantidad de estudiantes por grupo.

En DeLeón *et al*. (2009) se propone la conformación de grupos de estudio que sean interheterogéneos e intrahomogéneos, esto es, que al interior de cada grupo los integrantes sean diferentes, pero en promedio los grupos sean similares entre sí. Otra característica de este trabajo, y quizá la más importante, es que la medida de similaridad o disimilaridad no se toma respecto a un solo aspecto sino a varios, a diferencia de muchos trabajos en los que usualmente solo se considera el nivel académico.

TABLA 2. CARACTERÍSTICAS, ROLES Y RESPONSABILIDADES SEGÚN EL CEREBRO DOMINANTE

	Características	Rol	Responsabilidad
Cerebro dominante izquierdo	Verbal, numérico, analítico, lógico, descompositor, racional, abstracto, alerta, vigilante, articulador, crítico, investigador, visual, lineal.	Liderazgo, cronometrista, expositor.	Proponer temas, conceptuar, buscar información, presentar conclusiones, relatoría.
Cerebro dominante derecho	Proverbial, intuitivo, sintético, reintegrador, holístico, emocional, sensorial, espacial, espontáneo, libre, asociativo, artístico, contemplativo, auditivo, no lineal.	Visionario, prospector, creador.	Proyección, creatividad, recursividad, priorización de alternativas.
Cerebro dominante central	Instintivo, motor, concreto, agresivo para la supervivencia y la reproducción, trabajador, negociante, apropiador, planificador, económico, político, comerciante, administrador y regulador, ecosistémico.	Propositor, relator, *feedbacker*.	Concreta los proyectos, revisa y controla el cronograma, digitación y corrección de los trabajos escritos. *Feedback* y propuestas de mejoramiento.

Fuente: Ovalle *et al*. (2009).

En particular, en este trabajo se propone el uso de tres *test* de aptitud que dan cuenta de algunas de las aptitudes necesarias para llevar a cabo satisfactoriamente el aprendizaje colaborativo en un curso específico. Una vez conocidos los resultados de dichos *tests* para todos los estudiantes, se emplea un método de búsqueda heurística, más específicamente algoritmos genéticos para llevar a cabo la conformación de los grupos. Para organizar a los estudiantes en grupos, los profesores deben decidir sobre tres aspectos fundamentales: el tamaño de los equipos, la duración de los equipos y la forma de asignación de los estudiantes a los equipos (Johnson *et al*., 1991).

Los equipos pueden formarse al azar o por decisión de los estudiantes o del profesor. Los que han participado en actividades de AC concuerdan en que los equipos más efectivos son heterogéneos y formados por el profesor y no por los mismos estudiantes. Algunos profesores que han aplicado con éxito el AC, piden a sus estudiantes llenar cuestionarios el primer día de clase. A través de los cuestionarios se puede obtener información útil, por ejemplo: sexo, promedio de calificaciones, experiencia en alguna área de estudio, habilidades más relevantes, características más débiles. Estos cuestionarios pueden ayudar a los profesores a formar grupos con balance, variedad y compatibilidad.

7. Sistemas de CSCL

En Arteaga y Fabregat (2002) se muestra un modelo basado en Sistemas Hipermedia Adaptativos integrados con Ambientes Colaborativos de Aprendizaje, llamado SHACA. Este sistema tiene como principal fortaleza que provee adaptatividad, tanto en la parte individual como en la colaborativa del estudiante. Su limitación consiste en que no tiene un método para conformar grupos de aprendices de forma eficiente, teniendo en cuenta las características colaborativas de cada estudiante, además no implementa estrategias para realizar un plan instruccional en el proceso de aprendizaje.

El sistema Synergeia, como lo describe Stahl (2002) está diseñado para apoyar la construcción colaborativa de conocimientos en las aulas de las escuelas. Proporciona una visión común, estructurada, basada en la web de tal forma que es un espacio de trabajo colaborativo en el que el aprendizaje puede tener lugar, los documentos y las ideas pueden ser compartidas, los debates se pueden almacenar, los artefactos y los conocimientos pueden ser desarrollados y presentados. Este sistema basado en BSCW (*Basic Support Cooperative Work*) se presenta como una herramienta que implementa el "trabajo cooperativo", lo cual se debe diferenciar de

nuestro enfoque de "aprendizaje colaborativo" para lograr el aprendizaje individual a partir de la colaboración del grupo.

En Mora *et al*. (2004) se propone un entorno colaborativo basado en las técnicas de CSCL para el aprendizaje de fenómenos físicos llamado COLAB (*Collaborative Laboratories*). Es básicamente un entorno de simulación en el que se presentan diferentes espacios por los que los estudiantes se pueden desplazar para ver diferentes fenómenos físicos. Provee herramientas tanto asíncronas como síncronas para la comunicación y la colaboración. Sin embargo, este sistema carece de una forma de monitoreo del avance en el conocimiento de los estudiantes, así como no se plantea ninguna estrategia para llevar a cabo todo el proceso. Finalmente, no tiene una manera de distribuir los estudiantes en los grupos teniendo en cuenta sus características individuales y grupales.

Otro ejemplo de entorno educativo donde se integra un Ambiente Colaborativo de Aprendizaje a uno Individualizado se propone en Jiménez (2006). El modelo propuesto llamado ALLEGRO, es un STI (Sistema Tutorial Inteligente) que utiliza un (Sistema Multi-Agente) SMA pedagógico y la técnica de Razonamiento Basado en Casos (CBR) para cambiar la planificación de acuerdo con el conocimiento que va adquiriendo el estudiante. El ambiente colaborativo que se integra con el STI carece de estrategias para guiar el trabajo de aprendizaje grupal; adicionalmente, la conformación de grupos y la evaluación grupal son realizadas casi en su totalidad por el profesor sin intervención del sistema.

Una aplicación de los sistemas de enseñanza y aprendizaje colaborativos, llamada DomoSim-TPC, es la que se propone en Bravo *et al*. (2006), la cual tiene como objetivo que los estudiantes aprendan el diseño domótico a través de la interacción sincrónica. Este sistema tiene como limitación que no implementa estrategias para llevar a cabo el proceso de aprendizaje, además de no tener mecanismos para hacer un seguimiento del progreso del estudiante diferente al profesor. También se puede agregar que no hay una manera de armar grupos de acuerdo con las características individuales de cada estudiante.

ESCOLE (*Environment for Supporting Collective Learning Enthusiasts*) (Lonchamp, 2006) es una plataforma que proporciona servicios básicos de cooperación, instrumentos y herramientas de CSCL accesibles desde diferentes lugares de trabajo. El conjunto de herramientas básicas de cooperación incluye herramientas de producción (editores compartidos), herramientas de comunicación (correo electrónico, noticias, foros, chat, wiki), y la coordinación de herramientas (calendario compartido, servicio de votación).

Un modelo colaborativo se propone en Ruiz *et al*. (2007) para implementar un plan de estudios en la educación superior española de acuerdo con los nuevos lineamientos que se debían cumplir para el año 2010. El modelo implementa únicamente la estrategia colaborativa JIGSAW para el trabajo grupal y no especifica un mecanismo para hacer un seguimiento del progreso de los estudiantes ni de una forma de evaluación que sea independiente del profesor.

El sistema gStudy (Winne *et al*., 2007) permite a un alumno o un instructor crear o importar contenidos sobre casi cualquier tema. La información acerca de los temas se hace utilizando el lenguaje HTML, el cual puede incluir texto, gráficos, fotos, gráficos, tablas, audio y clips de video,es decir, los formatos de información común para la biblioteca de recursos impresos y en internet. El sistema gStudy proporciona las herramientas cognitivas de los alumnos para crear, compartir, intercambiar información y objetos.

Tanto en el proyecto ESCOLE como en gStudy no se observa claramente cómo el hecho de simplemente proporcionar herramientas colaborativas logre los resultados de aprendizaje de los individuos. Esto se puede referir mejor a trabajo cooperativo, además de no implementar estrategias de aprendizaje definidas para los individuos. Aunque logran resultados a partir del aprendizaje colaborativo, Synergeia, ESCOLE y gStudy carecen de mecanismos específicos para la conformación de los grupos que se requieren para el apoyo de estrategias de aprendizaje colaborativo.

I-MINDS (Soh, 2008) es un sistema multiagente para el Aprendizaje Colaborativo y la administración de aulas de clase. Este sistema está compuesto por un SMA, herramientas colaborativas e implementa la estrategia de aprendizaje JIGSAW. Además, uno de los agentes que compone el sistema, El Agente Grupal, se encarga de construir perfiles de los alumnos, basado en el comportamiento que estos tienen al interior del sistema (con quien se comunican, tipos de preguntas, etc.); perfil que posteriormente utiliza para ejecutar la conformación de grupos.

El trabajo anterior tiene varias similitudes con el proyecto CIA (Cursos Virtuales Inteligentes Adaptativos) (Ovalle, 2008) que se ha desarrollado en la Universidad Nacional de Colombia - Sede Medellín en cuanto al enfoque a partir de SMA, CSCL y estrategias de aprendizaje colaborativo. Sin embargo, difieren en que en el proyecto CIA se implementa, además de la estrategia JIGSAW, otra estrategia de aprendizaje (técnica de cuestionamiento por pares) y un enfoque que se utiliza para la conformación de grupos, el cual se basa en el test RCMT. Adicionalmente, los modelos de conocimientos utilizados (dominio, tutor, alumno) tienen en cuenta las dos perspectivas; individualizada y colaborativa.

El trabajo presentado en Ferrarini *et al*. (2008) propone la metodología *LuCoA* (Sitema de Aprendizaje Lúdico-Colaborativo) para el aprendizaje de contenidos a través de la generación de actividades colaborativas para los alumnos de educación superior. Dicha metodología es una combinación de técnicas colaborativas como el JIGSAW, compartir pensamiento en pares, aprendizaje basado en problemas, *Talking Chips* y *Cooperative Controversy*, ya reconocidas por su aplicación para el aprendizaje. *LuCoA* cuenta con 4 etapas, de las cuales la primera está orientada al trabajo individual y las tres subsiguientes al trabajo grupal. Se espera que estas puedan evitar los problemas que pudieran suscitarse durante un proceso de aprendizaje colaborativo, tales como *free-riding*, rigidez, distribución de asimetrías de participaciones, entre otros.

Para poder validarla, se ha generado una herramienta computacional *LuCoAS* que la soporta. *LuCoAS* contempla los diferentes roles: el rol del tutor como observador y guía; y el rol de los alumnos como participantes activos en su proceso de construcción del conocimiento.

En Ovalle *et al*. (2009) el sistema de gestión de conocimiento *KnowCat* diseñado y desarrollado en la Universidad Autónoma de Madrid, España, es utilizado para realizar el seguimiento y evaluación de actividades colaborativas a través de estudios experimentales, realizados en el aula de clase, en cursos de pregrado en ingeniería de la Universidad Nacional de Colombia - Sede Medellín y de la Universidad del Cauca, en Popayán. *KnowCat*, acrónimo de *Knowledge Catalyser* o "catalizador de conocimiento", es un sistema distribuido para la Gestión del Conocimiento que permite la creación incremental de conocimiento estructurado. La motivación subyacente es permitir la generación de materiales educativos de alta calidad como resultado de la interacción de los estudiantes con los propios materiales. *KnowCat* se considera un sistema *groupware* de trabajo asíncrono por proporcionar un entorno de colaboración sobre la web, basado en el concepto de "cristalización de conocimiento" soportado por comunidades virtuales de expertos. Permite la construcción de sitios de conocimiento a los que se accede vía web, denominados nodos *KnowCat*, los cuales organizan el conocimiento sobre un área o tema a través de una estructura en árbol, donde cada una de sus ramas contiene un conjunto de documentos que describen un tema específico. Cada documento puede recibir anotaciones que expresan opiniones, explicaciones, comentarios y sugerencias acerca de su contenido. En un momento dado, todos los documentos ubicados en una misma rama compiten entre sí para obtener el título de mejor descripción del tema, el cual lo obtiene el documento con mayor aceptación social. La aceptación social de los documentos es calculada por el mecanismo de cristalización de conocimiento que posee el sistema, el cual tiene en

cuenta todas las interacciones que reciben los documentos tanto explícitas (votos y anotaciones) como implícitas (accesos) y cómo han sido dichas interacciones a lo largo del tiempo.

Los autores proponen entonces una guía metodológica para la utilización del sistema *Knowcat*. La guía que se propone está compuesta por 10 etapas, que involucran aspectos fundamentales sobre la generación colaborativa de conocimiento tales como: estructuración del árbol de conocimientos, conformación de grupos, votación, valoración y anotación de informes individuales, votación de mejores documentos, producción de documentos grupales, realización de encuestas y análisis de resultados. Finalmente, se presentan los resultados obtenidos en las experimentaciones con algunos grupos trabajando con la versión de *KnowCat* que proporciona los servicios de consciencia de grupo (*Awareness*) y otros utilizando la versión tradicional de *KnowCat*, que no brinda dichos servicios. Se ha elegido este diseño para el estudio con el fin de evaluar el desempeño de los mecanismos de consciencia de grupo en procesos de aprendizaje colaborativo asistido por computador.

8. Monitoreo y evaluación de sistemas de CSCL

En los últimos años, a través de una gran cantidad de estudios científicos se ha demostrado el alto grado de aprendizaje que es posible lograr cuando los estudiantes colaboran en sus tareas de aprendizaje. Sin embargo, falta realizar investigación en torno al impacto de la tecnología en los sistemas de evaluación y monitoreo del proceso de colaboración.

8.1. *Metodología para evaluación y monitoreo en ambientes de CSCL*

En la tesis de doctorado de Collazos (2003) se propone el desarrollo de una plataforma (*framework*) que permita monitorear y evaluar el proceso de colaboración, tanto por parte de los estudiantes como por parte de los profesores dentro de una actividad grupal. Lo anterior se busca con el fin de entender el proceso de colaboración que se genera al momento de interactuar por parte de un grupo de trabajo, para posteriormente poder evaluar computacionalmente y mejorar este proceso.

8.2. *Evaluando y monitoreando actividades colaborativas en dispositivos móviles*

En el trabajo de Collazos *et al*. (2008) se expone un modelo que permite obtener la información necesaria para evaluar correctamente el desarrollo de una actividad

colaborativa trabajando con dispositivos móviles. Los autores señalan que es necesario tener en cuenta el contexto en el que se desarrolla la actividad puesto que en todos los casos la forma de interactuar es distinta, ya sea que se haga cara a cara, a través de computadores en forma sincrónica o asincrónica, o a través de dispositivos móviles celulares.

Cabe señalar que, además de evaluarse los resultados finales de la actividad colaborativa, se debe tener en cuenta el proceso que fue utilizado por el grupo para cumplir los objetivos planteados al inicio de la actividad. Por esto, la ejecución de un monitoreo constante sobre la actividad durante su ejecución, basado en indicadores, permite obtener información de aquellos datos que en una actividad no colaborativa serían despreciados. Para ello los autores definen un nuevo indicador relacionado con la movilidad y la eficiencia de recursos, debido al hecho de manejar dispositivos móviles. Una limitación del modelo propuesto es que no considera actividades colaborativas sincrónicas, las cuales serán incluidas en la experimentación que se realizará como trabajo futuro.

8.3. *Thinklets: un artefacto útil para evaluar la usabilidad colaborativa*

La evaluación de la usabilidad es una característica fundamental al momento de realizar monitoreo y evaluación de actividades colaborativas debido a que sus resultados permiten determinar qué mejoras o correcciones deben realizarse. Por lo anterior en Méndez *et al*. (2008) se propone el diseño de métodos de evaluación de la usabilidad haciendo uso de *thinklets* y la metodología de desarrollo de procesos colaborativos. Los *thinklets* se definen como técnicas facilitadoras o "patrones de colaboración", los cuales son predecibles, repetibles y transferibles ayudando a un grupo a alcanzar los objetivos acordados. Se han identificado así algunos patrones de colaboración tales como generación, reducción, clarificación, organización, evaluación, construcción del consenso entre otros.

8.4. *Servicios de* Awareness *para la mejora del escenario educativo soportado sobre* KnowCat

Con el propósito de monitorear y evaluar constantemente las actividades colaborativas realizadas por un grupo de trabajo se implementan los Servicios de *Awareness* (SAw). Dichos SAw, dentro de un sistema *groupware*, crean consciencia del trabajo que realizan los demás miembros del grupo en cuanto a participación, aportes, historial de colaboración, usuarios registrados y en línea en un momento dado, etc. Se ha demostrado que dicha consciencia de grupo ayuda a mejorar el desempeño del

grupo en el desarrollo de actividades colaborativas. En el artículo propuesto por Claros *et al*. (2009) se presentan los principales SAw que han sido implementados en el sistema de gestión del conocimiento *KnowCat*, tales como: usuarios registrados, usuarios en línea, vista histórica, vista radar, participómetro y mapa de interacción. Se concluye por los autores que la utilización de estos SAw ha dado como resultado una mayor interacción entre los individuos, acompañado de un mayor grado de satisfacción de los productos realizados por el grupo, al igual que una mejora del escenario educativo soportado sobre el sistema *KnowCat*.

9. Conclusiones

La enseñanza y el aprendizaje colaborativos se fundamentan en la idea que el aprendizaje es un acto social natural en el cual los participantes interactúan entre sí, y es mediante esta interacción que el aprendizaje se obtiene. En efecto, en el aprendizaje colaborativo los estudiantes tienen la oportunidad de conversar, proponer y defender ideas con sus compañeros, pero adicionalmente, intercambiar diversas creencias y puntos de vista y de esta forma estar involucrado activamente.

El principal objetivo de implementar en el aula de clase estrategias de aprendizaje colaborativo es estimular en los equipos conformados la discusión, el análisis crítico sobre un tema específico, el desarrollo de habilidades individuales y sociales, el logro de metas de tipo académico, el mejoramiento de las relaciones interpersonales, la construcción de conocimiento –donde cada estudiante actua como participante activo–, la comunicación y la negociación, entre otros.

Finalmente, en este capítulo se presentaron las generalidades de la enseñanza y aprendizaje colaborativos, los ambientes Colaborativos de Aprendizaje Asistidos por Computador (CSCL), las herramientas utilizadas para construir sistemas de CSCL, así como el monitoreo y evaluación de tales sistemas.

Referencias

Arteaga, C. y Fabregat, R. (2002). Integración del aprendizaje individual y del colaborativo en un sistema hipermedia adaptativo. Congreso Iberoaméricano Informática Educativa, Vigo.

Bravo, C., Redondo, M., Ortega, M. y Verdejo, M. (2006). Collaborative Environments for the Learning of Design: A Model and a Case Study in Domotics. *Computers & Education*, *46*, 152-173.

Calzadilla, M. (2000). *Aprendizaje colaborativo y tecnologías de la información y la comunicación*. Venezuela: Universidad Pedagógica Experimental Libertador.

Claros, I., Collazos, C. y Cobos, R. (2009). Servicios de awareness para la mejora del escenario educativo soportado sobre KnowCat. *Revista Avances en Sistemas e Informática*, *6*(1), 127-133.

Collazos, C., Alvira, J., Martínez, D., Jiménez, J., Cobos, R. y Moreno, J. (2008). Evaluando y monitoreando actividades colaborativas en dispositivos móviles. *Revista Avances en Sistemas e Informática*, 5(1), 82-93.

Collazos, O. y César, A. (2003). *Una metodología para el apoyo computacional de la evaluación y monitoreo en ambientes de aprendizaje cooperativo* (Tesis doctoral). Universidad de Chile.

DeLeón, A., Gómez, S. y Moreno, J. (2009). Uso de tests de aptitud y algoritmos genéticos para la conformación de grupos en ambientes colaborativos de aprendizaje. *Revista Avances en Sistemas e Informática*.

Domingo, J., Llumà, J., Manzanares, M., Ruiz, C. y Camps, G. (2004). Estrategias para el trabajo con grupos cooperativos. Grupo EEEI (EcoEnginys Escola Industrial).

Dorneich, M. (2002). A System Design Framework-Driven Implementation of a Learning Collaboratory. *IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans*, *32*(2).

Ellis, C., Gibbs, S. y Rein, G. (1991). Groupware Some Issues and Experiences. En *Communications of the ACM*, *34*(1).

Ferrarini, C. y Collazos, C. A. (2008). Metodología LuCoA para el aprendizaje lúdico-colaborativo en la educación superior. *Revista Avances en Sistemas e Informática*, *5*(1), 1-8.

Gros, B. (2000). *El ordenador invisible*. Barcelona: Gedisa.

Guitert, M. y Siménez, F. (2000). Aprender a colaborar. En Campiglio, A. y Rizzi, R. (Eds.), *Cooperar en clase. Ideas e instrumentos para trabajar en el aula*. Madrid: Publicaciones del MCEP.

Jiménez, J. (2006). *Un modelo de planificación instruccional usando RBC en SMA para entornos integrados de STI y CSCL* (Tesis de Doctorado en Ingeniería) Medellín: Universidad Nacional de Colombia - Sede Medellín.

Johnson, D. W., Johnson, R. T. y Smith, K. A. (1991). Cooperative Learning: Increasing College Faculty Instructional Productivity. *ASHE-ERIC Higher Education Report*, *4*.

Lonchamp, J. (2006). A Platform for CSCL Practice and Dissemination. Advanced Learning Technologies. Sixth International Conference, 66-70.

Lucero, M. (2004). Entre el trabajo colaborativo y el aprendizaje colaborativo, *Revista Iberoamericana de Educación*.

Maldonado, M. I. (2007). El trabajo colaborativo en el aula universitaria. *Laurus, Revista en Educación*, *13*(23), 263-278.

Méndez, Y., Jiménez, J., Collazos, C., Granollers, T., González, M. (2008). Thinklets: un artefacto útil para el diseño de métodos de evaluación de la usabilidad colaborativa. *Revista Avances en Sistemas e Informática*, 5(2), 147-154.

Mora, G., Gómez, S., Martínez, C. y Martínez, G. (2004). COLAB: una plataforma para la simulación en entornos colaborativos en laboratorios virtuales. *Revista Iberoamericana de Inteligencia Artificial*, *24*, 45-53.

Ovalle, D., Jiménez, J. y Moreno, J. (2009). *Group Conformation Method Using TQD Test as a Support for Collaborative Learning Strategies Development*. Proceedings of 9th World Conference on Computers in Education – WCCE. Bento Gonçalves, Brasil.

Ovalle, D., Jiménez, J., Collazos, C., Claros, I., Pantoja, L., Cobos, R., Moreno, J., Pifarré, M. y Argelagos, E. (2009). Guía metodológica para el seguimiento y evaluación de aprendizaje colaborativo asistido por el sistema KNOWCAT. Congreso ACOFI.

Ovalle, D., Jiménez, J., Moreno, J., Arias, F., Jiménez, M. y Salazar, C. (2008). *Modelo de sistema Multi-agente de cursos adaptativos integrados con ambientes colaborativos de aprendizaje*. Medellín: Universidad Nacional de Colombia - Sede Medellín.

Panitz, T. y Panitz, P. (1998). *Encouraging the Use of Collaborative Learning in Higher Education*. Nueva York: Garland Publishing.

Ruiz, I., Jorrín, I. M. y Villagrá, S. L. (2007). Análisis de competencias en un entorno CSCL: aportaciones de una experiencia utilizando un JIGSAW. *Revista Latinoamericana de Tecnología Educativa*, *6*(2), 29-40.

Salinas, J. (2000). El aprendizaje colaborativo con los nuevos canales de comunicación. En Cabero, J. (Ed.), *Nuevas tecnologías aplicadas a la educación*. Madrid: Síntesis.

Sancho, P., Fuentes, R. y Fernández, B. (2008). NUCLEO: *Adaptive Computer Supported Collaborative Learning in a Role Game Based Scenario*. Eighth IEEE International Conference on Advanced Learning Technologies.

Soh, L. K., Khandaker, N. y Jiang, H. (2008). I-MINDS: A Multiagent System for Intelligent Computer-Supported Collaborative Learning and Classroom Management. *18*, 119-151.

Stahl, G. (2002). Groupware Goes to School. Design, Implementation, and Use. Proceedings of the 8th international workshop, CRIWG 2002, La Serena, Chile, September 1-4.

Surthers, D. (2001). Architectures for Computer Supported Collaborative Learning". En *Proceeding of the IEEE International Conference on Advanced Learning Technologies* – ICALT.

Winne, P., Hadwin, A. y Gress, C. (2007). The Learning Kit Project: Software Tools for Supporting and Researching Regulation of Collaborative Learning. Computers in Human Behavior.

3
SELECCIÓN DE ESTRATEGIAS EDUCATIVAS EN UN CONTEXTO DE APRENDIZAJE COLABORATIVO

Demetrio Arturo Ovalle Carranza, Julián Moreno Cadavid, Francisco Arias Sánchez

En este capítulo se caracterizan inicialmente las estrategias educativas más utilizadas en un contexto de Aprendizaje Colaborativo que permiten a grupos de personas llevar a cabo tareas colaborativas de forma planificada. Luego se propone un método para una selección de la estrategia más adecuada al problema que se quiere resolver y que se ajusta al grupo conformado. El método establece inicialmente una serie de criterios desde tres dimensiones o enfoques: interactividad, educacional y adaptatividad, los cuales son evaluados con el fin de obtener las estrategias más relevantes de acuerdo con las dimensiones establecidas. Los resultados obtenidos al aplicar el método para un caso de estudio arrojaron una clasificación de estrategias conveniente y esencial para el correcto desarrollo de los procesos de enseñanza y aprendizaje.

1. Introducción

El aprendizaje colaborativo se basa en la idea de que el aprendizaje es un acto social natural en el cual los participantes interactúan entre sí, y que es a través de la interacción que el aprendizaje se obtiene. En este enfoque grupal se busca que las habilidades, destrezas y conocimientos de cada uno de los miembros de los equipos conformados (homogéneos u heterogéneos) se complementen para lograr una meta común (Moreno *et al*., 2012). Los participantes de cada equipo asumen roles y responsabilidades definidos dentro de la estrategia colaborativa seleccionada, los cuales garantizan el éxito del aprendizaje colaborativo.

Hoy en día existe gran variedad de estrategias y procesos que permiten a los estudiantes llevar a cabo una tarea o actividad de forma colaborativa, pero en muchas

ocasiones la selección de las estrategias más apropiadas para impartir conocimiento se convierte en una tarea compleja debido a la variedad de estrategias disponibles y a la gran cantidad de criterios que se deben tener en cuenta para llevar a cabo dicha selección de forma apropiada. En este capítulo se propone un método para la selección de estrategias de interacción en ambientes virtuales colaborativos de aprendizaje. Adicionalmente, se presenta un caso de estudio, en el cual se seleccionan las estrategias más relevantes según diferentes criterios.

2. Marco teórico

Para lograr aprendizaje colaborativo en los estudiantes, se requiere de la aplicación de estrategias de aprendizaje (Domingo *et al*., 2004; Domingo, 2010). Dichas estrategias de aprendizaje se refieren al conjunto de actividades o procesos que se planifican de acuerdo con las necesidades de los estudiantes, los objetivos que se desean lograr y la naturaleza de los conocimientos, con la finalidad de hacer efectivo el proceso de enseñanza-aprendizaje. En la tabla 1 se describen diversas estrategias colaborativas con su respectivo protocolo de planificación de actividades colaborativas.

TABLA 1. ESTRATEGIAS COLABORATIVAS

Nombre de la estrategia en español	Metodología
Resolución de problemas	1. Agrupar a los estudiantes
	2. En cada grupo se evalúa el problema, identificando y caracterizando los componentes con los que estén familiarizados
	3. Identificar lo que conocen y lo que desconocen sobre el problema
	4. Asignar tareas y tiempos en cada grupo
	5. Compartir en cada grupo el conocimiento que cada alumno ha adquirido
Resolución estructurada de problemas	1. Realizar una exposición breve sobre el problema que se desea resolver y sobre ideas generales para la solución del problema

Nombre de la estrategia en español	Metodología
Resolución estructurada de problemas	2. Agrupar a los estudiantes (grupos de 4 personas)
	3. Asignar un problema y un tiempo para dar solución al problema
	4. Asignar a cada estudiante del grupo un número del 1 al 4
	5. El grupo busca una solución del problema, de tal manera que todos los miembros estén de acuerdo con la solución y puedan explicar la respuesta y la estrategia utilizada para resolver el problema
	6. Finalizado el tiempo especificado, el profesor anuncia el número (por ejemplo, 2) del alumno que representará al grupo y quien se encargará de hablar sobre la solución del problema
Método del descubrimiento	1. Agrupar a los estudiantes (grupos de 4 personas)
	2. Asignar un problema y un tiempo para dar solución al problema
	3. Asignar a cada estudiante del grupo un número del 1 al 4
	4. El grupo busca una solución del problema, de tal manera que todos los miembros estén de acuerdo con la solución y puedan explicar la respuesta y la estrategia utilizada para resolver el problema
	5. Finalizado el tiempo especificado, el profesor anuncia el número (por ejemplo, 2) del alumno que representará al grupo y quien se encargará de hablar sobre la solución del problema
Envío de problemas	1. El profesor o los alumnos definen un problema
	2. Agrupar a los estudiantes (grupos de 3 personas)
	3. El grupo 1 toma el problema y lo engancha en la parte exterior de una carpeta
	4. El grupo 1 genera varias soluciones del problema y las coloca en el interior de la carpeta
	5. Se entrega la carpeta al grupo 2. Este grupo puede ver el problema, pero no las soluciones
	6. El grupo 2 genera varias soluciones del problema y las coloca en el interior de la carpeta
	7. Se entrega la carpeta al grupo 3, que debe seleccionar las 2 mejores soluciones

Nombre de la estrategia en español	Metodología
Cuestionamiento por pares	1. Realice una exposición sobre un tema específico (10 a 15 minutos)
	2. Entregue a los alumnos un conjunto de esquemas de preguntas, tales como: • ¿Cuál es la idea central de? • ¿Qué pasa si? • ¿Cómo afecta a? • ¿Cuál es otro ejemplo de? • Explique por qué • Explique cómo • ¿Cómo se relaciona ___ con lo que aprendí antes? • ¿Qué conclusiones puedo sacar acerca de? • ¿Cuál es la diferencia entre____ y _____? • ¿En qué se asemejan____ y _____? • ¿Cómo puedo emplear ____ para____? • ¿Cuáles son las fuerzas y debilidades de? • ¿Cuál es el mejor ___ y por qué?
	3. Los estudiantes trabajan individualmente para escribir sus preguntas basados en el material que se ha cubierto. Los alumnos pueden no ser capaces de responder a preguntas que ellos plantean
	4. Agrupados en pequeños grupos, los estudiantes someten en forma individual una pregunta a discusión
Parejas de revisión de ejercicios	1. Se agrupan los 4 estudiantes en parejas
	2. Se asigna a cada pareja 2 problemas
	3. A cada estudiante del grupo se le asigna un rol de explicador (persona encargada de describir paso a paso cómo solucionar el problema) y un rol de verificador (persona encargada de verificar la solución del problema)
	4. Cambiar los roles para el otro problema
	5. Agrupar a los 4 estudiantes para que revisen las soluciones propuestas
Parejas de resolución de problemas de alta comprensión	1. Se agrupa a los estudiantes en parejas
	2. Se asignan roles de solucionador y de oyente a los estudiantes
	3. El solucionador lee en voz alta el problema y habla acerca de la solución del problema
	4. El oyente escucha atentamente cada paso para la solución del problema e identifica los posibles errores. El oyente puede realizar preguntas si el proceso para la solución del problema es confuso

Nombre de la estrategia en español	Metodología
Cuestionario recíproco guiado	1. El profesor imparte una miniconferencia sobre la temática y da una serie de preguntas abiertas tales como: ¿Cuál es la idea principal de _____?, ¿Qué pasa si _____?, ¿Cuáles son las fortalezas y debilidades de _____?, etc.
	2. En forma individual los estudiantes, a partir de las preguntas genéricas, preparan otras preguntas más concretas que no están en obligación de contestar inmediatamente
	3. Se forman grupos de estudiantes
	4. Los estudiantes discuten las preguntas elaboradas y discuten las posibles respuestas
Mesa redonda	1. El profesor expone una pregunta que pueda tener diferentes respuestas correctas
	2. En pequeños grupos los estudiantes toman turnos para exponer sus respuestas. El primer estudiante expone la suya y se discute en el grupo. Luego el estudiante de su izquierda (o derecha) hace lo mismo y así sucesivamente hasta que se acabe el tiempo
	3. Los resultados grupales se discuten al final entre todos
Rompecabezas (*Puzzle* o JIGSAW)	1. Divida a los estudiantes en grupos de 6 personas
	2. Nombre a un alumno del grupo como el líder
	3. Divida el tema en secciones (cantidad de secciones iguales a la cantidad de estudiantes en el grupo)
	4. Asigne a cada estudiante una sección
	5. Dé tiempo a los estudiantes para que lean su sección
	6. Agrupe a los estudiantes que poseen la misma sección (grupos de expertos), esto se hace con el fin de que discutan los puntos principales de su sección y de que generen una presentación que harán a su grupo
	7. Solicite a los estudiantes que retornen a los grupos originales
	8. Pida a cada estudiante que presente su sección al grupo
	9. Pase periódicamente por cada grupo, observando el proceso y resolviendo conflictos
	10. Entregue un examen a los estudiantes

Nombre de la estrategia en español	Metodología
Rompecabezas entre grupos (*Puzzle* o JIGSAW entre grupos)	1. Divida a los estudiantes en grupos de 6 personas
	2. En cada grupo agrupe a los estudiantes en parejas
	3. Divida el tema en secciones (cantidad de secciones iguales a la cantidad de parejas en el grupo)
	4. Asigne a cada pareja una sección
	5. Dé tiempo a las parejas para que lean su sección
	6. Agrupe a las parejas con la misma sección (grupos de expertos), esto con el fin de que discutan los puntos principales de su sección y de que generen una presentación que harán a su grupo
	7. Solicite a los estudiantes que retornen a los grupos originales
	8. Pida a cada pareja que presente su sección al grupo
	9. Pase periódicamente por cada grupo, observando el proceso y resolviendo conflictos
	10. Entregue un examen a los estudiantes
Rompecabezas II (*Puzzle* II o JIGSAW II)	1. Divida a los estudiantes en grupos de 6 personas
	2. Nombre a un alumno del grupo como el líder
	3. Divida el tema en secciones (cantidad de secciones iguales a la cantidad de estudiantes en el grupo)
	4. Asigne a cada estudiante una sección y el documento. Sin importar la sección asignada, los estudiantes deben leer todo el documento
	5. Dé tiempo a los estudiantes para que lean el documento
	6. Agrupe a los estudiantes que poseen la misma sección (grupos de expertos), esto se hace con el fin de que discutan los puntos principales de su sección y que generen una presentación que harán a su grupo
	7. Solicite a los estudiantes que retornen a los grupos originales
	8. Pida a cada estudiante que presente su sección al grupo
	9. Pase periódicamente por cada grupo, observando el proceso y resolviendo conflictos
	10. Entregue un examen a los estudiantes
Grupos de investigación	1. Los estudiantes seleccionan un tema para investigar
	2. Los estudiantes dividen el tema en partes más pequeñas
	3. Cada estudiante selecciona una parte del tema
	4. Cada estudiante investiga
	5. Los estudiantes se reúnen en grupo y comparten su información
	6. Los estudiantes sintetizan la información para producir un producto final
	7. Cada miembro del grupo participa en la presentación en clase

Nombre de la estrategia en español	Metodología
Controversia estructurada	1. El profesor selecciona un tema con dos puntos de vista
	2. Los estudiantes forman grupos de 2 y cada uno adopta un punto de vista
	3. Si el profesor lo aprueba, los estudiantes de diferentes grupos, pero con la misma posición, pueden comparar sus ideas una vez se hayan familiarizado con su posición
	4. Cada uno de los grupos iniciales realiza una breve presentación al resto del grupo
	5. Los estudiantes toman nota de las presentaciones, pero se les prohíbe realizar preguntas, estar en desacuerdo o debatir
	6. Los estudiantes se reúnen y discuten sus posiciones
	7. Con sus notas como guía realizan una nueva presentación
	8. Los estudiantes abandonan su posición de defensa y generaran un informe de consenso sobre la problemática planteada
Equipo con roles de aprendizaje estructurado	1. El profesor explica y asigna los roles
	2. Explicación y asignación del rol Líder: es responsable de mantener activo al grupo en la tarea asignada. Debe asegurarse de que todos los miembros del grupo tengan la oportunidad de participar, aprender y tener respeto por todos los miembros del equipo. Finalmente, debe cerciorarse de que los miembros del grupo dominan los conceptos aprendidos a partir de la actividad colaborativa
	3. Explicación y asignación del rol Líder Documentador: mantiene toda la documentación del grupo, como archivos digitales, carpetas, informes, tanto del grupo como de los otros grupos de trabajo. También mantiene el registro de las actividades realizadas por el grupo. El Documentador escribe las soluciones a los problemas del grupo para ser usadas como notas de clase o para ser entregadas al profesor. Finalmente, puede también preparar las presentaciones cuando el grupo debe realizar una presentación oral
	4. Reportero: suministra respuestas orales a la clase sobre las actividades grupales realizadas o resultados obtenidos y conclusiones generadas
	5. Monitor: coordina que el área de trabajo para el grupo esté disponible y controla el tiempo del cronograma de actividades colaborativas
	6. Comodín (*Wildcard*): colabora y asiste al líder del grupo y asume el rol del miembro que falta en la actividad que se va a realizar
Aprender escribiendo	1. El profesor plantea una problemática
	2. Se arman grupos heterogéneos para escribir sobre la dicha problemática
	3. El profesor evalúa los ensayos obtenidos y califica

Nombre de la estrategia en español	Metodología
STAD	1. El profesor presenta un nuevo concepto
	2. Se arman grupos para estudiar y cómo aplicar el concepto
	3. Se evalúa a los estudiantes sobre los nuevos conceptos aprendidos
	4. Se reconocen los grupos ganadores (con base en lo que han mejorado con respecto a la última prueba)
Torneos entre grupos	**Fase 1 (pretorneo)**
	1. Presentación de un contenido específico
	2. Formación de grupos de 4 a 5 alumnos, con adecuados niveles de heterogeneidad
	3. Definir una lista de preguntas sobre el contenido y numerarlas. Dar la lista de preguntas a un miembro de cada grupo
	4. En cada grupo se comienzan rondas de preguntas de la siguiente manera: un participante selecciona un número (no repetido) entre 1 y el número de preguntas. A ese estudiante se le hace esa pregunta, si la responde bien recibe un punto y si no, otro integrante del grupo puede "robarse" el punto respondiendo correctamente. Se pasa al siguiente estudiante y así sucesivamente hasta que se acaben las preguntas
	5. Los puntos de cada estudiante se suman y se determina quién de cada grupo es el primero, quién el segundo, etc.
	Fase 2 (torneo)
	6. Asignación de los alumnos a mesas de torneo de 4 a 5 miembros cada mesa. La asignación se realiza colocando en la mesa 1 a los estudiantes que obtuvieron el puesto 1 en sus respectivos grupos en el pretorneo, en la mesa 2 a los del puesto 2, y así sucesivamente
	7. Para cada mesa se realiza un torneo con características similares al pretorneo y los puntos obtenidos por cada estudiante se suman a su equipo correspondiente
Entrevista en 3 pasos	1. Organizar los participantes en grupos de 3
	2. Asignar a cada participante una letra y un rol. Por ejemplo, A = Entrevistador, B = Entrevistado, C = Reportero
	3. Rotar los roles después de cada entrevista
	4. Poner a los participantes de todos los grupos a compartir la información que recopilaron mientras tuvieron el rol C
Lista de interés	1. Se les solicita a los estudiantes de manera individual que creen una lista de frases o palabras que describan o definan algo específico, como por ejemplo: ¿Qué significa ambientalmente amigable?
	2. Los estudiantes organizan su listado
	3. Se conforman grupos pequeños para discutir los listados individuales. Opcionalmente se le solicita a cada grupo generar un listado común

Nombre de la estrategia en español	Metodología
Anotaciones contrapuestas	1. Los estudiantes se agrupan en parejas a las que se les asigna un artículo, capítulo o un contenido específico para su lectura y reflexión
	2. En esas parejas los estudiantes discuten los puntos clave y buscan las ideas individuales convergentes y divergentes
	3. Cada pareja elabora un documento resumiendo la discusión
Co-op Co-op	1. Se les centra a los estudiantes en la discusión de clase. Se diseñan experiencias iniciales, incluyendo discusiones de clase, para descubrir y estimular su curiosidad
	2. Se selecciona a los estudiantes para los grupos de aprendizaje, de forma que se maximice la heterogeneidad dentro de los equipos en variables como capacidad, sexo o pertenencia étnica
	3. Se construye el equipo: se hace de tal forma que se aumenten las habilidades de cooperación y de comunicación dentro del equipo
	4. Se selecciona el tema para el equipo: los estudiantes dividen la unidad de aprendizaje en temas, de forma que cada equipo se haga responsable de un aspecto de la unidad y el trabajo de cada equipo complemente el de los otros
	5. Selección de subtemas: como en el JIGSAW, cada estudiante se hace experto en un aspecto del tema; pero aquí (diferente al JIGSAW) son los estudiantes los que determinan cómo dividir el tema y los subtemas, y no por el profesor
	6. Preparación de los subtemas: los estudiantes individualmente recogen y organizan el material
	7. Presentación del subtema: como en el JIGSAW, cada estudiante presenta al grupo lo que ha aprendido del tema elegido. Una segunda ronda de presentaciones del subtema supone una oportunidad para responder a la discusión de grupo de cada minitema individual y su relación con todo el tema
	8. Preparación de las representaciones de los equipos: las presentaciones se hacen a toda la clase. Se prefieren presentaciones no leídas, tales como demostraciones, juego de roles, o el uso de medios audiovisuales
	9. Evaluación: se evalúan las presentaciones individuales hechas ante el grupo (generalmente por parte de los compañeros de equipo), de las presentaciones del grupo o equipo a toda la clase (generalmente por parte de los compañeros de clase), y de cada trabajo o proyecto individual de cada estudiante sobre su subtema (generalmente por parte del profesor)
Grupos de cuatro	1. Los estudiantes conforman grupos pequeños
	2. Se asignan y resuelven problemas complejos con poca guía del profesor
	3. Los estudiantes trabajan en busca de principios con material de ayuda específico al problema

Nombre de la estrategia en español	Metodología
Descubrimiento	1. Se forman grupos pequeños de trabajo
	2. Se presenta el material de apoyo a las actividades; las actividades están bien diferenciadas y presuponen diferentes habilidades
	3. El profesor asigna competencias a los estudiantes de bajo nivel para ayudar a equilibrar su interacción en el grupo
	4. Cada participante completa la actividad interactuando en su grupo
	5. El profesor busca las contribuciones valiosas de los estudiantes de bajo nivel y públicamente las comunica al grupo entero
Expectativas de equipo	1. Los grupos listan los comportamientos (expectativas) deseables de sus miembros dentro del grupo. La información se recoge de manera individual, en pares y grupal
	2. Se levanta una hoja de "expectativas"
	3. Las actividades del grupo se pueden evaluar contra la hoja de expectativas usando la autoevaluación, evaluación de pares o evaluación de grupo
	4. Las reglas se revisan de manera periódica a medida que se avanza en el aprendizaje colaborativo
Cuatro esquinas o rincones	1. Etiquete cada esquina con los nombres "Muy de acuerdo", "De acuerdo", "en desacuerdo", "muy en desacuerdo"
	2. Entrégueles una lista de declaraciones o creencias, sea claro en que ellos no deben poner sus nombres en estas. Ellos privadamente deben escoger su nivel de acuerdo con la declaración
	3. Lea una declaración a los estudiantes y escríbala en el tablero para que los estudiantes la recuerden
	4. Luego de escuchar la declaración, los estudiantes deben escoger su grado de acuerdo
	5. Los estudiantes se unirán a otro estudiante para discutir su posición. Los estudiantes deben discutir entre ellos antes de socializar
	6. Luego de la discusión por pares, la esquina completa discute su elección. Todos deben hacer parte del proceso de socialización de la decisión. Se debe compartir luego con el resto de la clase
	7. El profesor realiza una discusión con todo el grupo. Luego de la discusión, los estudiantes podrían cambiar su posición frente a la declaración. Deben defender su decisión
	8. Los estudiantes deben escribir un párrafo o un ensayo acerca de su posición o declaración

Nombre de la estrategia en español	Metodología
Diario de entrada doble	1. Los estudiantes dividen su espacio, en el que realizan su registro de lectura en dos columnas
	2. Llamar a una columna Citas del texto y a la otra Reflexiones
	3. Mientras el estudiante lee, solicitarle que copie una cita que le haya parecido interesante
	4. Solicitar al estudiante leer nuevamente las citas y escribir sus reflexiones relacionadas con cada cita
Técnica del grupo nominal	1. Divida en grupos de 5 o 6 miembros, preferiblemente alrededor de una mesa
	2. Realice una pregunta abierta
	3. Cada persona debe encontrar su respuesta de manera individual
	4. Registre en el tablero todas las respuestas
	5. Discusión de cada una de las ideas registradas
	6. Cada persona debe evaluar y luego votar por las mejores
	7. Realice el conteo de los votos y escoja las ideas que obtuvieron el mayor número
	8. La decisión se toma por orden de categorías o puntuación
Diagrama causa-efecto	1. Dibuja un diagrama en blanco
	2. Escribe de forma breve el problema o efecto
	3. Escribe las categorías que consideres apropiadas a tu problema
	4. Realiza una lluvia de ideas (tormenta de ideas) de posibles causas y relaciónalas con cada categoría
	5. Pregúntale ¿por qué? a cada causa, no más de dos o tres veces
	6. Empieza por enfocar tus variaciones en las causas seleccionadas como fácil de implementar y de alto impacto
Diagrama de afinidad	1. Agrupar a los estudiantes
	2. Seleccionar un líder de cada grupo
	3. Establecer el problema que se va a atender
	4. Realizar lluvia de ideas en el grupo
	5. Transferir datos a notas (frases independientes con un solo significado)
	6. Colocar las notas en un lugar visible
	7. Agrupar las notas
	8. Dibujar un diagrama de afinidad

Fuente: elaboración propia basada en Domingo *et al*., 2004; Domingo, 2010.

3. Trabajos relacionados

Según Badilla y Meza (2015) la selección de una estrategia de aprendizaje supone determinar previamente la actividad cognoscitiva que implica el aprendizaje o, lo que es lo mismo, el tipo de habilidades, destrezas y técnicas a desarrollar. Esta selección dependerá de dos factores: la situación sobre la que se ha de operar (tipo de problemas a resolver, datos a analizar, conceptos a relacionar, etc.); y de los propios recursos cognitivos o "metacognición" de los alumnos (habilidades, capacidades, destrezas, etc.). Además del conocimiento de los diferentes recursos didácticos para el trabajo con sus alumnos, el docente debe conocer y saber aplicar los criterios para seleccionar la estrategia o técnica más adecuada para su curso. Incluso el profesor debe tener la posibilidad de adaptar o crear y documentar sus propias estrategias y técnicas didácticas a través de plantillas con actividades estructuradas.

Según Sanz (2013) el concepto de estrategias didácticas se involucra con la selección de actividades y prácticas pedagógicas en diferentes momentos formativos, métodos y recursos de la docencia. Hacer una distinción conceptual entre método, técnica y estrategia permite asumir coherentemente el Aprendizaje Colaborativo como una propuesta para los espacios mediados o de orden tutorial. El criterio empleado para seleccionar una estrategia de aprendizaje colaborativo para procesos de enseñanza y aprendizaje son: (1) la facilidad de aprendizaje, por parte de los alumnos, de las normas de funcionamiento de la actividad colaborativa, (2) la facilidad de su aplicación en entornos virtuales y (3) la posibilidad de utilizar, de forma simultánea y enlazada, otras estrategias de aprendizaje.

Adicionalmente, el autor enuncia las siguientes ventajas encontradas en la aplicación de la estrategia:

- La adopción de un enfoque de aprendizaje por problemas facilita un proceso de aprendizaje significativo, donde el alumnado va construyendo las soluciones a partir de las herramientas básicas que ya conoce.
- Se potencian las habilidades y competencias relacionadas con la búsqueda, revisión y síntesis de información.
- La estructuración de las tareas de aprendizaje en entornos virtuales facilita la superación de barreras de localización y tiempo, permitiendo el trabajo desde sitios distintos al aula o, incluso, desde otros países, como es el caso de estudiantes que están en el extranjero con becas Erasmus de la Unión Europea.

Por su parte, Collazos *et al*. (2010) proponen una metodología para el aprendizaje de contenidos a través de la generación de actividades colaborativas para

alumnos de educación superior en Ingeniería. Con el fin de mejorar la interacción de los estudiantes y obtener mejores resultados en el Aprendizaje Colaborativo, la metodología integra las estrategias colaborativas JIGSAW, Compartir pensamiento en pares, Aprendizaje basado en problemas, *Talking Chips* y *Cooperative Controversy*, ya reconocidas por su aplicación para el aprendizaje. La metodología consta de 4 etapas, de las cuales la primera está orientada al trabajo individual y las tres subsiguientes al trabajo grupal. Para poder validarla, se ha generado una herramienta computacional la cual contempla los diferentes roles: el rol del tutor como observador y guía; y el rol de los alumnos como participantes activos en su proceso de construcción del conocimiento.

El trabajo de investigación presentado en Yuan-Hsuan (2015) analiza el efecto de facilitar el pensamiento crítico mediante un *script* de colaboración de tipo QRAC (del inglés cuestionamiento, lectura, respuesta y comprobación) para la estrategia de competencia lectora de los estudiantes de ciencias utilizando aprendizaje colaborativo (CSCL), apoyada en ambiente virtual. Los participantes fueron 76 estudiantes universitarios (71 % mujeres) matriculados en una universidad de Taiwan. Los estudiantes se agruparon ellos mismos por afinidad, y luego fueron asignados aleatoriamente a una condición de aprendizaje colaborativo con o sin el apoyo del *script* de colaboración QRAC. Los resultados mostraron que dicho guion de colaboración mejoró significativamente la competencia lectora de los estudiantes. Además, el efecto era dependiente de género y del formato de las pruebas. Los estudiantes varones se beneficiaron más de la parte de escritura del *script* de colaboración de respuestas mientras que las mujeres se beneficiaron más de ítems de opción múltiple. Al final los autores suministraron algunos consejos que se deben tener en cuenta para las prácticas de enseñanza y aprendizaje para mejorar competencias lectoras sobre la base de los hallazgos arrojados del estudio.

4. Método propuesto para la selección de estrategias

Debido a que existe una gran variedad de estrategias de aprendizaje colaborativo se hace necesaria la selección de dichas estrategias según un método de evaluación estándar. A continuación, se presentará el método de evaluación aplicado para la selección de estrategias. Dicho método de evaluación se compone de diferentes etapas:

1. *Etapa de identificación de estrategias*: en esta etapa se deben listar y caracterizar las estrategias. En dicha caracterización se debe tener en cuenta: su definición, los pasos o protocolos de la estrategia, los actores involucrados y los roles que participan en dicha estrategia.

2. *Etapa de especificación de criterios*: En esta etapa se deben especificar los criterios. En este artículo se definen diversos criterios desde el enfoque educacional, el interactivo y el adaptativo.

Ahora, los criterios tomados en cuenta en la dimensión "educacional" son los siguientes:

- Población objetivo: con este criterio se pretende identificar el tipo de personas hacia las que podemos dirigir la estrategia.
- Pensamiento crítico: para formar estudiantes con pensamiento crítico las estrategias colaborativas deben promover el desarrollo de habilidades expresivas y comprensivas. Algunas estrategias incentivan el sentido crítico al aplicar procesos de discusión o de opinión entre los estudiantes.
- Capacidad de análisis: la capacidad de análisis está relacionada con todo aquello que nos permite extraer conclusiones como resultado de relacionar datos recogidos del presente y extrapolarlos con los del pasado. Algunas estrategias incentivan la capacidad de análisis al aplicar procesos de análisis.
- Capacidad de síntesis: la capacidad de síntesis se refiere a la capacidad que poseen los estudiantes para resumir la información, identificando las ideas principales del tema que están abordando. Algunas estrategias incentivan la capacidad de análisis al aplicar procesos que involucran el desarrollo de informes parciales o finales, en los que los estudiantes deben exponer lo aprendido de forma resumida.
- Aplicabilidad del conocimiento: en el proceso de formación de los estudiantes se adquieren conocimientos propios que pueden ser aplicados o reforzados en el desarrollo de una estrategia colaborativa. Existen estrategias que incentivan más la aplicación o el refuerzo de dichos conocimientos.
- Incentivo a la creatividad: la creatividad se refiere a la generación de nuevas ideas o conceptos, o de nuevas asociaciones entre ideas y conceptos conocidos, que habitualmente producen soluciones originales. Se puede decir entonces que las estrategias que incentivan la creatividad son aquellas que proporcionan actividades relacionadas con la solución de problemas a partir de conocimientos previamente aprendidos y, por ende, está directamente relacionada con la aplicabilidad del conocimiento.
- Incentivo a la autonomía: la autonomía se refiere a la posibilidad de actuar y pensar sin depender del deseo de otros. Este criterio está directamente relacionado con el criterio de nivel de interacción, ya que un nivel de interacción bajo incentiva la autonomía y un nivel de interacción alto frenan la autonomía.

Por otra parte, los criterios de interacción son:

- Protocolo bien definido: cada estrategia está compuesta por un conjunto de pasos (protocolo o flujo de trabajo), pero en algunas de estas no son muy claros o no es tan fácil ajustarlos en un ambiente virtual. Con este criterio se pretende evaluar si dicho protocolo está bien definido.
- Roles claros: en cada estrategia los participantes deben tener roles según el desarrollo de aquella. En algunas de estas estrategias no son muy claros los roles que deben tener los participantes. Con este criterio se pretende evaluar si los roles están bien definidos para cada participante.
- Tiempo de vida de la estrategia: en todas las estrategias se debe llevar a cabo un conjunto de procesos que requieren de tiempos predeterminados. Con este criterio se pretende evaluar el tiempo total que requiere la estrategia para llevar a cabo todos sus procesos.
- Nivel de interacción: en todos los protocolos que se deben aplicar para cada estrategia deben existir interacciones. En algunos casos dicha interacción abunda y en otros casos dicha interacción es escasa. Con este criterio se pretende evaluar la cantidad de interacciones que se debe realizar en la estrategia para alcanzar sus objetivos.

Por su parte, los criterios de adaptación son:

- Conformación de grupos: con este criterio se pretende evaluar si en la estrategia de aprendizaje se incentiva el trabajo grupal.
- Reasignación de roles: cuando los estudiantes participan en una estrategia colaborativa deben llevar a cabo diversos roles según el protocolo definido por la estrategia de aprendizaje. En algunos casos, dichos roles pueden ser adaptados según las características propias de los estudiantes.
- Adaptación de recursos de interacción: en muchos procesos propios de cada estrategia se requiere el uso de herramientas y recursos. En algunos casos dichos recursos pueden ser adaptados según el perfil de los estudiantes o según otro criterio.
- Adaptación del dispositivo de acceso: cuando los participantes van a realizar un trabajo grupal de forma virtual se requieren diferentes dispositivos de comunicación, que en muchos casos no son aptos para llevar a cabo las actividades propias de la estrategia. Con este criterio se pretende evaluar si la estrategia se debe adaptar al *hardware* que será utilizado por los participantes.

Para la especificación de cada criterio se debe tener en cuenta:

- Prioridad: describe la importancia del criterio. Debe ser un valor entre 1 y 5.

- Tipo de evaluación: pueden ser rangos numéricos (ej. 0-10), etiquetas lingüísticas (ej. bajo, medio, alto), unidades de medida (ej. GB), temporal (ej. horas – minutos – segundos) entre otras.
- Forma de evaluación: explicación puntual de los niveles de gradación utilizados para la valoración de cada criterio.

Cabe resaltar que para cada criterio se debe completar una plantilla de especificación según las necesidades de cada persona o proyecto. A modo de ejemplo, se presentan en las tablas 2 a 4 las especificaciones realizadas para 3 de los criterios utilizados.

TABLA 2. ESPECIFICACIÓN DE CRITERIO DE PENSAMIENTO CRÍTICO

Prioridad (0-5)		4
Tipo de evaluación		Numérica (0-10)
Forma de evaluación	10	Se pueden identificar varios procesos que incentivan claramente a la discusión o a la opinión entre los estudiantes
	5-9	Se puede identificar por lo menos un proceso que incentiva claramente a la discusión o a la opinión entre los estudiantes
	0-5	No se pueden identificar claramente procesos que incentiven a la discusión o a la opinión entre los estudiantes

Fuente: Domingo *et al*. (2004).

TABLA 3. ESPECIFICACIÓN DE CRITERIO DE PROTOCOLO BIEN DEFINIDO

Prioridad (0-5)		5
Tipo de evaluación		Numérica (0-10)
Forma de evaluación	8-10	Se pueden identificar los procesos de la estrategia y su interacción de forma secuencial
	5-7	Los procesos no son claros, pero se puede identificar una interacción secuencial
	3-6	Se pueden identificar los procesos de la estrategia, pero no es clara su interacción
	0-3	Los procesos y su interacción no son claros

Fuente: Domingo *et al*. (2004).

TABLA 4. ESPECIFICACIÓN DE CRITERIO DE ADAPTACIÓN PARA LA CONFORMACIÓN DE GRUPOS

Prioridad (0-5)		5
Tipo de evaluación		Numérica (0-10)
Forma de evaluación	8-10	Se pueden identificar los procesos de la estrategia y su interacción de forma secuencial
	5-7	Los procesos no son claros, pero se puede identificar una interacción secuencial
	3-6	Se pueden identificar los procesos de la estrategia, pero no es clara su interacción
	0-3	Los procesos y su interacción no son claros

Fuente: Domingo *et al*. (2004).

3. *Etapa de valoración de los criterios*: previa identificación de las estrategias y teniendo en cuenta las especificaciones realizadas en cada criterio, se debe realizar la valoración de los criterios. Para la valoración se propone una plantilla (figura 1). Los componentes de la plantilla de valoración son:
   1. Nombre de la estrategia que se desea valorar.
   2. Nombre de los criterios que se desea valorar.
   3. Campos para la valoración de cada criterio según la estrategia.
   4. Resultados de la valoración de los criterios (para cada estrategia) según el enfoque de adaptación. Cabe resaltar que dicho resultado se obtiene al promediar los resultados obtenidos para los criterios relacionados con el enfoque de adaptación.
   5. Resultados de la valoración de los criterios (para cada estrategia) según el enfoque de cooperación. Cabe resaltar que dicho resultado se obtiene al promediar los resultados obtenidos para los criterios relacionados con el enfoque de cooperación.
   6. Resultados de la valoración de los criterios (para cada estrategia) según el enfoque de educación. Cabe resaltar que dicho resultado se obtiene al promediar los resultados obtenidos para los criterios relacionados con el enfoque de educación.
   7. Resultados de la valoración de los criterios (para cada estrategia) según todos los enfoques. Cabe resaltar que dicho resultado se obtiene al promediar los resultados obtenidos por cada enfoque.

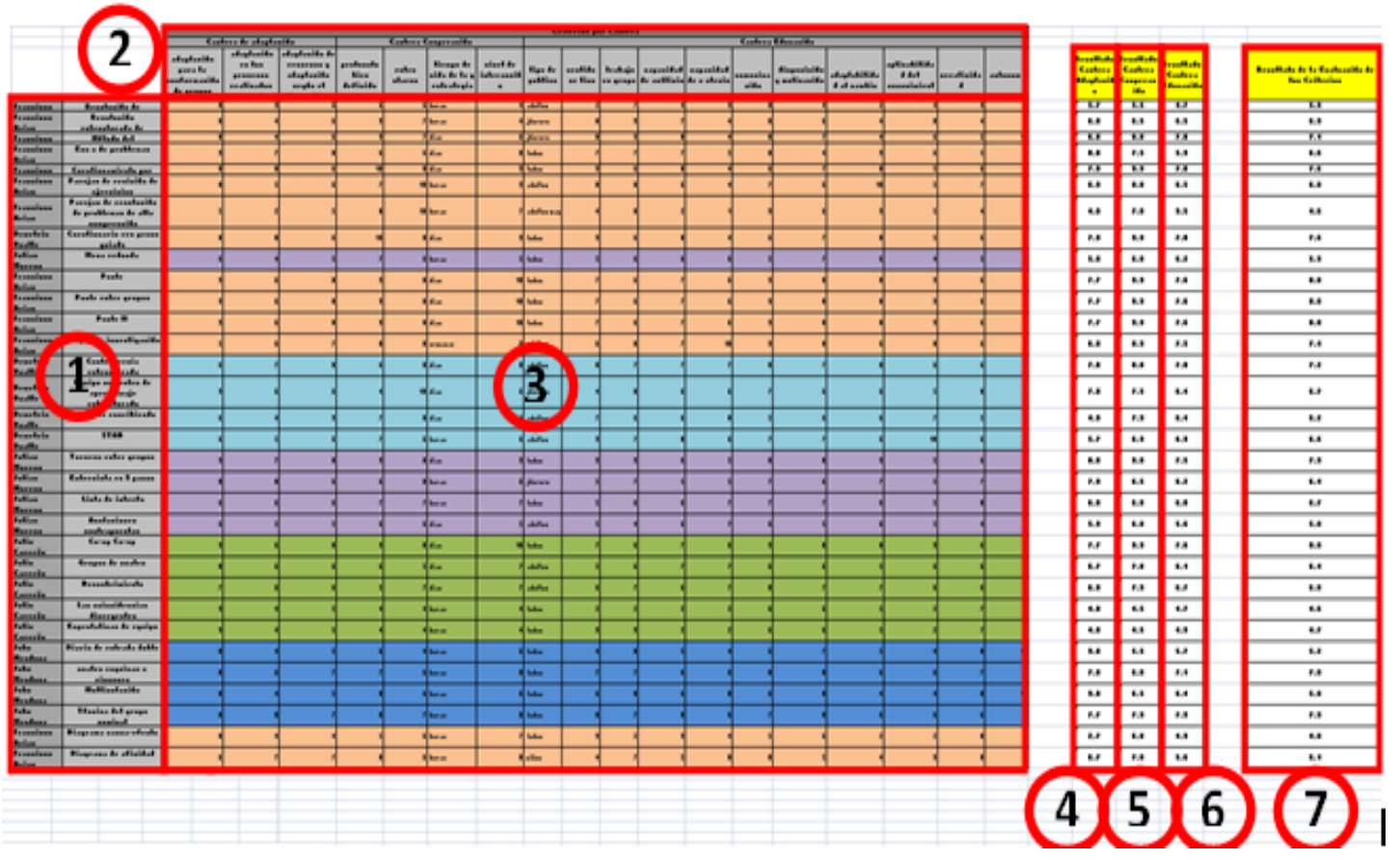


FIGURA 1. PLANTILLA DE EVALUACIÓN DE ESTRATEGIAS.

Fuente: elaboración propia.

5. Resultados

En la tabla 5 se presenta un ejemplo del resultado obtenido al aplicar la valoración de criterios por grupos en 9 estrategias. Cabe resaltar que dicha valoración fue aplicada sobre 32 estrategias.

TABLA 5. EJEMPLO DE RESULTADOS POR ESTRATEGIA

	Resultado de Enfoque Adaptación	Resultado de Enfoque Cooperación	Resultado de Enfoque Educación
Resolución de problemas	5.7	5.5	4.7
Resolución estructurada de problemas	6.0	6.5	6.5
Método del descubrimiento	6.0	8.0	7.0
Envío de problemas	8.0	7.5	5.9
Cuestionamiento por pares	7.3	9.3	7.0
Parejas de revisión de ejercicios	6.3	8.0	6.5
Parejas de resolución de problemas de alta comprensión	4.0	7.8	3.5
Cuestionario recíproco guiado	7.3	9.3	7.0

Fuente: elaboración propia.

En las tablas 6 a 8 se presentan las 10 estrategias colaborativas más relevantes, según la valoración aplicada de los criterios por cada uno de los enfoques.

TABLA 6. TOP 10 DE ESTRATEGIAS GRUPO DE ADAPTACIÓN

1	Envío de problemas
2	Técnica del grupo nominal
3	Torneos entre grupos
4	Cuestionamiento por pares
5	Cuestionario recíproco guiado
6	JIGSAW
7	Co-op Co-op
8	Entrevista en 3 pasos
9	Equipo con roles de aprendizaje estructurado
10	Cuatro esquinas o rincones

Fuente: elaboración propia.

TABLA 7. TOP 10 DE ESTRATEGIAS GRUPO DE COOPERACIÓN

1	Cuestionamiento por pares
2	Cuestionario recíproco guiado
3	JIGSAW
4	Co-op Co-op
5	Torneos entre grupos
6	Parejas de revisión de ejercicios
7	Grupos de investigación
8	Parejas de resolución de problemas de alta comprensión
9	Método del descubrimiento
10	Controversia estructurada

Fuente: elaboración propia.

TABLA 8. TOP 10 DE ESTRATEGIAS GRUPO DE EDUCACIÓN

1	Torneos entre grupos
2	JIGSAW
3	Co-op Co-op
4	Grupos de investigación
5	Técnica del grupo nominal
6	Método del descubrimiento
7	Cuatro esquinas o rincones
8	Cuestionamiento por pares
9	Controversia estructurada
10	Controversia estructurada

Fuente: elaboración propia.

En la tabla 9 se presentan las 10 estrategias colaborativas más relevantes según la valoración general de criterios.

TABLA 9. TOP 10 DE ESTRATEGIAS

1	Torneos entre grupos
2	JIGSAW
3	Cuestionamiento por pares
4	Técnica del grupo nominal
5	Grupos de investigación
6	Controversia estructurada
7	Envío de problemas
8	Parejas de revisión de ejercicios
9	Cuatro esquinas o rincones
10	Método del descubrimiento

Fuente: elaboración propia.

6. Conclusiones

En este capítulo se presentó un método para la selección de estrategias colaborativas que aplica para cualquier tipo o cantidad de estrategias. Para el método propuesto se definieron diversos criterios desde el punto de vista educacional, interactivo y adaptativo. Los criterios tenidos en cuenta por la dimensión "educacional" fueron los siguientes: población objetivo, pensamiento crítico, capacidad de análisis, capacidad de síntesis, aplicabilidad del conocimiento, incentivo a la creatividad e incentivo a la autonomía. Los criterios desde el punto de vista de la "interacción" fueron: protocolo bien definido, roles claros, tiempo de vida de la estrategia y nivel de interacción. En cuanto a los criterios tenidos en cuenta desde el punto de vista de adaptación fueron los siguientes: conformación de grupos, reasignación de roles, adaptación de recursos de interacción y adaptación del dispositivo de acceso.

Aunque en el caso de estudio considerado en esta investigación los criterios anteriores fueron los más significativos, el usuario del método, de ser necesario, podría definir otros criterios que crea importantes para su proceso de selección de estrategias colaborativas.

El método propuesto es muy sensible a la prioridad dada para cada criterio, por lo cual se debe ser muy cuidadoso al momento de asignar las prioridades respectivas.

Cabe señalar finalmente que los resultados obtenidos al aplicar el método para un caso de estudio arrojaron una clasificación de estrategias conveniente y esencial para el correcto desarrollo de los procesos de enseñanza y aprendizaje.

Referencias

Badilla, M. y Meza, G. (2015). A Pedagogical Model to Develop Teaching Skills. The Collaborative Learning Experience in the Immersive Virtual World TYMMI. *Computers in Human Behavior*, *51*, 594-603.

Collazos, C., Ordónez, Ochoa, S. y Guerrero, L. (2010). Addressing Computer-Supported Collaborative Learning in the Classroom: Experiences in Engineering Education. *Social and Behavioral Sciences*, *2*(2), 2685-2688.

Domingo, J. (2010). El aprendizaje cooperativo y las competencias. *Revista d'Innovació Docent Universitària*, *2*, 1-9.

Domingo, J., Llumà, J., Manzanares, M., Ruiz, C. y Camps, G. (2004). Estrategias para el trabajo con grupos cooperativos. Grupo EEEI (EcoEnginys Escola Industrial).

Moreno, J., Ovalle, D. y Vicari, R. (2012). A Genetic Algorithm Approach for Group Formation in Collaborative Learning considering Multiple Student Characteristics. *Computers & Education*, *8*(1), 560-569.

Sanz, J. (2013). Construcción colaborativa del aprendizaje y del conocimiento mediante Google Docs en el ámbito de la educación superior universitaria. *Revista de Formación e Innovación Educativa Universitaria*, *6*(2), 109-119.

Yuan-Hsuan, L. (2015). Facilitating Critical Thinking Using the C-QRAC Collaboration Script: Enhancing Science Reading Literacy in a Computer-Supported Collaborative Learning Environment. *Computers & Education*, *88*, 182-191.

4
OBJETOS VIRTUALES DE APRENDIZAJE

Demetrio Arturo Ovalle Carranza, Julián Moreno Cadavid, Francisco Arias Sánchez

Considerando la acogida que dentro de la comunidad académica y científica tienen los Objetos Virtuales de Aprendizaje, en el presente capítulo se describen, de manera breve pero concisa, varios de los principales temas alrededor de este tipo de recursos. Más específicamente, se parte de la discusión de varias definiciones brindadas por diversos autores y organizaciones. Igualmente, se presentan los estándares e iniciativas más usados para los metadatos que los describen. Finalmente, se describen las aplicaciones generalmente utilizadas para almacenarlos, esto es, los repositorios y las federaciones, incluyendo algunos ejemplos específicos.

1. Introducción

Dentro del amplio mundo del aprendizaje asistido por computador, conocido también como *e-learning*, los Objetos Virtuales de Aprendizaje (OVA) surgen como una alternativa para "encapsular" material educativo. Así, por ejemplo, para el Ministerio de Educación Nacional de Colombia se trata de

> una entidad digital, autocontenible y reutilizable, con un claro propósito educativo, constituido por al menos tres componentes internos editables: contenidos, actividades de aprendizaje y elementos de contextualización. Tal material puede ser luego desplegado al usuario, generalmente el estudiante, sea directamente o a través de diversos medios como, por ejemplo, un sistema gestor de aprendizaje o LMS (por sus siglas en inglés de *Learning Management Systems*). (Ministerio de Educación Nacional, 2006)

El uso de estos OVA supone numerosas y variadas ventajas. La primera es que permiten separar la estructura del dominio de conocimiento del contenido, brindando flexibilidad y reusabilidad. La segunda es que los metadatos que los describen no solo pueden ser usados para efectos de caracterización, sino también para otros diversos (ej. la adaptación). Y la tercera es que existe actualmente un número considerable de repositorios, muchos de libre acceso, donde profesores y desarrolladores pueden acceder a ellos y de esta forma enriquecer aún más sus clases como apoyo a procesos de enseñanza-aprendizaje de determinadas temáticas. Considerando este panorama, este capítulo presenta una descripción general de los OVA y de algunos de los principales temas relacionados, como son los metadatos, los repositorios y las federaciones.

2. Definiciones

Existen numerosas y diversas definiciones acerca de lo que puede ser un OVA. Para Kurilovas y Dagiene (2014), por ejemplo, se trata de "cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos para facilitar las búsquedas semánticas, que pueda ser utilizado y reutilizado dentro de un entorno *e-learning*" (figura. 1). Para el Ministerio de Educación Nacional de Colombia se trata de: "a manera de complemento, los objetos de aprendizaje han de tener una estructura (externa) de información que facilite su identificación, almacenamiento, búsqueda y recuperación: los metadatos" (López *et al*., 2014). Por su parte, Moreno *et al*. (2010) de manera más concisa lo definen como "cualquier recurso digital empleado de manera simple o compuesta para soportar el proceso de enseñanza-aprendizaje y que puede ser reutilizado".

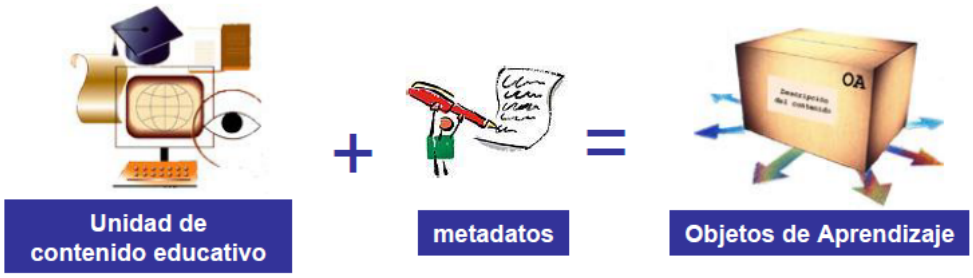


FIGURA 1. CONCEPTUALIZACIÓN DE UN OVA.

Fuente: tomado de Kurilovas y Dagiene (2014).

Una metáfora muy común empleada para definir los OVA y ampliar estas definiciones es que pueden entenderse como piezas de LEGO: pequeñas piezas de instrucción (LEGO) que pueden ser ensambladas entre sí en alguna estructura instruccional más grande (ej. un castillo) y luego ser reusadas en otras estructuras (ej. una nave espacial). Esta analogía, aunque es muy ilustrativa, tiene los siguientes problemas conceptuales que entran en conflicto con las propiedades de los OVA: a) cualquier bloque de LEGO puede ser combinable con cualquier otro; b) los bloques de LEGO pueden ser ensamblados de cualquier manera; c) los bloques de LEGO son tan simples que incluso un niño puede combinarlos.

Considerando estos problemas, es indispensable proponer al átomo como una nueva metáfora para los OVA: un átomo es algo pequeño que puede ser combinado y recombinado para formar estructuras más grandes. Esta metáfora es más acorde con las propiedades de los OVA, pues: a) no todo átomo puede ser combinado con otros átomos; b) los átomos solo pueden ser ensamblados entre sí en ciertas estructuras dependiendo de su propia estructura interna; c) algún entrenamiento es requerido para combinar átomos. Estas propiedades, en resumen, quieren decir que la estructuración de contenido educativo a partir de OVA es posible siempre y cuando exista un diseño instruccional apropiado de por medio (Zapata, 2015).

3. Estándares de metadatos

De manera general puede decirse que el proceso de estandarización sirve para garantizar el funcionamiento de elementos construidos independientemente, así como para avalar la calidad de dichos elementos y asegurar su acoplamiento. En el caso de los OVA, una de las principales funciones de los estándares de metadatos es servir como garantes de su durabilidad, reutilización e interoperabilidad.

Uno de los principales estándares para metadatos de OVA, si no el más importante, es el *Learning Object Metadata* (LOM) de la IEEE (IEEE, 2015). Este estándar considera en total 45 atributos, algunos atómicos y otros compuestos, algunos con cardinalidad unitaria y otros con cardinalidad múltiple, repartidos en 9 categorías como se describe a continuación.

- *General*: como su nombre lo indica, agrupa la información general que describe el OVA de manera global. Incluye los atributos: *identificador*, *título*, *lenguaje*, *descripción*, *palabra clave*, *alcance*, *estructura* y *nivel de agregación*.
- *Ciclo de vida*: agrupa las características relacionadas con el origen y estado actual del OVA, así como aquellas que le han afectado durante su evolución. Incluye los atributos: *versión*, *estado* y *contribución*.

- *Meta-metadatos*: agrupa la información sobre la propia instancia de los metadatos en lugar del OVA al que hace referencia. Incluye los atributos: *identificador*, *contribución, esquema de metadatos* y *lenguaje*.
- *Técnica*: agrupa los requerimientos y características técnicas del archivo donde está encapsulado el OVA. Incluye los atributos: *formato*, *tamaño*, *ubicación*, *requerimiento*, *notas de instalación*, *otros requerimientos de plataforma* y *duración*.
- *Uso educativo*: comprende las características pedagógicas del OVA. Incluye los atributos: *tipo de interactividad*, *tipo de recurso de aprendizaje*, *nivel de interactividad*, *densidad semántica*, *rol del usuario objetivo final*, *contexto*, *rango típico de edad*, *dificultad*, *tiempo de aprendizaje típico*, *descripción* y *lenguaje*.
- *Derechos*: reúne la información sobre propiedad intelectual y condiciones de uso del OVA. Incluye los atributos: *costo*, *copyright y otras restricciones* y *descripción*.
- *Relación*: concentra las características que definen la posible relación entre este OVA y otros. Incluye los atributos: *tipo* y *recurso*.
- *Anotación*: permite incluir comentarios sobre el uso del OVA, así como información sobre cuándo y por quién fueron creados dichos comentarios. Incluye los atributos: *entidad*, *fecha* y *descripción*.
- *Clasificación*: describe el OVA en relación con un determinado sistema de clasificación. Incluye los atributos: *propósito*, *ruta taxonómica*, *descripción* y *palabra clave*.

Después de LOM, quizá el estándar más utilizado es Dublin Core (Powel *et al*., 2015), el cual, más que ser específico para el OVA, puede ser utilizado para describir recursos digitales de diversa índole. Este estándar se encuentra compuesto de 15 atributos, que pueden ser clasificados en tres grupos que indican la clase o el ámbito de la información que contienen, tal como se describe a continuación.

Atributos relacionados con el contenido del recurso:

- *Título*: el nombre dado a un recurso.
- *Palabras clave*: frases clave o códigos de clasificación que describen el tema del recurso.
- *Descripción*: descripción puede ser detallada por medio de un resumen, tabla de contenidos, referencia a una representación gráfica de contenido o una descripción de texto libre del contenido del recurso.
- *Fuente*: referencia a otro recurso del cual se deriva el recurso actual.
- *Lenguaje*: idioma del contenido intelectual del recurso.
- *Relación*: referencia a un recurso relacionado.

Atributos relacionados con la propiedad intelectual:

- *Autor*: entidad primariamente responsable de la creación del contenido intelectual del recurso. Entre los ejemplos de un creador se incluyen una persona, una organización o un servicio.
- *Editor*: entidad responsable de hacer que el recurso se encuentre disponible.
- *Colaboradores*: entidad responsable de hacer colaboraciones al contenido del recurso.
- *Derechos*: información sobre los derechos de propiedad y sobre el recurso.

Atributos relacionados con la instanciación del recurso:

- *Fecha*: típicamente la fecha será asociada con la creación o disponibilidad del recurso.
- *Tipo de recurso*: incluye términos que describen las categorías generales, funciones, géneros o niveles de agregación del contenido. Algunos tipos de recurso pueden ser: animación, simulación, gráfico interactivo, glosario, ejercicio.
- *Formato*: puede incluir el tipo de archivo o dimensiones del recurso. Podría usarse para determinar el *software*, *hardware* u otro equipamiento necesario para ejecutar u operar con el recurso. Algunos tipos de formatos pueden ser: pdf, doc, txt, swf, ppt, entre otros. Se recomienda seleccionar un valor de un vocabulario controlado (ej. MIME).
- *Identificador*: referencia no ambigua para el recurso dentro de un contexto dado.

Además de los estándares, también existen diversas iniciativas, conocidas como especificaciones, que procuran capturar un consenso entre investigadores resumiendo o extendiendo ciertos aspectos de un estándar ya existente. Una lista de iniciativas particulares para regiones específicas, países, centros de investigación o universidades, se presenta en la tabla 1.

TABLA 1. LISTADO DE INICIATIVAS PARA METADATOS DE OVA

Nombre	Comunidad	URL
ANZ-LOM	Australia y Nueva Zelandia	http://www.thelearningfederation.edu.au
OBBA	Brasil	http://www.portalobaa.org
CanCore	Canadá	http://cancore.athabascau.ca
ISRACORE	Israel	http://www.iucc.ac.il
NORLOM	Noruega	http://www.itu.no/no/NSSL
UK LOMCore	Reino Unido	http://metadata.cetis.ac.uk

Fuente: elaboración propia.

Dentro de estas iniciativas vale la pena destacar en el ámbito regional a OBAA, acrónimo de *Objetos de Aprendizagem Baseados em Agentes*. Esta iniciativa fue desarrollada por la Universidade Federal do Rio Grande do Sul (UFRGS) en colaboración con la Universidade do Vale dos Sinos (UNISONOS). La base de OBAA es LOM, del cual "hereda" todos sus metadatos, complementando las categorías *Técnica* y *Educacional*, al tiempo que agrega dos nuevas categorías (Vicari, 2010).

Más específicamente, en la categoría *Técnica* OBAA agrega el atributo atómico *plataforma soportada* y los atributos compuestos *características específicas de la plataforma* y *servicio*. En la categoría *Educacional* agrega los atributos atómicos *tipo de contenido educativo*, *interacción* y *estrategia didáctica*. Así mismo, define una categoría *Accesibilidad* con los atributos *visual*, *auditivo*, *textual*, *táctil*, *declaración inicial* y *recurso equivalente*. Finalmente, define una categoría *Tabla de información de segmento* en la que incluye los atributos *lista de segmento* y *lista de grupo de segmento*.

Como anotación final respecto a los estándares e iniciativas, es importante mencionar que en la mayoría de los casos estos se representan a través de lenguajes abiertos como XML, puesto que esto facilita el intercambio de información y de contenidos entre los diferentes sistemas que los administran, sobre los cuales se hará hincapié en la siguiente sección. A manera de ejemplo, en la figura 2 se muestra parte de una instancia de metadatos LOM en XML.

4. Repositorios y federaciones

Para que la reutilización sea un beneficio real de los OVA es necesario un lugar destinado para su almacenamiento y clasificación que facilite posteriormente su mantenimiento, localización y también permita compartir ese OVA con otros sistemas para aplicaciones diversas. Este lugar es lo que se conoce como Repositorio de Objetos de Aprendizaje (ROA). Según Kurilovas y Dagiene (2014), los ROA son un tipo de bibliotecas digitales especializadas en recursos educativos (los OVA) que utilizan estándares específicos de metadatos.

Principalmente se identifican dos tipos de ROA (Rodríguez *et al*., 2013; Downes, 2015): el primer tipo son los que contienen los OVA y sus metadatos; en estos, los objetos y sus descriptores se encuentran dentro de un mismo sistema e, incluso, dentro de un mismo servidor. El segundo tipo son los que contienen solo los metadatos; en este caso, el repositorio contiene solo los descriptores y se accede al objeto a través de una referencia a su ubicación física que se encuentra en otro sistema o repositorio de objetos (figura 3).

```
<lom>
    <general>
        <identifier>
            <catalog>Ingeniería</catalog>
            <entry></entry>
        </identifier>
        <title>
            <string language='en'>Comparing Agent-Oriented Methodologies</string>
        </title>
        <language>es</language>
        <description>
            <string language='en'>Este trabajo presenta una comparacion entre tres
            metodologias orientadas a agentes: MaSE, Prometeo y Tropos. </string>
        </description>
        <keyword>
            <string language='en'>Sistemas multiagente</string>
        </keyword>
        <coverage>
            <string language='en'>Siglo XXI</string>
        </coverage>
        <structure>
            <source uniqueElementName='source'>LOM-ESv1.0</source>
            <value uniqueElementName='value'>5</value>
        </structure>
        <aggregationlevel>
            <source uniqueElementName='source'>LOM-ESv1.0</source>
            <value uniqueElementName='value'>1</value>
        </aggregationlevel>
    </general>
    <lifecycle>
        <contribute>
            <role>
```

FIGURA 2. EJEMPLO DE METADATOS LOM EN XML.

Fuente: elaboración propia.

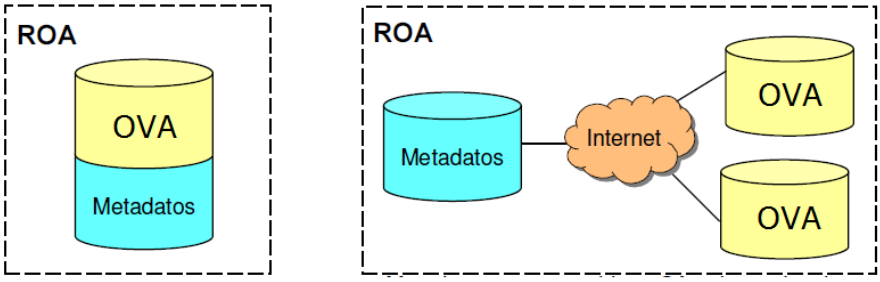


FIGURA 3. TIPOS DE ROA.

Fuente: Kurilovas y Dagiene (2014).

Algunos ejemplos de ROA son: MERLOT, CAREO, CLOE, GEM, POOL, CeLeBraTe, ELENA, RABID, Alexandría, APROA (Rodríguez *et al*., 2014). En el ámbito nacional, vale la pena mencionar el Banco Nacional de Objetos de Aprendizaje e Informativos promovido por el Ministerio de Educación en el marco del proyecto Colombia Aprende (figura 4).

En un nivel de agregación mayor respecto a los ROA se encuentran las federaciones, que tienen como objetivo la centralización de diversos ROA a partir de un único punto de acceso. El esquema de acoplamiento entre los ROA dentro de una federación es generalmente débil. Esto significa que cada ROA individual mantiene su modelo de datos intacto y lo que se hace es una traducción a un modelo común. Precisamente esta característica es la que diferencia principalmente a una federación de un ROA distribuido.

Algunos ejemplos de federaciones son: SMETE, ARIADNE, LAFLOR y AGREGA (Rodríguez *et al*., 2014). En el ámbito regional, vale la pena mencionar la Federaçao Educa Brasil (FEB) promovida por la Rede Nacional de Ensino e Pesquisa del Gobierno brasilero (figura 5).



FIGURA 4. BANCO NACIONAL DE OBJETOS DE APRENDIZAJE E INFORMATIVOS.

Fuente: tomado de https://www.colombiaaprende.edu.co/.

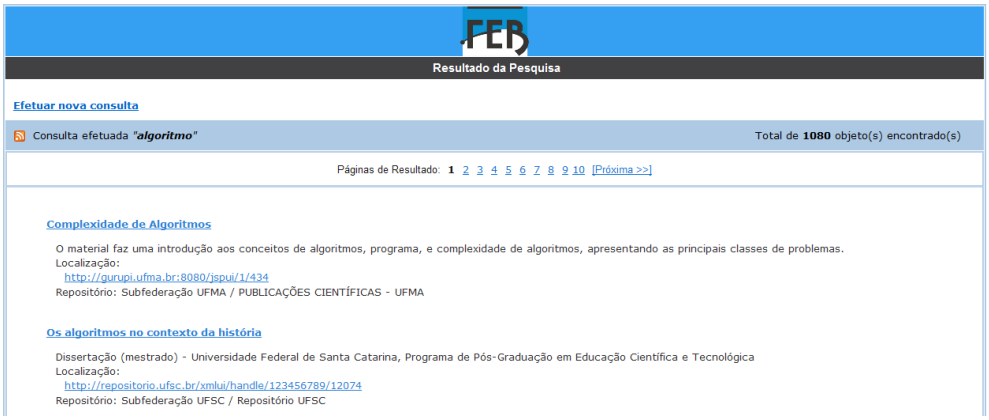


FIGURA. 5. FEDERAÇAO EDUCA BRASIL.

Fuente: tomado de https://aberta.org.br/federacao-educa-brasil-feb/.

5. Conclusiones

La utilización de OVA para apoyar los procesos de enseñanza-aprendizaje en entornos virtuales ha ido creciendo en los últimos años, máxime cuando cada vez existen más repositorios y entidades que los promueven. Sin embargo, para que su uso se masifique y se cumpla la promesa de reusabilidad que muchos autores comparten, es necesario primero que la comunidad, tanto académica como científica, esté al tanto de sus principales características. Precisamente en este sentido, en este capítulo se comenzó por comparar varias definiciones de interés. Si bien no existe un consenso único en tales definiciones respecto al alcance que deben tener los OVA, sí existen dos elementos claves que todas comparten: a) su finalidad como apoyo en un entorno educativo; y b) su caracterización por medio de metadatos.

Respecto a este segundo elemento, en este mismo capítulo se describieron algunos de los estándares de metadatos e iniciativas más usados para los OVA. De nuevo en este caso existen diversos puntos de vista, los cuales difieren básicamente en extensión. Así, pese a que podría decirse que LOM de la IEEE sería el estándar de base, lo cierto es que muchas organizaciones emplean por ejemplo Dublin Core, el cual es mucho más reducido (hablando del número de metadatos), mientras que otras emplean OBAA, que es más amplio.

Finalmente, respecto a los esquemas para el almacenamiento y administración de OVA, el panorama nuevamente es de diversidad. Por una parte, se encuentran los repositorios, dentro de los cuales existen diversas arquitecturas. Asimismo, existen las federaciones, las cuales surgen precisamente para suplir la necesidad de enlazar repositorios de diversas naturalezas (arquitecturas y modelos de datos). La diversidad de la que se ha hablado respecto a los OVA, más que ser un impedimento para su utilización, es una potencialidad que puede explotarse. De esta forma, investigadores, educadores, estudiantes y demás actores involucrados pueden acoger las alternativas que más se ajusten a sus necesidades sin que ello suponga un aislamiento del resto de la comunidad. Como comentario final es importante señalar que el siguiente capítulo trata sobre la forma como los OVA son adaptados a los perfiles de los estudiantes en contextos colaborativos a partir de los metadatos que los describen.

Referencias

Downes, S. (2015). *The Learning Marketplace. Meaning, Metadata and Content Syndication in the Learning Object Economy*. Disponible en http://www.downes.ca/files/book3.pdf. Última consulta: agosto de 2015.

IEEE. (2015). *Draft Standard for Learning Object Metadata (LOM)*. Disponible en http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf. Última consulta: agosto de 2015.

Kurilovas, E. y Dagiene, V. (2014). Web 3.0 - Based Personalisation of Learning Objects in Virtual Learning Environments. *Computers in Human Behavior*, *30*, 654-662.

López, V., De la Prieta, F., Ogihara, M. y Wong, D. (2012). A Model for Multi-Label Classification and Ranking of Learning Objects. *Expert Systems with Applications*, *39*, 8878-8884.

Ministerio de Educación Nacional Colombiano. (2006). Objetos Virtuales de Aprendizaje e Informativos. Disponible en Portal Colombia Aprende http://www.colombiaaprende.edu.co/html/directivos/1598/article-172369.html. Última consulta junio 6 de 2009.

Moreno, J., Ovalle, D. y Vicari, R. (2010). Método para la selección de objetos de aprendizaje a partir de características cognitivas del estudiante. En *Memorias del 21 Simpósio Brasileiro de Informática na Educação*, João Pessoa, Brasil.

Powell, A., Nilsson, M., Naeve, A., Johnston, P. y Baker, T. DCMI Abstract Model. Disponible en http://dublincore.org/documents/abstract-model. Última consulta: agosto de 2015.

Rodríguez, P. A., Moreno, J., Duque, N. D., Ovalle, D. y Silveira, R. (2014). Un modelo para la organización semiautomática de contenido educativo desde repositorios abiertos de objetos de aprendizaje. *Revista Electrónica de Investigación Educativa*, *16*(1), 123-136.

Rodríguez, P., Duque, N., Ovalle, D. y Tabares, V. (2013). Validación de un sistema inteligente de recomendación híbrida en federaciones de repositorios de objetos de aprendizaje. *Novas Tecnologias na Educaçao*, *11*(3).

Vicari, R., Bez, M., Marques, J., Ribeiro, A., Gluz, J., Passerino, L. "Proposta Brasileira de Metadados para Objetos de Aprendizagem Baseados em Agentes (OBAA)". Novas Tecnologias na Educação, *8*(2), 2010.

Zapata, A., Menéndez, V., Prieto, M. y Romero, C. (2015). Evaluation and Selection of Group Recommendation Strategies for Collaborative Searching of Learning Objects. *Human-Computer Studies*, 76, 22-39.

5
SISTEMAS DE SOPORTE A LA COLABORACIÓN

Luis Guillermo Torres-Ribero, Enrique González

El presente capítulo presenta el estado del arte sobre los sistemas de soporte a la colaboración realizado en el marco del proyecto de investigación AYLLU. En este proyecto se desarrolló un *framework* genérico sobre el cual se pueden construir servicios colaborativos (Cantor *et al*., 2006), soportados a través de agentes de *software* y apoyados en servicios como agendas, directorios, etc., con el fin de lograr objetivos comunes entre un grupo de usuarios. A continuación, después de unas definiciones básicas y unas consideraciones preliminares, se analizan con detalle las principales problemáticas encontradas en el desarrollo de sistemas colaborativos. A partir de las problemáticas identificadas surgen nuevos cuestionamientos que pretenden ser respondidos con el desarrollo de los componentes que son presentados en los capítulos siguientes.

1. Definiciones básicas de sistemas colaborativos

Las definiciones utilizadas para el proyecto AYLLU podrían ser muy diversas y depender del contexto; por tal motivo, lo primero que se requiere es unificar el lenguaje. En esta sección se introducen las definiciones generadas por algunos de los autores consultados, para luego unificar y presentar las definiciones acogidas y utilizadas en este libro.

El CSCW (acrónimo en inglés de *Computer Supported Cooperative Work*) se refiere a un conjunto de personas trabajando juntas en un producto, área de investigación, tema o esfuerzo escolar con la ayuda de un computador (Palmer y Fields, 1994). Según estos autores, en el contexto de CSCW las definiciones más importantes son las siguientes:

- *Software* de colaboración: *software* que permite la interacción y el manejo de contenidos con el fin de producir un resultado o entregable.
- Cooperación: en este contexto se utiliza de la misma forma que colaboración, como si fuesen sinónimos.

La problemática de colaboración no solo ha sido abordada para las personas, sino también para entidades artificiales en el marco de los sistemas multiagentes. Por ejemplo, los agentes actúan como representantes en un sistema para ayudar a responder peticiones sencillas y grabar peticiones no respondidas. En una forma más amplia, se pueden concebir sistemas de agentes organizados a través de roles que pueden ayudar a las personas que colaboran (Yin *et al*., 2014). De acuerdo con los autores, en el contexto orientado a agentes las definiciones más importantes son:

- Colaboración: división de trabajo entre muchos agentes del mismo tipo para alcanzar un objetivo.
- Cooperación: interacción y comunicación entre agentes para cumplir con una tarea.
- Rol o tipo: agente especializado en una tarea en particular; por ejemplo, un agente de interfaz, un agente de transacción, etc.

Por otro lado, la Ingeniería de Colaboración surge como un método facilitador para diseñar mecanismos de interacción entre personas (Vreede y Briggs, 2005). En este contexto las definiciones base son:

- Colaboración: esfuerzos conjuntos de las personas para llegar a un resultado.
- Cooperación: se utiliza como sinónimo de colaboración, sin hacer mayor distinción.

Además de tener en consideración las definiciones aportadas por autores representativos, en el marco de este libro, se tiene en cuenta que uno de los objetivos primordiales de la plataforma AYLLU es lograr adaptar el paradigma de agentes, propuesto por Ferber (1999) y ampliado por Cantor *et al*., (2006), a la interacción y colaboración entre personas. Dado que el paradigma de agentes ha demostrado ser eficiente en el contexto de agentes, se busca llevar al contexto de la colaboración entre personas sus aspectos principales, como lo son:

- La interacción como medio para alcanzar metas comunes.
- La existencia de roles que permitan asumir una responsabilidad acorde con las habilidades de cada uno y tener acceso controlado a determinados recursos.
- Las interacciones se realizan de manera estructurada, intencional y con semántica bien definida.

El paradigma de las 5C define la cooperación entre agentes como la suma de la colaboración, la coordinación y la gestión de conflictos soportada por mecanismos de comunicación estructurada. Al tomar este paradigma y aplicarlo en el trabajo conjunto entre personas, se encuentran múltiples similitudes entre algunas de estas palabras; por ejemplo, en los contextos relacionados con personas, la palabra cooperación, muy similar en definición a la palabra colaboración, puede generar confusión si se compara con las mismas palabras en el contexto de agentes. Por esta razón, en el contexto 5C, se utilizará la expresión "asignación de tareas" (*allocation*, en inglés) para referirnos a lo que tradicionalmente se le denomina colaboración en el marco de los agentes, eliminando así la posible confusión al utilizar las palabras antes mencionadas. Una vez analizadas estas definiciones provenientes de diferentes contextos, a continuación, se presentan las definiciones que se han tomado como generales en el marco del proyecto AYLLU.

- *Colaboración*: trabajo conjunto entre personas, es decir, interacción directa entre personas mediante mecanismos estructurados de comunicación con el fin de llegar a un resultado asociado a una tarea común compartida.
- *Cooperación*: se toma como un sinónimo de colaboración, aunque se aplicará principalmente a contextos de colaboración entre agentes.
- *Asignación de tareas*: se refiere a la forma en que se distribuyen las tareas entre los agentes o personas para conseguir un objetivo.
- *Coordinación de acciones*: se refiere a la forma en que se sincronizan y planifican las acciones de un conjunto de personas en un contexto colaborativo.
- *Gestión de conflictos*: se refiere a los mecanismos para manejar los conflictos por el acceso concurrente a recursos por un conjunto de personas en un contexto colaborativo.
- *Grupo de trabajo*: un grupo de trabajo colaborativo es un conjunto de personas que buscan alcanzar una meta común mediante la asignación de tareas, la coordinación y la gestión de los conflictos.
- *Servicio cooperativo*: son servicios que apoyan y facilitan la interacción, bien sea en forma síncrona o asíncrona, entre los integrantes de un grupo de trabajo para el cumplimiento de un objetivo común.
- *Rol*: ente abstracto que establece las responsabilidades, recursos y habilidades que se requieren para que una persona desempeñe una tarea especializada. En el marco de un servicio cooperativo, en un grupo de trabajo se puede requerir de uno o varios roles para el cumplimiento del objetivo. La definición y gestión de roles son aspectos claves y necesarios para la creación de un sistema colaborativo.

- *Awareness de grupo*: en el contexto colaborativo, implica tener información útil sobre otros usuarios que se encuentran en el sistema, las actividades colaborativas que ocurren y las interacciones que se presentan en un determinado espacio de tiempo.

2. Problemáticas principales de los sistemas colaborativos

En esta sección se identifican las problemáticas que se han encontrado durante la investigación y actualización del estado del arte sobre sistemas colaborativos. Estas problemáticas han surgido del análisis de diversos artículos que se han encontrado en el proceso de revisión de trabajos relacionados sobre las temáticas identificadas como claves para la colaboración entre personas. Dada la evolución deseada para AYLLU y la necesidad de adaptarlo a un ambiente educativo, se organizaron las problemáticas claves en las siguientes categorías: *awareness* de grupo, servicios de soporte a la colaboración, servicios cooperativos, manejo de *workflows*, servicios web, arquitecturas para *groupware*, ingeniería de la colaboración, movilidad, racionalidad, gestión de artefactos y recursos, gestión de grupos.

Así pues, en las secciones siguientes, para cada problemática presentada se realiza una introducción de las definiciones y aspectos que tienen en cuenta los diferentes autores, y una descripción de cada una de las problemáticas y los aspectos esenciales que cada problemática incluye. Finalmente, se presentan las diferentes preguntas y cuestionamientos que cada aspecto le aporta al proyecto AYLLU.

3. *Awareness* de grupo

Para describir esta problemática es necesario definir el concepto de *awareness*, el cual traducido textualmente como *sensibilidad* no alcanza para entender su alcance y utilidad. Según Young (2009), el *awareness* consiste en tener información sobre los usuarios que se encuentran en el sistema, las actividades que ocurren y las comunicaciones que se presentan en un determinado espacio. En un contexto educativo, el *awareness* puede ser utilizado para conocer el estado y el progreso de cada tarea o cada estudiante y tomar decisiones respecto a ese estado; por ejemplo, si hay una tarea no cumplida por algún estudiante, los mecanismos de *awareness* pueden informar al docente y tomar decisiones sobre si realizar una observación, una alerta o un llamado de atención.

En general, la sensibilidad al grupo de trabajo permite conocer los datos de cooperación y las diferentes actividades que realizan los usuarios de la herramienta en un momento determinado. En el contexto educativo, este *awareness* puede aprovecharse para conocer las formas de trabajo de un grupo de estudiantes, los mecanismos de comunicación más utilizados, los integrantes del grupo con los que hay mayor comunicación, los artefactos que son utilizados, con el fin de generar servicios colaborativos que se acoplen a las formas de interacción dentro del grupo de trabajo.

Por otra parte, se debe tener en consideración el análisis sobre la importancia de mantener las maneras de colaboración ya existentes antes de imponer nuevas formas de colaboración (Hirlehei y Hunger, 2011). En efecto, el tiempo invertido en aprender a utilizar una nueva herramienta afecta en gran medida el rendimiento y los resultados de los elementos relevantes al contexto. Adicionalmente, es necesario que el diseño de una aplicación de *groupware* tenga la posibilidad de integración con esquemas propuestos por herramientas existentes. La creación de una plataforma de *groupware* debe realizarse de tal forma que permita modificar y acoplar los servicios de una herramienta colaborativa a las necesidades de los usuarios. Es conveniente que esas herramientas sean comúnmente utilizadas por los usuarios, está demostrado que la colaboración entre las personas mejora cuando se manejan herramientas previamente conocidas y se elimina la necesidad de migrar a nuevos paradigmas de comunicación (Mogan y Wang, 2010).

Los interrogantes que surgieron para el desarrollo del proyecto fueron: ¿es suficiente usar métodos ya definidos por otros autores para conocer la forma en que cooperan los usuarios?, ¿o será necesario desarrollar métodos propios para realizar las tareas colaborativamente?, ¿cómo lograr que un usuario se sienta parte de un equipo de trabajo?, ¿qué información debe conocer un usuario de los otros usuarios? Las respuestas para estas preguntas se orientan a la generación de servicios cooperativos inspirados de estrategias de colaboración ya validadas y que se adapten en lo posible a las formas de comunicación ya existentes. El diseño de estos servicios parte del conocimiento de las actividades de los usuarios buscando identificar patrones de interacción que permitan la automatización de procesos de colaboración. En el proyecto AYLLU se realizó el análisis de estas interacciones, definiendo las más comunes en el contexto educativo, las formas de crear nuevos servicios y la reutilización de servicios ya existentes. Para estas interacciones se identificaron y crearon mecanismos de *awareness* para grupo que permiten que una persona conozca el nivel de avance de las tareas de sus compañeros de grupo en el marco de un servicio cooperativo.

4. Servicios de soporte a la colaboración

La colaboración, según Cantor (2006), define las interacciones entre los diferentes usuarios de un sistema con base en las acciones que cada uno de ellos lleva a cabo para cumplir con sus objetivos. En un contexto educativo, por ejemplo, la colaboración incluye la asignación de tareas entre diferentes miembros de un grupo de estudiantes que buscan llegar a una conclusión o resultado. Dichos estudiantes se basan en la comunicación para generar una correcta coordinación de sus acciones, además de mecanismos que les permiten acceder a recursos compartidos y llevar a un feliz término su objetivo.

Un servicio que soporte la colaboración debería estar enfocado a ser recurrente, es decir, a brindar servicios repetitivos, y no *ad hoc* (Vreede y Briggs, 2005). Los servicios temporales que se generen únicamente durante una interacción no deberían ser considerados por dos razones: la primera es que al diseñar un servicio *ad hoc* que cumpla satisfactoriamente con los objetivos de la interacción solo beneficiará el proceso de colaboración ocasionalmente; la segunda razón, para evitar los servicios *ad hoc* es que los participantes no están obligados a dejar los resultados y experiencias de su interacción, debido a que es improbable que el servicio se repita. A partir de este análisis se generan los siguientes cuestionamientos: ¿qué tan conveniente es permitir los servicios *ad hoc*? y ¿se puede a partir del análisis de un servicio *ad hoc* generar un servicio recurrente? A partir del estudio de los sistemas CSCW se identificaron dos tipos de servicios claves para lograr un trabajo cooperativo más efectivo: los espacios interpersonales y los espacios de trabajo compartidos. Estos servicios son descritos a continuación.

Los espacios interpersonales son espacios que aumentan la confianza entre los usuarios y mejoran la colaboración, debido a que proporcionan información de los usuarios, tal como su ubicación, intereses y aspectos no relacionados directamente con el objetivo común (Yong, 2009). Estos espacios pueden facilitar una eficiente asignación de tareas y la creación de equipos con personas en diferentes lugares. Son muy similares a las redes sociales, donde se provee información a un grupo de personas pertenecientes a la misma red, en la que los usuarios pueden pertenecer a varios grupos de trabajo y compartir sus temas de interés con otros. Como ejemplo de uno de los beneficios de los espacios interpersonales, se toman dos personas dentro de un grupo de trabajo que se encuentran localizadas en diferentes zonas horarias; la distribución de trabajo entre estas personas puede verse influido por la diferencia de hora, de tal manera que mientras uno de los integrantes termina su horario de trabajo en un determinado tema, el otro inicia su horario laboral y continúa el trabajo de su compañero.

Los espacios de trabajo compartidos (*shared workspaces*) establecen una serie de tareas que deben ser cumplidas, los roles que deben cumplirlas y el progreso que cada tarea tiene. Estos espacios se complementan con los espacios interpersonales debido a que permiten a los usuarios no solamente interactuar, sino también controlar el trabajo asignado a cada persona. Un espacio de trabajo compartido permite realizar el seguimiento de los diferentes objetos relevantes para el contexto de un grupo y permiten el acceso concurrente a ellos; estos objetos pueden ser archivos, objetos virtuales de aprendizaje, entre otros. Dichos espacios ponen a disponibilidad de los miembros del grupo, además de los objetos, información sobre sus modificaciones, su estado y si la modificación del objeto ha alcanzado el estado deseado para cumplir con una tarea o subtarea.

Al analizar estos espacios surgen cuestionamientos sobre la manera más efectiva de distribuir y almacenar los datos, ¿será conveniente almacenar los datos de manera centralizada? ¿O será más conveniente realizar un intercambio de archivos estilo P2P (*peer to peer*)? En el marco del proyecto, como respuesta a estos interrogantes, se diseñó un mecanismo de gestión distribuida de recursos regulada por espacios de trabajo definidos para el contexto de cada servicio cooperativo.

5. Arquitecturas de *groupware*

Las arquitecturas de *groupware* son diseñadas para proveer servicios cooperativos a través de diferentes estrategias, bien sea a través de plataformas o de patrones que faciliten la colaboración entre los usuarios. Una de las primeras arquitecturas de *groupware* desarrollada fue Lotus Notes, una aplicación que trabajaba en comunión con un servidor que hacía las veces de integrador de la información. El servidor Lotus Domino permitía el acceso concurrente de personas a diferentes tipos de archivos y herramientas que facilitaran su trabajo en equipo.

Es importante acoplar las nuevas herramientas de colaboración a las herramientas o métodos ya existentes dentro de una organización o grupo de trabajo preestablecido (Hirlehei y Hunger, 2011). Por esta razón, se sugiere no construir una aplicación como tal (similares a la antes mencionada), sino un *framework* sobre el cual cada equipo de trabajo pueda elegir aquellos servicios que más le convengan y que mejor se acoplen con las herramientas anteriormente manejadas por los usuarios, para obtener como resultado una aplicación que satisfaga las necesidades particulares de la organización o grupo de trabajo. Por otra parte, no basta solamente con definir la herramienta que mejor se acople a las necesidades de un grupo de trabajo, sino que también es importante y fundamental la asignación de diferentes

roles a los usuarios, permitiendo un mejor conocimiento en las responsabilidades y recursos con los que cada uno cuenta (Zhu, 2010). Los roles son la base para la colaboración, la coordinación y la asignación de tareas y recursos. Los roles permiten determinar, incluso en forma dinámica, el trabajo que deben realizar los diferentes miembros del equipo, estableciendo con mayor claridad las tareas puntuales y sus responsables. En el proyecto AYLLU, los servicios cooperativos definen interacciones entre roles que posteriormente serán asignados en forma dinámica a las personas del grupo de trabajo.

Existen *frameworks*, como AGORA (Martínez-Prieto *et al*., 2008), que soportan los aspectos principales de la colaboración como la flexibilidad, el *awareness* de grupo, los mecanismos de comunicación, entre otros. Dichos *frameworks* se enfocan principalmente en el uso de diferentes herramientas como mecanismo para organizar los servicios cooperativos; en particular, se hace énfasis en herramientas consideradas fundamentales, primordialmente los mecanismos de comunicación y los espacios de trabajo donde se comparte información (Cantor, 2006; Yong, 2009; Martínez-Prieto *et al*., 2008). El principal objetivo que deben buscar los diferentes *frameworks* orientados a grupos es mantener una conversación fluida entre los usuarios y facilitar su interacción mediante múltiples mecanismos, sean síncronos o asíncronos, de comunicación que determinen políticas de comunicación que puedan ser reinterpretadas como mensajes estructurados y con una semántica específica, haciendo que los mensajes estén estandarizados y puedan ser llevados a otros contextos. Sin embargo, según algunos autores, la creación de plataformas de *groupware*, aunque brindan gran cantidad de herramientas para la interacción del usuario, no son una solución que cumpla completamente con los requisitos de un CSCW (Palmer y Fields, 1994).

Es importante, entonces, realizar una investigación sobre qué servicios o características son claves en un ambiente CSCW para lograr la construcción de un *software* colaborativo más robusto. En el proyecto AYLLU los elementos diferenciadores incluyen la utilización de agentes para soportar las interacciones y asistir a los usuarios en sus tareas, además de la implementación de mecanismos de adaptación que permitan personalizar el trabajo de acuerdo con la información como el perfil del usuario y su contexto y la gestión distribuida de recursos basada en un esquema de espacios de trabajo dinámicos. La arquitectura general de la plataforma AYLLU se muestra en la figura 1, en la cual se puede observar cómo el componente de adaptación complementa al de cooperación.

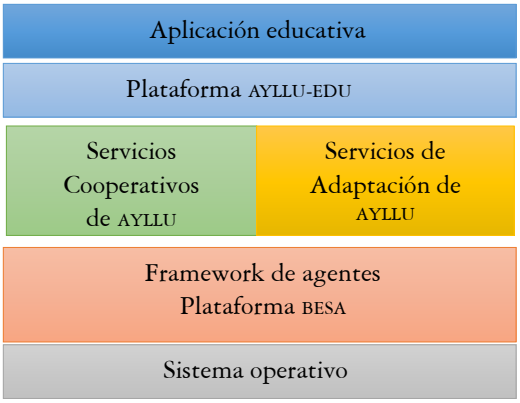


FIGURA 1. DISEÑO ARQUITECTURA BASE DE AYLLU INCLUYENDO LA CAPA DE APLICACIÓN AYLLU-EDU.

Fuente: elaboración propia.

Una aplicación e incluso un *framework* orientados a soportar *groupware* deben contar con buenas prácticas de implementación. Si bien no se recomienda proponer una herramienta que reemplace las ya utilizadas por los integrantes de un grupo de trabajo (Hirlehei y Hunger, 2011), es necesario tener en consideración las ventajas de construir un *software/framework* que tenga numerosos módulos desacoplados que presten servicios lo más independientes posible pero con la posibilidad de ser integrables entre ellos sin generar conflictos (Hirlehei y Hunger, 2011; Derrick *et al*., 2013). En la literatura pueden encontrarse algunos patrones que pretenden ser una guía de implementación para servicios cooperativos y se enfocan en el uso de herramientas como *workflows* y servicios orientados a roles (Vreede y Briggs, 2005; Knuplesch *et al*., 2013). En AYLLU cada servicio cooperativo es definido por un patrón denominado protocolo de interacción; en la plataforma base se crean servicios generales aplicables a múltiples dominios. Adicionalmente, se deja abierta una capa especializada, en la cual se implementen estrategias validadas y de común aplicación en el dominio; por ejemplo, en el caso de educación se implementan patrones inspirados de las estrategias presentadas en el capítulo 3 de este libro.

6. *Workflows*

Un *workflow* es un procedimiento predefinido para ser usado por actores humanos dentro de un contexto organizacional en el que rigen normas y valores mientras desempeñan un trabajo colaborativo (Knuplesch *et al*., 2013). Los *workflows* en

general buscan integrar los flujos de energía, flujos de información, roles organizacionales, tiempos, recursos, entre otros, de un determinado proyecto u objetivo que debe ser alcanzado. Inicialmente los *workflows* se vieron limitados a registrar la información de un proyecto en curso, pero rápidamente avanzaron hasta convertirse en herramientas de control que pueden ser utilizadas ampliamente para incrementar el trabajo en equipo. En un sistema de *workflows* se define un conjunto de patrones recurrentes de comportamiento, asimilables a procesos repetitivos, regidos por unas normas y procedimientos establecidos que tienen que cumplir y seguir aquellas personas que deseen hacer uso de dicha rutina para lograr un objetivo. Las reglas y pasos establecidos por un *workflow* que apoya un sistema *groupware* tienen en cuenta mecanismos de comunicación, cuyo uso facilita la interacción entre personas y les permite alcanzar una comunicación sin ambigüedades y coordinar de manera efectiva un trabajo colaborativo.

Los *workflows* orientados a roles definen tareas y procesos para los diferentes roles que tiene disponible el sistema; las tareas pueden ser cumplidas no por una persona en particular sino por un rol. Esta forma de definir tareas facilita en gran medida su asignación y la distribución de trabajo en el caso tal que se modifique la estructura de una organización (Anzures-García *et al*., 2010); la organización depende de los roles, mas no de las personas específicas que asumen estos roles. De igual forma, es importante el manejo sesiones para definir la estructura organizacional de un equipo de trabajo basado en las funciones que desempeñarán cada uno de los miembros del equipo. Las obligaciones que son asignadas a cada rol limitan y orientan las acciones de dicho rol dentro de una estructura o grupo de trabajo. Una tarea es una actividad llevada a cabo por una o muchas personas (que pueden asumir uno o varios roles) para alcanzar un objetivo. Estas tareas utilizan recursos compartidos, los cuales representan los recursos necesarios por los diferentes usuarios para cumplir satisfactoriamente con una tarea.

En conclusión, una herramienta del estilo *workflow* orientada a roles puede ser de gran utilidad por su flexibilidad. Dentro del contexto del proyecto se incluye un servicio cooperativo basado en esta aproximación; sin embargo, el contar con esta opción genérica y abierta no excluye que se incluyan servicios específicos inspirados en los existentes en los sistemas multiagente o en las estrategias de colaboración de dominio. En todos los casos el modelo de servicios se basa en el concepto de roles organizacionales para la creación dinámica de equipos de trabajo y la asignación de responsabilidades.

7. Servicios web

Muchos *frameworks*, hoy en día, ofrecen la posibilidad de una interfaz vía web con el fin de asegurar la colaboración entre personas en diferentes lugares, con unos requisitos mínimos del sistema como una conexión a internet y un navegador web. Dichos *frameworks* contienen funcionalidades almacenadas en uno o varios servidores web, con interfaces que definen *web services* que pueden ser utilizados e integrados en una aplicación web creada según las necesidades de los usuarios. La principal ventaja de los servicios web es la utilización de un lenguaje estándar que puede ser fácilmente traducido, en la práctica, a muchas herramientas que utilizan nuevas tecnologías orientadas a la web e incluso puede ser aplicado a herramientas existentes diseñadas y creadas con tecnologías anteriores que no estaban pensadas para funcionar en la web. En el ambiente empresarial se utilizan en gran medida los servicios web para reutilizar las herramientas habituales con las que la empresa trabajaba (sistemas *legacy*) sin la necesidad de hacer una reimplementación (Gomes *et al*., 2011).

Dada esta ventaja de la reutilización de herramientas existentes surgen las siguientes preguntas: ¿son los servicios web una manera adecuada de integrar los servicios ofrecidos en AYLLU? ¿Los servicios web deberían ser accedidos a través de las herramientas ya utilizadas por las personas para la colaboración? Los servicios web parecen ser una forma de alcanzar el objetivo propuesto, sin embargo, es importante mencionar que, debido a las limitaciones de los estándares y protocolos web, no es posible guardar el estado de los usuarios e información como los gustos, preferencias e intereses, que faciliten a la herramienta cumplir con el objetivo de ser adaptativa, como lo establece la arquitectura general propuesta (figura 1). Por estas razones, en el proyecto se adopta el paradigma de los sistemas multiagentes, tanto para el modelamiento de servicios de colaboración, como para su implementación; adicionalmente, cabe resaltar que el paradigma de agentes es más amplio y flexible, con un nivel de abstracción más alto y cercano a la solución de problemas que implican cooperación. En el futuro podrían utilizar los servicios de AYLLU como *web services*, aunque posiblemente se limitarían algunas de sus funcionalidades.

8. Ingeniería de Colaboración

La Ingeniería de Colaboración se define como un método facilitador para el diseño de esquemas de interacción entre personas. Para que exista una colaboración correcta es necesario el uso de protocolos de interacción bien estructurados, los cuales

deben ser recurrentes; en consecuencia, no cabe la posibilidad de procesos *ad hoc*. Dichos protocolos de interacción deben incluir en su definición los diferentes roles que pueden tomar los usuarios, cada rol con sus permisos sobre recursos y sobre las actividades que se deben realizar (Vreede y Briggs, 2005). Además, en esta propuesta se definen procesos en común que se concretan a través de cinco pasos asociados al diseño de un servicio cooperativo:

- Divergir: evolucionar para pasar de tener una menor a una mayor cantidad de conceptos.
- Converger: entender y enfocarse en un concepto particular que sea prioritario entre otros conceptos.
- Organizar: aumentar el nivel de entendimiento de las relaciones entre conceptos.
- Evaluar: aumentar el nivel de entendimiento del beneficio de los conceptos hacia la obtención de una meta común relativa a uno o más criterios.
- Generar consenso: aumentar el nivel de acuerdo entre los involucrados (*stakeholders*) para llegar mutuamente a compromisos acordados.

Estos pasos permiten realizar un proceso de colaboración entre personas que puede ser llevado a la construcción de servicios cooperativos. Por ejemplo, en un contexto educativo pueden aplicarse todos estos pasos durante el desarrollo de un proyecto en una asignatura. Divergir es el proceso que el estudiante hace cuando el profesor muestra los diferentes conceptos dentro de la asignatura. Posteriormente, el estudiante debe elegir un tema de interés, de todos los que hay, para generar su proyecto; en este punto, el estudiante está realizando el proceso de converger. Una vez el estudiante ha definido los elementos clave de su objeto de estudio, establece las relaciones entre dichos temas, y la manera en la que se aplican a su interés y realiza el proceso de organizar. Finalmente, el docente hace el proceso de evaluación y determina si ha alcanzado satisfactoriamente o no el entendimiento y realiza una retroalimentación, con la que se genera un consenso. Surge entonces la pregunta sobre la forma en la que estos subprocesos de colaboración se interconectan, es decir, si pueden ser ofrecidos como servicios colaborativos individuales que se juntan para ofrecer un servicio compuesto. En el proyecto AYLLU, los servicios se inspiran de estrategias colaborativas ya existentes que, si bien podrían relacionarse con el marco de la Ingeniería de la Colaboración, por sí mismos incluyen algunos de estos pasos, pero utilizando la nomenclatura que los inspira.

9. Servicios cooperativos

Existe una serie de servicios cooperativos que se consideran en múltiples fuentes que mencionan trabajo colaborativo entre personas a través de un *software* o una serie de herramientas que faciliten la interacción entre ellas. En varias propuestas se observan herramientas que cuentan con una serie de servicios que ayudan al usuario, como por ejemplo, servicios de comunicación síncronos y servicios asíncronos como foros, correos electrónicos, entre otros (Cantor *et al*., 2006; Yong, 2009; Jeners *et al*., 2009). Aunque muchos de los servicios cooperativos se enfocan en la comunicación de información, existe otro tipo de servicios como lo son *Shared Workspaces* e *Interpersonal Spaces* (Yong, 2009), los cuales ya fueron introducidos anteriormente este capítulo. El desarrollo de una tarea colaborativa no solo requiere contar con un espacio donde se compartan los objetos y se definan las tareas, sino también con un espacio donde se comparta información de los integrantes de los grupos de trabajo que utilizan la herramienta con el fin de generar confianza e incrementar la efectividad de la colaboración.

En un contexto educativo, los servicios cooperativos pueden verse como aquellas herramientas que facilitan la interacción entre personas y la generación de espacios de trabajo concurrentes que permiten a los estudiantes generar los resultados mediante un trabajo cooperativo. Las herramientas como talleres educativos, espacios virtuales de reunión, mecanismos de intercomunicación, entre otros, pueden ser vistos como servicios cooperativos. Como ya se mencionó, la clave para lograr una correcta interacción y colaboración entre personas es el uso de roles dentro de las interacciones y tareas; estos roles son asumidos por una persona, la cual debe interactuar mediante mecanismos de interacción definidos y estructurados (Yin *et al*., 2014). Para establecer dichos mecanismos estructurados que cumplan con una semántica establecida se crean servicios cooperativos. En esta problemática se han generado cuestionamientos sobre la forma de ofrecer un servicio cooperativo, debido a que puede interpretarse como un servicio general compuesto de varios servicios o una conjunción de servicios colaborativos más pequeños que pueden ser interconectados. En la plataforma AYLLU los servicios se pueden generar ya sea para implementar protocolos de colaboración o mediante composición de otros servicios ya existentes.

10. Movilidad

Es importante considerar la colaboración en un ambiente móvil. Dicho ambiente debe tener en cuenta los dispositivos cambiantes, su heterogeneidad y el entorno

en el que se desempeñan. La movilidad facilita la colaboración gracias a la facilidad de compartir información. Los aspectos más importantes dentro del contexto de movilidad son la ubicación de las personas en el espacio (dónde se encuentran), las personas que comparten ese espacio (con quién se encuentra), los recursos que se encuentran en ese mismo espacio (con qué se cuenta), y el contexto de acceso limitado por la calidad de la conexión y las capacidades del dispositivo utilizado.

Es importante notar que, en un contexto móvil, la ubicación del usuario no sea lo único a tener en cuenta, debido a que también hay que considerar aquellos elementos del entorno que constantemente están cambiando de posición y de estado como resultado de su interacción con el entorno (Schilit *et al*., 2014). Otro aspecto importante de la movilidad es la portabilidad de los recursos de cómputo, es decir, a diferencia de los equipos estáticos (o de escritorio) en donde el procesamiento de información ocurre siempre en el mismo lugar, la computación móvil depende, aún en sus funciones básicas, de lo que suceda en el entorno, de las capacidades del dispositivo y su estado. Incluso el usuario de un dispositivo móvil puede cambiar de estado y verse sujeto a aspectos externos que modifiquen su interacción con el dispositivo.

Al considerar las diferentes implicaciones que tiene la computación móvil, es importante también analizar cómo dichas implicaciones modifican la manera de colaborar de las personas y cómo estas pueden influir en la manera en la que se ofrecen servicios de colaboración por la plataforma. Existen patrones que definen una forma de compartir la información entre diferentes usuarios; paradigmas como P2P en conjunto con patrones tipo cliente servidor, repositorios centralizados o distribuidos, entre otros que pueden ser utilizados para la gestión de los recursos. La proximidad entre usuarios del mismo equipo de trabajo puede tener efectos sobre la colaboración. Las herramientas móviles pueden ayudar a coordinar espacios de interacción interpersonales mediante los diversos métodos de localización que poseen, permitiendo la ubicación de miembros del equipo de trabajo cercanos entre sí y generando posibles tareas colaborativas *in situ* con las personas frente a frente. Existen al menos tres tipos de interacción posible al tener proximidad entre diferentes dispositivos (Schilit *et al*., 2014):

- Aquellos elementos de entrada y salida que pueden interactuar con los dispositivos móviles (pantallas, impresoras, cámaras, entre otros).
- Objetos con los que los dispositivos ya se encuentran interactuando y son direccionados por procesos de *software*.
- El tipo de lugares sobre los que se busca información.

Cuando hay sistemas que tienen en cuenta la proximidad se da un problema de presentación de la información al usuario. Por ello, se deben tomar decisiones sobre si se muestran los lugares, dispositivos, personas, etc., en orden alfabético o en orden de distancia (del más cercano al más lejano). Aplicando este tema al enfoque educativo del proyecto, se presenta el problema de la presentación de los usuarios cercanos. Una posible solución es ordenar a los usuarios por proximidad o por orden alfabético, pero haciendo especial énfasis en aquellos usuarios del mismo grupo o con aquellos con los que haya una tarea colaborativa pendiente.

La ubicuidad ofrece la posibilidad de contar con la información en todo momento desde un dispositivo móvil. La disponibilidad permanente de información compartida puede ser útil dentro de un contexto de trabajo en grupo. En algunos trabajos se ha demostrado que la computación móvil permite hacer grupos de trabajo y aprovechar la forma en la que fluye la información. Debido a que los usuarios en un ambiente móvil cambian de posición y es posible que no puedan seguir reunidos en un mismo espacio, debe asegurarse la continuidad del trabajo gracias a la disponibilidad de la información en todo lugar. El modelo cíclico considera dos estados "trabajo autónomo" y "trabajo colaborativo"; la transición del trabajo autónomo al trabajo colaborativo se llama "involucrarse" y la transición del trabajo colaborativo al trabajo autónomo se llama "aislarse" (Herskovic *et al*., 2009). En este esquema, una vez los usuarios se encuentran reunidos generan conocimiento y se realiza el proceso de asignación de tareas que pueden ser realizadas dentro del grupo de trabajo colaborativo. En el instante en el que el trabajo colaborativo se termina, ocurre el aislamiento y empieza el trabajo autónomo; se desarrollan las tareas pendientes por realizar asignadas a cada rol. Este modelo implica un estilo de trabajo tanto síncrono como asíncrono enfocado hacia el desarrollo de una tarea. Para el modelo AYLLU la asignación de tareas se puede realizar en forma asíncrona y ubicua, sin necesidad de reuniones presenciales.

Uno de los requisitos de la movilidad es la disponibilidad de la información y las herramientas para el procesamiento en los lugares donde se encuentre el usuario. La ubicuidad de la información requiere contar con mecanismos para obtener la información. También hay que tener en consideración que debido a las limitaciones de los dispositivos móviles no es posible almacenar grandes cantidades de información o poseer gran velocidad de procesamiento. Para lograr este requerimiento de obtención de información existen diversas herramientas: paradigmas como P2P o el uso de servidores centralizados permiten la obtención de la información en cualquier momento; para resolver la problemática del procesamiento de la información se recurre a técnicas más complejas como agentes móviles, los cuales pueden

aprovechar los recursos disponibles en el entorno donde se encuentren para realizar una tarea determinada. Como ejemplo de este aspecto, se puede imaginar un trabajo de campo en donde un estudiante recolecta información geográfica de un terreno en un dispositivo móvil, el cual almacena la información en forma de una lista de puntos y coordenadas. Una vez finalizada la recolección de datos, el agente puede viajar a un dispositivo con mayor capacidad; por ejemplo, aprovechando la capacidad de despliegue gráfico del nuevo dispositivo donde se aloja, el agente realiza una representación tridimensional de las características de un terreno. Otra posible aplicación de los agentes en la distribución de la información es que el agente conozca la lista de personas involucradas en un grupo de trabajo y, al recibir la orden, envíe la información a dichas personas de manera que todos los usuarios tengan la información más reciente en tiempo real (Dewi *et al*., 2009).

11. Racionalidad y adaptación

La racionalidad en un entorno de *groupware* puede definirse como las interacciones que realizan elementos de *software*, como los agentes, los cuales actúan como elementos de *software* que emulan un comportamiento humano (Dewi *et al*., 2009). Estos agentes se comunican entre sí y facilitan la toma de decisiones, bien sea analizando las opciones existentes o tomando una decisión basada en criterios definidos por el usuario. En algunos *frameworks* se propone el manejo de diferentes componentes que actúan en representación del usuario al ingresar a un servicio colaborativo y le permiten una personalización (Hirlehei y Hunger, 2011). Dichos componentes interactúan mediante la ejecución de servicios cooperativos que facilitan la colaboración entre los usuarios; por lo tanto, es necesario que los representantes de cada usuario puedan guardar sus preferencias, su historial y tengan un empoderamiento controlado de las acciones que pueden realizar en nombre del usuario que representan.

Las técnicas propuestas por la Inteligencia Artificial pueden ser utilizadas para construir sistemas racionales capaces de manejar altos niveles de abstracción. En particular, los agentes pueden ser vistos como las entidades racionales y concurrentes básicas que componen un sistema (Tello *et al*., 2014). Estos agentes colaboran con el usuario brindándole información sobre, por ejemplo, en una búsqueda en internet, los vínculos que ya ha visitado y hacer una sugerencia sobre los vínculos que podría visitar. Por otra parte, un proceso de adaptación puede aumentar el nivel de inteligencia y autonomía de algunos de los componentes de una plataforma colaborativa. Uno de los usos de la adaptación es realizar una búsqueda refinada

basada en el perfil del usuario (Dewi *et al*., 2009). Además, una característica de los agentes en esta propuesta es que cuentan con una lista de amigos, los cuales pueden hacer sugerencias que el agente acepta para futuras búsquedas y permite refinarlas con más detalle. En efecto, la construcción del módulo de adaptación que hace parte de AYLLU beneficia, como se ha visto a lo largo de esta sección, en gran medida el uso de agentes que recolectan información, que, basados en el perfil del usuario, reinterpretan la información o la distribuyen de acuerdo con las necesidades.

12. Artefactos y recursos

Los artefactos son aquellos objetos, físicos o virtuales, que complementan la colaboración entre personas; también pueden asimilarse a los recursos producidos y consumidos a lo largo de un servicio cooperativo. Estos objetos pueden ser documentos, elementos multimedia, entre otros, que facilitan la realización de las tareas de las personas. En el caso educativo, el ejemplo más usual de estos recursos son los OVA (Objetos Virtuales de Aprendizaje), utilizados para soportar el aprendizaje. Los artefactos pueden ser considerados los objetos mediante los cuales se realizan las interacciones y las modificaciones entre los miembros del grupo y que generan los resultados esperados al finalizar las tareas que el equipo de trabajo tiene definidas para la obtención de un objetivo.

Por otro lado, un artefacto puede ser un elemento que tiene una evolución determinada por la necesidad del momento y que transmite información nueva o presenta una serie de pasos a seguir para alcanzar un objetivo. Dichos artefactos estáticos no son modificados frecuentemente y son reutilizados por los usuarios sin modificarlos, ya que son como una guía o una instrucción. Por otra parte, existen artefactos de colaboración, los cuales son usualmente transformados de manera concurrente para generar otro artefacto, es decir, tienen un estado inicial y mediante la intervención de los usuarios estos artefactos se modifican hasta alcanzar el estado final requerido. Tales artefactos son de utilidad bien sea para servir de entrada a una nueva tarea, como finalización de una actividad o simplemente como actualización del mismo artefacto (es decir, aumentar las funcionalidades de un artefacto original sin modificarlo totalmente) (Yong, 2009). En el contexto del proyecto AYLLU los artefactos son modelados como recursos compartidos almacenados en forma distribuida y accesibles a través de los espacios de trabajo definidos para la realización de un servicio cooperativo.

13. Gestión de grupos de trabajo

La gestión de grupos de trabajo, iniciando desde su creación, es uno de los factores más influyentes en una comunidad colaborativa, ya que es el grupo de trabajo precisamente el que ejecutará las tareas que se requieren dentro de este para cumplir los objetivos comunes. Para completar un objetivo se debe ir más allá de asignar tareas a diferentes personas y seguir el modelo "divide y vencerás", ya que también es necesaria la interacción de las diferentes personas en el contexto global del trabajo; es decir, la colaboración ofrece mejores resultados si los integrantes del grupo de trabajo no se limitan a hacer una lista de tareas, sino que también apoyan a otros en tareas en las que puedan colaborar (Tan *et al*., 2014). Aunque el aprendizaje colaborativo puede ser una forma efectiva de aprender, en la colaboración en línea y las aplicaciones colaborativas la efectividad del aprendizaje no está garantizada. En este trabajo se concluye que, tanto en la investigación como en la práctica, el aprendizaje colaborativo es un aprendizaje que mantiene a largo plazo el conocimiento obtenido y que genera la aplicación del conocimiento. Estas afirmaciones se sustentan en la evaluación de la calidad de los resultados que se obtienen al final de un trabajo colaborativo contrastados con los mismos resultados generados por un trabajo individual.

En este contexto, es importante establecer como objetivo la definición de las características de los usuarios y cómo la conformación de grupos entre ellos influye en la colaboración y los resultados. Dichas características, son otro punto clave de interacción con el módulo de adaptación, debido a que es necesario tener en cuenta múltiples elementos cambiantes del contexto para una generación dinámica de los grupos de trabajo. Como ejemplo aplicado al contexto educativo, al tener un grupo de trabajo en un curso se puede utilizar la adaptación para determinar -de acuerdo con las habilidades de cada uno de los integrantes del grupo de trabajo- el rol y la posición en la estructura de colaboración en la que existe un mayor aprovechamiento de las habilidades de los integrantes. Otro aspecto que se debe tener en cuenta es el análisis histórico de aquellos integrantes que, cuando forman un grupo, alcanzan los objetivos esperados con una calidad igual o superior a la esperada. Para AYLLU la gestión de grupos está muy ligada con los perfiles de los usuarios que componen el grupo general, por ello es importante definir ¿cuáles son los criterios que se utilizan para la conformación de grupos? y ¿cuáles elementos del entorno o del perfil de usuario deben ser tenidos en cuenta para la conformación de grupos? Las respuestas a estas preguntas se desarrollan en el modelo MATEO, que se expondrá en el capítulo 9 de este libro.

14. Conclusiones

La colaboración entre personas implica tener en cuenta una serie de aspectos relevantes. Los más importantes son los siguientes: *awareness* de grupo, servicios de soporte a la colaboración, servicios cooperativos, manejo de *workflows*, servicios web, arquitecturas para *groupware*, ingeniería de la colaboración, movilidad, racionalidad, gestión de artefactos y recursos, y gestión de grupos. La discusión presentada a lo largo de este capítulo ha permitido conocer y analizar el estado del arte en cada uno de estos aspectos, y sentar las bases conceptuales que sustentaron el desarrollo de la plataforma AYLLU, que será presentada en los siguientes capítulos.

Referencias

Anzures-García, M., Hornos, M. J., Paderewski, P. y Sanchez-Galvez, L. A. (2010). Service-Based Models to Provide Group Interaction and Dynamism in Collaborative Applications. *Networking and Services (ICNS), Sixth International Conference*, 67-72.

Cantor, O. J., Mancilla, L. E. y González, E. (2006). AYLLU: A Cooperative Approach to Groupware Development. *IEEE Int. Conf. Commun.*, *1*(c), 447-452.

Derrick, D. C., Read, A., Nguyen, C., Callens, A. y De Vreede, G. (2013). Automated Group Facilitation for Gathering Wide Audience End-User Requirements. *System Sciences (HICSS), 46th Hawaii International Conference*, 195-204.

De Vreede, G. y Briggs, R. O. (2005). Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference*, 17c-17c.

Dewi, A. R., Suryadi, D. H. S., Bernhard, J. y Mahardito, A. (2009). Study Case: Distributed Groupware and Collaborative Web Agents. *Software Engineering, WCSE, WRI World Congress*, *1*, 195-198.

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Boston: Addison-Wesley Longman Publishing.

Gomes, R. L., Rivera, G. J., Willrich, R., Lima, C. V y Courtiat, J. (2011). A Loosely Coupled Integration Environment for Collaborative Applications. *Syst. Man Cybern. Part A Syst. Humans, IEEE Trans.*, *41*(5), 905-916.

Herskovic, V., Ochoa, S. F. y Pino, J. A. (2009). Modeling Groupware for Mobile Collaborative Work. *Computer Supported Cooperative Work in Design, CSCWD,13th International Conference*, 384-389.

Hirlehei, A. y Hunger, A. (2011). In Designing an End-User Tailorable Synchronous Groupware.*Collaboration Technologies and Systems (CTS)*, 373-380.

Jeners, N., Budweg, S. y Prinz, W. (2009). Portal modules for Groupware systems. *Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom, 5th International Conference*, 1-10.

Knuplesch, D., Reichert, M., Pryss, R., Fdhila, W. y Rinderle-Ma, S. (2013). Ensuring Compliance of Distributed and Collaborative Workflows. *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 9th International Conference Conference*, 133-142.

Martínez-Prieto, M. A., Fuente, P. y Cuesta, C. E. (2008). AGORA: a Cooperative Services-Based Architecture for Groupware Environments. *Lat. Am. Trans. IEEE*, 6(4), 308-316.

Mogan, S. y Wang, W. (2010). The Impact of Web 2.0 Developments on Real-Time Groupware. *Social Computing (SocialCom), 2010 IEEE Second International Conference*, 534-539.

Palmer, J. D. y Fields, N. A. (1994). Computer Supported Cooperative Work. *Computer*, *27*(5), 15-17.

Schilit, B., Adams, N. y Want, R. (1994). Context-Aware Computing Applications. *Mobile Computing Systems and Applications, WMCSA. First Workshop*, 85-90.

Tan, C. S. S., Luyten, K., Van Den Bergh, J., Schöning, J. y Coninx, K. (2014). The Role of Physiological Cues during Remote Collaboration. *Presence Teleoperators Virtual Environ.*, *23*(1), 90-107.

Tello Leal, E., Chiotti, O. y David Villarreal, P. (2014). Software Agents for Management Dynamic Inter-Organizational Collaborations. *IEEE Lat. Am. Trans.*, 12(2), 330-341.

Yin, S., Haibin, Z., Xianzhong, Z. y Luming, F. (2014). Effective Approaches to Adaptive Collaboration. *Networking, Sensing and Control (ICNSC)*, 297-302.

Yong, L. T. (2009). Collaborative Awareness for Translation Groupware. *Information and Multimedia Technology, 2009. ICIMT '09. International Conference*, 47-51.

Zhu, H., Member, S., Hou, M. y Zhou, M. (2010). Establishing The Foundation of Adaptive Collaboration, pp. 546-554.

6
AYLLU: PLATAFORMA BASADA EN AGENTES PARA APOYO A LA COLABORACIÓN

Enrique González, Ángela Carrillo-Ramos, Alexandra Pomares Quimbaya, Luis Guillermo Torres, María Paula Arias

En este capítulo se presentan los componentes principales de la plataforma AYLLU. Cada uno de los componentes se introduce en forma general y en capítulos posteriores se realiza su presentación detallada. Antes de presentar la forma en que se organizan y articulan todos los componentes de AYLLU, se identifican los requerimientos más importantes que debe cumplir este tipo de plataformas. Los fundamentos del sistema propuesto se encuentran en el modelo de colaboración basado en agentes. Este modelo base se complementa con el sistema de personalización y adaptación considerando el contexto y la administración distribuida de recursos. Finalmente, se presenta el modelo de integración de los componentes de la plataforma AYLLU en el marco de una aplicación en el contexto educativo.

1. Introducción

El modelo de colaboración de AYLLU se inspira en los modelos de cooperación entre agentes. La idea que soporta la plataforma es proporcionar los mecanismos que faciliten la interacción entre las personas, de forma tal que su comportamiento sea similar al de los agentes racionales cooperativos. Este objetivo se alcanza haciendo que las personas sean apoyadas por agentes de *software* que trabajan como mediadores de la colaboración. De esta forma se garantiza que los protocolos de interacción usados con fines de colaboración posean semántica bien definida y estructura acorde con el objetivo de cada tarea colaborativa. Si bien este principio es la base del modelo colaborativo, como se analizó en el capítulo anterior, en situaciones del mundo real estas características no son suficientes para soportar los requerimientos del trabajo en

grupo. Por lo tanto, la arquitectura distribuida de AYLLU presentada en forma general en este capítulo, y en forma más detallada a lo largo del libro, no solo incluye los componentes que soportan la realización de tareas en forma cooperativa, sino que, además, provee mecanismos especializados para la personalización, la adaptación y sensibilidad al contexto, así como la gestión de recursos compartidos. Finalmente, es importante resaltar que la plataforma AYLLU es un *framework* de colaboración general que puede ser aplicado en dominios específicos. Así, este capítulo presenta la arquitectura general de AYLLU y su estructura es la siguiente: inicialmente, se describen los trabajos relacionados que motivaron la creación de AYLLU; luego, se presenta su arquitectura general haciendo énfasis en cómo se articulan múltiples componentes para soportar todas las facetas del trabajo colaborativo; posteriormente, son descritos los principales componentes de AYLLU. Finalmente, se presenta la aplicación de AYLLU a un caso de estudio en el contexto educativo y se concluye el capítulo.

2. Requerimientos para una plataforma colaborativa

En prácticamente todas las actividades humanas la colaboración juega un papel muy importante; nadie logra grandes metas sin la ayuda de otros. El logro de objetivos comunes es un factor fundamental en cualquier organización; las interacciones son complejas y se requieren sistemas que soporten y contribuyan al desarrollo de las interacciones que se llevan a cabo entre las personas. Los modelos de colaboración existentes ofrecen recomendaciones sobre cómo facilitar la cooperación entre los participantes en términos de comunicación, manejo de la información y conocimiento con el propósito de ayudar a lograr de manera conjunta el objetivo común. Un ejemplo de estos modelos es el *software* colaborativo que fomenta la colaboración entre los participantes. *Groupware* es un tipo de *software* colaborativo que facilita el uso de la tecnología (*hardware*, *software*, servicios y técnicas) y permite a las personas trabajar juntas (Rama y Bishop, 2006). Aprovechando los beneficios de utilizar *Groupware* para colaborar, Stewart *et al*. (1999) presentan *Single Display Groupware* (SDG), un modelo para apoyar el trabajo colaborativo entre las personas que están físicamente cerca. Este modelo proporciona un dispositivo para cada usuario con el propósito de que dos o más personas trabajen juntas en la misma tarea, accedan y modifiquen los datos al mismo tiempo. Otro enfoque para apoyar la colaboración plantea la necesidad de definir un rol de facilitador con el fin de coordinar el trabajo colaborativo (Kolfschoten *et al*., 2008). El facilitador es responsable del diseño e implementación de procesos de colaboración dentro del grupo de trabajo para mejorar la comunicación. A su vez, dicho rol elige la tecnología más adecuada para soportar

la interacción entre las personas, con lo que proporciona mecanismos para la colaboración. Desafortunadamente, puede ser riesgoso confiar todas las actividades de colaboración a un solo rol.

Uno de los modelos de colaboración más reconocidos es el CSCW (*Computer-Supported Cooperative Work*) (Rama y Bishop, 2006). Este modelo se basa en el estudio de cómo la gente usa la tecnología para trabajar conjuntamente en tiempo y espacio compartidos. CSCW tiene dos dimensiones principales: espacio y tiempo, que, a través de su relación, establecen cómo los participantes comparten información: a) mismo lugar - misma hora, b) mismo lugar - diferentes tiempos, c) diferentes lugares - misma hora y d) diferentes lugares - diferentes tiempos. Cada uno de estos escenarios requiere de diferentes herramientas que ayudan a los participantes a distribuir información. (ej., chat, videoconferencias, *blogs*, correos electrónicos, etc.). Un enfoque diferente centrado en las herramientas que mejoran la colaboración es We-LCoME (Ferretti *et al*., 2008). Esta herramienta se orienta a contextos educativos en los que se necesite cooperar y compartir recursos de Synchronized Multimedia Integration Language (SMIL). Sus usuarios son estudiantes y profesores, que pueden enriquecer los materiales de aprendizaje a través de la inclusión de elementos multimedia con el fin de generar una mayor participación de los estudiantes en su proceso de aprendizaje (*e-learning*) y la descentralización de la información. El contenido multimedia es editado colaborativamente por todos los usuarios, siguiendo el enfoque de la Wiki como editor web. Como puntos fuertes, este enfoque mejora el poder de la inteligencia colectiva y permite que todos los miembros puedan ser parte del proceso de aprendizaje a través de su contribución sobre el objeto de aprendizaje para ser distribuidos.

A partir del análisis de estos trabajos se hace evidente la necesidad de crear herramientas tecnológicas para soportar la colaboración entre personas. Esta herramienta se soporta en la comunicación, y para este fin ya existen herramientas basadas en diferentes paradigmas síncronos y asíncronos a través de variados dispositivos, a menudo en ambientes distribuidos y nómadas. Desafortunadamente, la información llega de múltiples fuentes en forma no estructurada y sin contexto, con lo cual, la cooperación se vuelve compleja e ineficiente. Por otro lado, la capacidad de compartir información y recursos no es suficiente; el paradigma actual de intercambio de mensajes en lenguaje natural no es lo más adecuado para lograr que un equipo logre metas comunes. Tomando en cuenta lo anterior, el modelo AYLLU propone que la comunicación sea bien estructurada y que, la intencionalidad y semántica sean explícitas en los mensajes; en resumen, que la colaboración entre personas imite los mecanismos de cooperación de los sistemas multiagente (FIPA, 2012).

Si bien los protocolos de los agentes se inspiran a menudo en los humanos, en el cotidiano las personas los usan en forma implícita y poco estructurada. La propuesta de AYLLU consiste en hacer que los protocolos se hagan explícitos; que cada mensaje tenga una semántica y un propósito claro. Gracias a esta visión, la cooperación entre personas se hace más efectiva; no obstante, en situaciones reales no es suficiente gestionar las interacciones, también es indispensable proporcionar mecanismos de adaptación (considerando el contexto), conciencia de trabajo en equipo y gestión de recursos en un ambiente distribuido y móvil.

3. Arquitectura general de la plataforma AYLLU

Considerando las necesidades al realizar un proceso colaborativo, una plataforma que lo soporte debería cubrir al menos tres aspectos principales:

- La comunicación y cooperación entre personas.
- La diversidad y el contexto.
- La gestión de los recursos compartidos.

En la figura 1 se ilustra cómo estos tres aspectos se conjugan en una comunidad de personas que colaboran. En un primer nivel se debe asegurar una comunicación estructurada y semántica en un marco en el cual los participantes tengan conciencia de grupo. En un segundo nivel, la plataforma debe proporcionar los mecanismos para personalizar y adaptarse (tomando en cuenta el contexto), de forma tal que se tenga en consideración la diversidad de la gente, debido a que no todas las personas tienen las mismas necesidades, ni acceden desde contextos iguales. En el tercer nivel se gestionan el acceso y el almacenamiento de los recursos que comparten las personas en forma concurrente; estos recursos se encuentran por la naturaleza del problema en un ambiente distribuido y nómada.

La comunicación facilita la cooperación directa e indirecta; sin ella, las personas no podrían alcanzar metas comunes. Los mecanismos sincrónicos (ej., chats) permiten realizar acuerdos colaborativos en forma directa. Por su parte, la comunicación asincrónica (ej., foros) tiene la ventaja de no requerir la disponibilidad simultánea de las personas para llegar a acuerdos sobre las tareas que se van a realizar. Si bien, la asignación de tareas y recursos se puede alcanzar por estos medios, es muy complejo lograr únicamente a través de ellos la consolidación de comunidades cooperativas. El modelo AYLLU propone un paradigma de comunicación basado en protocolos de interacción. En estos protocolos, los mensajes intercambiados, además de tener secuencias muy bien definidas, también incluyen en forma explícita una *performativa*

que facilita su comprensión y procesamiento. De esta forma, basándose en un protocolo de interacción, en AYLLU se pueden implementar servicios cooperativos, en los cuales las personas asumen roles con semántica clara en el contexto de las metas comunes que debe alcanzar el grupo. Dado que el nivel semántico de la comunicación es alto, es posible utilizar agentes de *software* como mediadores de los protocolos, asegurando así la participación oportuna de cada participante. Los agentes además proporcionan información del desarrollo de la tarea cooperativa, con lo cual se logra la conciencia de grupo (*awareness*). La sección 4 muestra con más detalle el modelo de cooperación de AYLLU.



FIGURA 1. ELEMENTOS FUNDAMENTALES DE UN AMBIENTE COLABORATIVO.

Fuente: Arias-Báez,Torres-Ribero, Carrillo, Pomares, y González (2012a).

Al trabajar en grupo una persona adquiere un rol. Gracias a este la persona adquiere responsabilidades y capacidad de acceso a los recursos que requiere. El proceso de asignación de roles debe manejar la diversidad, debido a que no todas las personas tienen las mismas habilidades y preferencias. De igual forma, se debe tener en consideración el contexto de la persona, debido a que el dispositivo y las características del sitio de acceso también son diversos. Por tal motivo, la plataforma AYLLU incluye mecanismos de adaptación que facilitan la personalización y contextualización. El componente AES, presentado en el capítulo siguiente, ha sido creado para este fin (Arias-Báez *et al*., 2011). Antes de realizar la solicitud de un servicio, este es enriquecido gracias al uso de los perfiles que coinciden con el usuario y su situación. Dichos perfiles permiten hacer un filtrado acorde con la persona y con el contexto. Por su parte, el componente MATEO, presentado en el capítulo 9, permite soportar la formación de equipos. Para lograr las metas es indispensable construir buenos equipos de trabajo. Este componente, desarrollado sobre AES, conforma

grupos de trabajo en forma adaptativa, teniendo en cuenta los perfiles de los participantes y su contexto. En la sección 4 se presentan las ideas que fundamentan la adaptación en la plataforma.

El tercer componente fundamental para el soporte de la colaboración es el gestor de recursos AYPUY, presentado en el capítulo 11, el cual ha sido diseñado para el manejo de los recursos compartidos por los participantes en un grupo. Este componente aprovecha al máximo la información derivada del contexto cooperativo y soporta sus características de movilidad y distribución. El acceso a los recursos se hace en forma segura, el control de acceso tiene en cuenta el rol que una persona juega en el marco de un servicio cooperativo. A cada servicio se le asigna un espacio de trabajo, en el cual se almacenan virtualmente los recursos. Estos son mapeados en el ambiente distribuido por una segunda capa de agentes. Finalmente, los recursos reales son almacenados por una tercera capa bajo la coordinación de las anteriores. La sección 5 muestra cómo se articulan estas capas del gestor de recursos.

Por otro lado, la articulación de los servicios cooperativos de AYLLU, el componente para formación de equipos de trabajo MATEO, el componente para servicios de adaptación AES y el manejador de recursos AYPUY conforman la plataforma de colaboración AYLLU. La figura 2 muestra cómo se organizan y articulan estos componentes.

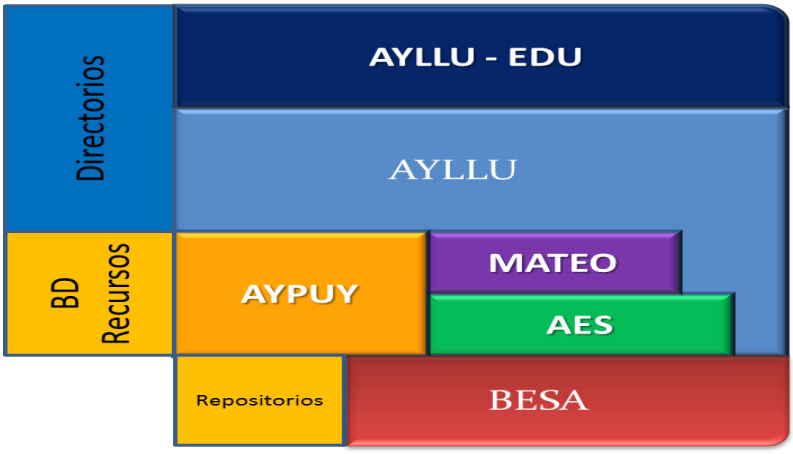


FIGURA 2. ORGANIZACIÓN DE LA ARQUITECTURA GENERAL DE AYLLU.

Fuente: Arias-Báez, Pavlich-Mariscal y Carrillo-Ramos (2013).

Todos los componentes de la plataforma AYLLU mencionados y mostrados en la figura 2 han sido diseñados usando el paradigma de agentes. Para su implementación se utilizó la plataforma de agentes BESA (González, 2003). Esta plataforma tiene la ventaja de que los agentes se construyen como un conjunto de hilos concurrentes; además, en cada agente hay un solo punto de entrada de eventos manejado por un mecanismo tipo *select* que facilita la gestión del no determinismo y la transparencia en la comunicación. En efecto, BESA es capaz de realizar el envío de eventos entre agentes en un ambiente distribuido y heterogéneo. En paralelo a la plataforma de agentes, se han construido los directorios de soporte a la cooperación, los perfiles para adaptación y las tablas para gestión de los metadatos y la información de localización asociadas a los recursos. La capa superior, en este caso AYLLU-EDU, es especializada para cada dominio de aplicación y se soporta en los servicios de las capas inferiores.

4. Modelo de colaboración AYLLU

La propuesta para la gestión de las interacciones de la plataforma AYLLU se basa en la creación de una serie de agentes que ayuden al usuario a seguir de manera correcta los protocolos de interacción que determinan un servicio cooperativo (González *et al*., 2012). Cuando un servicio cooperativo es instanciado, un Agente Manejador de Comunidad (CMA) es creado por medio del agente fábrica (FA). El agente CMA, especialista en el servicio que ha sido instanciado, se encarga de crear una serie de Agentes de Comunidad (CA), uno en representación de cada persona participante en un servicio cooperativo. Estos agentes CA son los que realmente están encargados de ejecutar los protocolos de interacción; es decir, a través de estos agentes CA se realizan los intercambios de mensajes estructurados entre los diferentes participantes en el marco del servicio cooperativo. Para lograr establecer comunicación con el usuario al que cada CA representa se utilizan otros agentes asociados a la sesión del usuario. El agente CMA se encarga de controlar la creación y destrucción de los CA y de monitorear la interacción entre ellos con el fin de proveer información sobre el estado actual del servicio cooperativo. Una vez se termina el servicio cooperativo, todos los CA y los CMA asociados al servicio desaparecen. Este comportamiento, en el que los agentes se crean al inicio de un servicio cooperativo y son destruidos al final de dicho servicio, da origen al concepto de "grupo volátil" de agentes.

Para cada usuario conectado a la plataforma existen básicamente dos agentes asociados a su sesión: el agente manejador de sesión SM y el agente representante RA. Adicionalmente, cada usuario cuenta con un agente interfaz IA en cada dispositivo

en el que el usuario ejecute un cliente de la plataforma AYLLU; esto supone que múltiples dispositivos pueden ser utilizados simultáneamente por un mismo usuario y que la sesión se mantiene activa incluso si hay intermitencia en la conexión. El agente manejador de sesión SM actúa como un puente entre el usuario y todos los agentes CA asociados con cada servicio cooperativo en el cual el usuario participa; cabe aclarar que lo más usual es que una persona participe simultáneamente en más de un servicio e, incluso, con equipos de trabajo diferentes. El agente SM aglutina todos los mensajes de los diferentes CA en cada servicio cooperativo y los transmite al agente representante RA. El agente RA puede responder en nombre del usuario a diferentes preguntas o peticiones; por ejemplo, si un usuario da permisos a su RA sobre su agenda de compromisos, el agente RA podrá manipular el calendario para agregar o modificar citas en la agenda del usuario. En los casos en los que el RA no posea una respuesta o no tenga la autorización para hacerlo y no pueda responder o tomar acciones en nombre del usuario, el agente RA redireccionará dicha petición a los agentes de interfaz IA. El agente IA mostrará la información al usuario y le solicitará al usuario que genere una respuesta o realice una acción.

Además de los agentes ligados directamente a los servicios cooperativos y a la gestión de las sesiones, la plataforma cuenta con algunos agentes de propósito específico. El agente administrador (AA) se encarga de realizar acciones administrativas, tales como la creación, actualización, eliminación de los diferentes usuarios, roles, habilidades, recursos y grupos de trabajo registrados en el sistema. Gracias a estas acciones se logra la gestión de los directorios de soporte a la colaboración y la inclusión de nuevos servicios cooperativos. El agente FA crea instancias de los diferentes servicios cooperativos disponibles en la plataforma; como ya se explicó, para ejecutar un servicio cooperativo se crea un agente CMA y este a su vez crea un grupo de agentes CA.

Como se ilustra en la figura 3, cada agente CA en un grupo volátil puede comunicarse con otros CA dentro del mismo grupo y reportar su estado al CMA cuando se requiera conocer el estado del grupo. Si un usuario necesita enviar un mensaje a otro usuario puede hacerlo mediante su CA que se encuentre dentro del mismo grupo volátil del receptor. El envío se realiza a través de su agente manejador de sesión SM. Una vez llegue el mensaje al CA del receptor, el mensaje es transmitido al destinatario a través del SM del receptor, el cual lo transferirá al RA del receptor. Si este RA puede responder por sí mismo lo hará, en caso contrario el mensaje se le enviará al agente IA activo que mostrará el mensaje al usuario receptor. Esta cadena de comunicación de soporte a la interacción ocurre para cada uno de los pasos del protocolo asociado a cada servicio cooperativo creado en el sistema.

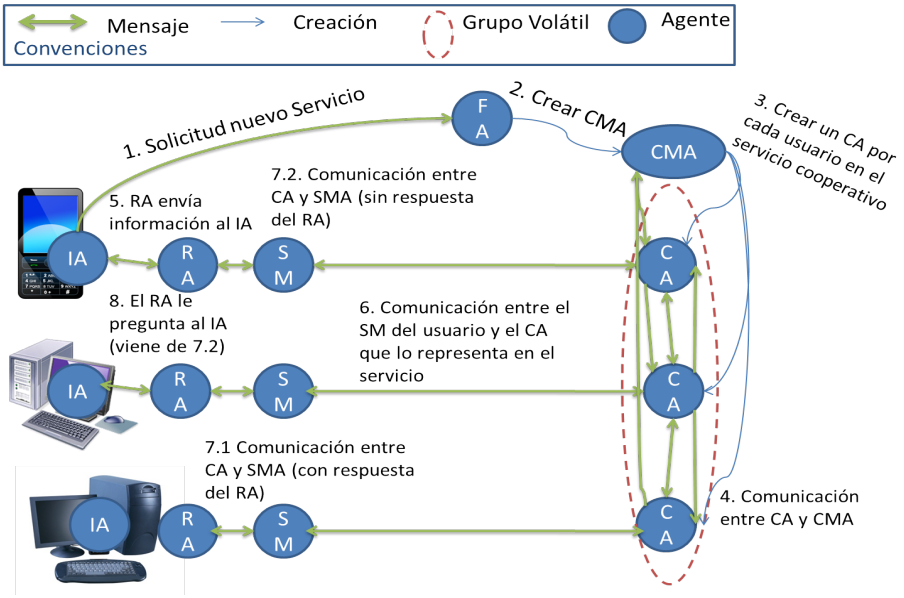


FIGURA 3. COLABORACIÓN MEDIADA POR AGENTES A TRAVÉS DE PROTOCOLOS DE INTERACCIÓN.

Fuente: Arias-Báez, Pavlich-Mariscal y Carrillo-Ramos (2013).

5. Personalización y adaptación considerando el contexto

Un reto importante en el manejo de información es la necesidad de tener acceso instantáneo a ella sin que esté limitado por características tales como la localización del usuario o su dispositivo de acceso. Las tecnologías actuales permiten a los usuarios tener acceso a una gran cantidad de información que no siempre les es relevante. De ahí surge la oportunidad de enriquecer la experiencia del usuario, es decir, adicionar a sus servicios criterios tales como sus preferencias y las características de su contexto de uso (ej., localización, tiempo de conexión y dispositivo de acceso) con el fin de ajustarle los resultados de la invocación de tales servicios a sus necesidades. Con todo lo anterior se podría reducir tanto el volumen de información recibido por el usuario como la sobrecarga cognitiva. Este proceso de enriquecimiento es conocido como adaptación.

Los sistemas que adaptan la información consideran características del usuario (preferencias, intereses, necesidades de información) y de su contexto cuando proveen servicios (Chebbine *et al*., 2005), lo que hace que la información sea presentada a

cada usuario de diferentes formas, orientada a sus necesidades y considerando todo aquello que puede afectar su interacción. Estos sistemas utilizan parámetros adicionales a los que tradicionalmente tienen los servicios que proveen; algunos de estos parámetros se obtienen usando procesos de alta abstracción (tales como inferencias, deducciones) y son utilizados para ajustar el servicio a las necesidades del usuario.

Con el ánimo de enriquecer los servicios de un sistema, en AYLLU se utiliza AES (*Agents for Enrichment Services*) (Arias-Báez *et al*., 2011), un *framework* que adapta la información en diversos contextos, incluso en ambientes móviles. AES puede ser usado por aplicaciones que necesitan enriquecer servicios considerando diversas características tanto del usuario como de su contexto. A continuación se explica el proceso de enriquecimiento de un servicio en AES:

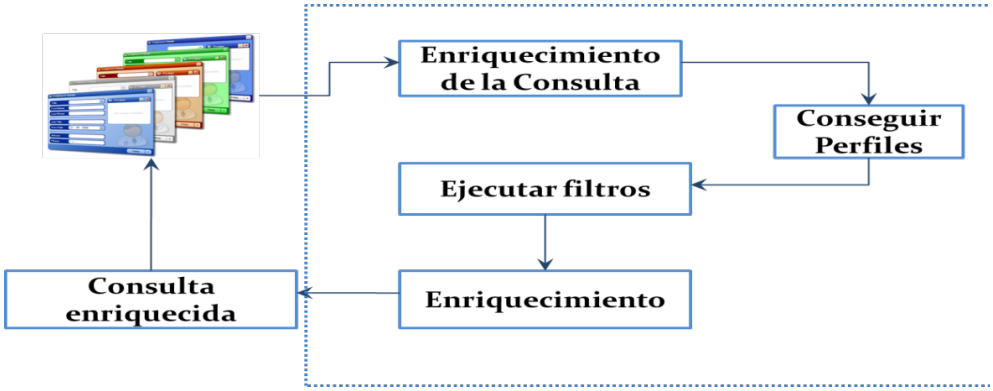


FIGURA 4. PROCESO DE ENRIQUECIMIENTO DE SERVICIOS DE AES.

Fuente: Arias-Báez, Pavlich-Mariscal y Carrillo-Ramos (2013).

Una aplicación externa (manejada por un usuario o por otro sistema) envía una "Consulta Inicial" a AES con el fin de que este *framework* la enriquezca (figura 4), modificándola o agregándole información que permita prestar un mejor servicio conforme al contexto del usuario. El módulo Conseguir Perfiles de AES obtiene los perfiles o características relevantes de la aplicación externa (usuario, contexto, dispositivo) e invoca funciones de filtro del módulo Ejecutar Filtros, en concordancia con los perfiles utilizados. El resultado es un conjunto de datos de alto nivel de abstracción que es utilizado para generar una Consulta Enriquecida a través del módulo Enriquecimiento; esta consulta será retornada a la Aplicación Externa.

Por ejemplo, en el dominio educativo un usuario alumno quiere seleccionar unos recursos (ej., archivos) que le ayuden a resolver una tarea asignada. Tal usuario accede al sistema a través de su teléfono celular. Los perfiles que se tienen en

AES son los de usuario y los de diversos dispositivos de acceso (celulares, PDA, tabletas, otros). A partir de estos perfiles se seleccionará el perfil correspondiente al del usuario y el perfil específico de su celular. El primer filtro recibirá el perfil del usuario que contiene información tal como su nombre, fecha de nacimiento, preferencias de despliegue y ciertas características de su personalidad (ej., si es comunicativo, crítico, riguroso, visual, verbal, etc.). El segundo filtro recibe el perfil de su teléfono celular que contiene información sobre la referencia, el fabricante, el nivel de batería y los formatos soportados (ej., texto, video, imágenes); ambos filtros separan, mediante el uso de reglas, la información que realmente es necesaria para la consulta que se desea enriquecer. El primer filtro deduce que el usuario, por sus preferencias de despliegue y sus características de personalidad, desea obtener archivos que contengan imágenes. El segundo filtro puede deducir, a partir de las características del celular (ej., pantalla pequeña, sensores, batería, memoria), que es un dispositivo que no puede desplegar archivos muy grandes (ej., solo archivos de tamaño en KB). A partir de la combinación de los dos filtros el sistema deduce que los archivos que le debe sugerir al usuario tendrían que contener imágenes; sin embargo, todo el archivo no debería tener un tamaño superior a 1MB. Para este ejemplo, se aplica primero el filtro de dispositivo (pues tiene más restricciones) y luego el del usuario. Vale la pena resaltar que el orden de aplicación de los filtros depende de las necesidades del usuario y de los requerimientos de la aplicación externa que utilice AES. Como aspecto final, es importante remarcar la posibilidad que ofrece AES para recibir realimentación, ya sea manual (proveniente del usuario) o automática (deducida por el sistema). Por ejemplo, el sistema puede preguntarle al usuario si los archivos le fueron de utilidad o evaluar el nivel de consulta que hace el usuario de los archivos que el sistema le propuso, todo esto con el fin de refinar cada vez los perfiles y los filtros que AES maneja.

6. Administración distribuida de recursos

AYPUY es un administrador de recursos creado para ambientes de colaboración distribuidos. Igualmente, soporta el almacenamiento de diversos tipos de recursos (ej., contratos, historias clínicas, objetos de aprendizaje) y garantiza su acceso cumpliendo con los requerimientos de confidencialidad, seguridad y escalabilidad de los sistemas que lo utilizan. Un recurso en AYPUY está compuesto de un conjunto de atributos $A = \{L, D, F\}$, donde L describe las características lógicas generales del recurso (ej., fecha de creación, autor, tipo de recurso), D establece las características lógicas relacionadas con el dominio (ej., salud, educación) al que pertenece

el recurso. Finalmente, F describe las características físicas de almacenamiento del recurso (ej., tipo de replicación, link de conexión); estas características le permiten a AYPUY establecer la estrategia de almacenamiento y acceso. Adicionalmente, los recursos tienen un conjunto de operadores O que determinan las acciones que pueden ser ejecutadas sobre los recursos. Los operadores provistos por AYPUY pueden ser extendidos para soportar los requisitos de un dominio específico.

El acceso a los recursos es controlado utilizando espacios de trabajo (ET). Un ET es un espacio lógico compartido entre usuarios participantes de una o más instancias de un proceso colaborativo. La figura 5 ilustra la relación entre instancias de procesos, recursos y ET.

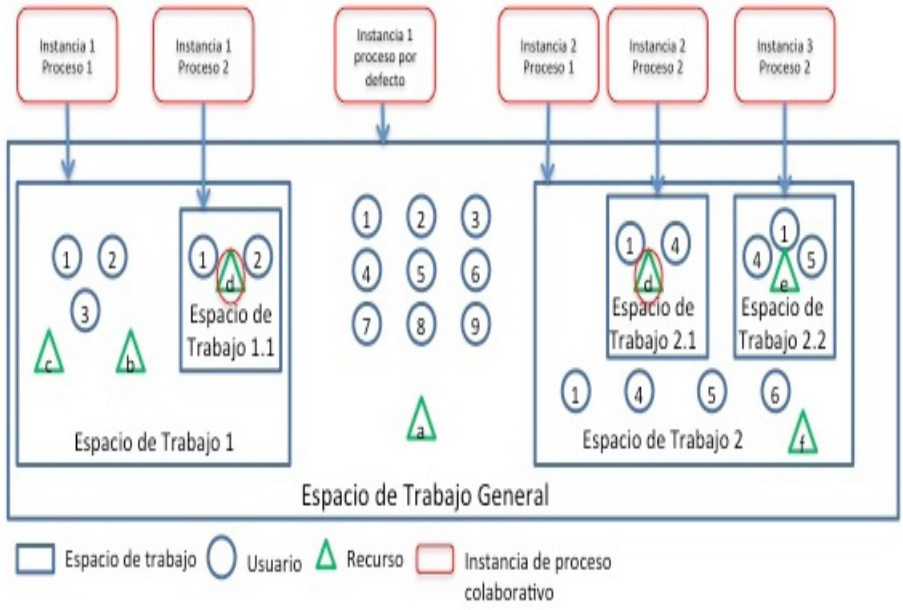


FIGURA 5. ESPACIOS DE TRABAJO EN AYPUY.

Fuente: Pomares-Quimbaya, Torres-Ribero, Carrillo-Ramos, Arias-Báez y González (2012).

Como puede observarse en la figura 5, cada sistema colaborativo tiene por defecto una instancia de proceso colaborativo. Esta representa al sistema y es asociada al ET General al que pertenecen todos los usuarios del sistema que tienen un rol específico. Si AYPUY requiere la especialización o modificación de los roles de acceso a los recursos, para todos o para un subconjunto de usuarios, este crea otro ET relacionado con una instancia de proceso específico. Por ejemplo, el ET 1 fue creado para

especializar los roles de los usuarios 1, 2 y 3. De forma similar, el ET 1.1 fue creado para cambiar el rol de los usuarios 1 y 2. En un ambiente empresarial, por ejemplo, el ET General representa a la empresa, el ET 1 corresponde a un departamento y el ET 1.1 a un proyecto específico. El acceso a los recursos es controlado por el rol que juega un usuario en la instancia donde el recurso es visible. En la figura 5, los usuarios 1 y 2 tienen acceso al recurso "d" con el rol que juegan en el ET 1.1. Por el contrario, tienen acceso a los recursos "c" y "b" con el rol que juegan en el ET 1. La política de acceso a los recursos en AYPUY usa los ET para asociar a los usuarios con los recursos.

AYPUY está implementado como un sistema multiagente. Los principales agentes y su interacción se ilustran en la figura 6. El objetivo del agente *WorkspaceAdmin* es administrar los ET junto con los roles de usuario y la lista de recursos visibles en cada uno de ellos. El agente *LogicResourceAdmin* establece la estrategia de distribución y replicación de los recursos a partir de sus características físicas de almacenamiento. Así mismo, mantiene un índice distribuido de características lógicas que usa para resolver consultas sobre los recursos. Finalmente, el agente *PhysicalResourceAdmin* gestiona los nodos que almacenan físicamente los recursos. Esta gestión incluye la traducción de consultas lógicas en consultas físicas.

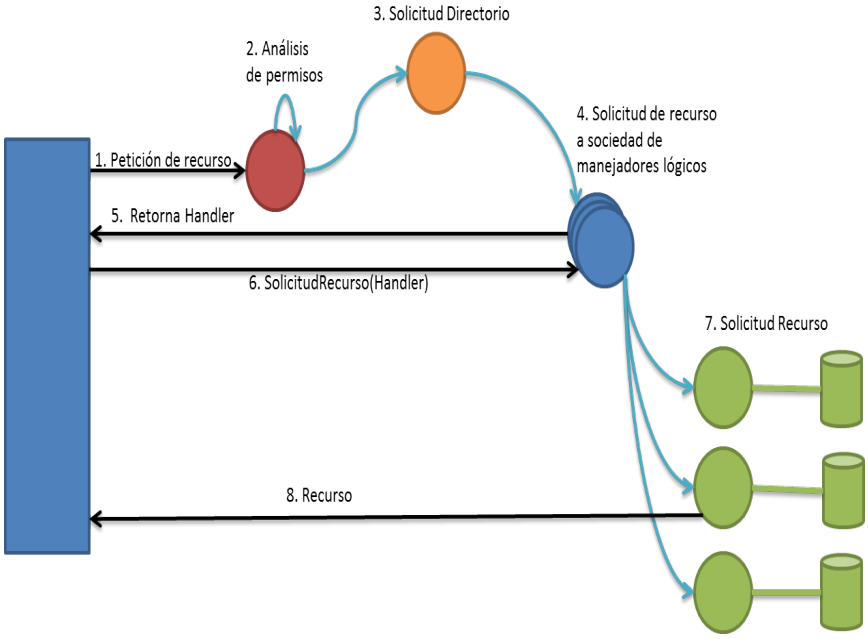


FIGURA 6. COMPONENTES DE AYPUY.

Fuente: Pomares-Quimbaya, Torres-Ribero, Carrillo-Ramos, Arias-Báez y González (2012).

7. Modelo de integración de los componentes de la plataforma AYLLU: aplicación a un caso de estudio en el contexto educativo

En la sección 2 se presentó una visión general de los componentes de la plataforma AYLLU y su articulación. El caso de estudio AYLLU-EDU, mostrado en la figura 7, incluye la construcción de múltiples servicios cooperativos en el contexto educativo. En esta sección, una vez ya conocidos con mayor detalle cada uno de los componentes, se ilustra el funcionamiento de la plataforma a través del ejemplo del servicio de presentación de un tema a los estudiantes.

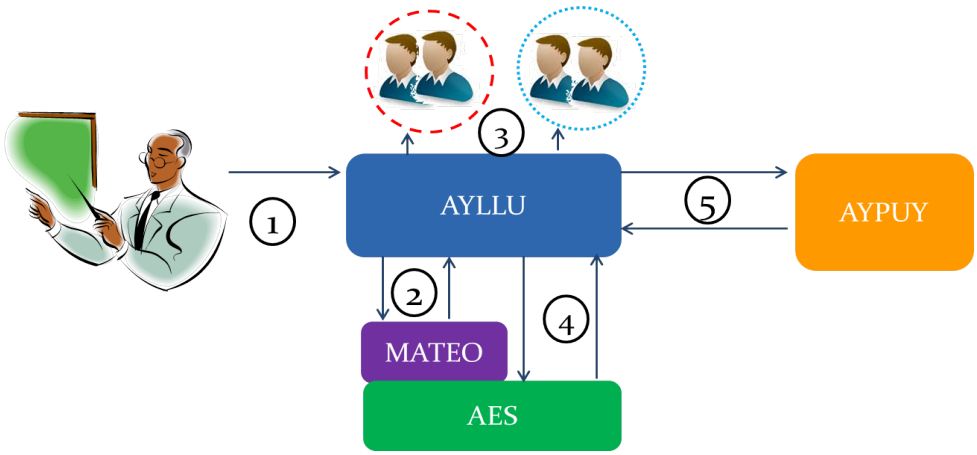


FIGURA 7. DESCRIPCIÓN DE LA INTERACCIÓN AYLLU-EDU.

Fuente: Arias-Báez, Torres-Ribero, Carrillo-Ramos, Pomares-Quimbaya y González (2012b).

Este servicio se modela mediante un protocolo de interacción en el que intervienen dos roles: el profesor y los aprendices. El profesor, una vez se han formado grupos heterogéneos, asigna un tema a cada equipo de trabajo; cada aprendiz recibe un OVA (Objeto Virtual de Aprendizaje), con el cual debe trabajar para aprender el tema; finalmente, todos los aprendices reportan al profesor que han realizado la tarea exitosamente; una vez todos han terminado, el servicio termina y el grupo está listo para continuar con una siguiente etapa de trabajo colaborativo alrededor del tema que les ha sido presentado. En este caso, un usuario profesor instancia el servicio y proporciona los parámetros requeridos como, por ejemplo, el número de equipos de trabajo, el grado de heterogeneidad de estos y la temática que se va a presentar, entre otros. El primer paso implica invocar a MATEO, que retorna los grupos acorde con los parámetros dados basándose en los perfiles de los estudiantes. Luego, a

nivel de cada equipo de trabajo se crea un conjunto de agentes de comunidad que representarán a cada uno de los participantes; todas las interacciones entre el profesor y los aprendices son mediadas por estos agentes. En este servicio el protocolo es muy simple, involucra solo dos tipos de mensajes: uno para enviar el tema a los aprendices y otro para notificar al profesor. Si bien estos son los mensajes principales, también se intercambian mensajes entre los aprendices con el fin de proporcionar información de *awareness*, que les permite saber cuántos aprendices del equipo de trabajo ya han realizado la tarea y cuánto tiempo queda para el cierre del servicio.

El agente que representa al profesor, antes de enviar el OVA a un estudiante, acude a los servicios de adaptación de AES. Teniendo en cuenta las preferencias y el modelo de aprendizaje de cada estudiante, así como también el dispositivo de acceso utilizado, AES determina las características del OVA que se va a suministrar. La consulta para encontrar el recurso queda así enriquecida, no solo se tiene la temática, sino además otras características del recurso que lo hagan más acorde a la persona y su contexto.

En el mensaje en el que se envía el tema a cada aprendiz no tiene sentido enviar un recurso, debido a que este puede ser de gran tamaño y puede estar replicado y localizado remotamente. En efecto, lo que realmente se envía es un *handler* que permite el acceso directo y seguro al recurso. El agente que representa al profesor, una vez tiene la consulta del OVA enriquecida por AES, invoca a AYPUY. Al interior de AYPUY se realiza el control de acceso teniendo en cuenta el rol y el espacio de trabajo asociados al servicio cooperativo. Si las restricciones de seguridad se cumplen, AYPUY busca y ubica el recurso más acorde con los metadatos de la consulta y retorna el *handler* que incluye, además de la información de ubicación del recurso, una contraseña cifrada requerida para acceder a este. Finalmente, cuando el agente que representa al aprendiz en el servicio recibe el mensaje, podrá recuperar directamente el OVA seleccionado desde el repositorio en el cual está almacenado, mediante una invocación a AYPUY.

El servicio termina cuando todos los agentes que representan a los aprendices han enviado el mensaje de notificación al profesor. Para evitar esperas infinitas y bloqueos, el protocolo incluye mecanismos de *timeout*. El servicio se cierra con la destrucción de todos los agentes de comunidad que fueron creados a su inicio.

8. Discusión y proyección

La arquitectura general de AYLLU presentada en este capítulo soporta los tres principales aspectos incluidos en una tarea colaborativa. La colaboración entre personas es un aspecto clave; en AYLLU, los protocolos de interacción permiten sobrepasar

las limitaciones en la semántica y la falta de estructura del intercambio básico de mensajes. El componente AES es el medio para alcanzar la diversidad de los participantes del equipo, tomando en cuenta perfiles de usuario con su correspondiente contexto. Finalmente, el administrador de recursos ha sido diseñado para soportar la naturaleza móvil y distribuida de los ambientes de colaboración actuales, brindando un acceso seguro y tomando ventaja de los roles de cooperación envueltos en el contexto de una tarea en equipo. Los componentes individuales ilustrados en la figura 2 han sido implementados y probados. Todos ellos están soportados en el *framework* BESA (González *et al*., 2003), creado en la Pontificia Universidad Javeriana (Bogotá) para la gestión de sistemas multiagentes que incluye: a) manejo de la arquitectura interna de un agente, ej., comportamientos concurrentes, b) comunicación basada en eventos asincrónicos entre agentes y c) su cooperación y estructura social. Actualmente, estos componentes han sido integrados en una plataforma unificada y fueron probados en el caso de estudio en el contexto educativo. Este caso de estudio permitió instanciar los componentes de AYLLU en la plataforma AYLLU-EDU.

Referencias

Arias-Báez, M., Pavlich-Mariscal, J. y Carrillo-Ramos, Á. (2013). Forming adapted teams oriented to collaboration: Detailed design and case study. *Rev. Avances en Sistemas Informática*, 10(1), 87-94.

Arias-Báez, M., Torres-Ribero, L., Carrillo- Ramos, Á., Pomares, A. y González, E. (2012a). AYLLU: *Architecture for Collaboration in a Distributed Environment.* Proc. Collaborative Technologies - IADIS Multi Conf. on Computer Science and Information Systems. Lisboa.

Arias-Báez, M., Torres-Ribero, L., Carrillo- Ramos, Á., Pomares, A. y González, E. (2012b). Platform based on agents for support to the collaboration of work teams. *7th Colombian Computing Congress (CCC).* DOI: 10.1109/ColombianCC.2012.6398037

Arias-Báez, M. P., Torres-Ribero, L.G., Carrillo-Ramos, A., Pomares-Quimbaya, A., Garzón, J. P., y González, E. (2011). *AES: A Generic Framework for Adapting Information in Multidimensional Contexts*. CLEI 2011. Quito.

Chebbine, M., Obaid, A., Chebbine, S. y Johnston, R. (2005). Internet Content Adaptation System for Mobile and Heterogeneous Environments. *2nd IFIP Int. Conf. on Wireless and Optical Communications Networks (WOCN)*, 346-350.

Ferretti, S., Mirri, S., Muratori, L., Roccetti, M. y Salomoni, P. (2008). E-learning 2.0: you are We-LCoME! *Proc. of the Int. Conf.*, 116-125.

FIPA. (2012). IEEE Foundation for Intelligent Physical Agents. Disponible en http://fipa.org.

González, E., Bustacara, C. y Ávila, J. (2003). BESA: Arquitectura para Construcción de Sistemas MultiAgentes. *XXIX Conferencia Latinoamericana de Informática (CLEI 2003)*.

González, E., Carrillo-Ramos, A., Pomares Quimbaya, A., Torres-Ribero, L. G. y Arias-Báez, M. P. (2012). AYLLU: Architecture for Collaboration in a Distributed Environment. *IADIS Collaborative Technologies Conference (CT 2012)*.

Kolfschoten, G. L., Niederman, F., Briggs, R. O. y De Vreede, G. J. (2008). Role Separation in Strategies for Collaboration Support. *Proc. of the 41st Annual Hawaii Int. Conf. on System Sciences. (HICSS-41)*.

Pomares Quimbaya, A., Torres-Ribero, L. G. Carrillo-Ramos, Á., Arias-Báez, M. P. y González, E. (2012). AYPUY: A resource management system for collaborative environments. *7th Colombian Computing Congress (CCC)*.

Rama, J. y Bishop, J. A. (2006). Survey and Comparison of CSCW Groupware Applications. *Proc. of the Research Conf. of the South African Institute of Computer Scientists and IT research in developing countries (SAICSIT '06)*.

Stewart, J., Bederson, B. y Druin, A. (1999). Single Display Groupware: A Model for Co-Present Collaboration. *Proc. of the Conf. on Human factors in computing systems (CHI '99)*.

7
AES: FRAMEWORK BASADO EN AGENTES PARA ENRIQUECER SERVICIOS COMPUTACIONALES EN CONTEXTOS MULTIDIMENSIONALES

Ángela Carrillo-Ramos, María Paula Arias-Báez, Luis Guillermo Torres-Ribero, Juan Pablo Garzón, Enrique González, Hervé Martin

1. Introducción

Debido a la gran cantidad de fuentes de información que existe, el usuario se ve inmerso en grandes volúmenes de información que en muchas ocasiones no se adecúan a sus preferencias, gustos o simplemente a lo que está buscando. Por tal razón, surge la necesidad de proveer información a la medida del usuario y para tal fin se debe adaptar la información; la adaptación es el proceso mediante el cual un servicio computacional (dirigido a un usuario) es enriquecido con una serie de características con el fin de ajustar la información obtenida de acuerdo con las características del entorno en el que se desarrolla la interacción del usuario con el sistema.

Sin embargo, no hay una herramienta computacional genérica que permita adaptar la información en cualquier tipo de escenario, ya que generalmente tales herramientas se encuentran ligadas de manera inseparable a un escenario en particular y no se puede aplicar a otros (Abeywickrama y Ramakrishnan, 2012; Koehl y Wang, 2012; Baur, 2015). Adicionalmente, no consideran características de entornos móviles (por ejemplo, el despliegue de la información en dispositivos móviles (Abeywickrama y Ramakrishnan, 2012; Cocea y Magoulas, 2011; Lamche *et al*., 2014) y los cambios de localización del usuario (Abeywickrama y Ramakrishnan, 2012; World Wide Web Consortium, 2015). En la literatura analizada no se encontró una herramienta computacional genérica que adapte la información en cualquier escenario, considerando, además, características de movilidad en entornos nómadas.

Por todo lo anterior surgió AES (Agentes para Enriquecer Servicios - *Agents for Enriching Services*), un *framework* genérico para adaptar información en contextos multidimensionales que considera la movilidad del usuario. AES puede ser usado por una aplicación que necesita enriquecer servicios con el fin de proveer al usuario información ajustada a sus necesidades y su contexto. Este enriquecimiento se puede ver como un proceso en el que una aplicación externa que quiere adaptar un servicio lo envía como una consulta inicial a AES. De allí AES obtiene información relevante para enriquecer la consulta, tal como las características del usuario, su dispositivo y su contexto, que son considerados el perfil del usuario, e invoca un conjunto de filtros que a través de reglas e inferencias afinan la información usada para enriquecer la consulta. Posteriormente, la consulta enriquecida es enviada a la aplicación externa que solicitó el servicio de adaptación. Finalmente, AES recibe retroalimentación de la aplicación externa con el fin de conocer si el enriquecimiento fue satisfactorio y así actualizar la información de los perfiles que contribuyeron a enriquecer la consulta. La estructura de este capítulo es la siguiente: la sección 2 presenta la arquitectura del *framework* AES; el formalismo de cada uno de sus componentes es presentado en la sección 3. Luego, la sección 4 presenta dos casos de estudio conceptuales y, finalmente, la implementación de AES es el tema central de la sección 5.

2. Arquitectura de AES

ASMA (Arquitectura para Sistemas MultiAgentes) es una plataforma para el desarrollo de aplicaciones basadas en sistemas multiagentes, la cual es soportada sobre BESA (González *et al*., 2003), que incluye: a) arquitectura interna de un agente, b) comunicación entre agentes basada en eventos asincrónicos y c) cooperación y estructura social entre los agentes. Tanto ASMA como BESA facilitan el desarrollo de aplicaciones complejas porque dividen el problema en módulos independientes autónomos, cuyas interacciones son estructuradas y claramente especificadas; además, permite el despliegue distribuido del sistema.

En el marco de ASMA surge la necesidad de brindarle al usuario información ajustada a sus características y necesidades, así como a su contexto. En este trabajo, contexto se refiere al conjunto de características que pueden influir en la interacción del usuario con el sistema (Arias-Báez *et al*., 2011). De allí nació la idea de crear un *framework* que considere diversas características pertenecientes al usuario, su dispositivo de acceso y en general de su contexto, con el fin de brindarle información relevante en el momento, en el lugar que la necesite, a través de su dispositivo de acceso (móvil o no), considerando siempre quién es el usuario y

qué es lo que verdaderamente necesita de un sistema o fuente de información. Tal *framework* fue creado en la Pontificia Universidad Javeriana y se denomina AES. Este *framework* de adaptación genérico es basado en agentes y puede ser utilizado como instancia para otros proyectos que necesiten enriquecer servicios con el fin de proveer al usuario con información adaptada. Para su utilización deben definirse los mecanismos de adaptación y los filtros a considerar con el fin de generar una nueva consulta enriquecida. A continuación se presenta el modelo de AES con sus entradas, salidas, componentes y relaciones (figura 1).

2.1. *Entradas y salidas*

Las entradas del sistema son la consulta que realiza el usuario al módulo de adaptación (Q). Adicionalmente, ingresa información externa (IA Contexto) como el tipo de dispositivo con que accede el usuario, las características del entorno en el que se desenvuelve la interacción del usuario con el sistema (tales como localización, dispositivo de acceso, entre otros). Debido a que las variables que se consideran del contexto pueden verse afectadas por ciertas limitaciones o restricciones, la interacción del usuario con el sistema puede ser diferente. Por tal razón, el sistema recibe dichas excepciones particulares (denominadas Restricción temporal) que afectan la manera de ejecutar los filtros (Filtro de las Fuentes de Información de Adaptación [FIA]).

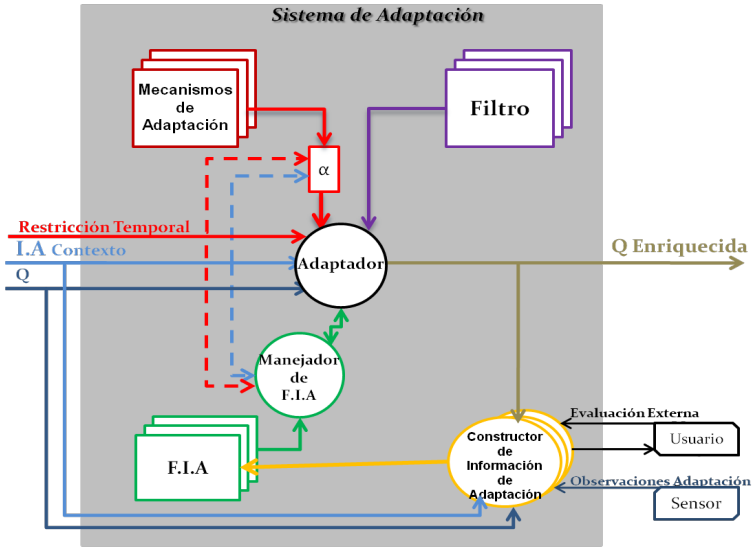


FIGURA 1. MODELO DE AES.

Fuente: Torres-Ribero *et al*., 2021.

Finalmente, es necesario que ingrese al sistema una retroalimentación brindada por iniciativa del usuario sobre la calidad y cumplimiento de expectativas de la respuesta obtenida (evaluación externa, por ejemplo, experiencia del usuario con respecto a la consulta enriquecida entregada) y las observaciones que brinda el sistema de los procesos de adaptación que ha realizado o preguntas que se resuelven sin la iniciativa del usuario (observaciones de adaptación, por ejemplo, deducciones del sistema). Como salida, el sistema genera una nueva consulta enriquecida con los elementos que contiene tal sistema, teniendo en cuenta sus entradas (creando una Q enriquecida). En el caso en el que la consulta ingresada no pueda ser enriquecida, la Q enriquecida será igual a la consulta inicial Q.

2.2. *Componentes estáticos*

Son aquellos componentes que son invocados al estilo de funciones y no son autónomos:

a. *Mecanismos de adaptación*: librerías de adaptación destinadas a enriquecer la consulta Q; se componen de una función de enriquecimiento y una lista de filtros que pueden ser manejados por dicha función.
b. *Selector de mecanismos de adaptación*: componente encargado de seleccionar una función de enriquecimiento.
c. *Filtro*: es una función cuyos parámetros corresponden a una lista de perfiles de una misma categoría y cuyo resultado es información de alto nivel que enriquecerá la consulta.
d. *Fuentes de información de adaptación* (FIA): cada FIA corresponde a un perfil (conjunto de características relacionadas con el fin de representar un concepto). Tal perfil tiene además una categoría (ej., usuario, localización, dispositivo, grupo) y un identificador. Los siguientes son algunos ejemplos de FIA:

- *Usuario*: perfil compuesto de las características relevantes del usuario (figura 2), como sus datos básicos, preferencias (información sobre el cómo quiere algo el usuario o entre varias opciones, cuál[es] escoge el usuario; ej., preferencias de resultado, de colaboración, de presentación, de localización y de actividad), gustos (sensaciones percibidas por los sentidos) e intereses (actividades que despiertan en el usuario algún tipo de inquietud como: motivaciones, pasatiempos y temáticas).
- *Localización*: perfil compuesto de características del entorno en el que se desenvuelve la interacción del usuario con el sistema (figura 3): ubicación, estructura (descripción física del lugar), recursos disponibles (insumos existentes que pueden ser de tipo personal, de cómputo o de comunicación), sociales (creencias o

costumbres presentes en el entorno del usuario) y políticas (condiciones, restricciones y sanciones implantadas).

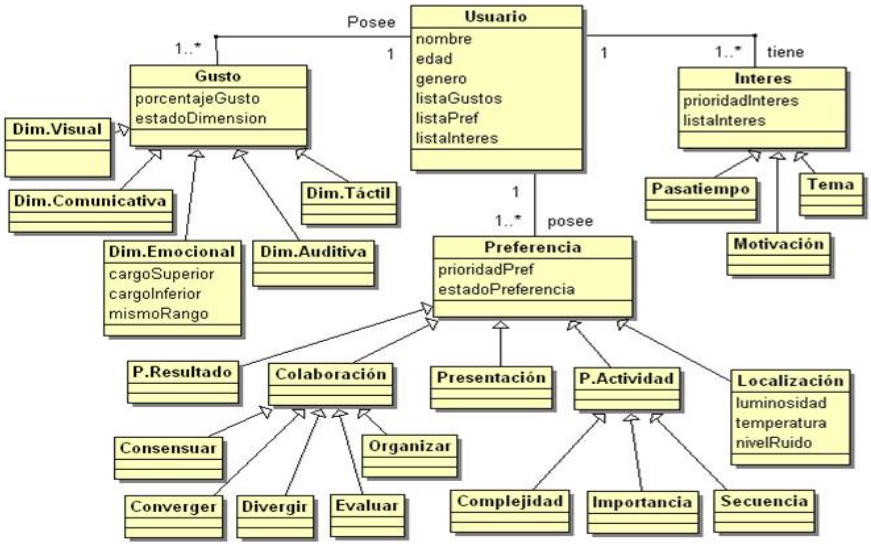


FIGURA 2. EJEMPLO DE UN PERFIL DE USUARIO.

Fuente: elaboración propia.

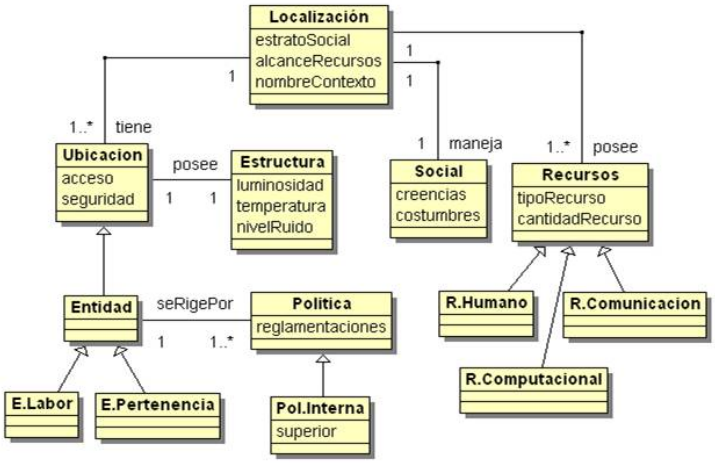


FIGURA 3. EJEMPLO DE UN PERFIL DE LOCALIZACIÓN.

Fuente: elaboración propia.

- *Dispositivo*: perfil compuesto de características relevantes que posee el dispositivo (figura 4) como el tipo de conexión que utiliza el dispositivo, *hardware* (resolución de pantalla, memoria, procesador, autonomía), *software* (*browser*, aplicaciones, sistema operativo).

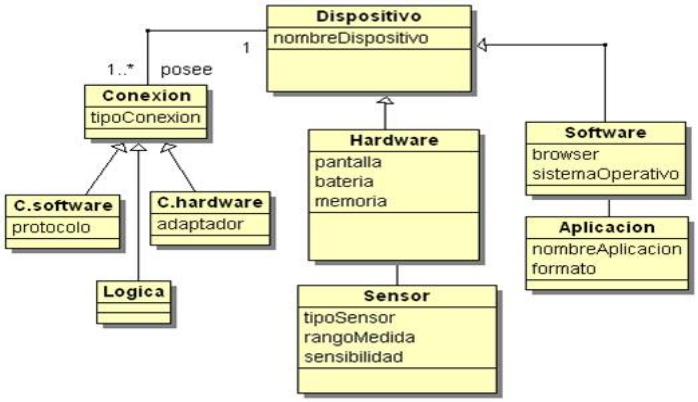


FIGURA 4. EJEMPLO DE UN PERFIL DE DISPOSITIVO.

Fuente: elaboración propia.

La definición de los perfiles depende del sistema que vaya a utilizar el *framework* AES. Internamente, cada una de las características de los perfiles se representa a través de una tripleta (atributo, valor, número), donde el atributo es el aspecto que será evaluado (ej. color), el valor es uno o varios de los posibles valores que puede tomar el atributo con el fin de darle significado en el contexto en el que se desenvuelva la interacción del usuario con el sistema (ej. rojo) y el número corresponde a la cuantificación de la prioridad de ese valor que tiene ese atributo con respecto a lo que el usuario opina o se deduce a partir de las interacciones del usuario con el sistema (ej. el usuario señala que le gusta el rojo en un 75 % de las ocasiones o, el sistema puede deducir que, como el usuario ha elegido cambiar el fondo de la pantalla por rojo en un 40 % de las veces, su color favorito es el rojo en un 40 %). Cabe señalar, que las FIA pueden tener múltiples valores por atributo.

2.3. *Componentes dinámicos*

Son componentes autónomos que se comunican entre ellos y tienen la capacidad de invocar los componentes estáticos, a saber:

a. *Manejador de* FIA: considerando las necesidades de información de adaptación del Adaptador, busca las FIA que provean tal información, la extrae y se la devuelve.

b. *Constructor de información de adaptación*: elemento cuyo fin es realimentar y actualizar las FIA. a través del procesamiento de la Evaluación Externa o a través de Observaciones de Adaptación (ver entradas del sistema, en sección 2.1). Tales actualizaciones pueden corresponder a un valor base o un conjunto de deducciones y son ejecutadas por el método "Actualizar" (*Update*). Dicho método tiene como parámetros la información proveniente de la Evaluación Externa u Observaciones de Adaptación (cuya fuente puede ser proveniente del usuario, de un sensor o de un recurso extra), la categoría de un perfil, el identificador de una de sus instancias y la(s) característica(s) del perfil a modificar.
c. *Adaptador*: es el componente principal del sistema y está encargado de utilizar todos los componentes para enriquecer la consulta realizada (es decir, generar la Q enriquecida a partir de Q y la información de adaptación). Este suministra los parámetros (FIA) al Filtro. Dicho adaptador obtiene los FIA mediante una solicitud al Manejador de FIA.

3. Formalismo de AES

Sea A el Adaptador, Am el Mecanismo de Adaptación, F el filtro, f la función de filtro, ∞ el selector de Am, P el Manejador de Perfiles, FIA la fuente de Información de Adaptación y e la Función de Enriquecimiento. Se utilizará la notación de paréntesis cuadrados para indicar listas, por ejemplo, $[Am]$ será tomado como una lista de Mecanismos de Adaptación. Por otro lado, el Adaptador es descrito como:

$$A = < [Am], \propto >$$

donde $[Am]$ contiene todos los mecanismos de adaptación que el adaptador puede utilizar para enriquecer un servicio y una función $\propto$ que ayudará a A a seleccionar el mejor de acuerdo con la información de contexto que se obtuvo en la consulta.

$$Am = < [F], e >$$

Cada uno de los Am contiene $[F]$ que será utilizada para enriquecer un servicio usando la función que puede ser descrita como $e = \sum_{i=0}^{n} F_i \cdot f$, donde corresponde al resultado de ejecutar la función de filtro en F_i, donde representa el número del filtro y n representa la cantidad total de filtros en Am.

$$F = < c, [FIA], f >$$

Cada *F* tiene una categoría (representada por c) y una lista FIA de la misma categoría; cada FIA contiene información básica de la categoría representada; por ejemplo, la categoría Dispositivo de FIA contendrá información básica sobre su resolución de pantalla, su capacidad, tipos de conexiones, etc. La función en cada filtro toma la información básica contenida en el respectivo FIA y produce información de alta abstracción; continuando con el ejemplo, ya que la FIA tiene la información básica del dispositivo, entonces, la función *f* producirá como información de alta abstracción los tipos de formato que soporta el dispositivo.

$$FIA = < c, [Info] >$$

Cada FIA contiene una categoría y la información básica acorde a dicha categoría.

4. Caso de estudio

Para realizar un seguimiento del funcionamiento y comprobar las interacciones del modelo, se realizaron los siguientes casos de estudio:

- *Primer escenario: Consulta de un estudiante acerca de un tema específico*

El estudiante Juan desea realizar una consulta sobre Macroeconomía (Q). Juan se encuentra en el edificio 1, accediendo al sistema a través de su celular (IA Contexto). En esta ocasión, solo requiere libros/artículos sobre el tema, cuyo formato electrónico tenga extensión *pdf* (restricción particular). Los perfiles (FIA) que el sistema posee almacenados son los de estudiante, localización (características ambientales y estructurales en auditorios, salones, laboratorios, etc.) y el de dispositivo (celular, computadores de escritorios, *laptops*, etc.). De estos perfiles, el sistema selecciona los que se van a utilizar para enriquecer la consulta (Manejador de FIA), que serían específicamente el perfil del usuario Juan, el perfil de localización acerca del edificio 1 y el perfil de dispositivo correspondiente al celular a través del cual tiene acceso al sistema.

El primer filtro recibe la instancia del perfil de usuario correspondiente al de Juan y arroja como información de alto nivel que Juan es un experto en el tema (Macroeconomía) y que aprende de una mejor manera cuando se le presenta la información en formato visual. El segundo filtro recibe la instancia del perfil de localización correspondiente al edificio 1 y retorna como información de alto nivel que tal edificio provee baja conectividad a los usuarios que intentan conectarse desde allí. El tercer filtro recibe la instancia del perfil de dispositivo correspondiente al celular y retorna como información de alto nivel que dicho dispositivo posee una

baja resolución de tipo CGA (Filtros). Considerando que el estudiante es experto, tiene aprendizaje visual, que hay baja conexión y que el dispositivo de acceso tiene baja resolución, el sistema considera que se deben traer libros de Macroeconomía especializados y solo se podrá enviar información almacenada en archivos pequeños (Función de Enriquecimiento). Como respuesta de la Función de Enriquecimiento y teniendo en cuenta los filtros (experticia del usuario, conectividad del celular), el sistema sigue la Función de Enriquecimiento que establece el orden de los filtros que se deben aplicar: Localización, Dispositivo y Usuario. Vale la pena resaltar que el sistema conoce que el usuario se encuentra en el edificio 1 (deduciendo baja señal de internet) y el dispositivo posee una resolución de 320 X 200 (*CGA*); además, que el usuario prefiere en esta ocasión documentos en *pdf* (Restricción Particular).

Debido a que Juan aprende de una mejor manera cuando se le presenta la información en formato visual, el sistema deduce que se le deben ofrecer libros digitales de Macroeconomía con gráficas y en formato *pdf*. También debido a que se encuentra en el edificio 1 y que su dispositivo de acceso no permite el despliegue de imágenes de más de cierto tamaño por su resolución, el sistema considera que se le deben ofrecer libros-artículos digitales con pocas imágenes o de baja resolución con el fin de que puedan ser descargados y visualizados en formato *pdf* únicamente (Adaptador). El Adaptador produce como consulta enriquecida: libros sobre Macroeconomía con gráficas, de pocos Kb y baja resolución, para ser descargados solo en formato *pdf*. Para finalizar y retroalimentar el sistema, es necesario evaluar-validar el nivel de experticia del usuario en el tema (Constructor de Información de Adaptación); para tal fin, el sistema revisa si Juan escogió alguno de los libros/artículos sugeridos (Observaciones de Adaptación); además, se podría evaluar a través de un *test*, teniendo en cuenta su experticia, qué tanto le sirvieron los libros-artículos digitales sugeridos por el sistema al estudiante (Evaluación Externa).

- *Segundo escenario*: *Conformación de grupos estudiantiles*

Un profesor desea conformar un grupo de trabajo con [1, n] roles (cada uno con [1, m] habilidades) y un número determinado de estudiantes (Q), para su clase (IA Contexto). En esta ocasión, el grupo será excluyente, es decir, solo se deben escoger los estudiantes cuyas características se ajusten más a las exigencias del grupo que se va a conformar (Restricción Particular).

Los perfiles (FIA) que el sistema posee almacenados son el perfil de estudiante (gustos, preferencias, habilidades, estilo de aprendizaje, etc.), el del profesor (gustos, preferencias, estrategias de enseñanza, etc.), el de clase (asignatura, temática, etc.) y el de grupo existente (participantes, roles, habilidades, metas, etc.). Conociendo

la clase en la que se va a conformar el grupo (IA Contexto), el sistema selecciona los perfiles que se van a utilizar para enriquecer la consulta (Manejador de FIA), que serían específicamente los perfiles de los estudiantes pertenecientes a dicha clase. El filtro recibe las instancias del perfil de estudiante y arroja como información de alto nivel las habilidades sobresalientes de cada estudiante (Filtros). El sistema deduce que, a partir de las habilidades sobresalientes de los estudiantes, se debe asignar el rol que se ajuste más a las habilidades requeridas (labor del Adaptador). Por esto, el Adaptador produce como consulta enriquecida: dentro de la clase, cuáles son los estudiantes que conformarán el grupo según las habilidades requeridas para cada rol establecido, tomando en consideración particularmente los perfiles de los estudiantes cuyas habilidades se ajusten a aquellas requeridas por cada rol establecido (Q enriquecida).

El constructor de información de Adaptación crea el perfil del grupo basado en las entradas dadas por el profesor (roles, habilidades, número de integrantes) y relaciona este con los estudiantes seleccionados para conformarlo. Adicionalmente, después de la conformación del grupo, se podría evaluar si se presentaron problemas dentro de él, si fueron resueltos (respondidas por los participantes del grupo) y si se logró la meta planeada (respondida por el profesor). Esto corresponde a la Evaluación Externa. Finalmente, el sistema revisa si el grupo conformado fue exitoso (se logró la meta planeada y se resolvieron sus problemas internos), además se podría deducir que, para ese tipo de proyectos, se debe armar de nuevo el mismo grupo conformado en esta ocasión. Esta deducción se podría hacer a partir de las Observaciones de Adaptación.

5. Implementación de AES con agentes

Esta sección presenta la implementación del *framework* AES basada en agentes de *software* y otras consideraciones necesarias para utilizar el *framework* por parte de un programador.

5.1. *Modelo físico: Arquitectura de Agentes*

Este *framework* fue implementado utilizando la plataforma BESA (González *et al*., 2003), la cual fue desarrollada en el Departamento de Ingeniería de Sistemas de la Pontificia Universidad Javeriana (Bogotá) con el fin de diseñar soluciones basadas en sistemas multiagente. Al implementar AES se estableció que los componentes

dinámicos fueran representados por agentes debido a que son entidades inteligentes, proactivas y llevan a cabo su trabajo; los componentes estáticos son recursos que no son autónomos, a los cuales acceden los agentes (estos componentes se convierten en recursos de tales agentes).

Para la implementación de AES solo se diseñaron dos agentes que podían desempeñar todas las funcionalidades: a) el agente Adaptador, que se encarga de la generación de información de alta abstracción a partir de los perfiles utilizados en el proceso de filtrado y a) el agente Manejador de FIA, que se encarga del almacenamiento y la actualización de las FIA. Cabe anotar que los recursos del agente Adaptador son los Filtros y los Mecanismos de Adaptación (componentes estáticos) y los recursos del agente Manejador de FIA son los FIA. Además, el agente Manejador de FIA se encarga de desempeñar las funcionalidades del Constructor de Información de Adaptación tomando como entradas la Evaluación Externa y las Observaciones de Adaptación. En la figura 5 se puede apreciar cómo dentro de dichos agentes aparecen elementos rectangulares que representan los componentes estáticos; estos componentes corresponden a la información que manejan para su labor (ej., sus recursos). El entorno se refiere a la aplicación externa que utiliza los servicios proporcionados por AES. Como prerrequisito, es necesario que el sistema que utilice los servicios provistos por AES, le proporcione los Filtros, los Mecanismos de Adaptación, los perfiles (con sus correspondientes instancias) y las funciones de filtrado que maneja, según sus propias necesidades. Esto es el insumo principal del agente Adaptador.

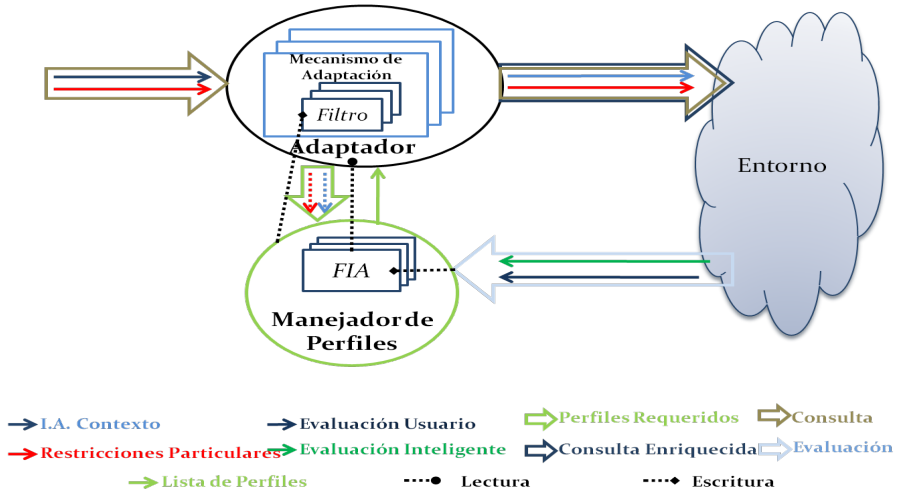


FIGURA 5. MODELO DE AGENTES.

Fuente: Torres-Ribero *et al*., 2011.

Inicialmente, el *framework* recibe como parámetros (por parte del usuario o de otro agente) la consulta (Q), las Restricciones Particulares y la información externa (IA Contexto). El proceso que sigue para enriquecer dicha consulta es el siguiente: los parámetros son recibidos por el agente Adaptador, el cual ejecuta la función de Filtros con el fin de seleccionar el mecanismo de adaptación más apropiado basado en los parámetros mencionados (el mecanismo de adaptación posee asociados ciertos Filtros, cada uno de ellos relacionado con una categoría de perfil). El agente Adaptador solicita al agente Manejador de FIA los perfiles que necesite por cada uno de los Filtros. El agente Manejador de FIA recibe la solicitud de los perfiles. Vale la pena destacar que el agente Adaptador hace tal solicitud considerando la información de contexto y restricciones. Una vez solicitados los perfiles, el agente Manejador de FIA los retorna al agente Adaptador, de tal modo que este último ejecuta la función de enriquecimiento para brindar como salida la consulta enriquecida. Al agente Manejador de FIA asume el rol de Constructor de Información de Adaptación ya que es este agente el que recibe información externa proveniente de la Evaluación Externa y las Observaciones de Adaptación.

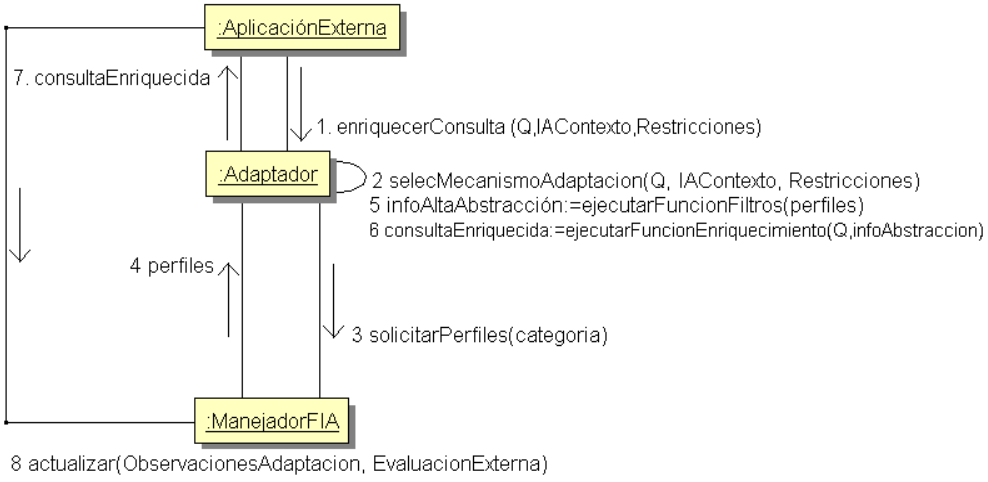


FIGURA 6. DIAGRAMA DE COLABORACIÓN AES.

Fuente: elaboración propia.

Con esta información, el agente Manejador de FIA modifica el perfil pertinente a través del método Actualizar o *Update*. Las figuras 6 y 7 presentan los mensajes entre los agentes y su respectivo orden de interacción para enriquecer una consulta.

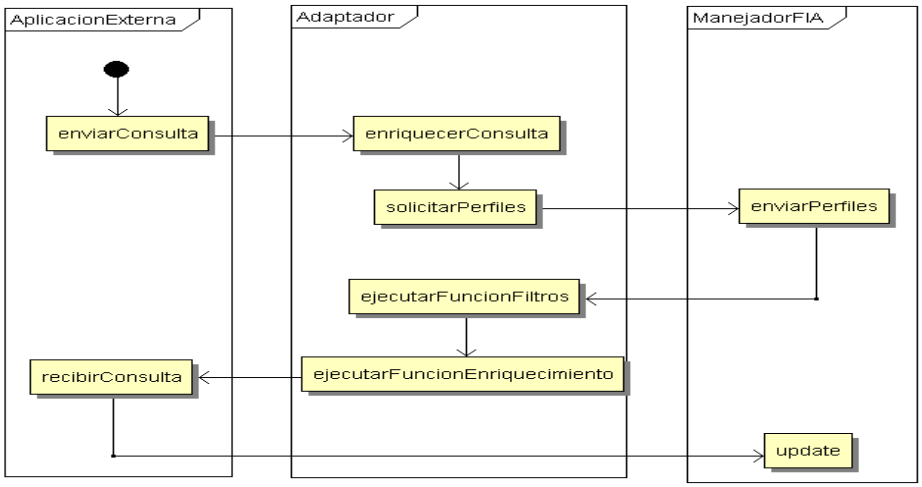


FIGURA 7. DIAGRAMA DE ACTIVIDAD AES.

Fuente: elaboración propia.

5.2. *Aspectos adicionales de implementación*

La instalación del paquete AES consiste en colocar la librería AES _*Framework*.jar en el proyecto que se desee utilizar. Es importante notar que la librería AES _*Framewrok*.jar ya incluye la librería BESA_3. Como parte fundamental de la configuración se requiere generar en el directorio raíz del proyecto el archivo ConfigBESA.properties, el cual debe tener las propiedades ajustadas al proyecto que se desea implementar. Un ejemplo de dicho archivo se puede observar en la figura 8. Cada una de las propiedades consignadas servirán para la configuración del *framework* de agentes BESA. Para mayor información en la configuración de BESA referirse a la documentación de dicho *framework*.

```
#########################################################################
# ConfigBESA.properties              2.0     11/01/11                   #
#                                                                       #
# Copyright 2011, Pontificia Universidad Javeriana, All rights reserved.  #
# Takina and SIDRe PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.  #
#########################################################################
#-----------------------------------------------------------------
# Alias container.
aliasContainer=Name
# IP address.
IP_ADD=127.0.0.1
# Mac address.
MC_ADD=230.0.0.1
# Mac port.
MC_PORT=2222
#-----------------------------------------------------------------
# Send event timeout.
sendEventTimeout=3000
# Sen event attemps.
senEventAttemps=3
#-----------------------------------------------------------------
# RMI port.
RMI_PORT=9876
# RMI timeout.
RMITimeout=5000
#########################################################################
```

FIGURA 8. EJEMPLO ESTRUCTURA ARCHIVO CONFIGBESA.PROPERTIES.

Fuente: elaboración propia.

A continuación, se presentan las estructuras que le proporciona AES al programador usuario para que este los implemente de acuerdo con su contexto y así pueda utilizar el enriquecimiento que brinda AES.

- FI.A: como se había mencionado, los FIA son las estructuras más básicas del *framework*. Dichos FIA deben ser implementados extendiendo las características básicas que provee el *framework*. Estas estructuras se definen como una clase convencional, hija de una estructura básica llamada ProfileBESA. Los FIA definen tanto los perfiles utilizados como el contexto, ya que este último está basado en información básica de los perfiles y se representa como una lista de FIA. Cada FIA tiene una categoría y un *id* de perfil único; por definición, la categoría debe ir en el constructor del perfil. El *id* de cada perfil se genera automáticamente con el fin de identificarlo de manera única (figura 9).

```
public class PerfilXYZ extends ProfileBESA {
    //.....
    public PerfilXYZ(String categoria) {
        super(categoria);
        //....
//...
}
```

FIGURA 9. ESTRUCTURA PARA DEFINIR PERFILES.

Fuente: elaboración propia.

- *Filtro*: se implementa heredando una clase básica implementada en el *framework*. Cada filtro tiene una categoría correspondiente a un tipo de perfil. Los filtros se aplican únicamente si el tipo de perfil coincide con el tipo de filtro. Adicionalmente, cada filtro contiene una lista de campos de los FIA que utilizará dentro de su función de generación de información de alta abstracción; los campos se conocen como subcategorías. Al generar información de alta abstracción pueden generarse los siguientes escenarios:
  * El filtro afecta más campos de los que tiene el perfil
    En este caso se genera un error que debe resolverse bien sea adecuando el filtro, reduciendo sus subcategorías o haciendo que los atributos del perfil aumenten y coincidan con las subcategorías definidas.
  * El filtro afecta menos campos de los que tiene el perfil
    Se generará una advertencia; el funcionamiento no se ve afectado, pero se recomienda utilizar la mayor cantidad de campos de un perfil en el análisis realizado por cada filtro.
  * El filtro afecta la misma cantidad de campos que tiene el perfil
    Este es el caso ideal, no genera errores ni advertencias.

Por otra parte, la implementación de los filtros se compone de dos partes: la primera es definir el filtro y la segunda es agregar las subcategorías. Esto último puede hacerse bien sea durante la primera parte, en la definición o en tiempo de ejecución (figura 10).

```
public class FiltroXYZ extends FilterBESA {
    public FiltroXYZ(String category, List<String> sub
    categories) {
        super(category, subcategories);
       //........
    }
    public FiltroXYZ(String category) {
        super(category);
    //........
    }
    @Override
    public FilterResultBESA doFilter(List<ProfileBESA>
    aList) {
        //.........
    }
}
```

FIGURA 10. ESTRUCTURA PARA DEFINIR FILTROS.

Fuente: elaboración propia.

Como puede observarse, el filtro requiere principalmente de dos constructoras y la implementación de un método básico llamado doFilter. Dicho método recibe como parámetro una lista de perfiles, los cuales deberá analizar y generar una respuesta de alta abstracción basándose en el valor de sus atributos.

- *Mecanismos de Adaptación*: heredan de la clase AdaptationMechanismBESA, se debe implementar el método enrichQuery. Este método recibe como parámetros la búsqueda inicial y una lista con los resultados de la información de alta abstracción generada por los filtros (figura 11).

```
public class MecanismoAdaptacion extends
AdaptationMechanismBESA {
   //....
   public MecanismoAdaptacion() {
   }
   @Override
   public EnrichedAdaptationQueryBESA enrichQuery
   (AdaptationQueryBESA theQuery, List<FilterResultBESA> filterRe-
   sults) {
     //...........
    }
}
```

FIGURA 11. ESTRUCTURA PARA DEFINIR FILTROS.

Fuente: elaboración propia.

Al finalizar el método enrichQuery se debe retornar un elemento de tipo EnrichedAdaptationQueryBESA; dicho elemento debe contener el análisis realizado a partir de los resultados de los diferentes filtros y de la consulta inicial hecha al servicio.

- *Guardas*: implementan servicios básicos que darán funcionamiento al *framework*. Inicialmente se deben implementar dos guardas: una establece las reglas de selección de un mecanismo de adaptación adecuada y la generación del contexto a partir del contexto actual y de las restricciones; la otra guarda define la obtención de los perfiles de acuerdo con el sistema de almacenamiento. La primera guarda debe extender de la clase AdaptQueryGuardBESA. Dicha guarda debe implementar los métodos: selectAdaptationMechanism y applyRestrictions (figuras 12 y 13).

```
public class MecanismoAdaptacion extends AdaptationMechanismBESA {
 //....
 public MecanismoAdaptacion() {
    }
    @Override
public EnrichedAdaptationQueryBESA enrichQuery(AdaptationQueryBESA
theQuery, List<FilterResultBESA> filterResults) {
   //...........
    }
}
```

FIGURA 12. ESTRUCTURA PARA DEFINIR GUARDAS.

Fuente: elaboración propia.

```
public class GuardRecieveQuery extends AdaptQueryGuardBESA{
    public GuardRecieveQuery(AgentBESA ag) {
       super(ag);
    }
     @Override
    public AdaptationMechanismBESA selectAdaptationMechanism(List
    <ProfileBESA> context, List<ProfileBESA> restrictions) {
      //...
     return super.selectAdaptationMechanism(context,restrictions);
 }
    @Override
   public List<ProfileBESA> applyRestrictions(List<ProfileBESA>
   context, List<ProfileBESA> restrictions) {
        //...
        return newContext;
    }
}
```

FIGURA 13. ESTRUCTURA PARA DEFINIR GUARDAS.

Fuente: elaboración propia.

El método selectAdaptationMechanism tiene una funcionalidad básica de selección; dicha funcionalidad revisa los diferentes tipos de perfil en el contexto y asigna como mecanismo de adaptación aquel que tenga más coincidencias de tipos de filtro. Dicho método puede ser sobrescrito y debe retornar el mecanismo de adaptación más adecuado con base en el contexto y las restricciones, representadas como listas de perfiles. El método applyRestrictions permite mezclar el contexto con las restricciones en un nuevo contexto; el nuevo contexto se representa como una lista de perfiles y es retornada por el método.

```
public class GuardReceiveFilter extends
GuardRecieveFilterQuery {
   public GuardReceiveFilter(AgentBESA ag) {
      super(ag);
}
  @Override
   public   List<ProfileBESA>   obtainProfiles(FilterBESA
   filter) {
      //...
      return resp;
    }
}
```

FIGURA 14. ESTRUCTURA CLASE GUARDRECEIVEFILTER.

Fuente: elaboración propia.

La segunda guarda debe implementar la obtención de los diferentes perfiles de acuerdo con el sistema de almacenamiento. Dicha guarda debe heredar de GuardRecieveFilterQuery e implementar el método obtainProfiles (figura 14).

Cada solución almacena de manera diferente sus perfiles, bien sea en archivos planos o en una base de datos, el método obtainProfiles debe implementar la lógica de obtención de perfiles que coincidan con el filtro que es dado por parámetro. El método retorna la lista de perfiles relevantes a su contexto y al filtro al que el perfil corresponde.

6. Conclusiones

AES es un *framework* que puede ser usado para adaptar información en diferentes clases de contextos debido al uso de filtros genéricos y mecanismos de adaptación. Todos estos componentes pueden ser extendidos dependiendo de los requerimientos particulares del servicio computacional (dirigido a un usuario) que va a ser enriquecido. Estas características hacen que AES enriquezca genéricamente un servicio utilizando entradas básicas, proveyendo una experiencia transparente y personalizada al usuario final. Como trabajo futuro se propone la utilización de AES en diversos escenarios. Cabe mencionar que AES fue utilizado en un ambiente educativo cuyo detalle se explica en el siguiente capítulo (titulado AES-EDU: enriquecimiento de servicios en un entorno educativo). Adicionalmente, AES se integrará a diversas aplicaciones basadas o no en agentes.

Referencias

Abeywickrama, D. y Ramakrishnan, S. (2012). Context-Aware Services Engineering: Models, Transformations, and Verification. *ACM Transactions on Internet Technology*, *11*(3), 1-28.

Arias-Báez, M.P., Torres-Ribero, L.G., Carrillo-Ramos, A., Pomares, A., Garzón, J. P. y González, E. (2011). *AES:* A Generic *Framework* for Adapting Information in Multidimensional Contexts. *XXXVII Conferencia Latinoamericana de Informática (CLEI 2011)*, 886-901.

Baur, T., Mehlmann, G., Damian, I., Lingenfelser, F., Wagner, J., Lugrin, B., André, E. y Gebhard, P. (2015). Context-Aware Automated Analysis and Annotation of Social Human--Agent Interactions. *Journal ACM Transactions on Interactive*

Intelligent Systems (TiiS) - Special Issue on Behavior Understanding for Arts and Entertainment, *5*(2), 1-33.

Cocea, M. y Magoulas, G. (2011). Context-Dependent Feedback Prioritisation in Exploratory Learning Revisited. *UMAP'11 Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*, 62-74.

González, E., Bustacara, C. y Ávila, J. (2003). BESA: arquitectura para construcción de sistemas multiagentes. *Conferencia Latinoamericana de Informática (CLEI 2003)*.

Koehl, A. y Wang, H. (2012). m.Site: Efficient Content Adaptation for Mobile Devices. *Proceedings of the 13th International Middleware Conference*, 41-60.

Lamche, B., Trottmann, U. y Worndl, W. (2014). Active Learning Strategies for Exploratory Mobile Recommender Systems. *Proceedings of the 4th Workshop on Context-Awareness in Retrieval and Recommendation*, 10-17.

Torres-Ribero, L. G., Garzón, J. P., Arias-Báez, M. P., Carrillo-Ramos, A. y González, E. (2011). Agents for Enriched Services (AES): A Generic Agent - Based Adaptation Framework. *International Conference on Collaboration Technologies and Systems (CTS)*, 492-499.

World Wide Web Consortium (W3C). (2015). *About W3C*. Disponible en: http://www.w3.org/Consortium/. Agosto de 2015.

8
AES-EDU: ENRIQUECIMIENTO DE SERVICIOS EN UN ENTORNO EDUCATIVO

Ángela Carrillo-Ramos, Luis Guillermo Torres-Ribero, María Paula Arias-Báez, Enrique González

Hoy en día los usuarios reciben una gran cantidad de información proveniente de diversos sistemas que no siempre es relevante y que no considera las características de estos usuarios ni las de su entorno. Adicionalmente, dicha información puede no ajustarse a sus necesidades y a las características de su dispositivo de acceso (móvil o no). Si los sistemas de información le proveyeran al usuario información de acuerdo con sus características, tales como preferencias, gustos e intereses, así como a las de su dispositivo de acceso, podría reducirse el problema de sobrecarga de información. Filtrar la información tomando en cuenta las características enunciadas es típico de los procesos adaptativos. Este capítulo presenta una aplicación de AES, un *framework* basado en agentes que adapta la información al usuario con el propósito de brindarle la información que necesita en el momento, lugar y a través del dispositivo de acceso con el que cuente. En este capítulo se muestra, entonces, el paso a paso para su aplicación en un contexto educativo para de esta forma obtener AES-EDU, el cual soporta las funcionalidades del proyecto AYLLU. Igualmente, este capítulo se enfoca en la utilización del *framework* AES para enriquecer el servicio de seleccionar un OVA[1] para un estudiante, continuando así con el primer escenario presentado a manera de caso de estudio de AES.

---

[1] OVA: Objeto Virtual de Aprendizaje.

1. Introducción

Generalmente, los sistemas de información (SI) presentan de la misma manera la información a todos los usuarios sin tener en cuenta sus características (preferencias, intereses) y el contexto que involucra sus interacciones con tales sistemas. Adicionalmente, se puede apreciar que, en ocasiones, la información no se ajusta a las características y restricciones del dispositivo de acceso que utiliza para el acceso al sistema. Por lo anterior surgió la Adaptación de la Información, proceso mediante el cual se intenta reducir la sobrecarga cognitiva en la que se ve envuelto el usuario al recibir grandes volúmenes de información, ya que filtra los resultados de invocar los servicios de los SI considerando las necesidades y características del usuario, el entorno de sus interacciones y el dispositivo de acceso que utiliza (Carrillo-Ramos *et al*., 2015).

La Adaptación es un proceso que enriquece un servicio mediante la adición de un conjunto de parámetros (grupo de características relevantes que podrían influir en la interacción del usuario con el sistema) con el fin de proveer la información al usuario de acuerdo con sus características y las de su contexto. Estos parámetros son utilizados para ajustar la información a los requerimientos del usuario y a su dispositivo de acceso. Por ejemplo, si el usuario prefiere obtener información de tipo visual, el sistema le debe proveer imágenes, animaciones o videos, siempre y cuando puedan ser desplegados y soportados por su dispositivo de acceso. Existen retos importantes que la adaptación de información aún debe superar: a) el diseño de herramientas existentes de adaptación son diseñadas para un dominio específico de aplicación (Nepal *et al*., 2015; Boudaa *et al*., 2014; van Riemsdijk *et al*., 2015) y b) la mayoría de ellas no consideran problemas en labores colaborativas (Velden *et al*., 2014).

Como se dijo, este capítulo presenta la utilización del *framework* AES para enriquecer el servicio de seleccionar un OVA para un estudiante, continuando así con el primer escenario presentado en la sección Caso de Estudio del capítulo anterior (Consulta de un estudiante acerca de un tema específico, ver capítulo anterior). AES es un *framework* genérico que adapta información en diversos contextos (móviles o no), enriqueciendo servicios con el fin de proveerle al usuario información ajustada a sus necesidades, las características del contexto y las de su dispositivo de acceso. El proceso de enriquecimiento consiste en los siguientes pasos: una aplicación externa (manejada por un usuario o por otro sistema) envía una consulta inicial a AES y solicita aumentar los parámetros con información que le permita filtrar los resultados que obtenga de dicha consulta. AES obtiene los perfiles con características relevantes de la aplicación externa (usuario, contexto, dispositivo) e invoca funciones de filtro de acuerdo con los perfiles seleccionados. El resultado es un conjunto de información

de alta abstracción que es utilizada para generar la consulta enriquecida que será enviada de nuevo a la aplicación externa.

La estructura de este capítulo es la siguiente: la sección 2 presenta el escenario paso a paso utilizando el *framework* AES, seguido de la especificación escogida para definir los metadatos de los OVA que se van a sugerir. La sección 4 presenta la implementación de los componentes utilizados para este escenario y finalmente las pruebas se exponen en la sección 5. Los trabajos relacionados se presentan en la sección 6 y en la sección 7 se concluye y se presenta el trabajo futuro.

2. Escenario en un Entorno Educativo

Suponga que el profesor de cierta asignatura desea seleccionar el OVA que mejor se ajuste a las características del estudiante Juan. Dicho estudiante accede al sistema a través de su celular (IA Contexto). Los diferentes perfiles (FIA) que el sistema almacena son: el perfil de Estudiante (figura 2) y el perfil de Dispositivo (figura1). De estos perfiles, el sistema selecciona aquellos que pueden enriquecer la Consulta inicial (a través del Manejador de Perfiles); para este caso se seleccionan el perfil del Estudiante Juan y el perfil del Dispositivo RT50 que corresponde al celular que utiliza Juan (figura 3).

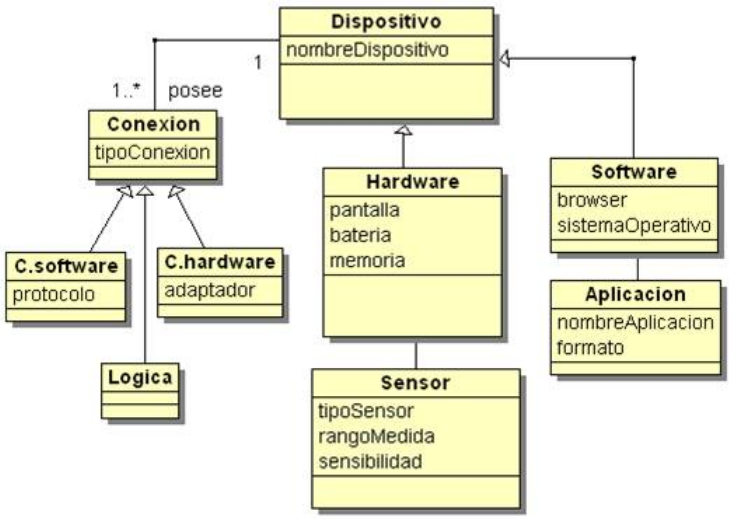


FIGURA 1. PERFIL DE DISPOSITIVO.

Fuente: elaboración propia.

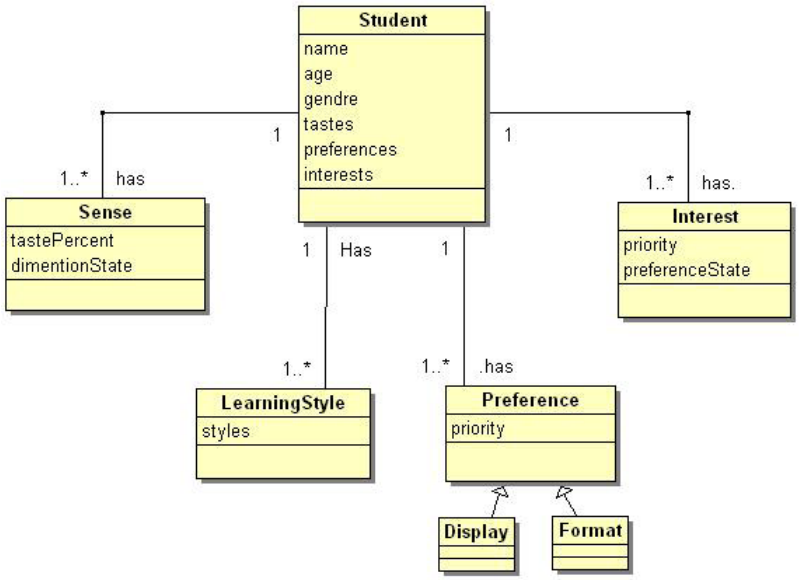


FIGURA 2. PERFIL DE ESTUDIANTE.

Fuente: elaboración propia.

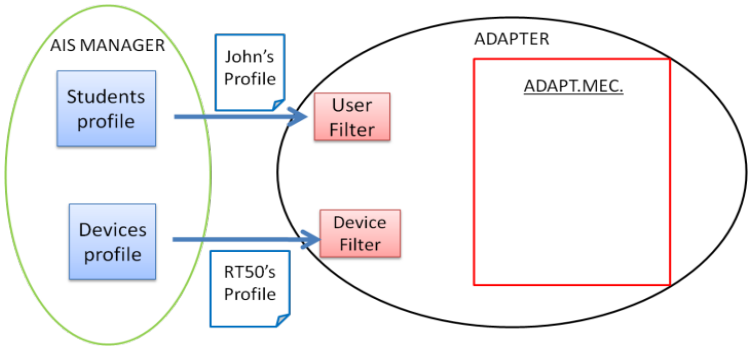


FIG. 3. SELECCIÓN DE PERFILES.

Fuente: elaboración propia.

El primer Filtro recibe el perfil de Juan, que contiene información como su nombre, fecha de nacimiento, estilo de aprendizaje, preferencias de despliegue,

de formato y color[2]. Tal Filtro toma la información más relevante de acuerdo con el contexto. El segundo Filtro recibe el perfil del dispositivo que contiene información sobre la referencia, niveles de batería y formatos soportados; lo anterior corresponde a la información más relevante que será utilizada por el Adaptador con el fin de generar información de alta abstracción (figura 4).

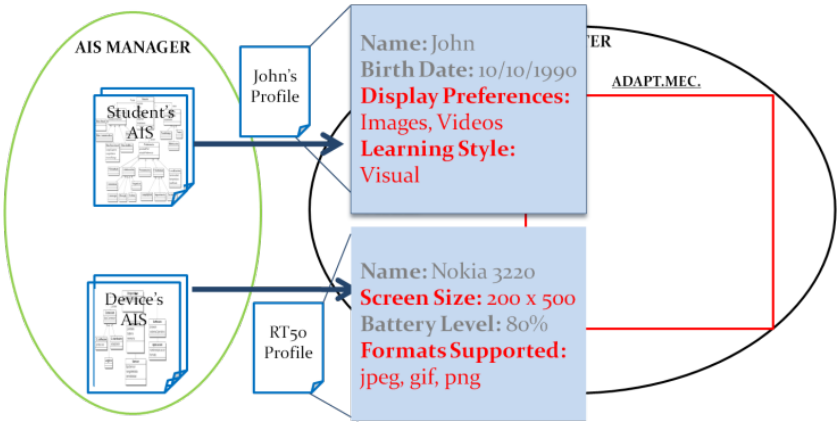


FIGURA. 4. SEPARACIÓN DE INFORMACIÓN RELEVANTE DENTRO DEL FILTRO DE ESTUDIANTE Y DENTRO DEL FILTRO DE DISPOSITIVO (EN NEGRILLA).

Fuente: elaboración propia.

La información de alta abstracción es generada a través de la utilización de reglas del negocio. El primer Filtro deduce que Juan posee un estilo de aprendizaje activo-visual de acuerdo con el test de Felder-Silverman (Kart, 2015) y retorna el tipo de material activo-visual que coincide con los OVA del color del estudiante y según las preferencias de despliegue de Juan, además de su grado de escolaridad. El segundo Filtro realiza el mismo proceso y retorna el tamaño del dispositivo de Juan (pequeño) a partir de su resolución (figura 5).

[2] Cuando se menciona "color", se hace referencia al color del estudiante obtenido a partir del test de Herrmann para dominancia cerebral y para conocer un poco más sobre el comportamiento del estudiante y su relación con el aprendizaje.

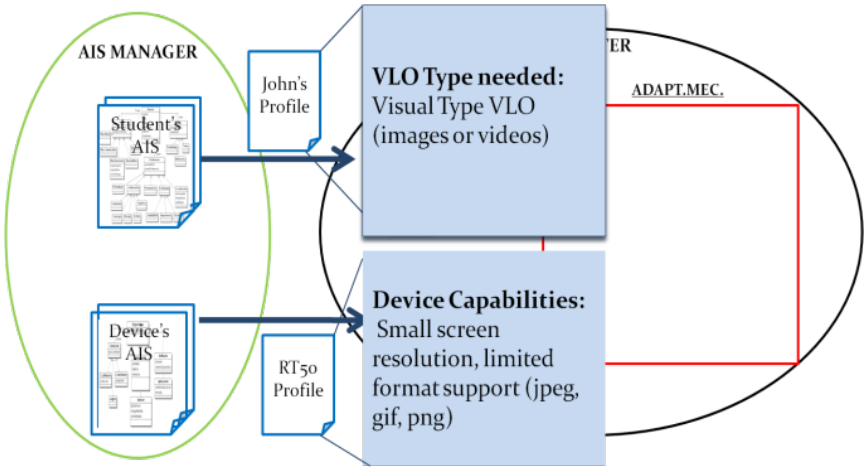


FIGURA. 5. RESULTADO DE CADA FILTRO DESPUÉS DE EJECUTAR LAS REGLAS.

Fuente: elaboración propia.

Teniendo en cuenta que el estilo de aprendizaje del estudiante es activo-visual, es decir, aprende a través de estímulos visuales de manera activa y su grado de escolaridad es educación primaria, y conociendo que su dispositivo tiene baja resolución y tamaño pequeño, el sistema deduce que se debe especificar que los OVA seleccionados deben ser de educación primaria, gráficos o dinámicos y su tamaño en Kb debe ser pequeño (Función de Enriquecimiento). Después de recibir el resultado de cada Filtro (tipo de material a partir del estilo de aprendizaje del usuario, su color, sus preferencias de despliegue, grado de escolaridad, tamaño del dispositivo), el sistema ejecuta un método en el Mecanismo de Adaptación sobre dichos resultados, el cual determina la importancia de cada Filtro para ser aplicado (figura 6).

Si el Filtro de dispositivo es ejecutado y luego el de estudiante, la resolución del dispositivo es baja, por lo tanto, el sistema deduce que los OVA que serán presentados deben ser de baja resolución y poco tamaño (del orden de Kb). Adicionalmente, de acuerdo con el hecho de que el tipo de información que mejor se adapta al método de aprendizaje de Juan es el visual y su color dominante es el amarillo, el sistema deduce que el OVA debe contener gráficos y ser interactivo para educación primaria (deducciones realizadas por el Adaptador). El adaptador produce una consulta enriquecida que comprende el material activo-visual-amarillo que coincida con las preferencias de Juan, teniendo en cuenta que está en Educación primaria (Consulta

Enriquecida). Por ejemplo: Simulación, Experimento, Autoevaluación, "Diagrama", "Figura", "Gráfico", de baja resolución con gráficos o en formato texto (figura 7).

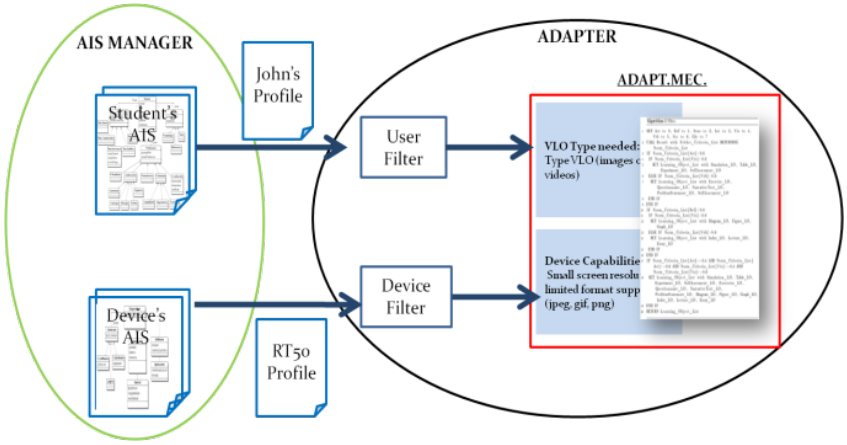


FIGURA 6. EL MECANISMO DE ADAPTACIÓN APLICA LAS REGLAS SOBRE LA RESPUESTA DE CADA FILTRO.

Fuente: Carrillo-Ramos *et al*., 2012.

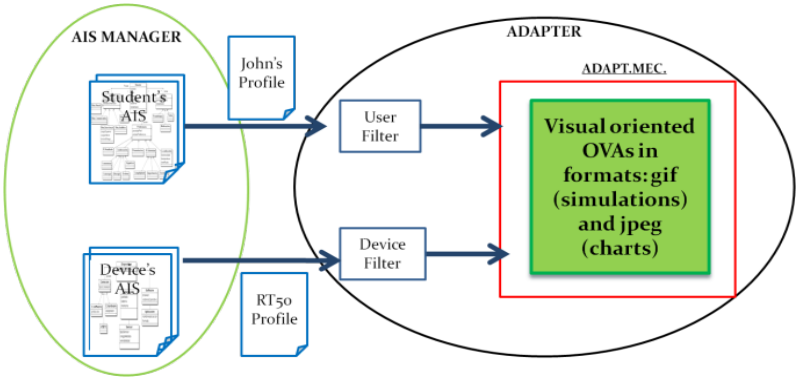


FIGURA 7. RESULTADO DEL MECANISMO DE ADAPTACIÓN.

Fuente: elaboración propia.

3. Estándar IEEE-LOM

Cada OVA posee sus respectivos metadatos que brindan información sobre sus propias características. Dichos metadatos fueron tomados del estándar IEEE-LOM[3], dentro de los cuales se tuvieron en cuenta:

- *Typical age range*: rango de edad para los que se enfoca el OVA.
- *Learning resource type*: tipo de recurso educativo. Especifica el tipo de material educativo (ej., ejercicio, figura, etc.). Un mismo material puede tener distintos tipos asociados. LOM propone el siguiente vocabulario para caracterizar el tipo de material: *exercise* (ejercicio), *simulation* (simulación), *questionnarie* (cuestionario), *diagram* (diagrama), *figure* (figura), *graph* (gráfico), *index* (índice), *slide* (diapositiva), *table* (tabla), *narrative text* (texto narrativo), *exam* (examen), *experiment* (experimento), *problemstatement* (enunciado de problema), *self assessment* (autoevaluación).
- *Format*: formato del OVA (ej.: image/jpeg).
- *Context*: entorno educativo típico en el que se usará el material. LOM propone el siguiente vocabulario:
  - *Primary education* (educación primaria)
  - *Secondary education* (educación secundaria)
  - *Higher education* (educación superior)
  - *University postgrade* (postgrado)

4. Implementación

Para llevar a cabo el escenario presentado al comienzo del capítulo, se implementaron las FIA, los Filtros y el Mecanismo de Adaptación. Las FIA utilizadas para el escenario fueron el perfil de usuario y el perfil de dispositivo (figura 8). Sin embargo, fue necesario crear el perfil de estudiante (figura 9) y añadir otros atributos relacionados con el contexto educativo: a) *Estilo de aprendizaje*: aproximación o medio de aprendizaje que mejora los resultados del estudiante. Dichos medios fueron tomados de los estilos de aprendizaje de Felder-Silverman (Kart, 2015); ii) *Color*: detección de formas pensar, crear, aprender y convivir a partir de las preferencias mentales mediante el test de Herrmann (Herrmann, 1989); a) *Preferencias de despliegue*: dependiendo del tamaño del dispositivo, qué tipo de material es el que prefiere el estudiante para desplegar la información; a) *Preferencia de formato*: de acuerdo con las preferencias de despliegue del estudiante (ej., video), qué tipo de formatos prefiere (ej., wmv, mp4, etc.).

---

[3] LOM: *Learning Object Metadata*, estándar de la IEEE para objetos de aprendizaje.

```java
package besa_adaptado.perfiles;

import adaptation.common.ProfileBESA;
import java.util.ArrayList;

public class PerfilEstudiante extends ProfileBESA {

    /**
     * @param nombrestudiante representa el nombre del estudiante
     *     @param edad
     *     @param estiloAprendizaje representa los estilo de aprendizaje del estudiante con su valor
     *     @param preferenciaDespliegue en posicion[0]: para dispositivo pequeño, posicion[1]: para dispositivo grande
     *     @param preferenciaFormatos representa los formatos preferidos por el estudiante
     */
    private String nombrestudiante;
    private int edad;
    private ArrayList <String> estiloAprendizaje;
    private String preferenciaDespliegue[] = new String[2];//
    private ArrayList <String> preferenciaFormatos;
```

FIGURA 8. PERFIL DE ESTUDIANTE.

Fuente: elaboración propia.

```java
package besa_adaptado.perfiles;

import adaptation.common.ProfileBESA;
import java.util.ArrayList;

public class PerfilDispositivo extends ProfileBESA {

/**
     * @param nombre representa la marca del dispositivo
     *      @param resolucion del dispositivo
     *      @param tamanio (deducido del ancho de la resolución del dispositivo)
     *      @param width representa el ancho de la resolución del dispositivo
     *      @param formatos que soporta el dispositivo
     */
    private String nombre;
    private String resolucion;
    private String tamanio;
    private int width;
    private ArrayList <String> formatos= new ArrayList<String>();
```

FIGURA 9. PERFIL DE DISPOSITIVO.

Fuente: elaboración propia.

El siguiente paso fue generar los Filtros. Cada Filtro posee una categoría que corresponde a un tipo de perfil. Los Filtros son aplicados únicamente si el tipo de Filtro coincide con el tipo de perfil. Por lo tanto, los Filtros implementados fueron el Filtro de estudiante y el Filtro de dispositivo, de acuerdo con las FIA creadas. Dichos Filtros fueron implementados utilizando un motor de reglas llamado Drools (Drools documentation, 2015), el cual permite crear y ejecutar reglas con el fin de generar deducciones. Las primeras reglas fueron basadas en el estilo de aprendizaje del estudiante teniendo en cuenta el estándar IEEE-LOM para OVA. El motor de reglas obtiene los tipos de material de OVA a partir del estilo de aprendizaje del estudiante. Dicho resultado es la entrada para el método que compara los tipos de OVA ya seleccionados con los que coinciden con el color del estudiante para filtrarlos a su medida. Finalmente, a este resultado adiciona las preferencias de despliegue del estudiante. El segundo filtro se basó en la edad del estudiante para deducir su grado de escolaridad teniendo en cuenta el estándar LOM para OVA. Dicho resultado es adicionado al procedimiento anterior. Observe que hasta este punto no se han considerado las características del dispositivo, por tanto, muchas preferencias de despliegue pueden aparecer junto con los tipos del material de OVA escogidos para el estudiante. A continuación, se presenta el llamado a las reglas y las reglas implementadas para este Filtro (figura 10 y 11):

```
public ArrayList<String> estiloAprendizajeXlom(List<ProfileBESA> aList) {
    try {
        // load up the knowledge base
        KnowledgeBase kbase = readKnowledgeBaseEaXlom();// leer archivo eaXlom.drl
        StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();
        KnowledgeRuntimeLogger logger = KnowledgeRuntimeLoggerFactory.newFileLogger(kse

        //agregando estudiantes a BK
        for (int i = 0; i < aList.size(); i++) {
            estudiante = (PerfilEstudiante)aList.get(i);
            ksession.insert(estudiante);
        }
        ksession.insert(ova);
        ksession.fireAllRules(); //se disparan las reglas
        logger.close();

    } catch (Throwable t) {
    }
    return ova.getMaterial();
}

//Función para leer el archivo eaXlom.drl
private KnowledgeBase readKnowledgeBaseEaXlom() throws Exception {
    KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
    kbuilder.add(ResourceFactory.newClassPathResource("eaXlom.drl"), ResourceType.DRL);
    KnowledgeBuilderErrors errors = kbuilder.getErrors();
    if (errors.size() > 0) {
        for (KnowledgeBuilderError error : errors) {
            System.err.println(error);
        }
        throw new IllegalArgumentException("Could not parse knowledge.");
    }
    KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
    kbase.addKnowledgePackages(kbuilder.getKnowledgePackages());
    return kbase;
}
```

FIGURA 10. LLAMADO A LAS REGLAS PERTENECIENTES AL FILTRO DE ESTUDIANTE BASADO EN EL ESTILO DE APRENDIZAJE.

Fuente: elaboración propia.

```
public String gradoEscolaridad(List<ProfileBESA> aList) {
    try {
        /*leyendo archivo de reglas edadEscolaridad.drl*/
        KnowledgeBase kbase = readKnowledgeEdadEscolaridad();// leer archivo edadEscolaridad.drl
        StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();
        KnowledgeRuntimeLogger logger = KnowledgeRuntimeLoggerFactory.newFileLogger(ksession, "test");

        /*agregando dispositivos a la base del conocimiento*/
        for (int i = 0; i < aList.size(); i++) {
            estudiante = (PerfilEstudiante)aList.get(i);
            ksession.insert(estudiante);
        }
        ksession.insert(ova);
        /*disparando las reglas*/
        ksession.fireAllRules();
        logger.close();

    } catch (Throwable t) {
    }
    return ova.getEscolaridad();
}

private KnowledgeBase readKnowledgeEdadEscolaridad() throws Exception {
    KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
    kbuilder.add(ResourceFactory.newClassPathResource("edadEscolaridad.drl"), ResourceType.DRL);
    KnowledgeBuilderErrors errors = kbuilder.getErrors();
    if (errors.size() > 0) {
        for (KnowledgeBuilderError error : errors) {
            System.err.println(error);
        }
        throw new IllegalArgumentException("Could not parse knowledge.");
    }
    KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
    kbase.addKnowledgePackages(kbuilder.getKnowledgePackages());
    return kbase;
}
```

FIGURA 11. LLAMADO A LAS REGLAS PERTENECIENTES AL FILTRO DE ESTUDIANTE BASADO EN LA EDAD DEL ESTUDIANTE.

Fuente: elaboración propia.

En cuanto al Filtro de dispositivo, se tuvo en cuenta su resolución con el fin de reducir el tamaño del dispositivo. A continuación, se presenta el llamado a las reglas y las reglas implementadas para este Filtro (figura 12):

```
public ResultadoFiltro tamanoDevice(List<ProfileBESA> aList) {
    ResultadoFiltro rf = null;
    try {
        // load up the knowledge base
        KnowledgeBase kbase = readKnowledgeBaseDevice();// leer archivo tipoDispositivo.drl
        StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();
        KnowledgeRuntimeLogger logger = KnowledgeRuntimeLoggerFactory.newFileLogger(ksession, "test");

        //agregando dispositivos a BK
        for (int i = 0; i < aList.size(); i++) {
            pd = (PerfilDispositivo) aList.get(i);
            ksession.insert(aList.get(i));

        }
        ksession.fireAllRules(); //se disparan las reglas
        logger.close();
        rf = new ResultadoFiltro(pd.getTamanio());

    } catch (Exception t) {
        System.out.println("ERROR EN REGLAS" + t);
    }
    rf.setCategory(this.getCategory());
    return rf;
}

//Función para leer el archivo tipoDispositivo.drl
private static KnowledgeBase readKnowledgeBaseDevice() throws Exception {
    KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
    kbuilder.add(ResourceFactory.newClassPathResource("tipoDispositivo.drl"), ResourceType.DRL);
    KnowledgeBuilderErrors errors = kbuilder.getErrors();
    if (errors.size() > 0) {
        for (KnowledgeBuilderError error : errors) {
            System.err.println(error);
        }
        throw new IllegalArgumentException("Could not parse knowledge.");
    }
    KnowledgeBase kbase = KnowledgeBaseFactory.newKnowledgeBase();
    kbase.addKnowledgePackages(kbuilder.getKnowledgePackages());
    return kbase;
}
```

FIGURA. 12. LLAMADO A LAS REGLAS PERTENECIENTES AL FILTRO DE DISPOSITIVO.

Fuente: elaboración propia.

Finalmente, se implementó el Mecanismo de Adaptación. Este une la información de alta abstracción obtenida por cada Filtro. En este caso, del Filtro de Estudiante recibe los tipos de material de ova a partir de su estilo de aprendizaje, así como el color con sus preferencias de despliegue, y recibe del Filtro de Dispositivo el tamaño del dispositivo. El Mecanismo de Adaptación toma ambas respuestas y las valida. Dicha

validación consiste en seleccionar los tipos de material de OVA con las preferencias de despliegue del estudiante que coincidan con el tamaño del dispositivo. Esto se hace para asegurar que el OVA seleccionado sea soportado por el dispositivo. Por ejemplo, si el dispositivo es pequeño, las únicas preferencias de despliegue del estudiante que se tendrán en cuenta serán aquellas que puedan ser desplegadas en dispositivos pequeños.

5. Pruebas

A continuación se presentan los resultados obtenidos de las pruebas aplicadas a este escenario. Se puede apreciar que cuando Juan hace la búsqueda en Google Académico puede recibir cerca de 70 500 resultados (figura 13). Estos resultados son generales y se presentan a cualquier usuario que realice la misma consulta. En figura 14 se puede apreciar la misma búsqueda en Google.

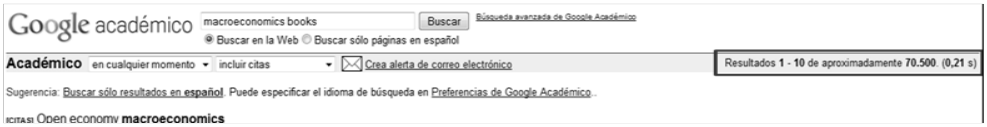


FIGURA 13. CONSULTA DE "MACROECONOMICS BOOKS" EN GOOGLE ACADÉMICO.

Fuente: tomado de https://scholar.google.es/schhp?hl=es.



FIGURA 14. CONSULTA DE "MACROECONOMICS BOOKS" EN GOOGLE.

Fuente: tomado de https://scholar.google.es/schhp?hl=es.

Al usar el *framework* AES, en este caso Google y Google Académico, el número de resultados decrece notablemente (figura 15). Los resultados son más específicos debido a la consulta enriquecida, ya que tiene más información acerca del formato del libro (.pdf), el nivel de experticia de Juan (avanzado) y restricciones del dispositivo de acceso (baja resolución).

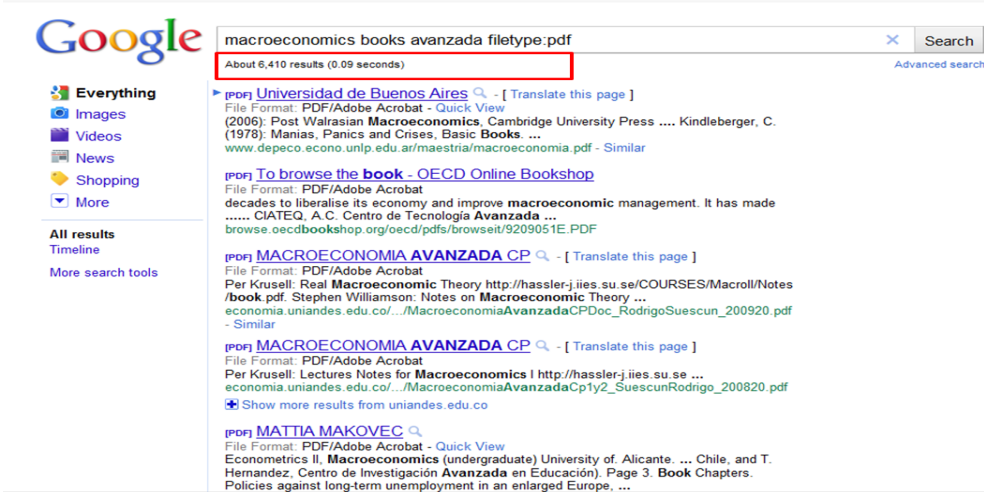


FIGURA 15. USO DE AES EN UNA APLICACIÓN EJECUTADA EN UN DISPOSITIVO MÓVIL.

Fuente: tomado de https://scholar.google.es/schhp?hl=es.

6.Trabajos relacionados

La adaptación de información es un proceso que considera múltiples variables con el fin de generar cambios en una entidad particular y cumplir con el propósito de enriquecer la consulta y así obtener información ajustada a las características del usuario y de su contexto (localización, dispositivo de acceso, etc.). Esta sección presenta algunos trabajos relacionados sobre sistemas que adaptan información (algunos de ellos en entornos móviles).

Algunos *frameworks* y aplicaciones son móviles, es decir, que toman en cuenta la localización para enriquecer el servicio. Un ejemplo es Waze (2015), un sistema que ofrece servicios de enrutamiento y mapeo. Los trabajos de Anantharam *et al*. (2015) y SmartRoad (Hu *et al*., 2015) presentan sistemas que también consideran la información de localización y de comunidad con el fin de definir el grado de cercanía o lejanía de un grupo o nicho específico. Aurigo (Yahi *et al*., 2015) es una guía

turística que toma en cuenta la localización para ofrecer una guía personalizada a sus usuarios, generándole una trayectoria particular.

La tabla I compara las características de AES con otros sistemas que adaptan información. Los criterios de comparación son: a) Sistema genérico: la aplicación puede ser usada en diferentes escenarios; b) Información contextual: la aplicación considera aspectos contextuales tales como infraestructura, características socioculturales, etc.; c) Información del usuario: la aplicación considera aspectos de usuario tales como preferencias, gustos e intereses; d) Información de dispositivo de acceso: la aplicación considera aspectos del dispositivo de acceso tales como características físicas, *hardware*, *software*, tipos de conexiones, etc.; e) Movilidad: la aplicación considera cambios en el entorno del usuario. La notación es '+' si el trabajo considera el aspecto; '-' si no lo considera; '++' si el trabajo explícitamente lo considera y hace énfasis en su utilización.

TABLA 1. TRABAJOS RELACIONADOS CON AES

	Waze, 2015	**Anantharam** *et al.*, 2015	**Hu** *et al.*, 2015	**Yahi** *et al.*, 2015	AES
Sistema genérico	-	-	-	-	++
Información contextual	++	++	++	+	++
Información del usuario	+	++	+	++	++
Información del dispositivo de acceso	+	-	-	-	++
Movilidad	+	+	+	+	++

Fuente: elaboración propia.

Adicionalmente a los criterios evaluados en la tabla 1, algunas aplicaciones no proveen la posibilidad de adicionar parámetros al servicio con el fin de enriquecerlo (Anantharam *et al.*, 2015; Yahi *et al.*, 2015) o no son flexibles para integrarse con otras aplicaciones o tecnologías (Hu *et al.*, 2015; Waze, 2015). AES considera estas limitaciones y puede ser aplicado en diferentes escenarios correspondientes a diversos dominios, considerando las características de los usuarios, la localización, las limitaciones del dispositivo de acceso, entre otras características contextuales.

7. Conclusiones

AES es un *framework* genérico que adapta servicios usando entradas básicas y provee al usuario una experiencia personalizada. AES considera diferentes perfiles que permiten adaptar la información en diversos escenarios. La información incluida en estos perfiles (tales como necesidades y características del usuario, de su dispositivo de acceso y en general de su contexto) se toman en cuenta para ajustar la información que va a ser desplegada en su dispositivo de acceso y que reducirá la información que sea redundante o innecesaria para el usuario. Su carácter genérico se basa en la implementación de una serie de filtros y de mecanismos de adaptación que el programador puede definir y aplicar en diversos contextos, dependiendo de los requerimientos de la aplicación que desea enriquecer.

En este capítulo se presentó la utilización del *framework* AES en un contexto educativo para así obtener AES-EDU, el cual soporta las funcionalidades del proyecto AYLLU. El objetivo de AES-EDU es enriquecer el servicio computacional de seleccionar OVA para un estudiante, explicando el paso a paso de la consulta de un estudiante acerca de un tema específico. El trabajo futuro incluye integrar AES con diferentes proyectos y ejecutar pruebas de desempeño, particularmente para el proceso de adaptación en otros escenarios (ej., salud, organizaciones y manejo de crisis).

Referencias

Anantharam, P., Barnaghi, P., Thirunarayan, K. y Sheth, A. (2015). Extracting City Traffic Events from Social Streams. *Journal ACM Transactions on Intelligent Systems and Technology*, 6(4), 1-27.

Boudaa, B., Hammoudi, S., Bouguessa, A. y Chikh, M. (2014). Supporting Runtime Adaptation of Context-Aware Services. *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications*, 24-30.

Carrillo-Ramos, Á., Arias-Báez, M. P., Torres-Ribero, L. G., Pomares-Quimbaya, A. y González, E. (2012). *Enriching Services Using Agents in Mobile Environments*. *ICMCS*, 821-826.

Carrillo-Ramos, A., Torres-Ribero, L., Arias-Báez, M., Pomares, A., González, E., Carreño, J., Garzón-Ruiz, J. y Martin, H. (2015). Generic Framework for Enriching Services: A Multiagent Approach. *International Journal of Web Information System*, *11*(2), 246-268.

Drools documentation. (2015). Página principal. Disponible en http://www.jboss.org/.

Herrmann, N. (1989). *The creative brain*. North Carolina: Brain Books.

Hu, S., Su, L., Liu, H., Wang, H. y Abdelzaher, T. (2015). SmartRoad: Smartphone-Based Crowd Sensing for Traffic Regulator Detection and Identification. *Journal of ACM Transactions on Sensor Networks*, *11*(4), 1-27.

Kart, M. (2015). Effects of Learning Styles on Student Outcomes in A General Education Computing Course. *Journal of Computing Sciences in Colleges*, *30*(4), 37-43.

Nepal, S., Paris, C., Pour, P., Freyne, J. y Bista, S. (2015). Interaction-Based Recommendations for Online Communities. *Transactions on Internet Technology*, *5*(2), 1-21.

Van Riemsdijk, M., Dennis, L., Fisher, M. y Hindriks, K. (2015). A Semantic Framework for Socially Adaptive Agents: Towards strong norm compliance. *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, 423-432.

Velden, T., Bietz, M., Diamant, E., Herbsleb, J., Howison, J., Ribes, D. y Steinhardt, E. (2014). Sharing, Re-use and Circulation of Resources in Cooperative Scientific Work. *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW'14)*, 347-350.

Waze. 2015. Página principal. Disponible en https://www.waze.com/.

Yahi, A., Chassang, A., Raynaud, L., Duthil, H. y Chau, H. (2015). Aurigo: An Interactive Tour Planner for Personalized Itineraries. *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI'15)*, 275-285.

9
MATEO: SISTEMA ADAPTATIVO PARA LA CONFORMACIÓN DE EQUIPOS DE TRABAJO

María Paula Arias-Báez, Ángela Carrillo-Ramos

1. Introducción

Se conoce como equipo al conjunto de personas que, en beneficio de sus propios intereses, influye en una organización, esfera o actividad social. Estos equipos llevan intrínsecamente una labor de colaboración entre sus miembros, a partir de sus capacidades complementarias y responsabilidades compartidas, para lograr la consecución colectiva de una meta. Existen diferentes tipos de equipos, caracterizados principalmente por su interacción y por el objetivo que persiguen como comunidad. De esta forma, se pueden reunir las personas idóneas para dicho equipo a partir de sus habilidades individuales y colectivas, las funciones que van a desarrollar, los intereses y preferencias individuales, además del nivel de conocimiento que maneja cada una sobre el tema que se va a trabajar; sin embargo, esto no es tan fácil de lograr (Gogoulou *et al*., 2014). Por lo general, la conformación de equipos de trabajo se realiza de tres maneras empíricas: a) se reúnen las personas que se conocen o han trabajado previamente; b) se imponen equipos de trabajo donde se desconoce completamente las fortalezas y debilidades de cada uno de sus integrantes; c) se reúnen personas conflictivas que pueden retrasar o entorpecer el trabajo del equipo. En estos casos, no se tienen en cuenta los perfiles de cada uno de los integrantes, cuáles son sus fortalezas y debilidades o demás características que podrían fortalecer al equipo al cual van a pertenecer. Lo anterior hace que se desaprovechen las habilidades de los participantes, lo cual es observable en el no cumplimiento satisfactorio de las metas, en aplazamientos de entregas y en problemas de tipo inter grupal, entre otras dificultades.

Se define *trabajo colaborativo* como "la nominación general y neutral de múltiples personas que trabajan juntas para producir un producto o servicio" (Kulkarni *et al*., 2012). Los participantes que intervienen en él comparten metas y las cumplen a partir de la realización de sus propias tareas y de las compartidas, se comunican a través de interacciones directas e indirectas, ya sean centralizadas o distribuidas (Prilla y Renner, 2014). Este tipo de trabajo se puede potencializar a través de herramientas como los sistemas de apoyo a la colaboración por computador (CSCW, trabajo cooperativo asistido por computadora por sus siglas en inglés) (Bjorn *et al*., 2014), las cuales brindan facilidades para compartir información, colaborar y ver el progreso de la labor que están llevando a cabo. Sin embargo, trabajar de manera colaborativa no es sencillo y en ocasiones, por acciones de los participantes, aparecen diversos aspectos que pueden entorpecer la meta definida, como por ejemplo: a) falta de coordinación del trabajo, lo cual genera pérdidas de tiempo productivo, b) influencias inapropiadas de las dinámicas de equipo, c) tendencia a "descansar en otros", d) responsabilidades no claras, e) uso incompleto o inapropiado de la información que generan costos de reunión, entre otros (Kulkarni *et al*., 2012).

Estos aspectos permiten pensar que si se escogiesen las personas que contaran con las competencias y comportamientos necesarios para colaborar y alcanzar la meta conjunta, dichos aspectos entorpecedores podrían ser minimizados. Por tanto, se puede pensar en analizar la conformación de los equipos de trabajo, fase inicial del trabajo colaborativo, que, a diferencia de las etapas de desarrollo y cierre (figura 1), no cuenta con suficientes herramientas que soporten dicha agrupación y beneficien las etapas siguientes.



FIGURA 1. FASES DEL TRABAJO COLABORATIVO.

Fuente: elaboración propia basada en Bjorn *et al*. (2014).

De llegarse a minimizar los aspectos entorpecedores, los beneficios podrían ser: a) un equipo entiende mejor un problema que una sola persona, b) responsabilidad de tareas compartidas, c) facilita la detección de errores, d) un equipo posee más información (conocimiento) que un solo miembro, tiene más alternativas en la resolución de problemas, e) se produce sinergia: la efectividad y calidad de la producción de un equipo es mayor que la suma de lo que pueda producir cada miembro en forma individual, entre otros (Bjorn *et al*., 2014). Considerando todo lo mencionado, surge la necesidad de crear un sistema genérico para adaptar la conformación de equipos de trabajo, a partir de las características relevantes de los candidatos, del contexto y los requisitos del equipo con el fin de minimizar los problemas descritos anteriormente. Dicho sistema se llama MATEO (acrónimo de *Making Adapted Teams Oriented to Collaboration*) y permite adaptar la conformación de equipos de trabajo con el fin de minimizar los problemas mencionados.

2. Modelo del sistema MATEO

MATEO es un sistema genérico que apoya el proceso de conformación adaptada de equipos de trabajo. Por "adaptada" se refiere a que el sistema utiliza adaptación de la información entendida como el proceso de enriquecimiento de un servicio, en este caso, el servicio corresponde a la conformación de equipos de trabajo considerando características del usuario o del equipo en un contexto específico (Belgharbi y Boufaida, 2015). El enriquecimiento será tomado como adición de características relevantes de las personas, contexto y equipo. Como consecuencia, la aplicación que utiliza el servicio de conformación podrá refinar tal servicio e, incluso, corroborar las respuestas de los candidatos (ya sea por cambios de la persona o por posibles incongruencias entre sus respuestas).

Los equipos conformados por MATEO pueden ser: a) Incluyente-Homogéneo: escogencia de personas con similitudes en el nivel de las características escogidas. Las personas que no clasifiquen dentro de dicha escogencia deben ser agregadas a los equipos ya conformados (ningún candidato es excluido); b) Incluyente-Heterogéneo: escogencia de personas con diferencias en el nivel de las características escogidas. Las personas que no clasifiquen dentro de dicha escogencia deben ser agregados a los equipos ya conformados (ningún candidato es excluido); c) Excluyente-Homogéneo: escogencia de personas con similitudes en el nivel de las características escogidas. Las personas que no clasifiquen dentro de dicha escogencia no se tienen en cuenta; d) Excluyente-Heterogéneo: escogencia de personas con diferencias en el nivel de las características escogidas. Las personas que no clasifiquen dentro de dicha escogencia

no se tienen en cuenta; e) Mixto: escogencia de personas que satisfagan una característica escogida y sean diferentes en otra característica.

2.1. *Modelo lógico*

El modelo lógico de MATEO consta de la integración de dos modelos: a) el proceso de conformación (JOIN): es una secuencia de pasos para realizar la agrupación de participantes a partir de los criterios requeridos por el sistema; y b) el perfil de equipo (MADA): son datos que enriquecen el proceso de conformación a partir del conocimiento de las características relevantes de colaboración, contexto y particulares de los candidatos. En las siguientes secciones se presentan JOIN, MADA y cómo se integran para crear MATEO.

2.1.1. JOIN: modelo de conformación de equipos

JOIN (figura 2) corresponde al protocolo de interacción necesario para conformar un equipo de trabajo sin importar el contexto de uso. Como precondición del sistema es necesaria la lista de personas (candidatos) y como poscondición, el sistema genera como respuesta la lista de equipos conformados con los participantes de cada uno. El proceso que se lleva a cabo es el siguiente:

1. *Completar información contextual*: se define el contexto dentro del cual trabajará el futuro equipo en términos de su contexto social, físico, temporal y particular.
2. *Ingresar requerimientos del equipo*: se definen las características del equipo que se desea conformar en términos de los roles necesarios, cuántas personas por rol son requeridas, qué competencias deben tener por rol, qué actividades deberán realizar y con qué recursos. Además, las características de sinergia y las normas.
3. *Escoger tipo de equipo a conformar*: selección de algún tipo de equipo: Incluyente-Homogéneo, Incluyente-Heterogéneo, Excluyente-Homogéneo o Excluyente-Heterogéneo.
4. *Completar información personal*: se diligencia la información de las personas candidatas a conformar los equipos. Para obtener dicha información es necesario recurrir a diferentes *test* que permitan adquirir información sobre las competencias, comportamientos, conocimientos, etc., de cada uno de los participantes.
5. *Preparar información obtenida*: realizar el *match* (correspondencia) entre las características requeridas por el equipo con las características de los candidatos.
6. *Ejecutar algoritmos según carácter*: según la elección del carácter del equipo (homogéneo, heterogéneo o mixto, figura 3a) se utiliza una tecnología que permita agrupar. El algoritmo 1 presenta el pseudocódigo que se sigue en este paso.

Algoritmo 1: Ejecutar según Carácter

Pre condición:

- RolesSeleccionados = conjunto de roles seleccionados
- archivoCC = archivo de competencias y comportamientos (información de los participantes)
- equipos = cantidad de equipos a conformar
- cantPartXrol = cantidad de participantes por rol seleccionado

Post condición:

- losEquipos = conjunto de equipos conformados. Cada equipo contiene una lista de participantes con su rol

1. *Para cada* rol de RolesSeleccionados *hacer*
2. caracterísitcasRol ← *llamar* escogencia de características *con* rol
3. rangos ← *llamar* preparación de datos *con* caracterísitcasRol
4. Lista<Cluster> clustersConformados ← *llamar* conformación de Clusters *con* evaluación de rangos y archivoCC
5. centroideEscogido ← *llamar* evaluar Centroides *con* clusterConformados
6. clusterSeleccionado ← *llamar* seleccionar Cluster *con* centroideEscogido
7. Lista<Participante> participantesRol ← *llamar* obtención de participantes del Cluster *con* clusterSeleccionado
8. *fin Para*
9. Lista<Equipos> losEquipos
10. *Mientras* i menor a equipos *hacer*
11. Lista<Participante> equipo ← *llamar* colocación de participantes en equipo *con* participantesRol y cantPartXrol
12. **guardar** equipo en losEquipos
13. i++
14. *fin Mientras*

Algoritmo 2: Ejecutar según Selección

Pre condición:

- losEquipos = conjunto de equipos conformados (provenientes del paso Ejecutar según Carácter)

Post condición:

- losEquipos = conjunto definitivo de equipos conformados (actualizado). Cada equipo contiene una lista de participantes

1. *Para cada* equipo de losEquipos *hacer*
2. distribucion ← *llamar* actualizar Distribución *con* losEquipos
3. cantFaltante ← *llamar* participantes faltantes por equipo *con* equipo
4. Lista<Iteracion> iteraciones ← *llamar* selección aleatoria de participantes *con* distribución
5. iteracion ← *llamar* evaluar Funcion Objetivo *con* iteraciones
6. participantes ← *llamar* participantes de la iteración escogida *con* iteracion
7. equipo ← *actualizar* equipo *con* participantes
8. **guardar** equipo **en** losEquipos
9. *Fin Para*

7. *Ejecutar según selección*: se ingresan, o no, todos los candidatos a los equipos (incluyente o excluyente) (figura 3b). Para esto se utilizan reglas que permitirán realizar el tipo de selección deseado. El algoritmo 2 presenta el pseudocódigo que se sigue en este paso.
8. *Generar respuesta*: se obtienen finalmente los diferentes equipos conformados (vacío, unitario, múltiples) a partir de la escogencia de tipo de equipo a conformar.

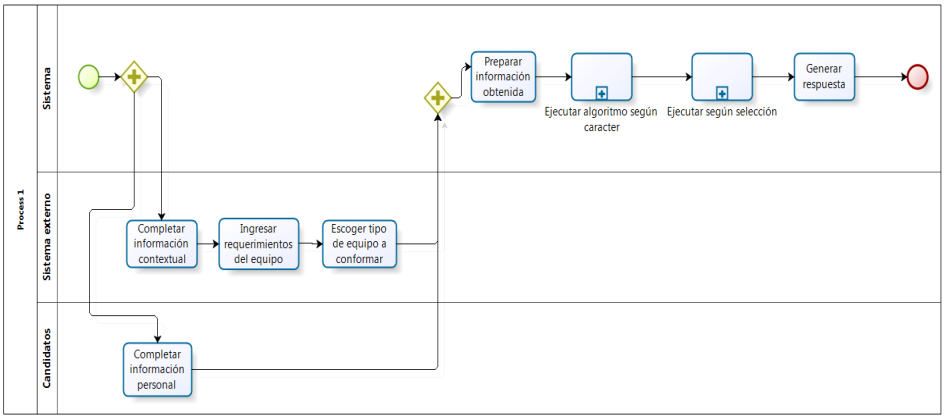


FIGURA 2. PROTOCOLO DE INTERACCIÓN DE JOIN.

Fuente: elaboración propia.

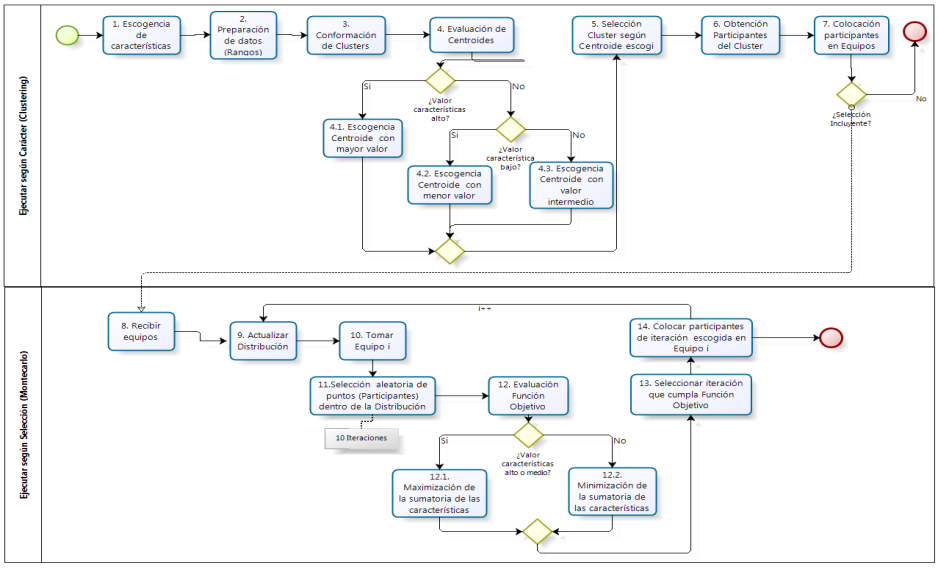


FIGURA 3. EJECUTAR SEGÚN A) CARÁCTER DEL EQUIPO; B) SELECCIÓN DEL EQUIPO.

Fuente: elaboración propia.

2.1.2. MADA: modelo de adaptación

A continuación se presenta MADA (figura 4), constituido por tres perfiles:

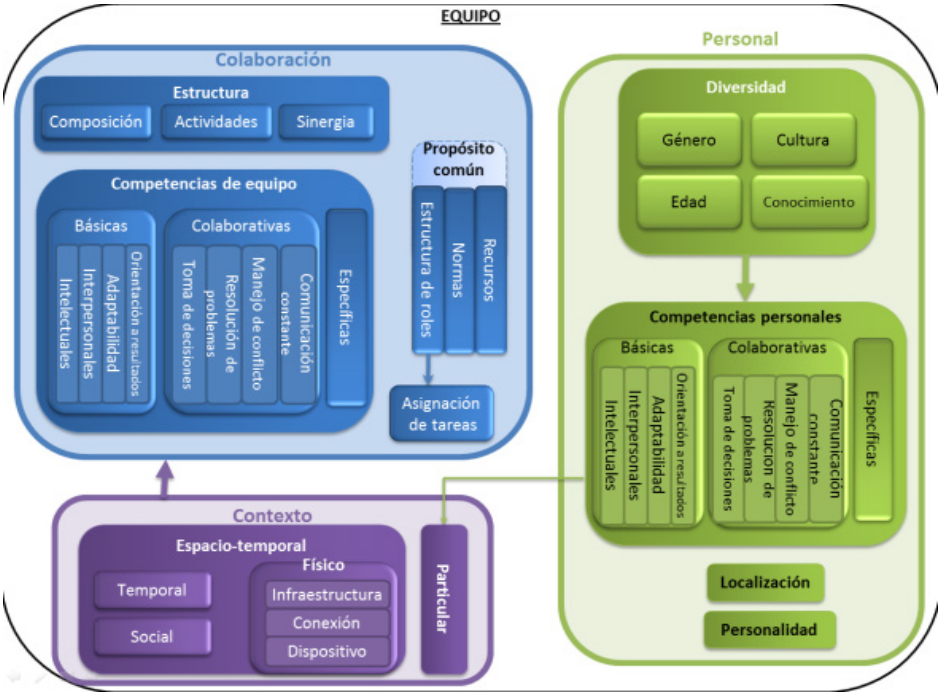


FIGURA 4 MADA: MODELO DE ADAPTACIÓN (PERFIL DE EQUIPO).

Fuente: elaboración propia.

El Perfil de Colaboración determina los valores requeridos por el equipo con los cuales se podrá realizar la selección de candidatos que coincidan con estos valores. Adicionalmente, este perfil también posee aspectos que contribuyen a propiciar colaboración. Dentro de los componentes de este perfil se encuentran:

i. Estructura: organización básica del equipo a conformar (qué tipo de equipo se va a conformar –homogéneo, heterogéneo o mixto–, las actividades que se llevarán a cabo dentro del equipo y su sinergia (características colaborativas, tanto de valores como de comportamientos, que deben tener los futuros miembros del equipo).
ii. Competencias de equipo: comportamientos que una persona puede tener, que es capaz de transformarla y de hacerla eficaz para una situación dada (Alles, 2008). Estas competencias pueden ser básicas (competencias que cualquier individuo

debe tener), competencias que fomentan la colaboración entre los participantes y competencias específicas del contexto de trabajo.

iii. Estructura de roles: cargos necesarios dentro del equipo, con sus permisos y responsabilidades asociados. A partir del rol se asignan tareas.
iv. Normas: reglas de conducta compartidas por todos sus participantes.
v. Recursos: conjunto de personas, bienes materiales y técnicos con que cuenta el equipo.

El Perfil Personal corresponde al perfil de un candidato que formará parte del equipo. Los perfiles personales que coincidan con los valores requeridos por el perfil de colaboración, inician el proceso de ser asignados. Dentro de los componentes de este perfil se encuentra: a) género (masculino o femenino); b) edad; c) cultura: conjunto de modelos o patrones, explícitos o implícitos, a través de los cuales una sociedad regula el comportamiento de las personas que la conforman; d) conocimiento: conjunto de información desarrollada en el contexto de una experiencia y transformada a su vez en otra experiencia para la acción. El conocimiento permite percibir escenarios nuevos, de cambio y tomar decisiones; e) competencias (el nivel que posee la persona en las competencias básicas, de colaboración y específicas requeridas por el equipo); f) localización: lugar donde se halla el participante; g) personalidad: suma total de las formas en que un individuo reacciona y se relaciona con los demás (tanto de valores como de comportamientos).

El Perfil de Contexto se compone de las características del ambiente en el que se crea el equipo que pueden condicionar su interacción. Dentro de los componentes de este perfil se encuentra: a) espacio-temporal: lugar e infraestructura en el que se desarrolla el equipo, tiempo que posee el equipo para interactuar; b) contexto particular: contexto específico en el que se desenvuelve el equipo; c) cultura: proceso de socialización que se da dentro de una organización, basado en valores, creencias, presunciones básicas, principios.

2.1.3. Integración de modelos

A partir de los dos modelos JOIN y MADA se construye MATEO (figura 5): la conformación de equipos de trabajo se inicia con la obtención de características relevantes de los participantes, del contexto y de los requisitos del equipo que se quiere organizar con el objetivo de ejecutar el protocolo de interacción que permite la creación de tales equipos, considerando criterios de decisión más sólidos para la asignación de roles y de participantes por equipo (Arias-Báez *et al*., 2012).

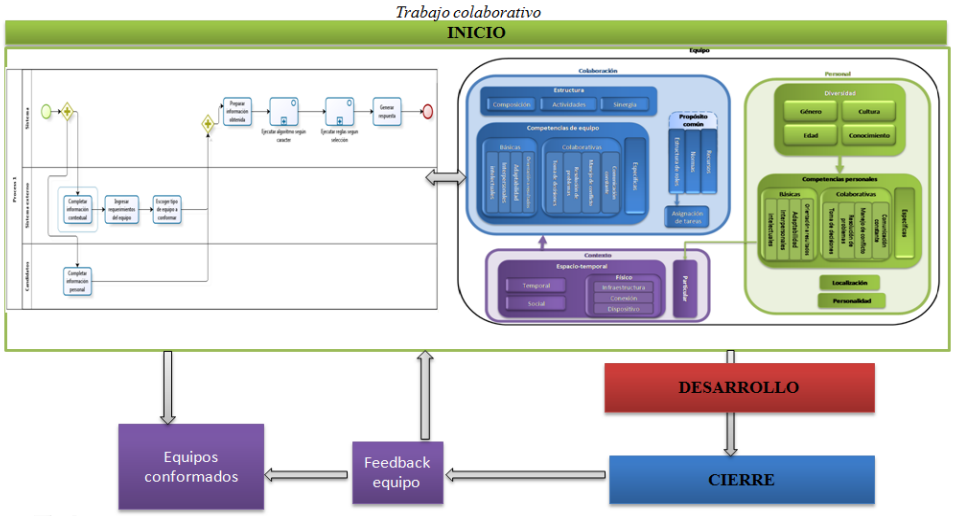


FIGURA 5. INTEGRACIÓN DE MATEO DENTRO DEL CICLO DE VIDA DEL TRABAJO COLABORATIVO.

Fuente: elaboración propia.

Los equipos son conformados con base en los requisitos requeridos por el solicitante de equipos (información del Perfil de Colaboración). Teniendo en cuenta dichos requisitos, se procede a evaluar los comportamientos y competencias de los participantes (de sus perfiles personales) para conocer quiénes poseen los valores requeridos para ser seleccionados. Una vez conformados los equipos, sus participantes pasan a trabajar conjuntamente durante el desarrollo y cierre del trabajo colaborativo. Al finalizar esta labor, el sistema se puede retroalimentar por parte de sus usuarios para conocer la usabilidad del sistema (Ounnas *et al*., 2007) (por ejemplo, evaluar si aparecieron o no los problemas sobre coordinación, cumplimiento de la meta, influencias negativas dentro del equipo, tiempo productivo, manejo de información, etc.), se puede además evaluar si el equipo logró alcanzar las metas propuestas y se podría medir la calidad del trabajo de dicho equipo. Una vez se hayan evaluado estos aspectos, el usuario del sistema podría tomar la decisión de almacenar la conformación del equipo para futuras ocasiones en las que se presente un contexto similar.

2.2. *Implementación de* MATEO

Inicialmente, MATEO fue implementado como una aplicación *stand-alone* bajo el lenguaje de programación Java. Su implementación se basó en el protocolo de

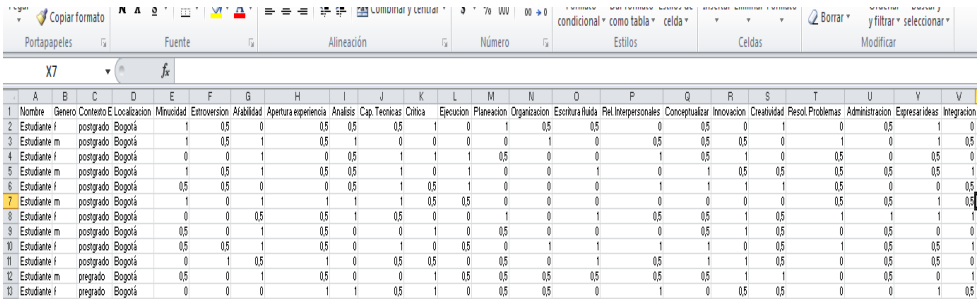
interacción definido en el modelo JOIN, obteniendo información de los perfiles definidos en el modelo MADA.

2.2.1. Configuración de MATEO como aplicación *Stand-Alone*

En cuanto a sus configuraciones, MATEO requiere:

i. El ingreso de un archivo con los valores de cada una de las características que el conformador de equipos desee evaluar. Dicho archivo debe tener extensión .csv (Excel delimitado por comas).
ii. Los datos requeridos en las interfaces de inicio y de selección de características/roles.

La figura 7 muestra un ejemplo del archivo .csv. Los valores de dicho archivo están entre 0 y 1.



	Nombre	Genero	Contexto E	Localizacion	Minucidad	Extroversion	Afabilidad	Apertura experiencia	Analisis	Cap. Tecnicas	Critica	Ejecucion	Planeacion	Organizacion	Escritura fluida	Rel. Interpersonales	Conceptualizar	Innovacion	Creatividad	Resol. Problemas	Administracion	Expresar ideas	Integracion
2	Estudiante f	postgrado	Bogotá		1	0,5	0	0,5	0,5	0,5	1	0	1	0,5	0,5	0	0,5	0	1	0	0,5	1	0
3	Estudiante m	postgrado	Bogotá		1	0,5	1	0,5	1	0	0	0	0	1	0	0,5	0,5	0,5	0	1	0	1	0,5
4	Estudiante f	postgrado	Bogotá		0	0	1	0	0,5	1	1	1	0,5	0	0	1	0,5	1	0	0,5	0	0,5	0
5	Estudiante m	postgrado	Bogotá		1	0,5	1	0,5	0,5	1	0	1	0	0	1	0	1	0,5	0,5	0,5	0,5	0,5	1
6	Estudiante f	postgrado	Bogotá		0,5	0,5	0	0	0,5	1	0,5	1	0	0	0	1	1	1	1	0,5	0	0	0,5
7	Estudiante m	postgrado	Bogotá		1	0	1	1	1	1	0,5	0,5	0	0	0	0	0	0	0	0,5	0,5	1	0,5
8	Estudiante f	postgrado	Bogotá		0	0	0,5	0,5	1	0,5	0	0	1	0	1	0,5	0,5	1	0,5	1	1	1	1
9	Estudiante m	postgrado	Bogotá		0,5	0	1	0,5	0	0	1	0	0,5	0	0	0	0,5	1	0,5	0	0,5	0	0
10	Estudiante f	postgrado	Bogotá		0,5	0,5	1	0,5	0	1	0	0,5	0	1	1	1	1	0	0,5	1	0,5	0,5	1
11	Estudiante f	postgrado	Bogotá		0	1	0,5	1	0	0,5	0,5	0	0,5	0	1	0,5	1	1	0,5	0	0,5	0,5	0
12	Estudiante m	pregrado	Bogotá		0,5	0	1	0,5	0	0	1	0,5	0,5	0,5	0,5	0,5	0,5	1	1	0	0,5	0	1
13	Estudiante f	pregrado	Bogotá		0	0	0	1	1	0,5	1	0	0,5	0,5	0	1	0	0,5	0,5	0	0	1	0,5

FIGURA 7. CAMPOS DEL ARCHIVO .CSV CON INFORMACIÓN DE PARTICIPANTES.

Fuente: elaboración propia.

Los campos que debe tener el archivo .csv deben basarse en el conjunto de características que utiliza MATEO para realizar agrupaciones, como son:

TABLA 1. CARACTERÍSTICAS PREDETERMINADAS

DATOS PERSONALES	Nombre Género Contexto educativo Localización	COMPETENCIAS Y COMPORTAMIENTOS	Minuciosidad Extroversión Afabilidad Apertura experiencia Análisis Capacidades técnicas Crítica Ejecución Planeación Organización Escritura fluida Relaciones interpersonales Conceptualizar Innovación Creatividad Resolución de problemas Administración Expresar ideas Integración

Fuente: elaboración propia.

Los roles predeterminados dentro de MATEO PURO fueron:

TABLA 2. ROLES PREDETERMINADOS

Rol	Características relacionadas con el rol
Secretario	Minuciosidad
	Ejecución
	Planeación
	Escritura fluida
Presentador	Extroversión
	Afabilidad
	Relaciones interpersonales
	Expresar ideas
Diseñador	Apertura de experiencia
	Conceptualización
	Innovación
	Creatividad
	Integración
Operativo	Análisis
	Capacidades técnicas
	Crítica
	Resolución de problemas
Coordinador	Organización
	Resolución de problemas
	Administración

Fuente: elaboración propia.

2.2.2. Funcionamiento

Como primera medida, se implementaron los perfiles de Equipo y de Estudiante (correspondientes al perfil de Colaboración y Personal presentados en MADA). Se redujo la cantidad de perfiles y sus respectivos atributos con el fin de realizar una comprobación conceptual de la conformación de equipos de trabajo teniendo en cuenta características de colaboración. El siguiente paso fue realizar la implementación de los mecanismos de *Clustering* apoyados en algunas funcionalidades del *software* WEKA (WEKA, 2015). Para realizar dichas agrupaciones, se prepararon los datos que recibe WEKA y se escogió la función de distancia euclidiana para identificar la similitud entre instancias, donde el término similitud hace referencia a la cercanía entre dos o más instancias a partir

de los valores evaluados. De esta manera, se identifica un *cluster* para cada instancia (se juntan las instancias con valores parecidos) y se obtienen los *centroide*s (promedio de todos los puntos) de cada *cluster*. Sobre el análisis de dichos *centroide*s, se realiza la escogencia del *cluster* que coincida mayormente con las características requeridas. El funcionamiento de *clustering* dentro del sistema se muestra el Algoritmo 1 y se ejemplifica dentro de la articulación de MATEO con el *framework* AES (pasos 1 al 7, proceso Ejecutar Según Carácter, figura 3a).

Posteriormente, se realizó la implementación del método de Montecarlo para la conformación de equipos incluyentes: los puntos aleatorios corresponden a los participantes escogidos de manera aleatoria dentro del conjunto total de participantes, donde todos tienen la misma posibilidad de ser escogidos (distribución uniforme); el óptimo global se toma como la maximización de la sumatoria de las características de colaboración de los escogidos aleatoriamente. Las características de colaboración escogidas fueron: a) resolución de problemas, b) administración, c) expresar ideas, d) integración. Este método se lleva a cabo con cada equipo requerido. El funcionamiento de Montecarlo se presenta en el Algoritmo 2 y se ejemplifica dentro de la articulación de MATEO con el *framework* AES (pasos 8 a 14, proceso Ejecutar Según Carácter, figura 3b). A continuación, se presentan algunas de las interfaces que despliega MATEO para realizar su proceso de conformación de equipos (para mayor detalle, por favor revisar la referencia) (Arias-Báez, s. f.):

A través de la interfaz de inicio (figura 8) se obtiene el archivo de los resultados del test realizado a los estudiantes. Adicionalmente, el conformador de equipos debe seleccionar el tipo de equipo que se va a conformar (Incluyentes-Homogéneos / Heterogéneos o Excluyentes-Homogéneos/ Heterogéneos), la cantidad de equipos que desea y cuántas personas debe haber por equipo.

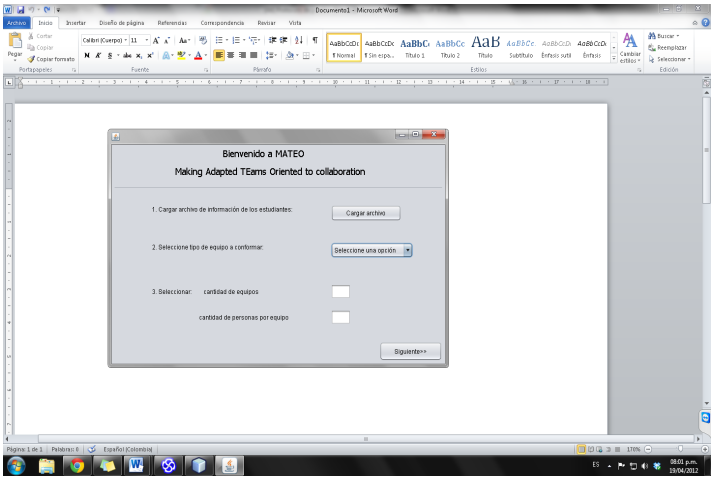


FIGURA 8. INTERFAZ DE INICIO.

Fuente: elaboración propia.

En el caso en el que el carácter del equipo sea Homogéneo, la siguiente interfaz (de características) es desplegada (figura 9) para que el conformador del equipo escoja las características que considera que deben tener las personas que conformen sus equipos, así como el nivel de cada una de dichas características (alto, medio o bajo puntaje).

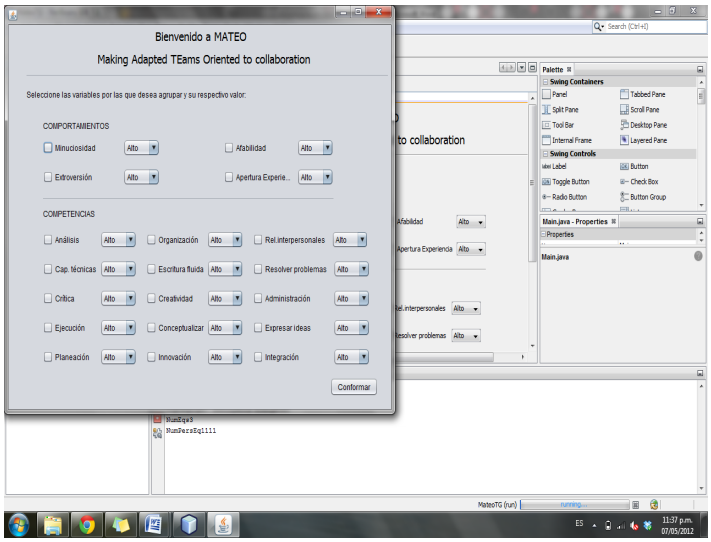


FIGURA 9. INTERFAZ DE CARACTERÍSTICAS (CARÁCTER HOMOGÉNEO).

Fuente: elaboración propia.

En el caso en el que el carácter sea Heterogéneo, la siguiente interfaz (de roles) es desplegada (figura 11) para que el conformador del equipo escoja los roles que necesita dentro de los equipos y cuántas personas por rol requiere. Internamente, cada uno de los roles posee una serie de características asociadas y el valor de cada una de ellas es *alto*.

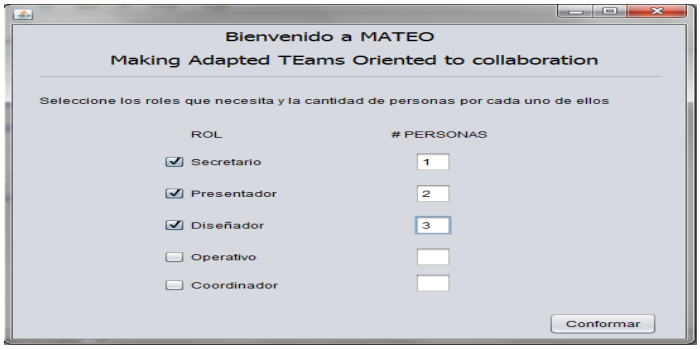
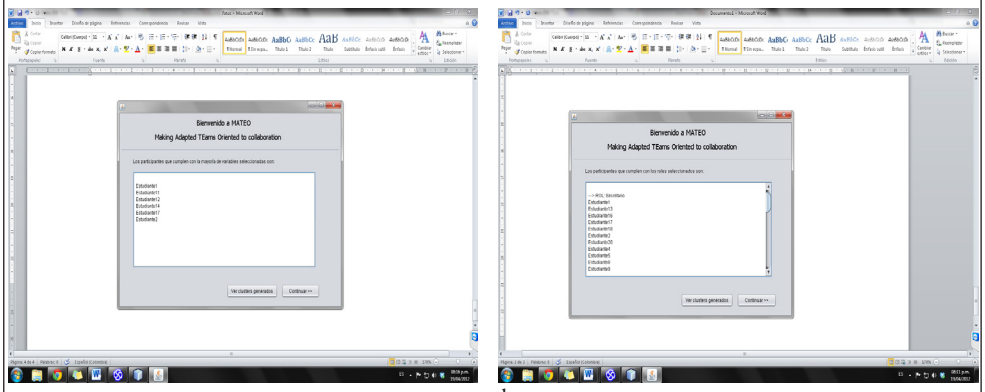


FIGURA 10. INTERFAZ DE ROLES (CARÁCTER HETEROGÉNEO).

Fuente: elaboración propia.

Esta interfaz (figura 11) presenta los participantes que cumplen con la mayoría de características seleccionadas en el nivel requerido por el conformador de equipos (figura 11a). En el caso de carácter Heterogéneo presenta los participantes que cumplen con las características requeridas por cada uno de los roles escogidos (figura 11b).



a. b.

FIGURA 11. INTERFAZ RESPUESTA DEL PROCESO EJECUTAR SEGÚN CARÁCTER; A) HOMOGÉNEO, B) HETEROGÉNEO.

Fuente: elaboración propia.

Esta interfaz (figura 12) presenta los participantes que pertenecen a cada equipo conformado, además de la descripción de los equipos (características/roles escogidos, cantidad de equipos y personas por equipo).

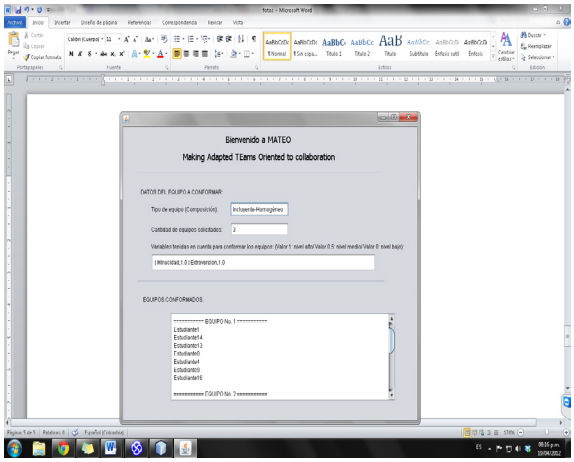


FIGURA 12. INTERFAZ RESPUESTA DEL PROCESO "EJECUTAR SEGÚN SELECCIÓN".

3. Trabajos relacionados

El equipo de trabajo es un conjunto reducido de personas con habilidades complementarias y comprometidas con un propósito común, una serie de metas de desempeño y un método de trabajo del cual todas ellas son mutuamente responsables. A su vez, cada persona responde por las responsabilidades asignadas relacionadas con

el rol que desempeñan. En la tabla 3 se presenta una comparación de los trabajos relacionados con Conformación de Equipos (CE) bajo diversos criterios. Según los resultados de esta tabla, la mayoría de tales trabajos se dedican particularmente a generar equipos homogéneos y heterogéneos. Adicionalmente, no es claro cómo las características de usuario pueden ayudar a enriquecer el proceso de conformación y tampoco consideran elementos contextuales o de colaboración influyentes para la conformación de equipos.

Por lo anterior, se detectó la necesidad, a la hora de conformar equipos de trabajo, de contar con mecanismos que consideren diversos aspectos (tanto de los participantes como de su entorno de trabajo y de colaboración) que puedan potenciar el alcance de las metas propuestas en un contexto sinérgico. Un posible mecanismo puede ser el uso de Adaptación de Información mediante la definición de perfiles (conjunto de características relevantes según el usuario, el contexto, el dispositivo, etc.) y la posibilidad de relacionarlos y obtener información de mayor abstracción (nuevos conocimientos a partir de los ya almacenados), con el fin de enriquecer el proceso de conformación de equipos de trabajo.

El trabajo colaborativo también permitirá observar posibles características relevantes dentro de un equipo como el comportamiento de sus participantes, las herramientas que utilizan para comunicarse y el grado de colaboración entre ellos para alcanzar la meta común. Finalmente se puede apreciar en los trabajos analizados la limitada cantidad de tipos de equipo que se pueden conformar de manera automática. Por tal razón, surge la idea de ampliar dicha conformación a incluyentes y excluyentes, lo cual será de gran aporte dependiendo del contexto en el que se conformarán los equipos de trabajo.

TABLA 3. COMPARACIÓN ENTRE TRABAJOS RELACIONADOS CON MATEO

	Christodoulopoulos y Papanikolaou, 2007	**Craig *et al*., 2010**	**Brusilovsky *et al*., 2007**	**Gogoulou *et al*., 2014**	**Kulkarni *et al*., 2012**	MATEO
CE homogéneos	+	+	+	+	+	+
CE heterogéneos	+	+	+	+	-	+
CE mixtos	-	-	+	+	-	+
CE incluyentes	-	-	-	-	-	+
CE excluyentes	-	-	-	-	-	+

	Christodoulopoulos y Papanikolaou, 2007	Craig *et al*., 2010	Brusilovsky *et al*., 2007	Gogoulou *et al*., 2014	Kulkarni *et al*., 2012	MATEO
Características de usuario	+	+	+	-	+	+
Características de contexto	+	-	-	-	-	+
Características de colaboración	-	-	-	-	+	+

Fuente: elaboración propia.

4. Conclusiones y trabajo futuro

MATEO representa una solución original a: a) la necesidad de mecanismos que consideren diversos aspectos adaptativos a la hora de conformar equipos de trabajo, b) la limitada cantidad de tipos de equipo que se pueden conformar de manera automática (agregación de incluyentes y excluyentes) y c) articulación con adaptación de la información, trabajo colaborativo y técnicas de agrupación y optimización. Además, MATEO facilita a la persona conformadora de equipos no gastar tiempo en analizar a los participantes para reconocer en ellos sus fortalezas y habilidades, ya que se encarga de adquirir la información necesaria y utilizarla de manera transparente para el conformador. Finalmente, le da a este una manera objetiva de hacer las agrupaciones, otorgándole la misma oportunidad a todos los participantes de ser seleccionados.

Como trabajo futuro se plantea la implantación de mecanismos de aprendizaje que brinden una retroalimentación a la conformación de equipos a partir de observaciones durante las etapas de desarrollo y cierre del trabajo colaborativo. También se propone la definición de mecanismos de evaluación que permitan medir la calidad de los equipos a partir de los resultados al final de su labor.

Referencias

Alles, M. A. (2008). *Diccionario de comportamientos. Gestión por competencias: cómo descubrir las competencias a través de los comportamientos*. Buenos Aires: Ediciones Granica.

Arias-Báez, M. P. (s. f). Sistema de apoyo al proceso de conformación adaptada de equipos de trabajo en un ambiente colaborativo. Disponible en http://pegasus.javeriana.edu.co/~PI111-05-SistemAdaptaTrabajo/.

Arias-Báez, M. P. y Carrillo-Ramos, A. Forming Collaborative Teams In An Adapted Way. *IADIS ICT, Conference on Society and Human Beings*, 19-26.

Belgharbi, K. y Boufaida, M. (2015). *A Context-Oriented Approach for the Adaptation of Service Compositions*.

Bjorn, P., Bardram, J., Avram, G., Bannon, L., Boden, A., Redmiles, D., de Souza, C. y Wulf, V. (2014). Global Software Development in a CSCW Perspective. *CSCW Companion '14: Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*.

Brusilovsky, P., Grigoriadou, M. y Papanikolaou K. (2007). *Forming Homogeneous, Heterogeneous and Mixed Groups of Learners*. *Proc. of 11th Int. Conference on User Modeling, Workshop on Personalisation in E-Learning Environments at Individual and Group Level*, 33-40.

Christodoulopoulos, C.E. y Papanikolaou K. A. (2007). Investigation of Group Formation Using Low Complexity Algorithms. *Proc. of 11th Int. Conf. on User Modeling*, 57-60.

Craig, M., Horton, D. y Pitt, F. (2010). Forming Reasonably Optimal Groups: (FROG). *Proc. of the 16th ACM Int. Conf. on Supporting group work (GROUP '10)*, 141-150.

Gogoulou, A., Gouli, E., Boas, G., Liakou, E. y Grigoriadou, M. (2014). *Forming Homogeneous, Heterogeneous and Mixed Groups of Learners*.

Kulkarni, D., Ahmed, T. y Tripathi, A. (2012) A Generative Programming Framework for Context-Aware CSCW Applications. *ACM Transactions on Software Engineering and Methodology*, *21*(2), 1-35.

Ounnas, A., Millard, D. E. y Davis, H. C. (2007). A Metrics Framework for Evaluating Group Formation. *Proc. of the 2007 Int. ACM Conference on Supporting Group Work*, 221-224.

Prilla, M. y Renner, B. (2014). Supporting Collaborative Reflection at Work: A Comparative Case Analysis. *Proceedings of the 18th International Conference on Supporting Group Work*.

Weka (2015). Weka 3: Data Mining Software in Java. Disponible en http://www.cs.waikato.ac.nz/ml/weka/.

10
MATEO-EDU: UN CASO DE ESTUDIO SOBRE CONFORMACIÓN DE EQUIPOS DE TRABAJO EN UN CONTEXTO EDUCATIVO

María Paula Arias-Báez, Ángela Carrillo-Ramos

Este capítulo expone MATEO-EDU, implementación de MATEO en el marco de un proyecto de investigación en el que la labor principal es colaborar con el ánimo de cumplir tareas más complejas. El rol de MATEO-EDU consistió en crear equipos de trabajo para una asignatura en la que todos los estudiantes debían participar en un equipo con el objetivo de llevar a cabo una actividad colaborativa diseñada por un profesor. La conformación de equipos de trabajo se realizó a partir de las características relevantes de los estudiantes, del contexto y los requisitos del equipo con el fin de minimizar los problemas propios del trabajo colaborativo (Kulkarni *et al*., 2012): a) falta de coordinación del trabajo, lo cual genera pérdidas de tiempo productivo, b) influencias inapropiadas de las dinámicas de equipo, c) tendencia a "descansar en otros", d) responsabilidades no claras, y e) uso incompleto o inapropiado de la información, lo que genera costos de reunión, entre otros.

1. MATEO-EDU

El escenario escogido está enmarcado dentro del proyecto AYLLU (Arias-Báez *et al*., 2012). Dicha plataforma se encuentra en su versión educativa (AYLLU-EDU), sobre la que se realizó una prueba piloto con estudiantes de la asignatura de Inteligencia Artificial, en un módulo consagrado al tema de Lógica Difusa en la Pontificia Universidad Javeriana y la Universidad Nacional de Medellín. AYLLU ofrece un conjunto de servicios cooperativos; algunos de estos servicios requieren de la conformación de equipos, con lo cual se hace uso de las funcionalidades de MATEO dentro del *framework* AES (Arias-Báez *et al*., 2011) para enriquecer esta acción. De esta manera se

aprovecha la facilidad que brinda la implementación de MATEO para ser utilizado por cualquier sistema que requiera enriquecer su proceso de conformación de equipos en ambientes colaborativos.

Vale la pena recordar que MATEO (Arias-Baéz y Carrillo, 2012a; 2012b) (ver capítulo anterior) está conformado por dos componentes: MADA que corresponde al conjunto de perfiles de donde se extrae información que permite adaptar la conformación de equipos y JOIN que es el proceso de conformación de equipos que utiliza tanto a MADA, como mecanismos de *clustering* y de optimización, con el fin de crear los equipos con los participantes, cuyas características se ajusten más a las necesidades tanto del equipo como del trabajo colaborativo. Los perfiles definidos por MADA corresponden a los perfiles almacenados por el *framework* AES (ver capítulo 7) de donde se adquiere información que permite enriquecer servicios. Adicionalmente, las funcionalidades de agrupación de MATEO (modelo JOIN) son invocadas por AES a través de su componente Mecanismo de Adaptación (figura 1).

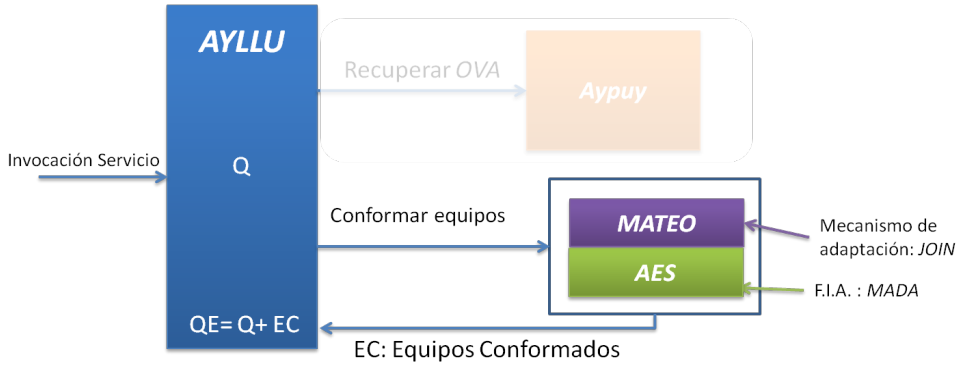


FIGURA 1. ARTICULACIÓN DE MATEO CON AES, ENMARCADO DENTRO DEL PROYECTO AYLLU.

Fuente: Arias-Báez *et al*. (2013).

1.1. *Instalación y configuración para un sistema externo*

El sistema externo que haga uso de las funcionalidades de MATEO debe agregar el archivo .jar de MATEO a su proyecto y utilizar sus métodos. Para articularse con un sistema externo, MATEO no despliega interfaces gráficas. El sistema externo puede hacer uso de los roles predeterminados o puede extenderlos según sus reglas de negocio. La única condición es que las nuevas características o roles deben coincidir con las columnas del archivo .csv, archivo entrada de información de los participantes.

Para que MATEO conforme equipos de trabajo, requiere que el sistema externo instancie un objeto de tipo *Perfilequipo* con el fin de diligenciar los siguientes datos:

```
private String composición;
private int cantEquipos, cantPersonasEquipo;
//(cuando sea homogeneo)
ArrayList<String> varsHomog = new ArrayList<String>();
ArrayList<String> varsColaboracion = new ArrayList<String>();
//Lista por Roles (cuando sea heterogeneo)
ArrayList<Rol> losRoles = new ArrayList<Rol>();
```

Donde *composición* corresponde al tipo de equipo que se va a conformar ("Incluyente-Homogéneo", "Excluyente-Homogéneo", "Incluyente-Heterogéneo", "Excluyente-Heterogéneo"), *cantEquipos* a la cantidad de equipos que se desean conformar, *cantPersonasEquipos* corresponde a la cantidad de personas por equipo, *varsHomog* al conjunto de características junto con su valor, *varsColaboracion* a las características que considere más importantes para que haya colaboración y finalmente, *losRoles* al conjunto de roles requeridos. Para añadir las características que se evaluarán en *Perfilequipo* se debe hacer uso del método:

```
public void setVarsHomog(String varsH) {
    this.varsHomog.add(varsH);
}
```

Donde el parámetro recibido corresponde a la característica y su valor deseado. Por ejemplo, una posible característica podría ser "Ejecución, 1". En el caso que se desee trabajar con roles, se debe crear un objeto de tipo Rol, el cual encapsula los siguientes atributos:

```
private String nombreRol= new String();
private int numPersRol;
private  ArrayList<String> varsRol = new ArrayList<String>();
```

Se diligencia el nombre del rol (*nombreRol*), la cantidad de personas por rol (*numPersRol*). Cada rol tiene una correspondencia con un conjunto de características. Adicionalmente, se debe tener listo un archivo .csv con los valores de las características evaluadas. Tanto el *Perfilequipo* como el archivo de valores deben ser enviados como parámetros al método:

```
private NuevosEquiposAES coreMateo(String resultFileEstudiante, PerfilEquipoAES resultPerfilEquipo) {
```

Su respuesta será encapsulada en un objeto de la clase *NuevosEquipos*, la cual contiene:

```
private int cantEquipos;
private ArrayList<ArrayList<String>> equiposConformados;
private ArrayList<ArrayList<String>> rolesEquipos;
```

La variable *cantEquipos* corresponde a la cantidad de equipos que se conformaron, *equiposConformados* corresponde al conjunto de participantes por equipo y *rolesEquipos* almacenan el rol de cada uno de los participantes por equipo.

1.2. *Caso de estudio*

A continuación, se presenta el flujo de información en AES para la conformación de equipos a través de un ejemplo, utilizando las funcionalidades de MATEO. Inicialmente, el conformador de equipos desea generar dos equipos con tres roles: Secretario (una persona), Presentador (dos personas) y Diseñador (tres personas) (Q). Las fuentes de información (FIA) son los perfiles implementados en el sistema MATEO: perfil de Equipo y Personal. La información de estos perfiles fue obtenida de la siguiente manera:

- *Perfil Equipo*: para conocer la información de los equipos que se desean conformar, la persona que utiliza MATEO (de ahora en adelante llamado *el conformador de equipos*), debe suministrar la siguiente información: a) archivo de los resultados del test de Hermann (Herrmann-Nehdi, 2011) realizado a los participantes (del que se obtienen los comportamientos predominantes de cada participante en cuanto a aspectos como crítico, comunicativo, organización, creatividad, entre otros), b) tipo de equipo que se va a conformar (Incluyentes- Homogéneos/Heterogéneos o Excluyentes-Homogéneos/Heterogéneos), c) cantidad de equipos, y d) roles requeridos y cantidad de personas por rol: cada rol posee un conjunto de características que lo diferencian. El término característica se refiere al comportamiento o competencia, relacionado con el test de Hermann (Herrmann-Nehdi, 2011).
- *Perfil Estudiante*: para conocer los comportamientos y competencias de cada uno de los estudiantes, en esta ocasión se decidió hacer una integración entre el test de Hermann (Herrmann-Nehdi, 2011) con el diccionario de competencias de Martha Alles (Alles, 2008) y el test de Los cinco grandes (Outofservices, 2011). Cabe aclarar que cualquier otro test puede ser tenido en cuenta para conocer los comportamientos y competencias y depende del conformador de equipos de su escogencia. A continuación se presenta una pequeña descripción de cada test.

* *El test de Hermann* (Herrmann-Nehdi, 2011) determina la dominancia cerebral del individuo mediante cuatro colores: a) amarillo: pensamiento imaginativo (actúa, siente, piensa), b) rojo: pensamiento interpersonal (siente, actúa, piensa), c) verde: pensamiento secuencial (actúa, piensa, siente), d) azul: pensamiento analítico (piensa, actúa, siente). Todos los individuos poseen estas cuatro dominancias, solo que en diferente nivel y es lo que marca su personalidad y su manera de interactuar.
* *El diccionario de competencias de Martha Alles* (Alles, 2008) define un conjunto de competencias y lo que vivencialmente significa tener o no dicha competencia. Las competencias escogidas en el modelo MADA fueron tomadas del extenso grupo que define dicho diccionario.
* *El test de Los cinco grandes* (Arias-Báez, 2012) busca responder por qué las personas se comportan de la manera en que lo hacen a través de las siguientes dicotomías: convencionalismo/ apertura a la experiencia, desorganización/ minuciosidad, introversión/ extroversión, antipatía/ afabilidad, estabilidad emocional/ neuroticismo.

Como entrada del sistema AES, se recibe el archivo del test de Hermann con las respuestas de todos los estudiantes. Adicionalmente, se selecciona el perfil de Equipo para enriquecer la consulta (Manejador de FIA). El filtro Estudiante recibe el archivo correspondiente al test de Hermann, deduce la dominancia de color de cada estudiante encuestado y arroja como información de alto nivel sus competencias y comportamientos, que almacena en un documento Excel (Filtro Estudiante) (figura 2).

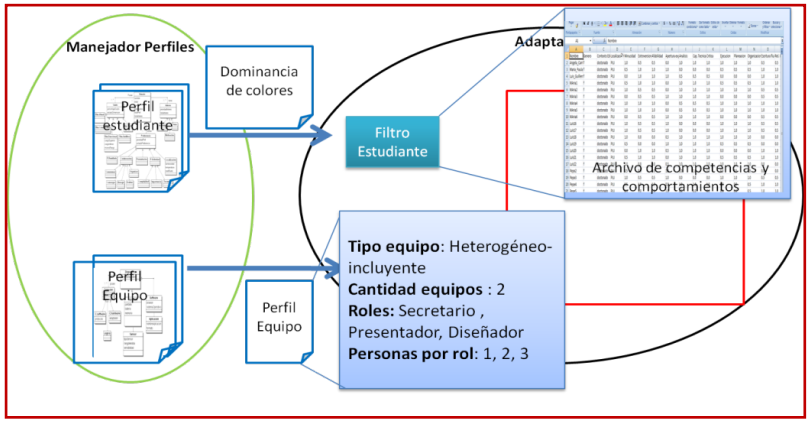


FIGURA 2. RESULTADO DEL FILTRO ESTUDIANTE E INFORMACIÓN DEL PERFIL EQUIPO.

Fuente: elaboración propia.

La información de alta abstracción se obtiene de la integración de los test mencionados anteriormente: se toma cada color y a partir de las características que predominan por cada uno, se asignan competencias y comportamientos que se le relacionen. Esta relación entre colores, comportamientos y competencias es validada por un experto en el tema. Considerando los roles seleccionados en el perfil de Equipo y las características de cada uno de los estudiantes (competencias y comportamientos resultado del proceso anterior), se busca que en cada equipo haya estudiantes que cumplan con alguno de los roles planteados (Mecanismo de Adaptación) (figura 3).

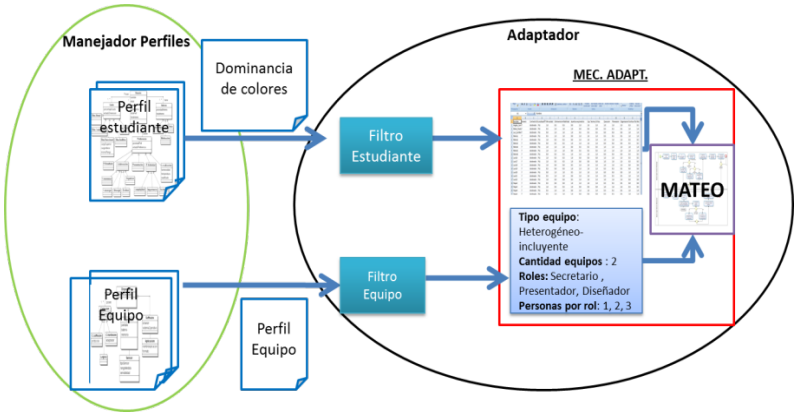


FIGURA 3. ENTRADAS DEL MECANISMO DE ADAPTACIÓN.

Fuente: elaboración propia.

A partir de este punto se inician las funcionalidades de MATEO (proceso JOIN) para conformar los equipos Heterogéneos-Incluyentes necesarios para el caso de estudio aplicado en AYLLU.MATEO; se inicia con el proceso Ejecutar Según Carácter para conocer las personas que cumplen con los roles requeridos (carácter Heterogéneo para el caso de AYLLU-EDU), del modo que se explica a continuación:

1. *Escogencia de características*: cada rol escogido por el conformador de equipos posee un conjunto de características relacionadas. Ejemplo: el conformador necesita tres diseñadores, un secretario y dos presentadores (tabla 1):

TABLA 1. ROLES ESCOGIDOS, CANTIDAD DE PERSONAS POR ROL Y SUS CORRESPONDIENTES CARACTERÍSTICAS

Rol	Cantidad de personas por rol	Características relacionadas con el rol
Secretario	1	Minuciosidad
		Ejecución
		Planeación
		Organización
		Administración
Presentador	2	Extroversión
		Afabilidad
		Escritura fluida
		Relaciones interpersonales
		Expresar ideas
Diseñador	3	Apertura de experiencia
		Conceptualizar
		Innovación
		Creatividad
		Integración

*Importante: los pasos 2 al 6 que se presentarán a continuación se realizan por cada uno de los roles escogidos por el conformador de equipos.

Fuente: elaboración propia.

2. *Preparación de datos (rangos)*: una vez seleccionados los roles y cargado y analizado dentro de MATEO el archivo del test de Hermann aplicado a los participantes, se realiza la preparación de los datos. Cada rol es evaluado de manera separada: del archivo de competencias y comportamientos se seleccionan las columnas correspondientes a las características relacionadas con el rol escogido, el cual será el rango (columnas dentro del archivo de competencias y comportamientos) sobre el que se desea aplicar *clustering*. Ejemplo (figura 4): se inicia con el rol Diseñador: se toman solo sus características relacionadas (columnas naranja), lo que corresponde a los rangos: columnas 4, 13-15, 19 y se aplica *clustering* sobre

estos rangos para todas las instancias[1]. De la misma manera se hace para el rol Secretario (rangos: columnas 2-3, 11-12, 18, columnas rojas) y el rol Presentador (rangos 1, 8-10, 17, columnas verdes).

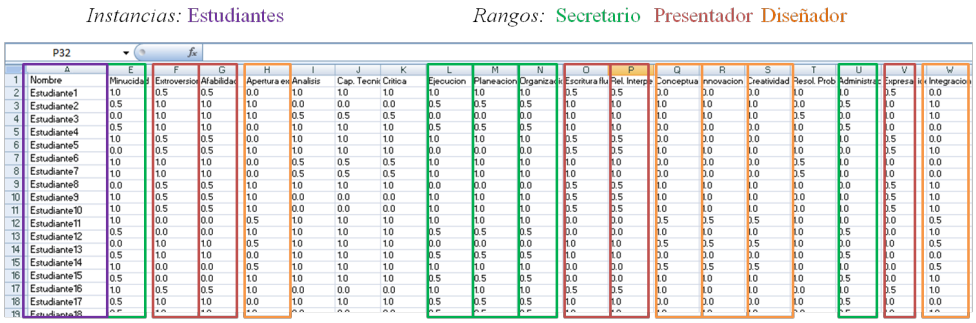


FIGURA 4. ARCHIVO CON LAS COMPETENCIAS Y COMPORTAMIENTOS DE LOS ESTUDIANTES.

Fuente: elaboración propia.

3. *Conformación de clústeres:* una vez estén seleccionados los rangos por cada rol, se procede a conformar las agrupaciones mediante mecanismos de *clustering* de datos proporcionados por el *software* WEKA (Weka, 2015). Como resultado, se identifica un *clúster* para cada instancia y se obtienen los *centroide*s de cada *clúster*, es decir, el promedio de todos los puntos del *clúster* (figura 5).

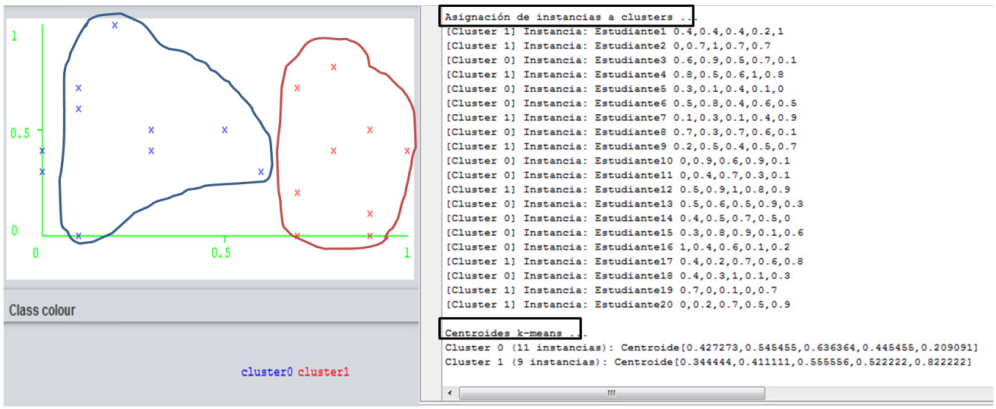


FIGURA 5. PRESENTACIÓN DE *CLÚSTERES* GRÁFICA Y TEXTUALMENTE.

Fuente: elaboración propia.

---

[1] Una instancia corresponde a un participante y sus respectivos valores para cada característica (fila del archivo del test).

4. *Evaluación de centroides:* después de obtener el *centroide* por cada *clúster* y sabiendo el valor deseado de cada característica relacionada con cada rol, se evalúan los valores del *centroide*: en caso de que se manejen roles, todas las características relacionadas con ese rol poseen una calificación alta, lo que significa que se buscan las personas que cuantitativamente tengan en las características los puntajes más altos. A partir de esta calificación, se escoge el *clúster* con la mayor cantidad de valores, seleccionando los más altos dentro de los *centroide*s. Estos valores indican que los participantes de dicho *clúster* poseen alto nivel en la mayoría de las características relacionadas con el rol.

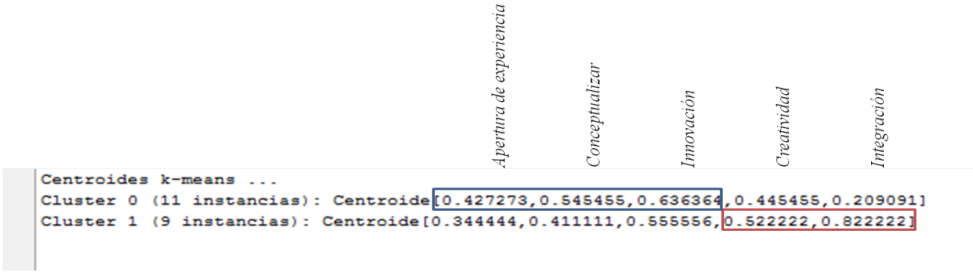


FIGURA 6. VALOR DEL *CENTROIDE* PARA CADA CARACTERÍSTICA DEL ROL DISEÑADOR.

Fuente: elaboración propia.

Ejemplo: la figura 6 muestra los valores del *centroide* correspondiente al *clúster* 0 y al *clúster* 1 al evaluar el rol Diseñador. Allí se compara entre *clústeres* cada característica, cuál es el que contiene la mayor cantidad de valores altos. Para el ejemplo, el *clúster* 0 tiene 3 características con alto valor (a comparación del *clúster* 1) y el *clúster* 1, solo tiene 2.

5. *Selección de clúster según centroide escogido:* se procede a seleccionar el *clúster* que posee mayor cantidad de valores altos dentro de su *centroide*. Ejemplo: según la figura 6, se debería tomar el *clúster* 0 ya que tiene los valores más altos para tres de las características requeridas (Apertura de experiencia, Conceptualizar e Innovación) y una en un valor no tan diferenciador del *clúster* 1 (Creatividad).
6. *Obtención de participantes del clúster*:

TABLA 2. ROLES Y ESTUDIANTES ESCOGIDOS

Rol: Secretario	Rol: Presentador	Rol: Diseñador
Estudiante 1	Estudiante 1	Estudiante 10
Estudiante 13	Estudiante 11	Estudiante 11
Estudiante 16	Estudiante 12	Estudiante 13
Estudiante 17	Estudiante 15	Estudiante 14
Estudiante 18	Estudiante 16	Estudiante 15
Estudiante 2	Estudiante 18	Estudiante 16
Estudiante 20	Estudiante 19	Estudiante 18
Estudiante 4	Estudiante 2	Estudiante 3
Estudiante 5	Estudiante 4	Estudiante 5
Estudiante 8		Estudiante 6
Estudiante 9		Estudiante 8

Fuente: elaboración propia.

Cada *clúster* posee un conjunto de instancias que son fruto del proceso de *clustering*. Las instancias del *clúster* seleccionado en el paso anterior corresponden a los estudiantes que cumplen con la mayor cantidad de características del rol escogido por el conformador de equipos. Ejemplo: como se escogió el *clúster* 0 en la etapa anterior, los participantes escogidos para el rol Diseñador serán los que pertenezcan a dicho *clúster*. Vale la pena recordar que el mismo proceso se hace para los otros dos roles (Presentador y Secretario). La tabla 2 presenta los estudiantes escogidos por rol. Cabe aclarar que un estudiante puede ser apto para uno o más roles. Esta información se le presenta al profesor para que, en caso de sustituciones o cambios dentro del equipo, él pueda tomar mejores decisiones.

7. *Colocación de participantes en equipos:* cada uno de los participantes que pertenecen al *clúster* seleccionado pasa a ser colocado dentro de un equipo de trabajo con el rol que cumple. Se colocan tantos participantes que cumplan con el rol como lo haya indicado el conformador. Ejemplo: se requiere un presentador, por tanto, cada equipo tendrá un presentador; se requieren dos secretarios, entonces, cada equipo tendrá dos secretarios y finalmente, se requieren tres diseñadores, en consecuencia, cada equipo tendrá tres diseñadores (figura 7).

```
========== EQUIPO No. 1 ==========
Estudiante1 - Rol: Secretario
Estudiante11 - Rol: Presentador
Estudiante12 - Rol: Presentador
Estudiante10 - Rol: Diseñador
Estudiante13 - Rol: Diseñador
Estudiante14 - Rol: Diseñador
========== EQUIPO No. 2 ==========
Estudiante16 - Rol: Secretario
Estudiante15 - Rol: Presentador
Estudiante18 - Rol: Presentador
Estudiante3 - Rol: Diseñador
Estudiante5 - Rol: Diseñador
Estudiante6 - Rol: Diseñador
```

FIGURA 7. ASIGNACIÓN DE ESTUDIANTES CON ROL DENTRO DE LOS EQUIPOS.

Fuente: elaboración propia.

Hasta este punto del proceso, se han incluido en los equipos los estudiantes que cumplen alguno de los roles seleccionados. Sin embargo, como el conformador de equipos necesita un equipo "Heterogéneos-Incluyentes", se debe proceder a incluir los demás estudiantes en los equipos ya conformados. El siguiente proceso que realiza MATEO es Ejecutar Según Selección para que, a partir de la respuesta generada en el proceso de Ejecutar Según Carácter se puedan generar los equipos Incluyentes para el caso de estudio. El método de Montecarlo se utiliza solo para casos donde se desee conformar equipos Incluyentes. A continuación, se presentan los pasos que se realizan:

8. *Recibir equipos:* se reciben los dos equipos conformados en el proceso anterior (procedentes del paso 7) y se verifica la cantidad de estudiantes que tiene cada equipo (figura 8).

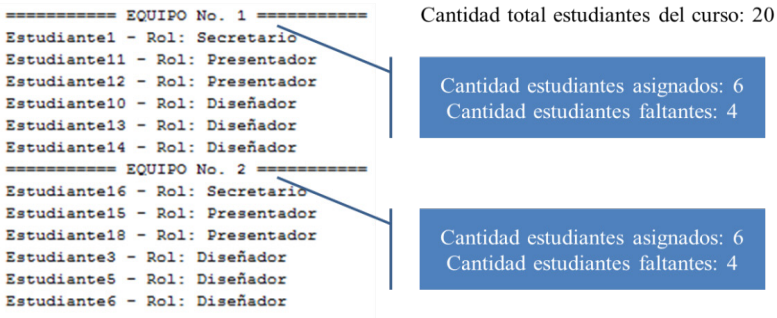


FIGURA 8. VERIFICACIÓN DE CANTIDAD DE ESTUDIANTES ASIGNADOS Y NO ASIGNADOS EN EQUIPOS.

Fuente: elaboración propia.

Los siguientes pasos se hacen para cada equipo requerido, es decir, en este ejemplo se harán dos iteraciones puesto que se desean conformar dos equipos:

9. *Actualizar Distribución:* una distribución es el conjunto de participantes que no han sido colocados dentro de un equipo hasta el momento, es decir, serán todos los estudiantes que no fueron seleccionados durante el paso 7 o paso 14. Ejemplo: recibiendo la respuesta del paso 7, se analiza qué participantes no han sido asignados al Equipo 1 o al Equipo 2 (figura 9). Sobre los estudiantes que no han sido asignados, se aplica *clustering*. En esta ocasión, el rango utilizado corresponde a las características de colaboración definidas en el modelo de MATEO. El resultado final es el valor de cada una de estas características para los estudiantes que no han sido asignados a algún equipo.

Estudiantes asignados	Estudiantes sin ser asignados
======== EQUIPO No. 1 ========	
Estudiante 1 - Rol: Secretario	
Estudiante 11 - Rol: Presentador	Estudiante2
Estudiante 12 - Rol: Presentador	Estudiante4
Estudiante 10 - Rol: Diseñador	Estudiante17
Estudiante 13 - Rol: Diseñador	Estudiante19
Estudiante 14 - Rol: Diseñador	Estudiante20
======== EQUIPO No. 2 ========	Estudiante7
Estudiante 1 - Rol: Secretario	Estudiante8
Estudiante 1 - Rol: Presentador	Estudiante9
Estudiante 1 - Rol: Presentador	
Estudiante 1 - Rol: Diseñador	
Estudiante 1 - Rol: Diseñador	
Estudiante 1 - Rol: Diseñador	

FIGURA 9. CREACIÓN DE LA DISTRIBUCIÓN CON BASE EN LOS ESTUDIANTES NO ASIGNADOS.

Fuente: elaboración propia.

10. *Tomar equipo*: en cada iteración se toma un equipo con participantes faltantes hasta haber evaluado la cantidad de equipos requerida. Ejemplo: se realizan los pasos 9 a 14 primero para el Equipo 1 y luego para el Equipo 2.
11. *Selección aleatoria de puntos (participantes) dentro de la Distribución*: de la distribución actual, se toman aleatoriamente participantes (la cantidad de participantes escogidos depende de la cantidad de personas faltantes en el equipo actual). Ejemplo: en el Equipo 1 hacen falta cuatro estudiantes por ser asignados, por tanto, se toman aleatoriamente cuatro estudiantes de los que no han sido aún asignados. Este paso se realiza iterativamente (10 iteraciones) y cada selección de participantes se almacena dentro de MATEO (figura 10).



FIGURA 10. SELECCIÓN ALEATORIA DE ESTUDIANTES POR ITERACIÓN.

Fuente: elaboración propia.

12. *Evaluación de Función Objetivo*: se toman todas las iteraciones almacenadas del paso 11 y por cada una de ellas se ejecuta el siguiente algoritmo:

Algoritmo 3: Función evaluar Funcion Objetivo (utilizada en Algoritmo 2: Ejecutar según Selección)

Pre condición:

- Lista<Iteracion> iteraciones = conjunto de iteraciones que contienen una cantidad de participantes escogidos aleatoriamente

Post condición:

- iteracion = iteración seleccionada como aquella que cumple con la Función objetivo establecida

1. Lista<entero> valoresFOiteraciones
2. *Para cada* iteración de iteraciones *hacer*
3. Lista<entero> valoresCaractsParticipantes
4. *Para cada* participante de iteracion *hacer*
5. C ← *llamar* sumatoria de Características Colaborativas *con* participante
6. **guardar** C **en** valoresCaractsParticipantes
7. **Fin para**
8. valorFO← *llamar* $\sum_{i=0}^{m} C_i$
9. *guardar* valorFO *en* valoresFOiteraciones
10. **Fin Para**
11. iteracion ← *llamar* seleccionar iteracion *con* mayor valor dentro de valoresFOiteraciones

Ya que el valor deseado para cada característica de colaboración fue definido como alto (es decir, se busca que las personas colaboren en un alto grado dentro del equipo), la función objetivo será maximizar la sumatoria de los valores de las características de colaboración de los estudiantes escogidos aleatoriamente. Ejemplo: por cada iteración se tienen los valores de las características de colaboración de cada estudiante. A partir de dichos valores se realiza la siguiente sumatoria:

$$\sum_{i=0}^{m} C_i$$

Donde i corresponde a la cantidad de características de colaboración, m la cantidad máxima de características de colaboración y Ci la característica de colaboración. Esto se realiza por iteración, por tanto, si se realizaron diez iteraciones, se tendrán diez valores obtenidos a partir de dicha sumatoria.

13. *Seleccionar iteración que cumpla con Función Objetivo:* la iteración que cumple con la Función Objetivo es aquella que busca maximización, es decir, que la sumatoria de las características de los participantes de la iteración sea la máxima entre todas las sumatorias de las iteraciones.
14. *Colocar participantes de iteración escogida en equipo actual:* de la iteración escogida, se toman los participantes escogidos y se colocan en el equipo actual. Posteriormente, se actualiza la distribución (se retorna al paso 9) eliminando los participantes que se acaban de colocar en el equipo actual y se pasa al siguiente equipo (paso 10). Una vez realizado el mismo proceso para el Equipo 2, la respuesta final para este ejemplo sería (figura 10):

```
=========== EQUIPO No. 1 ===========
Estudiante1 - Rol: Secretario
Estudiante11 - Rol: Presentador
Estudiante12 - Rol: Presentador
Estudiante10 - Rol: Diseñador
Estudiante13 - Rol: Diseñador
Estudiante14 - Rol: Diseñador
Estudiante9 - Rol: Participante
Estudiante19 - Rol: Participante
Estudiante2 - Rol: Participante
Estudiante4 - Rol: Participante
=========== EQUIPO No. 2 ===========
Estudiante16 - Rol: Secretario
Estudiante15 - Rol: Presentador
Estudiante18 - Rol: Presentador
Estudiante3 - Rol: Diseñador
Estudiante5 - Rol: Diseñador
Estudiante6 - Rol: Diseñador
Estudiante20 - Rol: Participante
Estudiante8 - Rol: Participante
Estudiante17 - Rol: Participante
Estudiante7 - Rol: Participante
```

FIGURA 11. EQUIPOS CONFORMADOS POR MATEO (HETEROGÉNEOS- INCLUYENTE).

Fuente: elaboración propia.

Entonces, como respuesta del sistema MATEO se presentan los dos equipos conformados, cada uno con los estudiantes que pueden desempeñar un rol de los requeridos, junto con los participantes que maximicen las características de colaboración. Cabe anotar que el rol de los últimos estudiantes ingresados en los equipos es Participante. Para finalizar, la figura 12 muestra cómo se integraría la respuesta generada por MATEO en cuanto a conformación de equipos a AES, enriqueciendo el servicio Conformar Equipos (Q enriquecida).

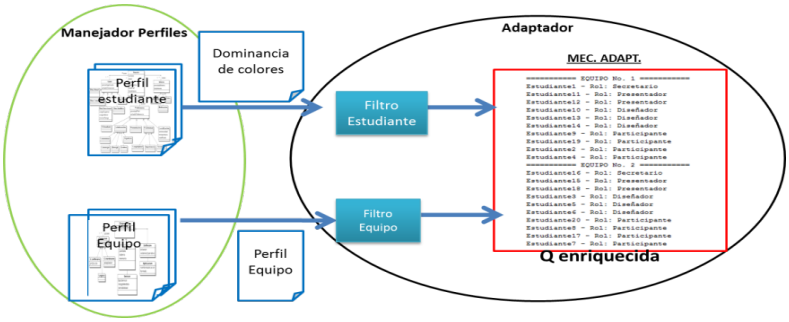


FIG. 12. RESPUESTA QUE ENTREGA MATEO AL *FRAMEWORK* AES.

Fuente: elaboración propia.

2. Pruebas

Esta sección presenta las pruebas realizadas al sistema MATEO en términos de implementación y usabilidad. Para esto se realizaron a) pruebas funcionales basadas en la ejecución de las funcionalidades básicas del sistema planteadas en el modelo de conformación de equipos de manera adaptada; b) pruebas de comparación con otro sistema generador de equipos de trabajo que realiza optimización con el fin de observar qué tan correlacionados están los equipos conformados por MATEO con el equipo que cumplió la función objetivo del sistema generador de equipos; c) evaluación por parte de los usuarios de la prueba piloto de AYLLU-EDU. Estas pruebas serán explicadas a grandes rasgos a continuación (para conocer con más detalle cada una de ellas, remítase a la referencia Arias-Báez, 2012 y al capítulo 13).

2.1. *Pruebas funcionales*

Inicialmente se plantearon cuatro contextos: conformación de equipos a) Homogéneo-Incluyente b) Homogéneo-Excluyente, c) Heterogéneo-Incluyente y d) Heterogéneo-Excluyente. En cada uno de ellos se realizó un cambio de cantidad de participantes totales y de características/roles escogidos. Para cada uno de estos contextos se corrió MATEO iterativamente, obteniendo así los tiempos por ejecución. Posteriormente, se obtuvo el tiempo promedio para cada contexto (puntos) y se procedió a realizar una comparación de la proyección del aumento de tiempo *vs.* la cantidad de características o roles (según el escenario) con los que se obtuviera un tiempo razonable de ejecución. Por tal razón, se utilizó la interpolación de *Lagrange*, en la cual, dado un conjunto de puntos (x0, y0), …, (xk, yk), se genera una ecuación

de grado k que permite observar la proyección de los puntos faltantes. Cabe aclarar que con este método se desea observar la tendencia que toman los datos en cada escenario. Para apoyar dicha tendencia, se evaluó chi^2 para conocer si hay asociación entre las variables estudiadas y cuál es su grado: cuanto más grande es el valor de chi^2, más dependientes son las variables y cuanto más se aproxime a cero, más independientes.

A continuación se presenta el análisis realizado a los cuatro escenarios y sus tres respectivos casos, los cuales serán graficados para observar su comportamiento. El eje *x* de la gráfica corresponde a la cantidad de características o cantidad de roles según el tipo de equipo, mientras que el eje *y* corresponde al tiempo de ejecución de MATEO hasta arrojar un resultado (en milisegundos). El color de cada función graficada corresponde a la cantidad de participantes P tenidos en cuenta, donde azul corresponde a P12, rojo a P50 y amarillo a P100. Por cuestiones de espacio solo se va a mencionar uno de los escenarios. El detalle de los otros se puede encontrar en la referencia (Arias-Báez, 2012).

En el caso de la conformación de equipos Heterogéneo-Incluyente:

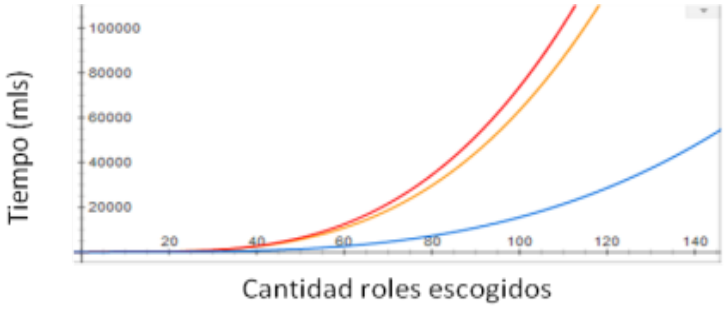


FIGURA 13. GRÁFICA RESULTADO DEL ANÁLISIS DE X3.

Fuente: elaboración propia.

Analizando *f(P12), f(P50)* y *f(P100)* en el dominio (figura 13), se observa que las tres funciones tienden a crecer: a medida que aumenta la cantidad de roles escogidos R, aumenta el tiempo en que se obtiene una respuesta. Aunque las tres funciones poseen valores muy cercanos entre ellas, aproximadamente en la porción de dominio (0,20) sus valores empiezan a diferenciarse, sobre todo para $f(P_{12})$, la cual obtiene valores menores de tiempo que $f(P_{50})$ y $f(P_{100})$ a medida que aumentan los roles escogidos. En cuanto a chi^2, se observa que, aunque es lejano de cero, para el caso de 12 y 100 estudiantes, las variables roles y tiempo son menos dependientes que para el caso de 50 estudiantes. Entonces, para conformar equipos

Heterogéneos-Incluyentes, se puede deducir que para obtener tiempos razonables de funcionamiento (menos de un minuto y medio para responder) se recomienda: a) escoger menos de 176 características de evaluación para una cantidad menor o igual a 12 personas, b) escoger menos de 109 características para una cantidad menor o igual a 50 personas y c) escoger menos de 114 características para una cantidad menor o igual a 100 personas.

2.2. *Pruebas de comparación*

A través de este tipo de pruebas, se compara MATEO con otro sistema generador de equipos de trabajo que realiza optimización con el fin de observar qué tan correlacionados están los equipos conformados por MATEO con el equipo que cumplió la Función Objetivo del sistema generador de equipos. Para observar la correlación entre los equipos conformados por MATEO y los conformados por otro sistema (de ahora en adelante, el sistema generador), que obtiene el equipo más equilibrado, se realizaron las siguientes pruebas. Dicha correlación se entenderá como qué tan lejos o cerca se encuentran los equipos conformados por MATEO con respecto al equipo equilibrado resultado del sistema generador. El sistema generador conforma equipos de trabajo de manera aleatoria, obteniendo una muestra de los posibles equipos que pueden resultar (ver figura 14 con un ejemplo de dos equipos conformados).

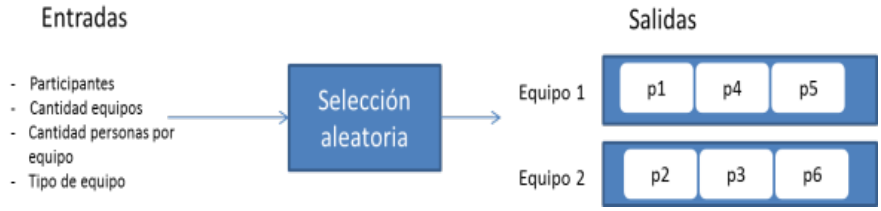


FIGURA 14. ENTRADAS Y SALIDAS DEL SISTEMA GENERADOR.

Fuente: elaboración propia.

Para cada uno de los equipos seleccionados se evalúa el valor de cada característica por estudiante (v en el ejemplo de la figura 15) y por cada una de ellas se obtiene un punto. En figura 15 se muestra un ejemplo del procedimiento con el Equipo 1. Esto se realiza por cada equipo conformado.

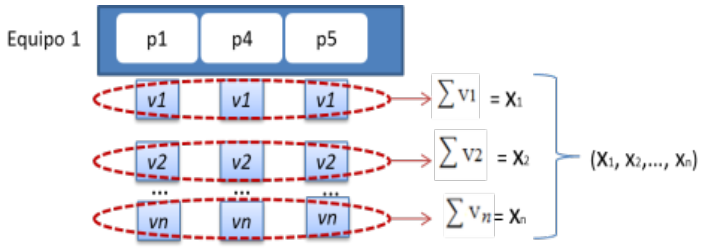


FIGURA 15. EVALUACIÓN DEL VALOR DE CADA CARACTERÍSTICA POR ESTUDIANTE.

Fuente: Arias-Báez *et al*. (2013).

Sobre el punto de cada equipo se evalúa una Función Objetivo (obtener el equipo más equilibrado) y se obtiene el punto óptimo. El término *óptimo* será utilizado en este caso como el punto que representa el equipo más equilibrado, donde el equilibrio se estableció como la mínima diferencia en el porcentaje de las características con respecto al valor del punto óptimo en cada una de ellas para que se presenten en similar grado de aporte por parte de todos los que conforman el equipo (Algoritmo 4).

Algoritmo 4: Evaluación Función Objetivo

Pre condición:

- ListaPuntos = lista que contiene el punto de cada equipo
- n = cantidad de características evaluadas dentro de un punto

Post condición:

- PuntoOptimo = punto del equipo que es similar al Punto deseado

1. puntoDeseado ← $\frac{100\%}{n}$
2. ***Para cada*** punto de ListaPuntos ***hacer***
3. Lista<entero> porcentajePunto
4. ***Para cada*** característica de punto ***hacer***
5. porcentajeCaracteristicaEquipo ← $\frac{(X_i * 100)}{(X_1, X_2 ... X_n)}$
6. Guardar porcentajeCaracteristicaEquipo en porcentajePunto
7. ***Fin Para***
8. ***Fin Para***
9. ***Para cada*** porcentajePuntos de ListaPuntos ***hacer***
10. puntoOptimo ← ***llamar*** evaluar porcentajeCaracteristicaEquipos **con** puntoDeseado
11. ***Fin Para***

Paralelamente, MATEO conforma los equipos de trabajo y de igual manera, para cada uno de los equipos obtenidos, se evalúa el valor de cada característica establecida (figura 16) y por cada una de ellas se obtiene un punto. Estos puntos son los que se van a comparar con el mejor punto del sistema generador.

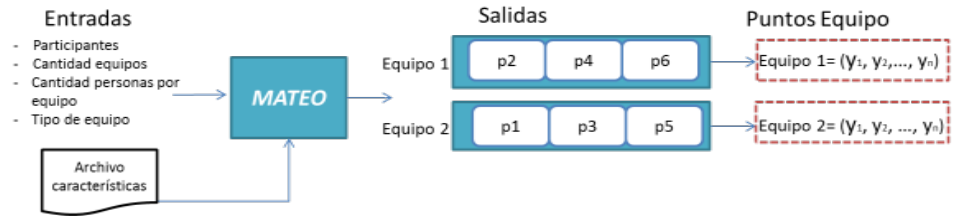


FIGURA 16. ENTRADAS, SALIDAS Y PUNTOS DE MATEO.

Fuente: elaboración propia.

Para evaluar la correlación entre el mejor punto del sistema generador con cada uno de los puntos de MATEO, se procede a evaluar la distancia euclidiana entre cada pareja de puntos, de la siguiente manera:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$

En las pruebas realizadas se observó que, para los cuatro contextos analizados, la distancia de los puntos de los equipos conformados por MATEO, en promedio, tiene un valor máximo cercano de 3 puntos de distancia (dentro de un rango de 0 a 6, como se observa en el eje *y* de las gráficas). Esto quiere decir que los equipos conformados por MATEO se pueden considerar equilibrados con respecto al punto óptimo ya que no discrepan en gran medida con el punto obtenido por el sistema generador, que correspondía al equipo más equilibrado. La figura 17 presenta un resumen de las gráficas mencionadas.

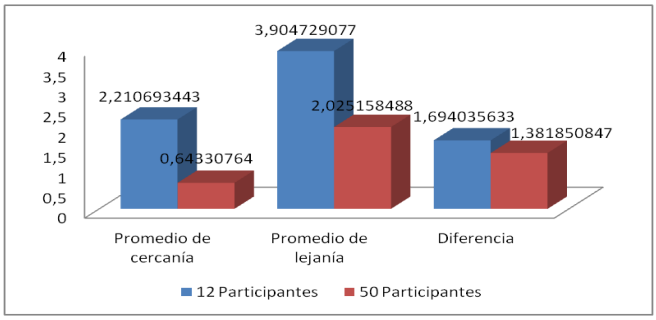


FIGURA 17. PROMEDIOS DE CERCANÍA Y LEJANÍA CON RESPECTO AL PUNTO ÓPTIMO.

Fuente: Arias-Báez *et al*. (2013).

El promedio de cercanía oscila entre 0,64 y 2,21 puntos de distancia con respecto al punto óptimo, mientras que el promedio de lejanía está entre 2,02 y 3,9 puntos de distancia con respecto al punto óptimo. Si se observa la diferencia entre P_{12} y P_{50}

se puede deducir que a mayor cantidad de participantes más equilibrio existe entre los equipos de trabajo conformados por MATEO. Adicionalmente, la distancia entre los puntos más cercanos y lejanos en cada contexto es menor para P_{50} que para P_{12}. Además del hecho de conformar equipos similares al equipo óptimo, MATEO considera características adicionales a las de colaboración que le permiten la asignación de roles con base en competencias y comportamientos de los participantes. A partir de este segundo criterio de comparación (asignación de roles) se observa la importancia de la adaptación, ya que se evidencia el enriquecimiento del proceso que desarrollan ambos sistemas (la conformación de equipos de trabajo): el sistema adaptado MATEO considera la correcta asignación de los roles requeridos mediante el procesamiento de los perfiles de los involucrados, mientras que el sistema generador solo asigna de manera aleatoria los roles a sus participantes, desconociendo si pueden o no desempeñarlos correctamente.

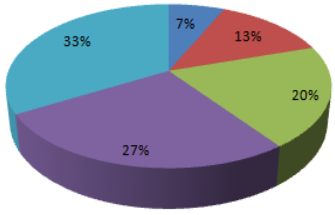
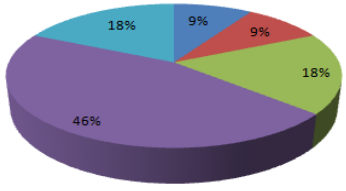
2.3. *Preexperimento*

Se realizó el estudio de un caso con una sola medición, ya que no se requirió de una referencia previa de cuál era el nivel que tenía el equipo de estudio (antes del estímulo) en las variables que se evaluaron. Se tomó este tipo de experimento como un primer acercamiento al problema de investigación en la realidad. Para esto, se realizó una encuesta a los estudiantes que participaron en el piloto AYLLU, con los cuales se conformaron equipos de trabajo. Adicionalmente, se realizó la misma encuesta al profesor de la clase para observar desde su punto de vista la dinámica de equipos a partir de su conformación. Al tener las respuestas de los involucrados, estas se triangularon para observar desde las dos perspectivas el mismo proceso. La encuesta fue diligenciada por once de los doce estudiantes del curso Inteligencia Artificial, en el módulo de Lógica Difusa en la Pontificia Universidad Javeriana, en el que se realizó una prueba de concepto de AYLLU.

En dicha prueba se requirió el proceso de conformar equipos dentro de la estrategia cooperativa *Jigsaw* en dos oportunidades: a) en la conformación de equipos base: primera reunión de estudiantes, donde se conformaron tres equipos de cuatro personas. A un estudiante de cada equipo se le asignó el rol de coordinador (aquel que coincidiera mayormente con las características de coordinador definidas en MATEO); b) en la conformación de equipos temáticos: reunión de estudiantes por subtema, donde se conformaron cuatro equipos de tres personas. En cada equipo, a un estudiante se le asignó el rol de secretario (aquel que coincidiera mayormente con las características de secretario definidas en MATEO). La figura 18 presenta algunos de los resultados a cada una de las preguntas realizadas al profesor y a sus

estudiantes dentro de la encuesta para evaluar el proceso de conformación de equipos (utilizando MATEO dentro del piloto). Esta misma encuesta fue diligenciada por los estudiantes del curso Inteligencia Artificial, en el módulo de Lógica Difusa en la Universidad Nacional de Colombia - Sede Medellín, grupo en el que también se realizó la prueba conceptual de AYLLU. Las respuestas de los estudiantes coinciden altamente con las dadas por los estudiantes de la Pontificia Universidad Javeriana.

En resumen, y para ambos escenarios, MATEO tuvo en cuenta los comportamientos y competencias de los estudiantes, los cuales fortalecieron el trabajo del equipo, ya que, mediante la asignación de los roles de acuerdo con el perfil de cada uno de los estudiantes, los encuestados consideraron que las decisiones fueron tomadas en equipo, todos los estudiantes participaron y su trabajo fue equilibrado y equitativo al interior de los equipos, se definieron y delimitaron claramente las responsabilidades, hubo un aprendizaje mayormente en equipo para el mejor entendimiento de un tema, se obtuvo nueva información tanto de manera individual como conjunta. Con esto se puede afirmar que se redujo el impacto de problemas tales como a) falta de coordinación del trabajo, b) influencias inapropiadas de las dinámicas de equipo, c) tendencia a "descansar en otros", d) responsabilidades no claras, e) uso incompleto o inapropiado de la información.

12. ¿El rol que le fue asignado concuerda con su perfil? 1 2 3 4 5  Escala: (1) poco, (5) bastante	Se observa que la mayoría de los estudiantes (33%) considera que el rol que le fue asignado concuerda bastante con su perfil. A la misma pregunta, el profesor respondió 4, lo que significa que el profesor considera que, de manera general, los roles asignados concuerdan con los perfiles de los estudiantes. De esta manera se observa que efectivamente se asignaron los roles de acuerdo con el perfil de los participantes.
13. ¿Se sintió a gusto con el rol que le fue asignado? 1 2 3 4 5  Escala: (1) poco a gusto, (5) muy a gusto	Se observa que la mayoría de los estudiantes (46 %) se sintió a gusto con el rol asignado. A la misma pregunta, el profesor manifiesta que quedó satisfecho con la asignación de los roles Secretario y Coordinador en la mayoría de los equipos. Por tanto, se puede deducir que tanto estudiantes como profesor quedaron a gusto con los roles asignados.

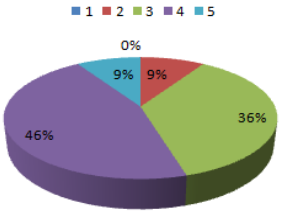
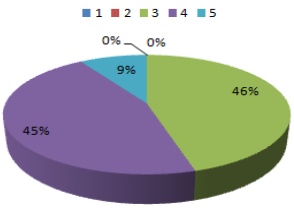
14. ¿Considera que fueron bien asignados los roles a sus compañeros? (de acuerdo a las competencias y comportamientos de ellos)  Escala: (1) mal asignado, (5) bien asignado	Se observa que la mayoría de los estudiantes (46 %) considera que los roles asignados a sus compañeros se hicieron de manera correcta. A la misma pregunta, el profesor respondió 4, lo que significa que el profesor considera que, de manera general, los roles asignados fueron bien asignados. De esta manera se observa que efectivamente se asignaron los roles de acuerdo con las competencias y comportamientos de los estudiantes.
15. ¿Considera que sus compañeros se sintieron a gusto con los roles asignados?  Escala: (1) poco a gusto, (5) muy a gusto	Se observa que la mayoría de los estudiantes (46 %) considera que sus compañeros se sintieron a gusto con los roles asignados. El profesor manifiesta estar satisfecho con la asignación de los roles Secretario y Coordinador en la mayoría de los equipos. Por tanto, se puede deducir que tanto estudiantes como profesor quedaron a gusto con los roles asignados.

FIGURA 18. RESULTADOS DE ENCUESTA DE PERCEPCIÓN POR PARTE DE USUARIOS FINALES DEL PROTOTIPO DE MATEO-EDU.

Fuente: elaboración propia.

3. Conclusiones

Con el fin de validar MATEO (Arias-Báez, 2012) y su posibilidad de conformar los tipos de equipos planteados se realizó MATEO-EDU. Dentro de las tecnologías consideradas en el prototipo se encuentran procedimientos de *clustering* de datos y la implementación del método de optimización Montecarlo, los cuales se llevaron a cabo mediante la revisión de la base del conocimiento (bibliografía y literatura en general sobre estos métodos) y asesoría por parte de expertos en inteligencia de negocios y métodos numéricos, hasta lograr el modelo que hoy se presenta. MATEO-EDU se implementó en dos entornos diferentes: el primero fue implementado como aplicación *standalone* y, posteriormente, fue articulado con un sistema externo (capítulo 7) que se proporciona como librería.

La validación de MATEO-EDU se realizó mediante tres tipos de pruebas: a) funcionales: ejecución de las funcionalidades del sistema planteadas en el modelo de conformación de equipos de manera adaptada, b) de comparación: evaluación con otro

sistema para ver qué tan equilibrados se encuentran los equipos de MATEO con respecto al óptimo obtenido a través del otro sistema y c) preexperimento: aplicación de MATEO en el proyecto AYLLU (capítulo 6). Para realizar esta validación se obtuvo asesoría por parte de expertos de la carrera de Educación y por expertos sobre métodos formales. De estas pruebas se obtuvo que MATEO permite tener en cuenta una gran cantidad de características o roles para hacer la selección de participantes sin comprometer negativamente el tiempo de respuesta del sistema, además de generar equipos con características necesarias para fomentar la colaboración entre los participantes.

Referencias

Alles, M. A. (2008) *Diccionario de comportamientos. Gestión por competencias: cómo descubrir las competencias a través de los comportamientos*. Buenos Aires: Granica.

Arias-Báez, M. P. y Carrillo, A. (2012). Forming Collaborative Teams in an Adapted Way. *Society and Human Beings (ICT 2012)*. Lisboa.

Arias-Báez, M., Torres-Ribero, L., Carrillo, A., Pomares, A. y González, E. (2012). AYLLU: Architecture for Collaboration in a Distributed Environment. *Proc. Collaborative Technologies (CT 2012)*. Lisboa.

Arias-Báez, M., Torres-Ribero, L., Carrillo-Ramos, A., Pomares, A., Garzón Ruiz, J. y González, E. (2011). AES: A Generic Framework for Adapting Information in Multidimensional Contexts. *CLEI2011*. Ecuador.

Arias-Báez, M.P. (2012). *Sistema de Apoyo al Proceso de Conformación Adaptada de Equipos de Trabajo en un Ambiente Colaborativo*. (Tesis de Maestría). Bogotá: Pontificia Universidad Javeriana.

Arias-Báez, M. P., Pavlich-Mariscal, J. A. y Carrillo-Ramos, Á. (2013). Forming Adapted Teams Oriented to Collaboration: Detailed Design and Case Study. *Rev. Avances en Sistemas Informática*, *10*(1), 87-94.

Herrmann-Nehdi, A. (2011). The Best of Both Worlds - Making Blended Learning Really Work by Engaging the Whole Brain. *Herrmann International*. Disponible en http://www.hbdi.com/Resources/whitepapers/index.cfm?viewall=1.

Kulkarni, D., Ahmed, T. y Tripathi, A. (2012). A Generative Programming Framework for Context-Aware CSCW Applications. *ACM Transactions on Software Engineering and Methodology*, *21*(11), 1-35.

Outofservices (2011). *Los 5 grandes. Modelo de personalidad*. Disponible en http://es.outofservice.com/bigfive/. Última revisión: septiembre 2011.

Weka. (2015). Weka 3: Data Mining Software in Java. Disponible en http://www.cs.waikato.ac.nz/ml/weka/.

11
AYPUY: ADMINISTRADOR DE RECURSOS EN AMBIENTES COLABORATIVOS

Alexandra Pomares Quimbaya, Luis Guillermo Torres-Ribero, Enrique González

Los sistemas de apoyo a la colaboración permiten la gestión de múltiples instancias de procesos colaborativos en cada uno de los cuales pueden estar involucrados múltiples participantes. De acuerdo con el rol desempeñado por cada participante, este puede proveer o requerir diversos recursos físicos o lógicos necesarios para desarrollar el proceso. Para garantizar la ejecución adecuada de un proceso colaborativo es necesario entonces gestionar los recursos involucrados, garantizando los requerimientos de confidencialidad, distribución y replicación definidos por cada proceso y por cada una de sus instancias. Las propuestas actualmente existentes para la gestión de procesos colaborativos presentan ciertas limitaciones, principalmente asociadas a su incapacidad de crear políticas específicas para la gestión de recursos teniendo en cuenta las necesidades de cada instancia de un proceso colaborativo. Para afrontar estas limitaciones, este capítulo presenta AYPUY, un sistema gestor de recursos creado específicamente para ambientes colaborativos. AYPUY se basa en el concepto de espacios de trabajo y en agentes inteligentes para proveer las funcionalidades de confidencialidad, distribución y replicación de recursos ajustados a cada proceso colaborativo que se ejecute dentro de un sistema de colaboración.

1. Introducción

Un proceso colaborativo corresponde a una labor en la que múltiples entidades (tales como personas, empresas, comunidades) llevan a cabo, de manera conjunta, la producción de un bien o servicio (Bannon y Schmidt, 1991). Aunque existen diferentes aproximaciones de procesos colaborativos, unos más estructurados que otros (Poltrock y Handel, 2009), las fases que los componen pueden resumirse en la

formulación del objetivo, el establecimiento de las entidades colaboradoras, la creación del espacio adecuado de colaboración, la previsión del tiempo requerido por las entidades y la utilización del resultado generado (Fuks *et al*., 2008). Los sistemas de apoyo a la colaboración por computador (CSCW) (Rama y Bishop, 2006) favorecen el desarrollo de estas fases. Estos sistemas proveen herramientas para garantizar la comunicación síncrona (ej., chat, videoconferencias) y asíncrona (ej., foros, wikis, tableros virtuales, repositorios) entre entidades (Demetriadis y Karakostas, 2008) y para generar y controlar los flujos de trabajo que dirigen el proceso colaborativo. Si bien estas herramientas han favorecido la comunicación constante, el aporte individual y la facilidad de compartir información entre entidades distribuidas físicamente, presentan diversas limitaciones frente al proceso colaborativo (Demetriadis y Karakostas, 2008). Una de estas limitaciones es la manera estática en la que se gestionan los recursos, es decir, su distribución, su replicación y el establecimiento de permisos de acceso, sin considerar la diversidad de requisitos de los procesos colaborativos frente a seguridad, métodos de acceso y tipos de recursos requeridos.

Frente a esta limitación se propone AYPUY, un sistema gestor de recursos dirigido a ambientes colaborativos. El principio de AYPUY es garantizar la gestión de los recursos respetando los requerimientos de seguridad, acceso y distribución de los diversos procesos colaborativos que pueden ser ejecutados en un sistema para el apoyo a la colaboración. AYPUY fue construido como un servicio dentro de AYLLU (Cantor *et al*., 2006), un sistema basado en agentes de apoyo a la colaboración entre personas; sin embargo, AYPUY puede ser utilizado por otros sistemas que requieran administrar recursos.

La estructura del presente capítulo es la siguiente: la sección 2 presenta el análisis de algunos sistemas existentes para gestionar recursos en ambientes colaborativos. Posteriormente, en la sección 3 se establecen los elementos más importantes del modelo AYPUY y su arquitectura. Luego, en la sección 4 se describe un escenario en el que se prueba e implementa AYPUY en un ambiente de colaboración educativo. Finalmente, en la sección 5 se concluye y se mencionan los retos en la gestión de recursos en ambientes colaborativos.

2. Trabajos relacionados

En esta sección se presenta el análisis de las principales propuestas existentes para la gestión de recursos en ambientes colaborativos. Para facilitar la comprensión, estas propuestas se dividieron en dos grupos: gestores de recursos autónomos y gestores de recursos que dependen de los sistemas que los contienen.

2.1. *Gestores de recursos autónomos*

Estos trabajos se caracterizan por permitir compartir recursos dentro de un proceso colaborativo. El análisis de cada uno de ellos fue realizado teniendo en cuenta las siguientes características: a) flexibilidad: el sistema administra diferentes tipos de recursos en diversos dominios de aplicación; b) escalabilidad: el sistema administra el almacenamiento y la disponibilidad de los recursos en un ambiente distribuido; y, c) confidencialidad: el sistema utiliza mecanismos de autenticación y autorización de usuarios (roles) para manejar los permisos de acceso a los recursos. Sinclair *et al.* (2014) proponen un administrador de recursos compartidos que permite que sus usuarios puedan acceder a dichos recursos desde diferentes posiciones geográficas con el fin de facilitar su acceso remoto al aula virtual.

Este administrador es flexible ya que el tipo de recurso que maneja puede ser diverso. El sistema está estructurado por *middlewares* con el fin de facilitar la colaboración entre los usuarios, y su modelo lógico está basado en la utilización de agentes, cada uno especializado en una tarea. Sin embargo, dichos agentes no son móviles, no se observan mecanismos de autenticación o autorización para manejar la confidencialidad de los recursos compartidos y su concepto de colaboración se limita a proporcionar acceso a los recursos en cualquier lugar o momento del estudiante y no se observan espacios de trabajo que amplíen dicha colaboración.

Otro administrador de recursos compartidos, denominado ROADS (Yang *et al*., 2007), se basa en el paradigma de los sistemas federados, en los que se comparten recursos, los cuales están en diferentes dominios y por tal razón requieren una "cabeza" que los maneje eficientemente. ROADS es un *middleware* que proporciona la localización de los recursos del sistema a través de servidores que forman una jerarquía de federados. Dicha jerarquía le permite a cada nodo determinar su alcance (resumen de sus recursos y de los nodos cercanos) y la manera de compartir sus recursos. Los datos de los recursos se propagan a lo largo de dicha jerarquía, prefiriendo ser almacenados en proveedores de servidores del mismo dominio con el ánimo de mantener la asociación entre ellos y su configuración. Cuando se elige un proveedor de otro dominio se lleva a cabo una negociación para llegar a acuerdos respecto al tipo de recursos y su propagación. No obstante, ROADS no presenta ningún modelo de confidencialidad que valide la autorización y autenticación de usuarios en dichos recursos, con sus permisos respectivos. Adicionalmente, limita su concepto de colaboración a proporcionar acceso público a todos los recursos con los que cuenta el sistema.

Por otro lado, MOCHA (acrónimo de *Middleware Based on a Code Shipping Architecture*) (Rodríguez-Martínez y Roussopoulos, 2010) es un *middleware* autoextensible para

base de datos, diseñado con el fin de interconectar gran cantidad de fuentes de datos distribuidos. MOCHA es flexible, ya que puede almacenar varios tipos de datos como imágenes, audio, texto, objetos geométricos y programas. La seguridad del sistema se enfoca en la implantación de mecanismos que garanticen que el código que se ejecute en sus máquinas no sea peligroso; sin embargo, no especifica la utilización de permisos para acceder a dichos datos con el objetivo de mantener la confidencialidad de estos y su concepto de colaboración se limita a compartir los recursos almacenados sin proporcionar un espacio de trabajo en el que se pueda acceder a los recursos relacionados con el proceso colaborativo que llevan a cabo sus participantes.

Finalmente, Machulak *et al.* (2010) proponen una aproximación para gestionar recursos web mediante la participación del usuario de tal modo que sea él quien se encargue de asignar permisos de acceso a sus recursos alojados en las aplicaciones web. Los usuarios comparten datos de manera más selectiva a través de un administrador de autorización centralizado que toma las decisiones de acceso basado en las instrucciones de uso (basado en el protocolo UMA - *User-Managed Access*). Los recursos protegidos se encuentran en diferentes *hosts*, los cuales almacenan y manejan dichos recursos y aseguran el cumplimiento de las decisiones de control de acceso expedidas por el administrador de autorización. Para acceder a los recursos del *host* existe un *Requester* autorizado que podrá hacer uso de ellos. Esta aproximación se enfoca principalmente en la confidencialidad de los datos; no obstante, no es claro si es capaz de manejar diferentes tipos de recursos ni tampoco hay información sobre si se puede utilizar en dominios de aplicación distintos. En la tabla 1 se presenta una comparación de los trabajos mencionados sobre manejadores de recursos. Se utilizaron como notaciones: '+' si el trabajo considera este aspecto, '-' si el trabajo no considera este aspecto.

TABLA 1. COMPARACIÓN DE TRABAJOS RELACIONADOS

	Sinclair	ROADS	MOCHA	Machulak
Flexibilidad	-	+	-	+
Escalabilidad	+	+	+	+
Confidencialidad	-	-	+	-

Fuente: elaboración propia.

Como se observa en la tabla 1, la escalabilidad es un aspecto clave para un sistema manejador de recursos, ya que todos los recursos deben ser gestionados sin importar su localización. Así mismo, es fundamental que el alcance del sistema pueda ir aumentando y que sea flexible en la gestión de diferentes tipos de recursos para

múltiples contextos (recursos educativos, financieros, historias clínicas, etc.). Sin embargo, se observan vacíos en cuanto a la confidencialidad de dichos recursos, es decir, no es claro el establecimiento de roles y permisos que cada uno de los sistemas analizados utiliza, lo que puede comprometer el manejo de los recursos dentro de los espacios compartidos y privados de los participantes del ambiente colaborativo.

2.2. *Gestores de recursos dependientes*

Herramientas para la gestión colaborativa de proyectos como GForge (GForge, 2015) poseen manejadores de documentos que gestionan sus archivos y carpetas (donde gestionar se relaciona con los procesos de organización, manejo de versiones y permisos de acceso) e incluyen herramientas para la creación y acceso autorizado a repositorios de código fuente (*Source Code Management* [SCM]). Adicionalmente, cuenta con espacios colaborativos representados como *Actividades* en las cuales un grupo de participantes puede cargar y descargar archivos (ej., documentos, presentaciones, etc.) asignados para cumplir con dicha actividad. Otro ejemplo de este tipo de herramientas es e-Task (HydraPSA, 2015), el cual, además de contar con funcionalidades para la ejecución de proyectos (manejo de reportes, de tiempo, tareas asignadas, etc.), asigna diferentes tipos de recursos a servicios o proyectos creados. Utiliza computación en la nube con el fin de generar y habilitar ambientes compartidos de trabajo donde los documentos pueden ser manejados de manera local o distribuida, gestionando su acceso y manejo de versiones.

Igualmente, Jazz (Jazz, 2105) es una plataforma para el desarrollo de *software* de colaboración con el fin de apoyar el ciclo de vida del *software* considerando la distribución geográfica de sus participantes y, por tanto, también los recursos que utilizan. La arquitectura de Jazz permite que haya concurrencia, lo que requiere de un manejador de datos y de recursos sólido y confiable; además, para que se pueda integrar con otros productos hechos en Jazz mediante el uso de *middleware,* por ejemplo, Tomcat.

Como se puede observar, si bien los trabajos existentes para compartir recursos han resuelto problemas asociados a la escalabilidad y en ciertos casos a la confidencialidad y flexibilidad, presentan falencias principalmente asociadas al establecimiento de políticas de seguridad, distribución y acceso generales y no a nivel de instancia del proceso colaborativo. Por esta razón, surgió la necesidad de crear AYPUY, un sistema manejador de recursos que permite soportar diferentes tipos de procesos colaborativos mediante el establecimiento de políticas específicas para cada instancia del proceso desde el punto de vista de seguridad, distribución y acceso a los recursos de información.

3. Arquitectura AYPUY

Esta sección describe los conceptos base de la arquitectura de AYPUY. Inicialmente, se describen los elementos del ambiente en el que se lleva a cabo el proceso colaborativo. Posteriormente, se realiza la caracterización de recursos y se introduce el concepto de espacios de trabajo en AYPUY.

3.1. *Ambiente colaborativo*

Los procesos colaborativos contemplados por AYPUY se desarrollan en ambientes donde las entidades que requieren los recursos se encuentran distribuidas en nodos de cómputo posiblemente heterogéneos. Cada nodo tiene la capacidad de almacenar recursos y de permitir el acceso a ellos. Los nodos pueden ser móviles o no; en el primer caso, la localización física del nodo puede variar con el tiempo. Adicionalmente, la pertenencia de un nodo al ambiente de colaboración puede ser catalogada como no volátil cuando la permanencia en el ambiente es estable, o como volátil cuando la permanencia en un mismo entorno no es permanente o es altamente intermitente.

Teniendo en cuenta que la intención de AYPUY no es gestionar la conexión física de los nodos al sistema sino la gestión de recursos en los nodos, los servicios para tener acceso a los nodos deben ser provistos por un sistema especializado. Para el caso de la implementación actual de AYPUY, los servicios son provistos por BESA (González, 2003), un *framework* de desarrollo de aplicaciones basadas en agentes, que en el caso de AYPUY le provee servicios tales como la comunicación, el manejo del ciclo de vida y la gestión de recursos relacionados con los agentes.

A. Recursos

En AYPUY un recurso es considerado como un medio para cumplir un objetivo. Un recurso puede ser de tipo físico (ej., chequera) o lógico (ej., presentación). Independientemente de su naturaleza, el recurso está compuesto de atributos y operadores. Los recursos son descritos a través de un conjunto de atributos $A = \{L, D, F\}$, donde L determina las características lógicas generales del recurso, D establece las características lógicas asociadas a su dominio y F describe las características de almacenamiento físico de dicho recurso. Adicionalmente, los recursos pueden tener asociado un conjunto de operadores O que determinan las acciones que se pueden ejecutar sobre ellos. Así, AYPUY provee un conjunto de operadores básicos que pueden ser extendidos. Las características lógicas generales L describen al recurso y pueden ser empleadas para realizar búsquedas sobre el repositorio de recursos. Las principales características lógicas se mencionan a continuación:

- Alias: es el nombre distintivo del recurso en el sistema.
- Naturaleza: indica si el recurso es lógico o físico.
- Tipo del recurso: cada sistema colaborativo puede predefinir los tipos de recursos que le interesan. Ejemplos clásicos de recursos son OVA (Objetos Virtuales de Aprendizaje), dispositivos, historias clínicas, entre otros.
- Fecha de creación: fecha en la que fue registrado el recurso en el sistema.
- Autor o propietario: identificador del autor en el sistema.

Las características lógicas asociadas al dominio del recurso D aumentan el conocimiento respecto al área particular en la que este recurso es relevante. Para el dominio educativo, el recurso tipo OVA tiene características clásicas definidas incluso por estándares internacionales como el IEEE LOM (IEEE, 2011). Los siguientes son algunos ejemplos:

- Rango típico de edad: indica el rango de edad al que está enfocado el OVA.
- Tipo de recurso de aprendizaje: establece el tipo de recurso educativo que puede ser *Exercise* (ejercicio), *simulation* (simulación), *questionnarie* (cuestionario), *diagram* (diagrama), *figure* (figura), *graph* (gráfico), *index* (índice), *slide* (diapositiva), *table* (tabla), *narrative text* (texto narrativo), *exam* (examen), *experiment* (experimento), *ProblemStatement* (enunciado de problema), *SelfAssessment* (autoevaluación).
- Formato: representa el formato en el que se almacena el OVA (ej*.,* image/jpeg).
- Contexto: define el entorno educativo típico en el que se usará el OVA: ej*.,* *Primary education* (educación primaria), *Secondary education* (educación secundaria), *Higher education* (educación superior).

Por último, las características físicas F permiten describir la manera en la que será almacenado el recurso. Dentro de las características más relevantes se encuentran las siguientes:

- Tipo de almacenamiento (almacenamiento abierto, almacenamiento cerrado): indica si el recurso puede ser almacenado en cualquier nodo del sistema o si su almacenamiento debe estar restringido a ciertos nodos por requisitos de privacidad, por ejemplo, los datos muy personales son recursos que solo pueden estar almacenados en el dispositivo del propietario.
- Tipo de réplica (única, parcial, total): en el caso de los recursos lógicos, indica si el recurso puede ser replicable y si las réplicas existentes pueden ser parciales o totales.
- Tipo de agregación (atómico, agregado): en el caso de los recursos lógicos, indica si el recurso está almacenado de forma atómica (ej*.,* presentación) o si

corresponde a la agregación de la información de varios lugares (ej., historia clínica proveniente de varias instituciones).
- Tamaño: tamaño en bytes para los recursos lógicos atómicos, no aplica para los recursos lógicos agregados.
- Fuentes de creación: esta característica solo aplica para recursos lógicos agregados. Contiene el listado de las fuentes de datos que proveen información para la conformación del recurso.
- Propiedades de conexión: establece los parámetros de acceso al recurso (ej., ruta de acceso al archivo). Para los recursos agregados, esta característica contiene los parámetros de acceso de cada una de las fuentes agregadas.
- Materialización: en el caso de los recursos lógicos, la materialización es el archivo que contiene al recurso.

Los operadores de los recursos permiten utilizar las características lógicas de un recurso para buscar, clasificar y unir recursos. Estas operaciones pueden ser extendidas para satisfacer los requerimientos del dominio que esté usando AYPUY. Los principales operadores ofrecidos por AYPUY son:
- Búsqueda de igualdad (Atributo = Valor): permite obtener los alias y las propiedades de conexión a los recursos que coinciden con el valor definido.
- Búsqueda de desigualdad (Atributo < ^ > Valor): permite obtener los alias y las propiedades de conexión a los recursos que coinciden con el valor numérico definido.
- Clasificación (Atributo con Valores Discretos): clasifica los recursos por grupos a partir del valor que tienen asociado en el atributo seleccionado para clasificar. Al finalizar, cada grupo representa el posible valor del atributo y sus integrantes son los alias y las propiedades de conexión a los recursos coincidentes.
- Parecido: permite obtener los alias de los recursos que contienen la cadena descrita y las propiedades de conexión a ellos.
- Unión (Listado de Alias): solo aplicable para recursos atómicos. Su función es unir los recursos definidos en un archivo comprimido.
- Intersección: solo aplicable a recursos agregados. Dados dos identificadores de recursos, identifica qué porciones tienen en común.
- Complemento: dado uno o más identificadores de recursos, retorna los identificadores de los recursos restantes en el sistema.
- Diferencia: solo aplicable a recursos agregados. Dados dos identificadores de recursos, identifica qué porciones tienen en común y retorna los elementos en los que difieren.

Adicionalmente, AYPUY permite extender los operadores para definir reglas de búsqueda o comparación más específicas para un tipo de recurso o para una característica lógica de dominio particular, por ejemplo, si un recurso representa las imágenes diagnósticas de una enfermedad. El operador *parecido* puede ser extendido para representar la similitud a nivel de imagen entre los diferentes recursos.

B. Espacios de trabajo

Para gestionar el acceso a los recursos, la propuesta de AYPUY consiste en crear espacios lógicos de trabajo (de ahora en adelante espacios de trabajo [ET]) que contengan y gestionen el acceso a recursos. Un ET representa un espacio lógico compartido entre los usuarios participantes de una o más instancias de un proceso colaborativo. Un ET puede ser accedido por una o más instancias de un proceso colaborativo. La figura 1 representa la relación entre espacios de trabajos, recursos, usuarios e instancias de procesos de colaboración.

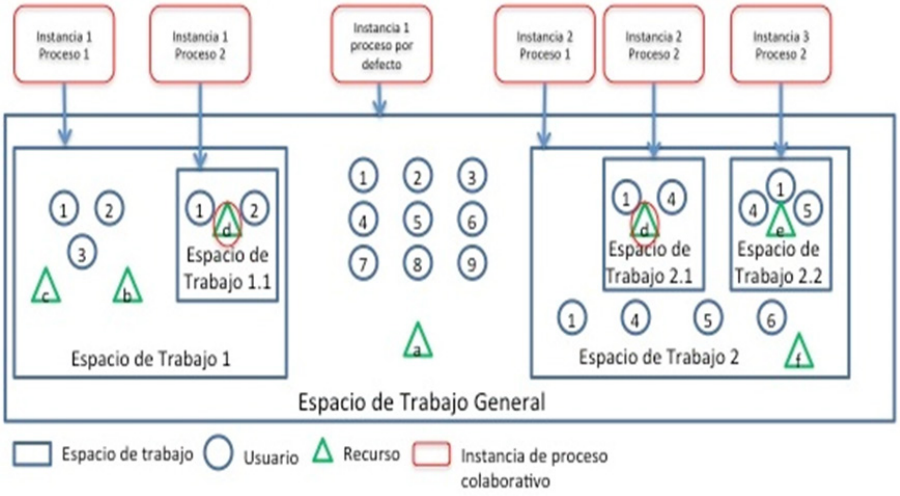


FIGURA 1. ESPACIOS DE TRABAJO.

Fuente: Pomares-Quimbaya, Torres-Ribero, Carrillo-Ramos, Arias-Báez y González (2012).

Como puede observarse, existe una instancia del proceso colaborativo por defecto. Esta instancia crea un ET general al cual tienen acceso todos los usuarios del sistema siguiendo un rol determinado por el proceso colaborativo. En la figura 1, esta instancia es llamada *Espacio de Trabajo General*. Si el sistema requiere crear una instancia de proceso colaborativo para especializar o modificar el rol de acceso a recursos de

todos o de un subconjunto de usuarios, esta instancia debe crear usuarios y su propio ET. Por ejemplo, el ET 1 fue creado para especializar el rol de los usuarios 1, 2 y 3. A su vez, el ET 1.1 fue creado para modificar el rol de los usuarios 1 y 2.

Cuando un ET tiene un subespacio de trabajo asociado, dicho subespacio se conocerá como ET hijo y el ET al que pertenece se le llamará padre o ancestro de ese ET. Por ejemplo, el usuario 5 tiene acceso a los recursos a, e, f, pero no a los recursos b, c ni d. El acceso al recurso del usuario 5 se rige bajo el rol que desempeña en el ET 2.2. De la misma manera, el acceso al recurso f se rige bajo el rol que desempeña el usuario 5 en el ET 2 y así sucesivamente, como ilustra la figura 2. Un usuario asignado a un ET hijo debe estar también asociado al ET padre.

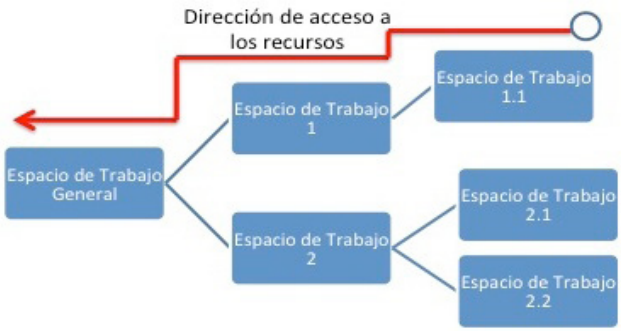


FIGURA 2. PROCESO RECURSIVO DE ACCESO A RECURSOS.

Fuente: Pomares-Quimbaya, Torres-Ribero, Carrillo-Ramos, Arias-Báez y González (2012).

Los atributos que describen a un espacio de trabajo son los siguientes:

- Identificador.
- Identificador de espacio de trabajo padre (opcional).
- Lista de recursos asociados.

C. Arquitectura

Los componentes creados para satisfacer los requerimientos de administración de recursos y espacios de trabajo se describen a continuación:

- *Administrador de Espacios de Trabajo*: es el responsable de administrar los ET pertenecientes al sistema. Permite crear un ET y asociarlo a una instancia del proceso colaborativo del sistema. De la misma manera, proporciona la manera de vincular recursos y usuarios a un ET. Finalmente, permite eliminar ET desvinculando los usuarios y recursos pertenecientes a él.

- *Administrador Lógico de Recursos*: es responsable de gestionar el almacenamiento y consulta de todas las características de un recurso. En particular, este componente realiza la gestión distribuida de los recursos; por ejemplo, se encarga del manejo de las réplicas de un mismo recurso.
- *Administrador Físico de Recursos*: responsable de almacenar la materialización de los recursos en el sistema.

La figura 3 presenta la interacción entre los componentes de AYPUY. Al ingresar una solicitud de usuario, el Administrador de ET (A) es el encargado de revisar que la petición entrante provenga de un usuario autorizado para realizar la acción solicitada. Posteriormente, el Administrador Lógico de Recursos (B) se encarga de realizar la distribución de datos o de realizar la obtención de un recurso, bien sea que provenga de una sola fuente lógica o sea un recurso agregado proveniente de varias fuentes. Finalmente, el Administrador de Recursos Físicos (C) se encarga de localizar el recurso de acuerdo con su lógica de dominio proveniente de la información transmitida por el Administrador Lógico de Recursos.

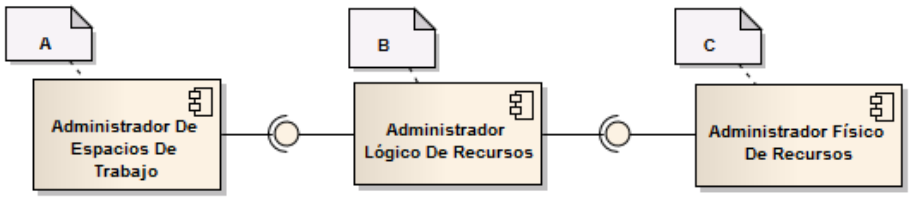


FIGURA 3. COMPONENTES DE AYPUY.

Fuente: elaboración propia.

El Administrador de ET es el punto de entrada para la solicitud de recursos. Es responsable de la creación y eliminación de ET y de la vinculación o desvinculación de los recursos, nuevos o existentes. En este componente se realiza el control de acceso a los recursos a través de la verificación del rol que desempeña el usuario en el ET al cual se quiere tener acceso.

El Administrador de Recursos Lógicos se encarga de la gestión de los recursos de manera lógica dentro del sistema. Esta gestión comprende la realización de consultas de los recursos, la ubicación de estos y la agregación de recursos heterogéneos en el caso que se requiera obtener un recurso agregado. A partir de las características de almacenamiento físico del recurso, se determina la estrategia de replicación o distribución a utilizar para almacenar el recurso. Para resolver una consulta, el

Administrador de Recursos Lógicos mantiene un índice distribuido que contiene los administradores de recursos físicos pertinentes, es decir, los que tengan dichos recursos almacenados.

Por último, el Administrador Físico de Recursos es el encargado de almacenar los recursos físicamente, es decir, de que se manejen como un conjunto de datos los recursos del sistema. Es importante mencionar que el administrador de recursos físicos maneja solamente recursos homogéneos, es decir, los recursos correspondientes a un mismo dominio, por ejemplo, en un ambiente educativo existen varios tipos de recursos, como edificios, aulas, equipos, objetos de aprendizaje, entre otros. Dado que estos recursos pueden tener naturalezas diferentes, por ejemplo, a) un aula y un edificio pueden ser considerados espacios inmuebles, b) un equipo puede ser un elemento móvil limitado y no replicable y, finalmente, c) un OVA maneja información sobre su contenido y es replicable en múltiples nodos. Cada uno de los recursos mencionados anteriormente tiene un administrador diferente de recursos físicos, uno encargado de inmuebles, otro de equipos, etc.

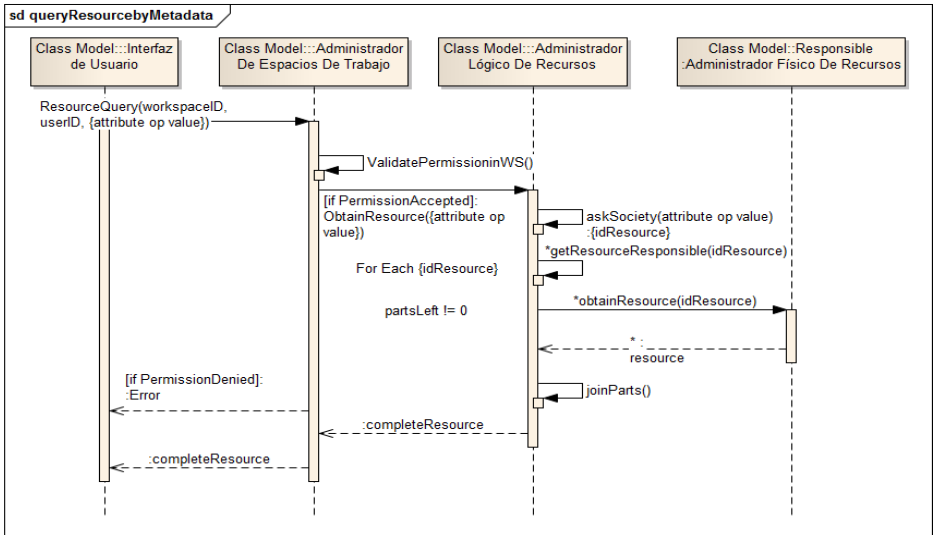


FIGURA 4. CONSULTA EN AYPUY.

Fuente: elaboración propia.

Este componente está conformado por un Manejador de Datos y un Manejador de Dominio. El primero permite obtener, actualizar, crear y borrar un recurso dentro de su almacén de datos. El segundo es el encargado de traducir la consulta realizada por el componente administrador de recursos lógicos al lenguaje nativo

de cada tipo de recurso. Debido a que puede existir una gran cantidad de dominios, la lógica del manejador de dominio debe ser configurada con las diferentes características y definiciones de cada dominio.

Cada uno de estos componentes puede replicarse con el fin de manejar de forma adecuada la concurrencia y la distribución; esto se ve reflejado especialmente en el administrador de recursos lógicos, el cual se replica con el fin de permitir múltiples consultas concurrentes en una arquitectura escalable. Un grupo de administradores de recursos lógicos se conoce con el nombre de "sociedad". La sociedad de administradores de recursos lógicos permite distribuir la gestión de consultas garantizando la escalabilidad con respecto al número de consultas realizando una asignación de responsabilidades de búsqueda a cada sociedad y a cada uno de los administradores dentro de la misma.

La figura 4 ilustra la secuencia para realizar una consulta en AYPUY. La secuencia inicia cuando el usuario envía al administrador de ET la solicitud de un recurso junto con información que permita identificar sus permisos. El administrador de ET verifica los permisos del usuario; en el caso en el que tenga los permisos necesarios, se hace la solicitud al administrador de recursos lógicos. El administrador de recursos lógicos verifica la localización (réplicas o fuentes para agregación) del recurso solicitado; posteriormente, pregunta a su sociedad por el identificador del recurso basándose en las combinaciones atributo-valor que se enviaron en la consulta. La sociedad identifica los administradores de recursos físicos requeridos para recuperar el recurso solicitado. Finalmente, los recursos son solicitados al/los administrador(es) de recursos físicos concerniente(s), integrados (si es necesario) por el administrador lógico de recursos y entregados al administrador de espacios de trabajo.

4. Caso de aplicación de AYPUY

Los componentes de AYPUY fueron implementados utilizando agentes inteligentes, los cuales, debido a sus características de proactividad y por contar con una gran facilidad de replicación y escalamiento, facilitan la gestión de los recursos. Cada uno de los elementos se implementó por medio de un conjunto de agentes *software* proporcionado por el *framework* de agentes BESA (González, 2003) y como servicio de soporte a la plataforma colaborativa AYLLU (Cantor, Mancilla y González, 2006). En la figura 5 se muestra cada uno de los componentes y sus interacciones: el Administrador de ET (AET), el Directorio de Recursos (DIR), el Manejador Lógico de Recursos (MLR) y el Manejador Físico de Recursos (MFR) fueron implementados como agentes de *software* que reaccionan a los diferentes eventos que aparecen en

dicha figura siguiendo protocolos de interacción. Estos protocolos garantizan un manejo de recursos efectivo que permite concurrencia, distribución y replicación. Los protocolos de obtención del *handler* y solicitud del recurso se ilustran en la figura 5.

La petición inicial de un recurso es realizada por un agente Au que representa a un usuario de la plataforma AYLLU en el contexto de un servicio cooperativo. Esta petición incluye metadata que permite ubicar el recurso en forma específica. El agente AET, al recibir la petición, teniendo en cuenta el rol del participante en el marco del agente ET asociado al servicio cooperativo, realiza el control de acceso. Si las restricciones de seguridad son respetadas, a través de los directorios se ubica al agente MLR acorde con la metadata recibida, que retorna directamente el *handler* al agente solicitante Au. El *handler*, además de la información de ubicación del recurso, también incluye una contraseña cifrada, requerida para acceder a él. Este *handler* puede ser intercambiado entre agentes de AYLLU hasta que llegue al agente Ar que realmente requiere usar el recurso. Este agente solicita el recurso pasando como parámetro el *handler* obtenido previamente. La solicitud es dirigida directamente al MLR que corresponde; la información que permite ubicar a este manejador lógico está incluida en el *handler*; nótese que no se requiere pasar por el AET ni los directorios. El MLR, teniendo en cuenta el contexto dentro del sistema distribuido de Ar, selecciona el agente MFR que sea más adecuado; por ejemplo, el que tenga la réplica más "cercana" al solicitante. Una vez el MFR recibe la solicitud del recurso, lo envía directamente al agente Ar que lo requiere.

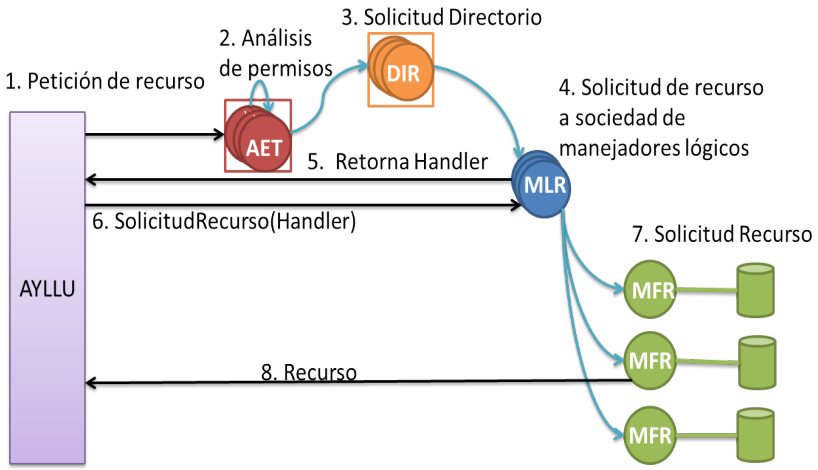


FIGURA 5. IMPLEMENTACIÓN POR AGENTES DE AYPUY: SOLICITUD DE RECURSO.

Fuente: Pomares-Quimbaya, Torres-Ribero, Carrillo-Ramos, Arias-Báez y González (2012).

5. Conclusiones

El sistema de gestión de recursos AYPUY presenta un modelo que distribuye, replica y asigna permisos a diferentes usuarios a partir de los requerimientos de cada instancia de un proceso colaborativo. AYPUY garantiza escalabilidad en el manejo de recursos a través de la construcción de sociedades de agentes que gestionan de forma distribuida los recursos. Aunque actualmente AYPUY hace parte de un sistema de apoyo a la colaboración llamado AYLLU, fue diseñado para ser utilizado por otros sistemas. Así mismo, la configuración de propiedades y operadores sobre los recursos pueden ser especializados para soportar los requerimientos en diferentes dominios. Actualmente AYPUY está siendo empleando como el gestor de recursos en la implementación para el apoyo al aprendizaje del sistema AYLLU, cuyo objetivo es fortalecer los procesos de enseñanza-aprendizaje.

Referencias

Bannon, L. J. y Schmidt, K. (1991). *Studies in Computer Supported Cooperative Work*. Amsterdam: North-Holland Publishing.

Cantor, O., Mancilla, L.E. y González, E. (2006). AYLLU: A Cooperative Approach to Groupware Development. *IEEE Int. Conf. Commun.*, *1*,447-452, 2006.

Demetriadis, S. y Karakostas, A. (2008). Adaptive Collaboration Scripting: A Conceptual Framework and a Design Case Study. *Complex, Intelligent and Software Intensive Systems*, 487-492.

Fuks, H., Raposo, A., Gerosa, M. A., Pimentel, M., Filippo, D. Lucena, C. (2008). Inter- and Intra-relationships Between Communication Coordination and Cooperation in the Scope of the 3C Collaboration Model. *Computer Supported Cooperative Work in Design*, 148-153.

GForge. (2015). Página principal. Disponible en https://gforgegroup.com/.

González, E., Bustacara, C. y Ávila, J. (2003). BESA: Arquitectura para Construcción de Sistemas MultiAgentes. *Conferencia Latinoamericana de Estudios en Informática*, 2003.

HydraPSA. (2015). Página principal. Disponible en https://site.hydrapsa.com/es/.

IEEE. (2011). IEEE Standard for Learning Object Metadata - Corrigendum 1: Corrigenda for 1484.12.1 LOM (Learning Object Metadata). IEEE Std 1484.12.1-2002/Cor 1-2011 (Corrigendum to IEEE Std 1484.12.1-2002), 1-24.

Jazz. (2015). Página principal. Disponible en https://jazz.net/. Machulak, M. P., Maler, E. L., Catalano, D. y van Moorsel, A. (2010). User-managed Access to Web Resources. *Proceedings of the 6th ACM Workshop on Digital Identity Management*, 35-44.

Poltrock, S. y Handel, M. (2009). Modeling Collaborative Behavior: Foundations for Collaboration Technologies. *System Sciences*, 1-10.

Pomares Quimbaya, A., Torres-Ribero, L. G. Carrillo-Ramos, Á., Arias-Báez, M. P. y González, E. (2012). AYPUY: A resource management system for collaborative environments. *7th Colombian Computing Congress (CCC)*.

Rama, J. y Bishop, J. (2006). A Survey and Comparison of CSCW Groupware Applications. *Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, 198-205.

Rodríguez-Martínez, M. y Roussopoulos, N. (2000). MOCHA: A Self-extensible Database Middleware System for Distributed Data Sources. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 213-224.

Sinclair, A., Finney,D., Martin, D. y Sharma, P. (2014). Distance Protection in Distribution Systems: How It Assists With Integrating Distributed Resources. *Ind. Appl. IEEE Trans*., *50*(3), 2186-2196.

Yang, H., Ye, F. y Liu, Z. (2007). Resource Discovery in Federated Systems with Voluntary Sharing. *Proceedings of the 2007 ACM/IFIP/USENIX International Conference on Middleware Companion*, 261-266.

12
PLATAFORMA BESA 3

Juan Pablo Garzón, Andrea Barraza, Enrique González

Este capítulo describe la plataforma (*framework*) BESA, en su versión 3, sobre el cual se ha realizado la implementación y la ejecución de los Sistemas Multiagente (SMA) que se utilizan en AYLLU. Esta versión incluye entre sus nuevas características: comunicación en redes WAN entre diferentes dominios de red llamados barrios (*broadcast*), especificación del archivo de configuración por medio del lenguaje de marcas extensible (XML), organización de la jerarquía de paquetes para facilitar al programador la construcción e implementación de un agente sin necesidad de creación de comportamientos (*behaviors*) internos, la eliminación del uso de constructoras para las guardas (*guards*), y la capacidad de extender el uso del sistema a dispositivos móviles. Estas características hacen posible que los analistas, los diseñadores y los programadores pueden utilizar la plataforma BESA para construir fácilmente nuevas aplicaciones basadas en agentes.

Para entender el funcionamiento de las nuevas características de BESA, este capítulo se divide en cuatro secciones y se basa en la introducción base de BESA 2 para la descripción general de la plataforma (Garzón *et al*., 2007). En la primera sección se describen los conceptos fundamentales que utiliza el *framework* para hacer la implementación. En la segunda se reseña la arquitectura del *framework* y se describen sus componentes principales y su interacción. En la tercera parte, por medio de ejemplos, se detalla la forma de usar BESA para crear una implementación de un SMA. Por último, en la cuarta sección, se describen los elementos principales que se deben tener en cuenta para escribir una solución con BESA. Después de leer este capítulo, el lector deberá estar en capacidad de implementar una solución de SMA en el contexto AYLLU con las nuevas características BESA, aprovechando las grandes ventajas que esta provee.

1. Introducción

BESA es un *framework* creado en el Departamento de Ingeniería de Sistemas de la Pontificia Universidad Javeriana de Bogotá, Colombia. Toma su nombre, en inglés, de cuatro de las principales características que provee: *behaviors*, *events*, *social* y *agents*. Estas características describen los conceptos fundamentales que utiliza el *framework* para diseñar la solución. En BESA, al igual que en cualquier *framework* de SMA, un agente es una entidad que incluye mecanismos para recibir percepciones de su ambiente y para modificarlo. El trabajo de un agente consiste en decidir o deducir cuál es la acción más adecuada para alcanzar una meta específica. Un agente tiene varios recursos y habilidades, y puede comunicarse con frecuencia con otros agentes (Rusell, 2009). Se selecciona la acción correcta usando una función de toma de decisiones que pueda ser expresada de diferentes maneras, desde un algoritmo simple en lenguaje procedural, pasando por reglas de tipo condición-acción y hasta mediante la utilización de mecanismos de inferencia más complejos. En algunos casos, la función de mapeo puede ser dada directamente por el diseñador; en agentes con mayor autonomía esta función puede ser incluso aprendida directamente por el agente.

Las capacidades de un agente aislado se limitan a sus recursos y habilidades. Cuando los objetivos son más complejos, la función de mapeo para seleccionar la mejor acción es menos eficiente, porque la complejidad de esta función aumenta. Así es más eficiente construir varios agentes, donde cada uno contribuye para alcanzar la meta general. Un SMA se puede definir como una colección de agentes que coopera para alcanzar un conjunto de metas comunes (Labidi y Lejouad, 1993). Un SMA tiene un conjunto de agentes heterogéneos que se ejecutan en una plataforma que respalda servicios de identificación e interacción entre agentes. En un contexto más extendido, un *framework* para la construcción de SMA debe considerar los siguientes elementos: comunicación entre los agentes para intercambiar la información o el conocimiento y localización de recursos (Ferber, 1999; Rueda, 1996). En un SMA, los agentes deben ser autónomos, adaptativos y cooperativos tanto en tiempo de diseño como en ejecución. La aproximación de SMA con BESA permite construir sistemas complejos en un ambiente distribuido y heterogéneo.

Por otro lado, la comunicación entre agentes requiere un lenguaje estándar que les permita interactuar mientras ocultan los detalles de su operación interna, a fin de generar comunidades de agentes que abordan los problemas que ningún agente individual podría solucionar. Un lenguaje de comunicación entre agentes (ACL) se refiere solamente a las actitudes proposicionales sin importar cómo se expresan (Labrou y Finin, 1999). KQML (*Knowledge Query and Manipulation Language*) es un

lenguaje de comunicación clásico de alto nivel orientado a mensajes e independiente de la sintaxis del contenido y de la ontología (Finin *et al*., 1994). Tiene una estructura que permite separar claramente los distintos componentes de la comunicación. Además, separa aquellos elementos de bajo nivel que rodean el envío de un mensaje (remitente, destinatario, etc.) del contenido real del mensaje, definido por un conjunto de palabras compartidas por los agentes (el vocabulario) y de la intención del mensaje (*performativa*). El lenguaje también promueve y toma en cuenta la autonomía de los agentes, al conceptualizar la comunicación como un intercambio de mensajes con intención entre sistemas independientes que persiguen diferentes metas y no como mensajes que realmente son comandos o llamadas a funciones entre elementos fuertemente ligados y dependientes.

Igualmente se encuentra FIPA ACL, que es un lenguaje de comunicación entre agentes como KQML que utiliza BESA (FIPA, 2017). Está basado en actos del habla: los mensajes son acciones o actos comunicativos, pues son intenciones para realizar una cierta acción en virtud de haber sido enviado. La especificación de FIPA ACL consiste en un conjunto de *performativas* y la descripción de sus intenciones, es decir, los efectos de las actitudes mentales de los agentes que envían y reciben los mensajes. La especificación describe cada acto comunicativo tanto en forma narrativa como en una semántica formal basada en la lógica modal (Labrou y Finin, 1999).

Otro factor importante en la comunicación es el que hace referencia a la operación de la plataforma en un entorno de cómputo distribuido. El *framework* BESA está compuesto de un conjunto de contenedores, típicamente uno en cada máquina, que se comunican con tecnologías de transmisión por difusión y P2P (*peer to peer*). En la tecnología de transmisión por difusión, la comunicación va desde un contenedor a todos los contenedores del sistema. Esta comunicación va dirigida a administrar el ciclo de vida de contenedores y agentes. Cuando los agentes desean comunicarse entre sí, se emplea una comunicación P2P. Todos los contenedores que utilizan la misma dirección IP y el mismo puerto *multicast* conforman un mismo grupo de agentes, denominado barrio, cada uno de estos grupos puede asociarse con otros para conformar un solo SMA.

Además de los conceptos fundamentales, la nueva versión BESA 3 extiende el alcance del sistema a diferentes dominios de *broadcast*, denominados barrios en BESA 3, y permite ejecutar contenedores compactos adecuados para dispositivos con restricciones en procesamiento, almacenamiento y red como son los dispositivos móviles. En las versiones anteriores se utilizaban, para el manejo del ciclo de vida de agentes y contenedores, exclusivamente mensajes que se difundían utilizando direcciones *multicast*. Estas direcciones tienen alcance en las redes de datos hasta el

dominio de *broadcast*. Este dominio es dividido por dispositivos de red de propósito específico como son los enrutadores. Para aumentar el alcance geográfico de una aplicación SMA, a partir de la versión 3 del *framework*, se propone un conjunto de nuevos componentes para lograr la comunicación entre diferentes barrios. Otra característica nueva es la de permitir interacción entre contenedores elaborados en sistemas autónomos estáticos y contenedores en dispositivos móviles con la plataforma Android. Esta característica permite además de la interacción, la ejecución de un contenedor BESA en dispositivos con fuertes restricciones en procesamiento, almacenamiento y conectividad a la red; en efecto, un contenedor que tiene este tipo de limitaciones es ayudado por otro contenedor que opera en un dispositivo con altas capacidades computacionales.

2. Conceptos fundamentales BESA

El modelo abstracto de BESA se basa en tres conceptos fundamentales: una arquitectura orientada a comportamientos del agente (*behavior-oriented*), una técnica de control orientada a eventos que implementa un mecanismo de selección (*event-driven*) y un respaldo basado en mecanismos sociales para lograr la cooperación entre los agentes (*social-based*).

2.1. *Control orientado a comportamientos*

Al construir agentes, uno de los problemas críticos por solucionar es la complejidad, pues el agente se piensa para ser más racional y autónomo; por lo tanto, los elementos involucrados pueden llegar a ser muy complejos. Para ocuparse de este problema cada vez mayor, se han propuesto diversas arquitecturas modulares. La idea fundamental consiste en dividir una entidad compleja en un conjunto de entidades más simples. Este proceso puede manejarse de una manera sistemática, por ejemplo, la metodología presentada por Foster para diseñar sistemas concurrentes, que permite obtener un conjunto de entidades capaz de reducir al mínimo la comunicación entre ellas (Foster, 1995). Sin embargo, al usar tal acercamiento, las entidades que resultan no poseen ninguna semántica asociada y, por lo tanto, no es posible analizarlas y modificarlas de una manera estructurada.

La comunidad de la inteligencia artificial ha propuesto una descomposición modular de la mente, donde un conjunto de entidades bien definidas coopera como una sociedad (Minsky, 1986). El concepto de comportamiento que se aplica en BESA, inspirado por la arquitectura *subsumption* (Brooks, 1986), define un *behavior*

como una entidad que encapsula lo necesario para garantizar la realización de un propósito bien definido. Este concepto se ha utilizado para construir sistemas inteligentes complejos. Un comportamiento tiene una semántica clara en el contexto de las metas de un agente, las cuales permiten crear relaciones semánticas entre ellas que conducen a sociedades cooperativas donde se obtiene una conducta racional. BESA incorpora esta técnica *behavior-oriented*: un agente se compone de un conjunto de comportamientos concurrentes. El conjunto de comportamientos concurrentes es parte fundamental del *framework*, sin embargo, para el desarrollador en la actual versión 3 no se tiene que hacer explícita esta creación. Los comportamientos (*behaviors*) se crean de forma implícita cuando se construye un agente.

2.1. *Dinámica basada en eventos*

En BESA, un evento se puede interpretar como una señal que puede ser percibida o usada por un agente que está interesado en ella, la cual puede incluir información sobre lo que ha sucedido en el ambiente. Lo relevante para los agentes no es la información, sino el hecho de haber recibido un estímulo al cual debe reaccionar para producir una respuesta. Un evento se puede producir cuando algo referido al agente ocurre en el ambiente; entre estos se incluyen también mensajes de otros agentes. Nótese que, en este contexto, el contenido del mensaje en sí mismo no es lo importante, sino el hecho de que hay un acto comunicativo con una intención bien definida.

Para el manejo de los eventos, BESA se basa en una técnica *event-driven*, en la cual un agente debe estar listo para reaccionar a las múltiples señales no determinísticas. En efecto, el ambiente de operación de un SMA es inherentemente no determinístico debido a que es imposible determinar el orden y las características temporales de los eventos que se producen. Un mecanismo tipo *select*, basado en el modelo propuesto por lenguajes de programación concurrente –por ejemplo, el definido en CSP (Bernstein, 1980)– permite hacer un manejo adecuado del no determinismo. Gracias a este mecanismo, BESA proporciona una semántica muy poderosa que permite implementar programas concurrentes de una manera elegante y clara.

El modelo abstracto de un mecanismo *select* es similar a una estructura *switch-case*, con la característica especial de ser bloqueante; es decir, mientras alguna de las condiciones asociadas a los *case* no se cumpla, la ejecución del programa permanece bloqueada. En el modelo formal, un mecanismo *select* incluye un conjunto de elementos conocidos como guardas. Una guarda está asociada a un tipo específico de evento y se compone de una función booleana y un bloque de instrucciones. Generalmente, la función booleana se relaciona con el estado del sistema. Para cada guarda se define un bloque de instrucciones, cuya función es realizar el tratamiento

adecuado dependiendo del tipo de evento asociado. Una guarda se dispara cuando al verificar una condición booleana resulta verdadera y el evento asociado a la guarda ocurre. Cuando la guarda se dispara, el bloque de instrucciones definido para esta es ejecutado.

2.3. *Mecanismos sociales de cooperación*

Para analizar y diseñar un SMA, el uso de un modelo basado en una aproximación social permite estudiar el sistema como una organización de entidades que interactúan entre sí. Ferber ha propuesto un conjunto de funciones esenciales y de dimensiones para analizar una organización de agentes; tal acercamiento tiene la ventaja de identificar de manera estructurada las relaciones de las entidades que componen el sistema, así como las conexiones con otros sistemas (Ferber, 1999). En un modelo social, una organización puede ser considerada un conjunto de organizaciones de bajo nivel. Esta descomposición recurrente de la organización en otras más simples proporciona una arquitectura para diseñar un SMA. En el nivel más alto de abstracción, el SMA se compone de un conjunto de organizaciones con una semántica y funcionalidades bien definidas. Estas organizaciones se subdividen de una manera recurrente. Al final de este proceso se obtiene un conjunto de entidades simples, las cuales son implementadas como agentes. Los agentes que resultan tienen una semántica clara, que incluye una meta en la organización y una interfaz bien definida con otros agentes y el ambiente.

En el nivel más bajo, un SMA se compone de una colección de agentes planos; no obstante, si se ha logrado una descomposición recursiva, el modelo incluye entidades abstractas de más alto nivel: por ejemplo, las organizaciones, que se pueden considerar agentes que sirven como facilitadores para promover la cooperación de las entidades de más bajo nivel. BESA proporciona elementos para construir patrones que manejen e implementen mecanismos de cooperación y protocolos entre los agentes de bajo nivel mediante agentes de más alto nivel. BESA utiliza, entre sus mecanismos de comunicación entre *frameworks,* la difusión del tipo *multicast*, lo cual provee la capacidad de crear y manejar correcta y eficientemente grupos de agentes. Basándose en este esquema, la comunicación puede llegar incluso a todos los agentes del sistema. Además, el programador puede utilizar agentes intermediarios como una alternativa para un manejo de grupos de agentes especializados, incluyendo en la lógica del agente las referencias de los agentes que componen el grupo.

El manejo *multicast* extendido tiene dos nuevos componentes llamados BAB y BABLocator que permiten conformar un grupo de contenedores limitados por el

alcance de la mensajería *multicast* llamado barrio y el BABLocator, que permite la interacción entre diferentes barrios.

Debido a las limitantes de los dispositivos móviles, se implementó adicionalmente un conjunto de componentes: BESA *PostOffice* llamado BPO y agentes llamados *AgentePostMan*, los cuales posibilitan la interacción de dispositivos móviles y dispositivos estáticos. Estas nuevas capacidades permiten tener agentes desarrollados en diversos lenguajes de programación y ejecutando en diferentes plataformas de *hardware* que se comuniquen y que forman un solo sistema.

3. Arquitectura BESA

La arquitectura BESA está compuesta por tres niveles: nivel de agente, nivel social y nivel de sistema. Cada uno de ellos describe los elementos básicos y la interacción que componen una solución de SMA con BESA.

3.1. *Nivel de agente*

La arquitectura interna de un agente integra dos características importantes: una composición modular de comportamientos y un mecanismo selector de eventos. Un agente está compuesto por un canal, un conjunto de comportamientos y un estado. El estado es una memoria compartida que puede ser utilizada por los comportamientos del agente usando sincronización de exclusión mutua para evitar problemas de concurrencia. En la figura 1 se presentan los componentes del agente y su interconexión.

El canal recibe en su buzón los eventos dirigidos al agente e implementa un mecanismo selector basado en guardas. El buzón constituye el único punto de entrada para los eventos recibidos por el agente. El selector de guardas transfiere los eventos entrantes a la guarda apropiada, dependiendo del tipo de evento. Una guarda implementa un mecanismo asincrónico, el cual verifica si esta puede ser disparada. La guarda se dispara cuando el evento asociado ocurre y el resultado de la evaluación de la función booleana es verdadero. Si la condición booleana no se cumple, el evento es almacenado temporalmente. Cuando una guarda se dispara, el primer evento almacenado en esta es enviado a los comportamientos adecuados. Para garantizar que la semántica del mecanismo selector sea respetada, todos los eventos, incluso los intercambiados entre comportamientos de un mismo agente, se reciben en el buzón y deben ser procesados por el selector de guardas.

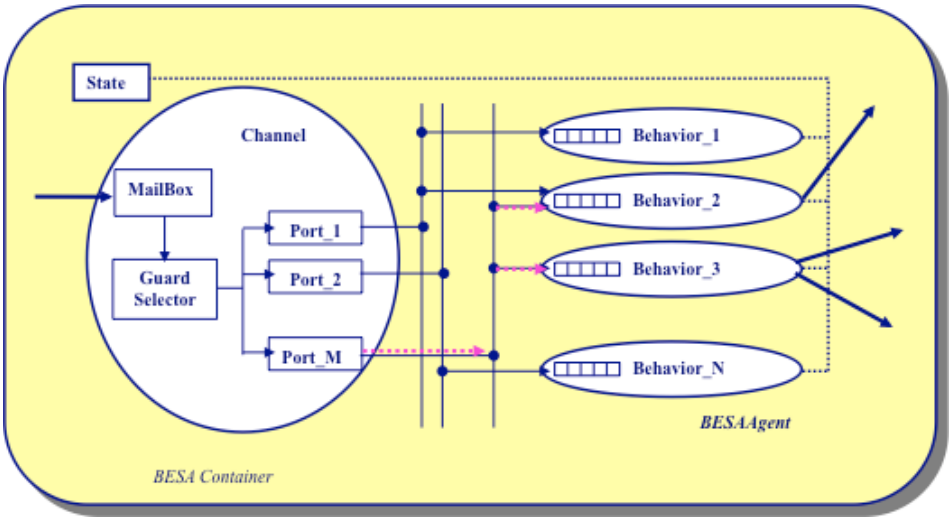


FIGURA 1. ARQUITECTURA INTERNA DE UN AGENTE BESA.

Fuente: elaboración propia.

Los comportamientos son procesos paralelos que trabajan juntos para lograr las metas del agente. Un comportamiento permite reaccionar a un conjunto bien definido de tipos de eventos; para demostrar su interés en un tipo específico de evento, el comportamiento es registrado ante la guarda asociada al evento. Un evento le será enviado cuando la guarda sea disparada. Cada comportamiento tiene una cola interna para recibir eventos de las guardas ante las cuales se registró. Cuando llega un evento, el comportamiento ejecuta automáticamente el bloque de instrucciones asociado a la guarda, el cual debe incluir el programa apropiado para reaccionar y para producir una respuesta al evento recibido.

Para la implementación, una guarda tiene dos funciones asociadas. La primera es ejecutada automáticamente por el mecanismo de disparo para verificar la condición booleana. La segunda, que es ejecutada por los comportamientos, incluye el bloque de instrucciones de la guarda. Nótese que hay una relación de 1 a 1 entre el tipo de evento, la guarda y las funciones asociadas. Una característica importante de este esquema es que el código del programa que implementa un comportamiento es genérico y se puede escribir independientemente de la especificidad del agente. De hecho, un comportamiento recibe un evento y dependiendo de su tipo ejecuta un programa. En consecuencia, el diseñador del agente tiene que describir solamente la estructura interna del agente, enlazando los comportamientos con las guardas y proporcionando las funciones asociadas a cada guarda.

En algunos tipos de aplicaciones se requiere que los agentes puedan detener su ejecución y continuar su trabajo en otra máquina. Esta característica se denomina movilidad de agentes –la estructura de BESA permite implementar fácilmente esta funcionalidad–. Para detener la ejecución de un agente se suspende la operación de todos sus comportamientos en un estado interno estable bien determinado; cuando se han detenido todos los comportamientos del agente, este puede ser movido con seguridad. Una vez que se haya restaurado el contexto del agente en la máquina destino, los comportamientos reasumen la ejecución en forma transparente; los eventos destinados al evento son redireccionados en forma transparente para el programador. Para permitir la comunicación entre dispositivos estáticos y móviles se adicionan nuevos componentes al *framework*, cuyos principales son: BESA *PostOffice* (BPO) que tiene por objetivo principal gestionar el intercambio de mensajes entre contenedores móviles y estáticos; y el BABLocator que permite ubicar un contenedor que se puede encontrar en un dominio de red *broadcast* distinto; cada grupo de agentes en este contexto se llama barrio.

3.2. *Nivel social*

En el modelo abstracto de BESA, los niveles intermedios de abstracción son incorporados para permitir la definición y la manipulación de organizaciones jerárquicas. Una organización se define como un conjunto de agentes asociados a un agente mediador del siguiente nivel más alto de la jerarquía global de la organización. Por su parte, el mediador actúa como facilitador del trabajo cooperativo de los agentes en la organización. Además, puede servir como un despachador de eventos que recibe eventos y los reenvía a los agentes adecuados o como puente a un conjunto de los agentes capaces de ejecutar o proveer un servicio. En este caso, el mediador es registrado en un directorio de páginas amarillas como el prestador de un servicio y cuando un evento de la petición llega a él, es dirigido al agente apropiado. En una situación más compleja, el mediador puede ser diseñado para manejar mecanismos de cooperación, proporcionando los servicios para asignación de tareas y recursos, resolución de conflictos y planeación y coordinación distribuidas.

En este modelo basado en una aproximación social, un agente mediador se ve como una entidad que incorpora un conjunto de agentes asociados. Sin embargo, cuando el sistema es desarrollado, no es compatible con el concepto de agente autónomo (un agente no puede estar dentro de otro). En efecto, el sistema estará compuesto de un sistema de agentes planos, donde los enlaces que describen relaciones organizacionales entre los agentes son solamente lógicos. En BESA, la implementación de las relaciones sociales se apoya en agentes sociales mediadores que ponen en

funcionamiento protocolos de interacción definidos e integran servicios de apoyo a la cooperación. Por ejemplo, la comunicación para un grupo de agentes utiliza un agente mediador social que se encarga de la gestión del grupo y de garantizar la entrega ordenada de los eventos a los agentes pertenecientes al grupo.

Un servicio del mediador se puede modelar e implementar por medio de un protocolo de interacción, que es una secuencia determinada de interacciones, o intercambio de eventos, entre los agentes involucrados. Una interacción se puede modelar por medio de un evento bien definido, que se puede asociar a una guarda. Cada agente involucrado en el protocolo debe incluir un comportamiento para manejar la secuencia de interacciones requeridas. Para el diseñador del sistema, el trabajo se reduce a proporcionar los procedimientos asociados a un conjunto predefinido de guardas; la semántica y los datos de estas guardas son determinados por su protocolo de interacción asociado.

3.3. *Nivel sistema*

Un sistema de agentes implementado usando el *framework* BESA se considera un sistema distribuido compuesto por uno o varios contenedores BESA, que pueden funcionar en diversas máquinas físicas o virtuales e, incluso, en diversos dominios *broadcast*. El sistema entero incluye un único puerto compatible con FIPA, que recibe la comunicación y las consultas de otros sistemas externos usando FIPA ACL. La figura 2 ilustra los componentes de un sistema típico BESA.

Un contenedor BESA es un espacio de ejecución donde los agentes operan; este maneja su ciclo vital. Un contenedor tiene un administrador local, que apoya la comunicación de los agentes y proporciona servicios de directorio de páginas blancas y amarillas. Los contenedores de un sistema BESA funcionan juntos e implementan un mecanismo de replicación que hace transparente la identificación de agentes, la localización y la migración. Cuando una nueva instancia de un agente se crea, el administrador local del contenedor lo registra en el directorio de páginas blancas. Si se requiere, un agente puede colocar los servicios que proporciona en el directorio de páginas amarillas. Cuando un cambio en el ciclo de vida de un agente se realiza o cualquiera de los directorios se modifica, un mecanismo de copia se activa para mantener la consistencia de los directorios en todos los administradores locales.

El directorio de páginas blancas está compuesto por una colección de manijas (*handlers*) de identificación de los agentes. Una manija incluye toda la información necesaria para localizar un agente en el sistema. Cuando un agente requiere enviar un evento a otro, realiza una solicitud al administrador local para que le retorne la manija del agente destinatario. Si el agente está contenido en el contenedor local,

la dirección del buzón del agente (donde el evento debe ser enviado) puede ser obtenida directamente de la manija. Si un contenedor externo contiene al agente destinatario, la información necesaria para realizar la solicitud está incluida en el *handler*, con lo cual BESA puede realizar el envío en forma adecuada. En todos los casos, el envío del mensaje se hace en forma transparente para el programador, que solo debe invocar el método de envío de eventos sobre el *handler*. Además, BESA es un *framework* que es utilizable en variedad de ambientes tanto de *software* como de *hardware*; en algunos casos, esta reutilización se basa en JAVA, pero en otros es necesario crear adaptadores debido a las limitaciones que existen (un ejemplo de este caso se aplica a los dispositivos móviles). Esta integración de otros ambientes es fácil en BESA debido a que se tienen actualmente variedad de adaptadores.

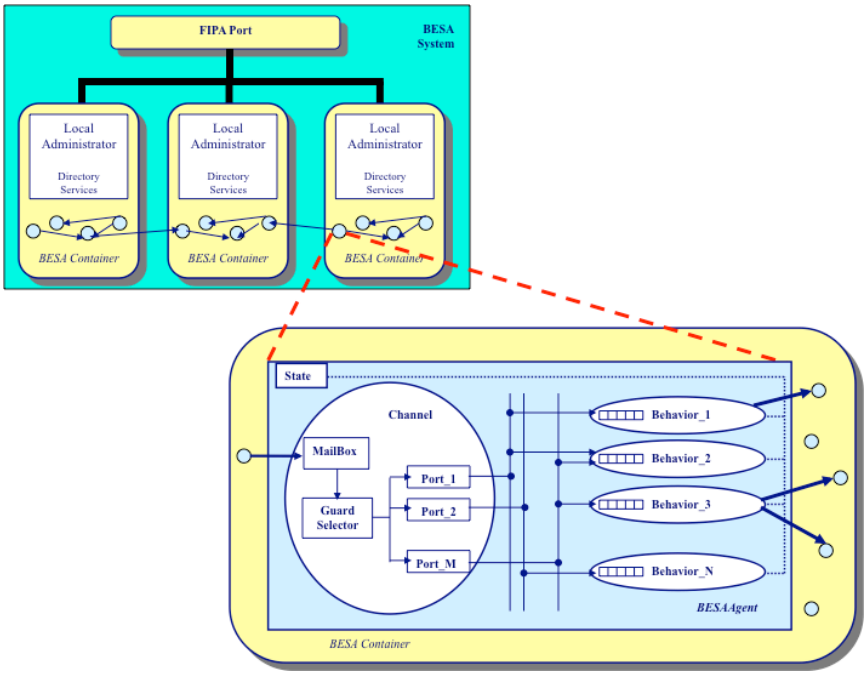


FIGURA 2. MODELO DEL NIVEL DEL SISTEMA EN BESA.

Fuente: elaboración propia.

En el ciclo de vida de un agente, en especial en la creación y destrucción, es necesario manejar un proceso complejo que es transparente al programador para mantener la integridad del sistema. Este proceso busca evitar, por ejemplo, la pérdida de eventos antes de la creación del agente y referencias a agentes ya destruidos.

Este proceso es muy similar al que se utiliza en la movilidad y que se explica al final de este capítulo.

En ambientes con dispositivos móviles, se tiene un contenedor especial *compacto* adecuado a las limitantes del dispositivo como, por ejemplo, baja capacidad y velocidad de memoria principal, memoria secundaria, procesamiento y comunicación inalámbrica que puede ser de bajas tasas de transferencia o intermitencia. El componente ligero, debido a estas capacidades limitadas, se apoya en un contenedor estático estándar como elemento para manejar estas limitantes; este contenedor no tiene las limitantes anteriores debido a que se ejecuta en un sistema de cómputo como un computador personal o servidor.

A nivel de sistema, se adicionan nuevos componentes que proveen la funcionalidad de conformación de barrios que corresponden a conjuntos homogéneos de contenedores BESA llamados BAB y BABLocator, y entre contenedores compactos y estáticos como el BESA *PostOffice* (BPO), que es el encargado de gestionar el intercambio de mensajes entre contenedores estándar y contenedores compactos. Para el BPO se utiliza adicionalmente un agente especial llamado el agente *PostMan* encargado del manejo de colas de mensajes y del canal de comunicación.

4. Modelo de implementación de BESA

El *framework* BESA está implementado usando el lenguaje de programación JAVA y el contenedor compacto en la plataforma Android. Esta selección permite ejecutar BESA en gran cantidad de plataformas heterogéneas y dispositivos móviles. El único requisito consiste en tener instalado en el equipo la versión adecuada para el sistema operativo de la máquina virtual de JAVA y en los dispositivos móviles tener mínimo la versión 4.0 de la plataforma Android. El canal y los comportamientos son implementados por hilos concurrentes, donde el método *run* procesa los eventos entrantes. Dicho método ya está codificado y no es accesible al diseñador del agente. Además, BESA proporciona una jerarquía de clases, entre estas, las clases abstractas básicas para los agentes, los estados, las estructuras, los comportamientos, las guardas, los adaptadores y los agentes sociales. Cabe destacar que la implementación de agentes sociales depende de la creación de agentes especializados o de código intermedio en el *framework*, basándose en el mecanismo de comunicación de difusión del tipo *multicast* y el BAB y BABLocator para que los contenedores y sus agentes puedan comunicarse en diferentes dominios *broadcast*. Para implementar un agente, las clases derivadas deben ser creadas y muy pocos métodos deben ser implementados. Estos

métodos incluyen procedimientos de *setup* y *shutdown* para los agentes, al igual que los procedimientos asociados para disparar y ejecutar cada una de las guardas.

En el arranque, un contenedor registra los servicios remotos en el registro de RMI; luego, un mensaje *multicast* se envía para informar a otros contenedores que estén en funcionamiento sobre la existencia de un nuevo contenedor; se hace un proceso similar para que los componentes del BAB y el BABLocator conozcan la existencia de este nuevo contenedor en un barrio y entre barrios. Cuando un contenedor recibe esta clase de mensajes, actualiza su tabla de contenedores, se registra en el nuevo contenedor y envía información de agentes residentes en él. De esta manera, el nuevo contenedor tiene noticia de todos los contenedores del sistema y de sus agentes. Este proceso se ilustra en la figura 3. Nótese que la utilización del mecanismo *multicast* limitaba el ámbito de operación del sistema de comunicación entre contenedores, pero con el BABLocator este limitante ya no se presenta.

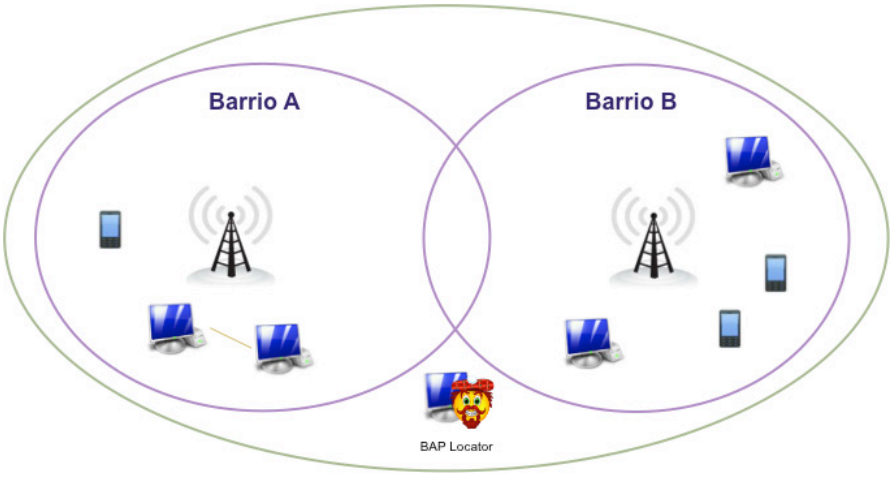


FIGURA 3. EJEMPLO DE NOTIFICACIÓN DE EXISTENCIA DE UN NUEVO CONTENEDOR.

Fuente: elaboración propia.

Para informar cualquier cambio en el ciclo de vida de un agente o en una entrada del directorio, el contenedor, y específicamente el proceso administrador local, invoca los servicios de duplicación de los otros administradores. Este esquema permite manejar tolerancia a fallas de tal forma que, si un contenedor colapsa, el sistema puede continuar funcionando adecuadamente. Sin embargo, en los contenedores donde los agentes se crean y se destruyen continuamente, el mecanismo de duplicación puede inducir una sobrecarga significativa.

El puerto FIPA es implementado igual que un adaptador con una fuente de datos tipo *socket*, como un hilo que recibe las peticiones en un puerto *multicast* bien conocido, el cual se considera como una parte integral de los administradores locales del sistema. Cuando llega una petición de comunicación, su contenido se verifica sintáctica y semánticamente antes de que se transforme en un llamado al contenedor. La comunicación entre agentes se hace por medio de otro puerto FIPA, que se utiliza para comunicarse con el servicio de registro RMI y comunicarse por medio de referencias remotas a los agentes. La dirección IP y el puerto del BABLocator permiten comunicación entre diferentes barrios.

5. Creación de aplicaciones con BESA

A continuación se describen los elementos principales que se deben tener en cuenta para escribir una solución con BESA. Entre los elementos se encuentran el contenedor, los agentes, los eventos y las características especiales como la movilidad. El código que se proporciona como ejemplo solo incluye las líneas más importantes y descarta algunas otras que no son de relevancia.

5.1. *Estructura del main*

El *main* de cualquier aplicación BESA debe estar estructurado de la siguiente manera: primero se debe inicializar el contenedor BESA haciendo uso de un archivo de propiedades en XML; luego se debe crear la estructura del agente en la cual se asocian las guardas. Después, se crea el agente y se pone en ejecución y, finalmente, se busca el agente con alias *AgentA* y se envía un evento con el mensaje "Hello [0]". Este proceso se ilustra en la figura 4.

```
package baptest;

import BESA.ExceptionBESA;
import BESA.Kernel.Agent.Event.EventBESA;
import BESA.Kernel.Agent.StructBESA;
import BESA.Kernel.System.AdmBESA;
import BESA.Log.ReportBESA;
import agenta.AgentA;
import agenta.AgentAGuard;
import agenta.AgentAState;
import agenta.SayHelloData;

public class AgentAMain {
```

```
    /**
     * Administrator of the local BESA container.
     */
    private static AdmBESA admLocal;

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        //-------------------------------------------------------------------//
        admLocal = AdmBESA.getInstance("res/Container1.xml");
        //-------------------------------------------------------------------//
        ReportBESA.info(admLocal.getAdmHandler().getAlias() + " [OK].");
        //-------------------------------------------------------------------//
        StructBESA structAgent = new StructBESA();
        structAgent.bindGuard(AgentAGuard.class);
        try {
            AgentA agentA = new AgentA("AgentA", new AgentAState(), structAgent, 77.77);
            agentA.start();
        } catch (ExceptionBESA ex) {
            ReportBESA.error(ex);
        }
        //-------------------------------------------------------------------//
        ReportBESA.info("Start [OK].");
        try {
            for (int index = 0; index < 1; index++) {
                SayHelloData sayHelloData = new SayHelloData("Hello[" + index + "]");
                admLocal.getHandlerByAlias("AgentA").sendEvent(new EventBESA(AgentAGuard.
                class.getName(), sayHelloData));
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException ex) {
                    ReportBESA.error(ex);
                }
            }
        } catch (ExceptionBESA ex) {
            ReportBESA.error(ex);
        }
    }
}
```

FIGURA 4. EJEMPLO DE UNA FUNCIÓN *MAIN* TÍPICA PARA UNA APLICACIÓN BESA.

Fuente: elaboración propia.

5.2. *Contenedor*

Antes de crear el contenedor se debe ejecutar la herramienta de registro RMI (*rmiregistry*) en el puerto especificado en la propiedad al inicializar el contenedor BAB*Locator* para establecer un entorno de interoperabilidad entre dos barrios y BAP que corresponde a cada barrio. En la tecnología de JAVA, la herramienta provee servicios de nombramiento indirecto, como la invocación remota de objetos, al permitir obtener referencias a estos por medio de directorios. Para la ejecución del contenedor BESA se debe definir un conjunto de propiedades que establezcan los parámetros de ejecución que están en un archivo en el lenguaje de marcas extendido (XML). Como mínimo se deben definir las propiedades en el contenedor: alias, clave para interactuar con el contenedor, dirección IP; en ambiente de ejecución: intentos de reenvío de un evento y tiempo de espera de confirmación de llegada de un evento; en interacción: puerto RMI, dirección IP y puerto *multicast*, dirección IP y puerto del *baplocator*, puerto del BAP del barrio y, finalmente, puerto del BPO. Todos los contenedores que tengan la misma dirección y puerto *multicast* se encuentran en el mismo barrio y, por lo tanto, tienen interacción con un BAP que se define por medio de un puerto BAP; cuando el contenedor y sus agentes necesiten utilizar e interactuar con contenedores y agentes de otros barrios se utiliza la dirección IP y puerto del BAPLocator.

Para lanzar una aplicación, se debe definir un archivo de XML con varias marcas como ya se mencionó: asociadas a la marca *container* son: *alias*, determina el nombre con que se conoce el contenedor en el sistema; *ipaddress*, determina la dirección de red de la máquina donde se ejecuta el contenedor y; *password*, determina la clave que se utiliza para aceptar mensajes dirigidos al contenedor. Asociadas a la marca *environment* son: *rmiport*, determina el puerto del componente del directorio RMI para elaborar invocaciones remotas; *mcaddress* y *mcport*, que corresponden a la dirección IP; puerto *multicast*, una dirección de red clase D y puerto que es la misma para el grupo de equipos que conforman el sistema; *baplocator* y *bloport*, que corresponden a la dirección IP; puerto del componente, que permite la comunicación e interacción entre contenedores y agentes de diferentes barrios; *bapport*, que corresponde al puerto donde se encuentra el componente del BAP que refiere al manejo del barrio; por último *bpoport*, que corresponde al puerto del BPO que hace referencia al puerto del componente que permite interacción con contenedores compactos. Después de especificar las propiedades, sus nombres se utilizan por medio del archivo de extensión XML en la clase donde se tiene el método principal (*main*) y donde se va a crear una instancia del contenedor. Un ejemplo de la ejecución para inicializar el contenedor BESA basándose en los nombres de las etiquetas anteriores en un archivo XML se muestra en la figura 5.

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
    <container alias = "MAS_1" password = "0.91" ipaddress= "127.0.0.1">
        <environment seneventattemps = "10" sendeventtimeout = "1" rmitimeout = "1000">
            <interop rmiport = "1098" mcaddress = "230.0.0.1" mcport = "2222" bapport =
            "7000" bpoport = "7005" bloport="8081" baplocatoradd = "127.0.0.1"/>
        </environment>
    </container>
</config>
```

FIGURA 5. ARCHIVO XML PARA LA EJECUCIÓN DE UN CONTENEDOR BESA.

Fuente: elaboración propia.

Para crear una instancia del contenedor se hace referencia al archivo de marcas y a una clave, la cual es un número que se utiliza para cambiar el ciclo de vida del contenedor. Un ejemplo para instanciar el contenedor se puede apreciar en la figura 4. Después de crear el nuevo contenedor, específicamente un administrador local, automáticamente se registra en los otros contenedores. Para sincronizar los directorios, todos los contenedores que conforman el sistema proveen información de sus agentes locales al nuevo contenedor.

5.3. *Agentes*

Para la construcción de un agente es necesario especificar el estado y la estructura donde se relacionan las guardas, el comportamiento se declara de forma implícita. Especificar estos elementos, extendiendo las clases adecuadas, es suficiente para crear un agente que hace parte de un sistema con múltiples contenedores interconectados. A continuación se describe un ejemplo muy sencillo de un agente A que envía una cadena de texto con un mensaje que es impreso por un agente B, que imprime el mensaje enviado por el agente A; además, se ilustra cómo construir el código del estado, la estructura y la creación de un agente.

Estado del agente: para especificar el estado de un agente en BESA es necesario extender la clase StateBESA y especificar el método initState, que inicializa el estado del agente. Este método recibe como parámetro un ArrayList, donde se pueden incluir todos los atributos, propiedades u otros objetos que componen el estado del agente. Además, debe implementar la interfaz *serializable*, pues en el momento en que un agente se mueve a otro contenedor, se debe llevar el estado interno del agente. Este proceso se ilustra en la figura 6.

```
package OdoAgent;

import BESA.Kernel.Agent.StateBESA;
import java.util.ArrayList;

public class StateOdo extends StateBESA {

/**
Enumeration that indicates the posible states of the odometer.
*/
public enum State {
Init, Start, Stop
}
/**
Current odometer state.
*/
private State state;
/**
Initializes into initState method using the profile.
*/
private int step;
/**
Counter of the odometer.
*/
private CounterOdo counter;

/** Creates a new instance
*/
public StateOdo() {
super();
}

/** Changues the states
*/
public void changueState(State state) {
this.state = state;
}

/** Initializes the state
*/
public synchronized void initState(ArrayList profile) {
step = ((Integer) profile.get(0)).intValue();
}

/** Increments the counter*/
public synchronized void incrementCounter() {
counter.addValue(step);
update();
```

```
}

/** Decrements the counter*/
public synchronized void decrementCounter() {
counter.addValue(-1 * step);
update();
}

/** Gets the step*/
public synchronized int getStep() {
return step;
}

/**
Sets step
*/
public synchronized void setStep(int step) {
this.step = step;
}

/**
*/
public CounterOdo getCounter() {
return counter;
}

/** Sets the counter
*/
public void setCounter(CounterOdo counter) {
this.counter = counter;
}

/** Sets the state
*/
public void setState(State state) {
this.state = state;
}

/** Gets state
*/
public State getState() {
return state;
}
}
```

FIGURA 6. CONSTRUCCIÓN DE UN ESTADO EN BESA.

Fuente: elaboración propia.

En la implementación del estado del agente *odómetro* se tiene un objeto *CounterOdo*, que también debe ser *serializable*, debido a que en el caso de movilidad todo lo que compone el estado debe enviarse al contenedor destino. *CounterOdo* tiene métodos para modificar el contador como *incrementCounter* y *decrementCounter*.

Comportamiento del agente: cada comportamiento del agente corresponde a un hilo concurrente donde se ejecutan las guardas que un agente puede procesar. Como se mencionó en la versión 3.x de BESA, esta creación se hace implícita.

Guardas de respuesta a eventos: después se definen las guardas que representan la acción que va a emprender el agente como respuesta a un evento. En la Figura 7 se describe el código de las guardas *AgentAGuard*. En el ejemplo de guarda, el código se ejecuta después de la evaluación de una condición booleana. La evaluación de la condición retorna dos posibles valores, *true* o *false*. Cuando la evaluación de la condición sea *true*, se ejecuta el código especificado en el método *funcExecGuard*.

El método de evaluación booleana se define en un método con nombre *funcEvalBool* que puede recibir cualquier objeto como parámetro para la evaluación. Por otra parte, el código de ejecución determinado en el método *funcExecGuard* puede recibir datos que se envían como parte del evento recibido, cuyo tratamiento se hace en este método. La condición en el ejemplo dispara la guarda siempre, debido a que no se recibe un evento asociado a esta guarda y no está condicionado a la ejecución del código. Al ejecutar el código asociado en el caso del *AgentA* se envía un evento al *AgentB* con el mensaje "Hello [0]". En el ejemplo de guarda (figura 7) no se requiere la verificación de la condición, por lo tanto, no se especifica el método de *funcEvalBool*. De esta forma, se utiliza el especificado en la clase de la cual hereda la guarda del *Agente A*, *AgentAGuard*, el cual siempre retorna verdadero.

```
/*
package agenta;

import BESA.Kernel.Agent.Event.EventBESA;
import BESA.Kernel.Agent.GuardBESA;
import BESA.Log.ReportBESA;
import agentb.AgentBGuard;
import java.util.logging.Level;
import java.util.logging.Logger;

public class AgentAGuard extends GuardBESA {

    public void funcExecGuard(EventBESA event) {
        try {
```

```
            this.agent.getAdmLocal().getHandlerByAlias("AgentB").sendEvent(new
            EventBESA(AgentBGuard.class.getName(), event.getData()));
              ReportBESA.info("The agent A is saying hello to extent agent B...");
          } catch (Exception ex) {
              Logger.getLogger(AgentAGuard.class.getName()).log(Level.SEVERE, null, ex);
          }
      }
  }
```

FIGURA 7. CONSTRUCCIÓN DE GUARDAS EN BESA SIN EVALUACIÓN BOOLEANA.

Fuente: elaboración propia.

Cabe destacar, como se explica a continuación, que para establecer la relación de guardas con comportamientos no se realiza desde un código en las guardas; en cambio, actualmente se ha creado una nueva clase llamada estructura, que provee implícitamente la relación guardas-comportamientos.

Estructura del agente: después de determinar estado y guardas, se especifica la estructura del agente. Esta determina la relación entre el agente y las guardas que lo conforman, así como los eventos que son procesados por cada uno de estos comportamientos. Después de crear una instancia de la clase *StructBESA*, se procede a utilizar el método *bindGuard*, para crear la relación guarda y agente. Con el método *bindGuard* se indica el nombre de la clase que especifica la guarda asociada al evento que se va a procesar por el hilo del comportamiento. En la figura 4 se demostró un ejemplo para construir la estructura del agente.

Construcción del agente: para la construcción de un agente se debe crear una clase que se extienda de *AgentBESA*. La clase debe especificar los métodos *setupAgent* y *shutdownAgent*. Estos métodos incluyen el código que se va a ejecutar al crear y destruir el agente, respectivamente. En estos métodos generalmente se debe especificar la reserva y liberación de recursos compartidos por varios comportamientos; por ejemplo, al crear el agente se puede especificar que debe abrir un archivo de texto a partir del cual podrá realizar sus tareas. Al momento de destruir el agente, se debe cerrar el archivo para liberar el recurso. En la figura 8 se ilustra la estructura general de la clase *AgentA*, para el ejemplo de enviar un mensaje entre dos agentes en barrios distintos.

```
/*
package agenta;
import BESA.Kernel.Agent.AgentBESA;
import BESA.Kernel.Agent.KernellAgentExceptionBESA;
import BESA.Kernel.Agent.StateBESA;
import BESA.Kernel.Agent.StructBESA;
```

```
public class AgentA extends AgentBESA {

    public AgentA(String alias, StateBESA state, StructBESA structAgent, double
    passwd) throws KernellAgentExceptionBESA {
        super(alias, state, structAgent, passwd);
    }

    public void setupAgent() {
    }

    public void shutdownAgent() {
    }
}
```

FIGURA 8. ESTRUCTURA DE UNA CLASE DE AGENTE EN BESA.

Fuente: elaboración propia.

Por último, para invocar la constructora del agente, se deben proporcionar como parámetros: el alias, el estado, la estructura y la clave. En la figura 4 se ilustró la forma en la que se debe crear un agente. Los parámetros para crear un agente se refieren a un alias, que corresponde al nombre lógico del agente; una clave, que se utiliza para cambiar el ciclo de vida del agente y para moverlo a otro contenedor; el estado y; la estructura el agente.

Envío de eventos

Para la comunicación entre agentes se utilizan eventos. Igualmente, para enviar este acto de lenguaje es necesario tener una referencia al agente destino y construir el evento, como se muestra en la figura 9. Para la construcción del evento se tiene la opción de enviar una estructura compleja de datos que se basa en una clase que provee el *framework* llamada *DataBESA*. En el uso de esta clase se obliga al programador a crear una clase que se extienda de esta y a escribir como mínimo el método *strToDataBESA*. Este método se utiliza para hacer impresiones por pantalla de los datos que se agrupan en esta estructura.

```
SayHelloData sayHelloData = new SayHelloData("Hello[" + index + "]");
admLocal.getHandlerByAlias("AgentA").sendEvent(new EventBESA(AgentAGuard.class.getName(),
sayHelloData));
```

FIGURA 9. ENVÍO DE EVENTOS EN BESA.

Fuente: elaboración propia.

Para especificar el agente destinatario se debe buscar en los directorios del administrador local la manija (*handler*) correspondiente, utilizando, por ejemplo, el alias. Después se debe crear el evento, donde el nombre corresponde al nombre de la guarda que se quiere ejecutar.

5.5. *Eventos entre contenedores ligeros y estáticos*

De forma muy sencilla se pueden comunicar agentes entre contenedores estándar y compactos. El código de la figura 10 muestra la interacción entre el componente BPO y el *Agente PostMan*. El BESA *PostOffice* (BPO) es un componente que hace parte de un contenedor BESA que tiene por objetivo principal la gestión del intercambio de mensajes con otros contenedores del mismo barrio por medio de un protocolo de comunicación. Por otra parte, es el encargado de interpretar los mensajes recibidos y ejecutar las acciones adecuadas. En el contexto móvil, el Contenedor Compacto y el Contenedor Estándar deben tener en sus respectivos contenedores componentes BPO que se comunican entre sí.

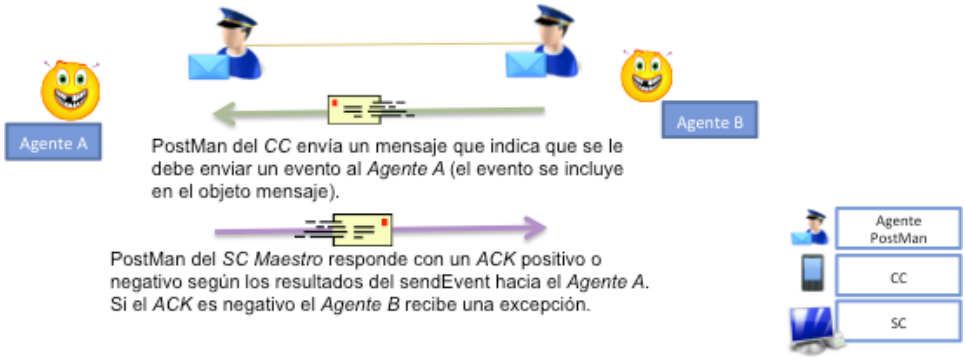


FIGURA 10. ENVÍO DE EVENTOS EN BESA ENTRE CONTENEDORES COMPACTOS Y ESTÁTICOS.

Fuente: elaboración propia.

El agente BESA *PostMan* que hace parte del BPO es el agente que tiene como objetivo elaborar el manejo de la comunicación entre dos contenedores específicos. Este agente debe mantener las colas de mensajes y administrar el canal de comunicación. Por cada contenedor compacto debe tener mínimo un agente *PostMan* que se encarga del manejo de la cola de mensajes y comunicación con un contenedor estándar llamado SC *Maestro*. En cambio, un contenedor estándar puede tener varios *PostMan* dependiendo de los contenedores compactos que lo tienen como maestro.

Por último, el *AgHanderBPO* que corresponde a un *handler* BESA especial, se encarga de transferir un evento dirigido a un agente que no es local por medio del *PostMan* adecuado. En el caso de encontrarse en un contenedor compacto, este *handler* será asociado con los agentes que no se encuentran de manera local. En el caso de un contenedor estándar será asociado a los agentes que se encuentran en algún contenedor compacto.

La arquitectura BESA presentada en este capítulo proporciona todas las funcionalidades para la creación de los agentes requeridos para el desarrollo de los diferentes componentes de AYLLU que han sido introducidos en los capítulos anteriores. Para cada uno de los agentes requeridos se han creado clases derivadas, acordes con el modelo de implementación presentado en esta última sección. El sistema resultante es modular y permite realizar una validación e integración incremental a nivel de cada agente y de cada uno de los protocolos de interacción utilizados.

Referencias

Bernstein, A. J. (1980). Output Guards and Nondeterminism in Communicating Sequential Processes. *ACM Trans. Program. Lang. Syst*., *2*(2), 234.

Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, *2*(1), 14-23.

Ferber, J. (1999). An Introduction to Distributed Artificial Intelligence. *MultiAgent Systems.* Longman: Addison Wesley.

Finin, T., Fritzson, R., McKay, D. y McEntire, R. (1994). KQML as an Agent Communication Language. *Proceedings of the Third International Conference on Information and Knowledge Management - CIKM '94*, 456.

FIPA. (2017).The Foundation for Intelligence Physical Agents. Página principal. Disponible en http://www.fipa.org. Consultado en julio de 2017.

Foster, I. (1995). *Designing and Building Parallel Programs.* Addison Wesley.

Labidi, S. y Lejouad, W. (1993). *De l'Intelligence Artificielle Distribuèe aux Systémes Multi-Agents.*

Labrou, Y. y Finin, T. (1999). The Current Landscape - *Agent Communication Languages.* IEEE Intelligent Systems.

Minsky, M. (1986). *The Society of Mind.* Touchstone Book.

Rueda, S. (1996). *Agentes y Sistemas multiagente.* Bogotá: Universidad de Los Andes-CIFI.

Rusell, N. (2009). *Inteligencia artificial: un enfoque moderno.* Prentice Hall.

13
CASO DE ESTUDIO DE AYLLU-EDU EN CONTEXTOS EDUCATIVOS

Ángela Carrillo-Ramos, Enrique González, Demetrio Ovalle

El objetivo de este capítulo es presentar la validación de la herramienta desarrollada en el proyecto AYLLU mediante su aplicación a un caso de estudio, en el contexto educativo, a través de dos escenarios de prueba: el curso de Inteligencia Artificial en la Universidad Nacional de Colombia - Sede Medellín y el de Sistemas Inteligentes en la Pontificia Universidad Javeriana de Bogotá. Se utilizó la estrategia colaborativa JIGSAW, la cual se desarrolló en dos fases principales: a) la conformación de equipos base y b) la conformación de equipos temáticos. La clave para la conformación de equipos está en saber conformarlos, ya sea homogéneos u heterogéneos, y asignarles los roles más acordes con su perfil, el cual está compuesto de competencias, conocimientos y habilidades de cada estudiante. La herramienta MATEO, que hace parte del proyecto AYLLU, realizó una apropiada conformación de equipos, lo cual se confirmó a través del grado de satisfacción expresado por los estudiantes y el profesor en diferentes ítems de la encuesta aplicada. Después de realizadas las pruebas y del análisis de resultados se puede concluir que se lograron los siguientes aspectos clave en aprendizaje colaborativo: (1) interdependencia positiva; (2) responsabilidad individual; (3) se promovió la interacción cara a cara. En efecto, la herramienta AYLLU propició diversos mecanismos de interacción (chat, correos, foros, etc.) facilitando la actividad del equipo; (4) se propició el uso de habilidades colaborativas y, finalmente, (5) se incentivó el mejoramiento continuo grupal.

1. Introducción

Para validar AYLLU se establecieron dos escenarios de pruebas: dos cursos de Inteligencia Artificial, uno en una universidad pública y otro en una universidad

privada. Al finalizar la experiencia, se realizó una encuesta a los estudiantes que participaron en el piloto AYLLU (Arias-Báez *et al*., 2012). Adicionalmente, se realizó la misma encuesta al profesor de la clase para observar desde su punto de vista la dinámica de la clase. En las pruebas se implementó la estrategia cooperativa JIGSAW (JIGSAW, s. f.), la cual se desarrolló en dos fases principales: a) en la conformación de equipos base: primera reunión de estudiantes, en la que se conformaron equipos de cuatro personas. A un estudiante de cada equipo se le asignó el rol de coordinador (aquel que coincidiera mayormente con las características de coordinador definidas en MATEO (Arias-Báez *et al*., 2013); y b) en la conformación de equipos temáticos: reunión de estudiantes por subtema, donde se conformaron cuatro equipos de tres personas. En cada equipo, a un estudiante se le asignó el rol de secretario (aquel que coincidiera mayormente con las características de secretario definidas en MATEO). Se realizó una encuesta tanto a los estudiantes como a los profesores con el fin de analizar la participación y grado de satisfacción al utilizar AYLLU, revisando específicamente aspectos relacionados con colaboración, trabajo en equipo, asignación de roles y mejora en su proceso de aprendizaje.

La estructura de este capítulo es la siguiente: la sección 2 presenta el proceso general del piloto de AYLLU (Arias-Báez *et al*., 2013) en el que se siguió como estrategia cooperativa JIGSAW (JIGSAW, s. f.). Luego, la sección 3 describe en detalle cada etapa del proceso seguido en la aplicación del mismo en la Pontificia Universidad Javeriana (PUJ). En la sección 4 se muestran las particularidades del piloto en la Universidad Nacional de Colombia - Sede Medellín (UNAL Medellín). En la sección 5 se presentan los resultados del preexperimento de ambos cursos, se concluye y se presentan las perspectivas de trabajo futuro en la sección 6.

2. Descripción general de la estrategia JIGSAW seguida

Para la prueba se utilizó el contexto de los cursos de Sistemas Inteligentes e Inteligencia Artificial que se ofrecían simultáneamente en las universidades participantes. En la UNAL Medellín este curso estuvo dirigido a estudiantes de pregrado, mientras en la PUJ, a estudiantes de maestría. En particular, el piloto se aplicó al módulo de Lógica Difusa. En ambos escenarios, antes de iniciar la prueba piloto, se les solicitó a los estudiantes diligenciar tres *tests*: uno en que se les preguntaba los datos básicos del estudiante, otro que definía el perfil del estudiante con base en la teoría de los colores de Hermann (Herrmann-Nehdi, 2015) y el último, que evaluaba las características de colaboración (Arias-Báez *et al*., 2012).

Para completar esta etapa de preparación antes del inicio de la aplicación de JIGSAW, se realizó una clase introductoria al tema de Lógica Difusa. En esta sesión presencial, además de la presentación del profesor, se hizo una introducción de los Objetos Virtuales de Aprendizaje (OVA) a los estudiantes sobre cuatro temas de conocimiento básico: a) Naturaleza de lo Difuso; b) Conjuntos Difusos; c) Variables Lingüísticas; y d) Operadores Difusos e Inferencia Básica. Adicionalmente, se les sugirió el artículo clásico introductorio a estos temas de John Yen (1999).

Después de la adquisición de la información que permitió alimentar los perfiles y de la introducción a la temática, se crearon los equipos base con un coordinador y tres aprendices. Se definieron y asignaron cuatro temas de investigación por subasta a todos y cada uno de los estudiantes; en cada equipo hay un estudiante asociado a cada temática. Con el fin de orientar el proceso de investigación individual, se le propuso un OVA personalizado a cada estudiante, tomando en cuenta sus características y competencias reconocidas y registradas en los perfiles. Previamente, se prepararon al menos dos OVA para cada tema de investigación, cada uno con características diferentes (por ejemplo, más o menos visuales).

Posteriormente, se generaron nuevos equipos por temas, un equipo por tema; en cada equipo se nombró un secretario temático. En los cuatro equipos generados por temáticas, la cantidad de estudiantes dependió del número de equipos base que había al inicio; el curso de UNAL Medellín tenía más estudiantes, entonces se requirieron más equipos base que en la PUJ. Luego, al interior de cada grupo temático y con el apoyo de las plataformas *e-learning* disponibles en las universidades, se generó una discusión con el fin de formular preguntas motivadoras, una por cada estudiante. El profesor hizo un refinamiento de dichas preguntas y asignó una calificación a esta actividad. Una vez depuradas las preguntas, estas se asignaron a través de subasta a los estudiantes del grupo temático. Cada estudiante trabajó sobre la pregunta asignada y generó un informe individual por tema-pregunta. A partir de este aporte individual, a nivel del equipo temático se realizó una compilación para producir el informe final temático; el resultado de estas actividades permitió tener cuatro informes compilados, uno por cada tema; estos informes fueron distribuidos a todos los estudiantes. En la siguiente etapa del proceso, los estudiantes regresaron a trabajar con su equipo base, de tal forma que en cada equipo base se disponía de un especialista en cada tema. En cada equipo base, a partir de los informes y del aporte del especialista, se compartió el conocimiento en cada tema a través de un foro de discusión (uno por cada equipo base); en este proceso de compartir siempre se contó con la mediación por parte del coordinador del equipo. Finalmente, se procedió a la evaluación presencial, en

la cual dos estudiantes escogidos al azar de cada equipo base respondían preguntas sobre los cuatro temas. Teniendo en cuenta las respuestas de los estudiantes seleccionados se le asignó una nota a todo el equipo.

3. Desarrollo del piloto en la Pontificia Universidad Javeriana

Esta sección se dedica a describir de manera detallada cada etapa del proceso seguido para emplear JIGSAW en el piloto de AYLLU en su aplicación en la PUJ. En este escenario participaron 12 estudiantes, que dieron lugar a la conformación de 3 equipos base de 4 estudiantes. Para apoyar las interacciones entre los estudiante, y las interacciones de estos con el profesor, se utilizó la plataforma Blackboard.

3.1. *JIGSAW – etapa 1*

En la etapa 1 se crearon los equipos base y se asignaron los temas ejecutando MATEO (ver capítulos 8 y 9). En cada equipo base se designó un coordinador. Se generó un foro de discusión por cada equipo base y el primer punto a discutir fue la asignación de temas por subasta. Teniendo en cuenta que los equipos base eran de 4 integrantes, a cada uno se le asignó un tema de investigación diferente y se seleccionaron los siguientes temas de Lógica Difusa:

Tema 1: Modelo Matemático Formal de la Lógica Difusa - Inferencia Difusa
Tema 2: Fuzzy Mappings y Control Difuso - Mamdani + Takagi-Sugeno
Tema 3: Identificación Fuzzy - Particiones + Dimensionalidad
Tema 4: Aprendizaje Fuzzy - NeuroFuzzy + Fuzzy Genético

Las ideas fundamentales de estos temas podían encontrarse en el artículo de referencia de John Yen (1999): el tema 1 se desarrolla con cierto detalle en las secciones 3 y 4; el tema 2 toma algunos fundamentos del final de la sección 4, y se desarrolla en forma general en las secciones 5 y 6; el tema 3 se presenta en forma general en la sección 7; y el tema 4 se introduce brevemente en la subsección 2.1. En esta primera etapa, el segundo paso fue la asignación de temas mediante un esquema de Subasta de Temas. Los integrantes del grupo escribían en el foro el tema que deseaban exponer. El coordinador esperaba que todos los grupos respondieran y confirmaba la asignación o resolvía los conflictos que aparecían. A través de un mensaje en el foro de discusión, el coordinador confirmaba a todos la asignación definitiva de los temas.

3.2. *JIGSAW – etapa 2*

La segunda etapa correspondió a la Generación de Preguntas Motivadoras. Una vez se organizaron los cuatro equipos temáticos, en cada uno de ellos se designó un secretario. Cada equipo temático incluye un integrante de cada uno de los equipos base acorde con la asignación previa de los temas. Para cada tema el equipo formuló las preguntas motivadoras alrededor de las cuales se centraría la investigación y el desarrollo de las temáticas. En el caso de este curso piloto, cada equipo temático debía producir seis preguntas; por ejemplo, en el caso de la PUJ en el cual solo había tres grupos base, se formularon dos preguntas por estudiante. El secretario debía compilar las preguntas en forma estructurada y someterlas a evaluación y ajuste por parte del profesor. Esta entrega generó una primera calificación a ponderar para la obtención de la nota final del módulo de Lógica Difusa.

La construcción del conjunto de preguntas generadoras se inició a partir de la identificación de los "subtemas claves"; estos subtemas debían ser aún generales, además, debían articularse de forma tal que cubrieran los aspectos fundamentales de la temática a desarrollar. Una vez identificados los subtemas, el secretario los repartía entre los miembros del grupo. Para cada subtema se hacía una pregunta general, cuya respuesta implicaba no solo compilar y presentar los conceptos base del subtema, sino también realizar un análisis riguroso (ventajas, desventajas, proyección, relación con otros conceptos, etc.). La respuesta a la pregunta motivadora implícitamente involucraba la respuesta a varias preguntas más específicas. Una vez se producía una primera versión de respuesta para una pregunta, lo más aconsejable era someterla a la crítica constructiva del grupo temático, para así refinarla y mejorarla, antes de incluirla en la compilación final que se sometía a consideración del profesor.

Para el proceso de formulación de las preguntas motivadoras, se siguieron estos pasos:

- Identificación y articulación de subtemas.
- Asignación de subtemas a los miembros del equipo temático.
- Discusión y refinamiento de las preguntas.
- Compilación y envío final de las preguntas para aprobación final del profesor.

El secretario debía estar pendiente del progreso de cada uno de los pasos y de la participación de cada uno de los miembros del grupo. Al final se debían generar un documento con las preguntas motivadoras definitivas. El profesor respondía a través de un mensaje sus comentarios y sugerencias acerca de las preguntas. Finalmente, el secretario debía responderle al profesor un mensaje con la versión final "mejorada".

3.3. *JIGSAW – etapa 3*

Esta etapa consistió en la Entrega de los Documentos Temáticos y la Discusión Final. Dicha etapa se inició con la entrega de los documentos que compilan los aportes de todos los integrantes a las preguntas motivadoras generadas en la etapa anterior. El documento debía articularse como si fuera un artículo, siguiendo un formato específico. Para cada temática se creaba una discusión en el foro, publicando el documento final y luego se iniciaba la discusión y la formulación de preguntas por parte de los equipos base. Para la discusión y preguntas por parte de los equipos base, se debía enviar un mensaje que reflejara el tema de discusión o pregunta. Luego, la respuesta debía ser enviada por el estudiante "especializado en la temática" del mismo equipo base de quien formuló la pregunta. Se esperaba que cada estudiante generase al menos dos preguntas o comentarios en temáticas diferentes. Gracias a este proceso, todos los estudiantes pudieron aprender sobre los temas en los cuales no eran especialistas, con la colaboración de sus compañeros.

Finalmente, se llevó a cabo la evaluación oral por parte del profesor utilizando el banco de preguntas generado a lo largo del proceso, que era complementado con otras preguntas elaboradas por el profesor. Para ello se escogieron dos estudiantes de cada equipo base; la evaluación de las respuestas de estos estudiantes aplicaba para todo el grupo base, lo que generaba un esquema de corresponsabilidad entre los integrantes del grupo.

4. Desarrollo del piloto en la Universidad Nacional de Colombia - Sede Medellín

En la Universidad Nacional de Colombia - Sede Medellín, Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión, se llevó a cabo el caso de estudio del piloto de AYLLU para un grupo de 49 estudiantes del curso de Inteligencia Artificial. Es importante señalar que antes de realizar la estrategia colaborativa fue realizada en sesiones presenciales una presentación inicial de conceptos básicos del Módulo de Lógica Difusa a cargo del profesor de la asignatura. A continuación, se describe el proceso utilizado bajo la estrategia JIGSAW. La metodología de trabajo se aplicó en forma similar a la PUJ, siguiendo las tres etapas descritas en la sección anterior.

En la etapa 1 se crearon los equipos base utilizando el módulo MATEO del sistema AYLLU y se obtuvieron 7 equipos, para los cuales se asignó un coordinador, quien a través de un foro de discusión con los demás integrantes del equipo procedió a la asignación de temas, mediante una estrategia de Subasta de Temas para cada uno

de ellos. Cabe señalar que para este escenario el número de temas se amplió a 7, según lo sugiere la estrategia JIGSAW, con el fin de tener un tema diferente por cada integrante. Los temas seleccionados fueron los siguientes:

Tema 1: Conceptos de aritmética difusa
Tema 2: Relaciones de implicación
Tema 3: SID-Mandani
Tema 4: SID-Takagi-Sugeno
Tema 5: Aplicación de LD: Fuzzy biometric system
Tema 6: Fuzzy Ascensores Bancolombia
Tema 7: Student Fuzzy Performance Assessment

Cada integrante de equipo base procedió a la lectura del tema asignado y como actividades siguientes, realizaron la preparación y la presentación a sus compañeros del Equipo Base del tema asignado. Para la etapa 2, inicialmente se reagruparon los estudiantes en Equipos Temáticos, según el número del tema asignado (1 a 7) y de esta forma se llevó a cabo el proceso de Generación de Preguntas Motivadoras a cargo de los equipos temáticos; una por cada integrante. El secretario de cada equipo compiló las preguntas en forma estructurada y las sometió a evaluación y ajuste por parte del profesor. Esta entrega generó una primera calificación a ponderar para la obtención de la nota final del módulo de Lógica Difusa. Cabe señalar que el secretario de cada equipo temático estuvo pendiente del progreso de cada uno de los pasos y de la participación de cada uno de los miembros del equipo para el proceso de construcción de las preguntas motivadoras que se definió en la sección anterior. Al final se generaron, en versión editable .doc, las preguntas motivadoras definitivas con sus correspondientes respuestas. Es así como se constituyó un banco de 49 preguntas con sus correspondientes respuestas filtradas, mejoradas y revisadas. Lo anterior se llevó a cabo satisfactoriamente debido a que el profesor interactuó con el secretario de cada equipo a través de mensajes con sus comentarios y sugerencias acerca de las preguntas. Es importante señalar que este banco de preguntas y respuestas realizado colaborativamente entre los estudiantes y el profesor puede ser utilizado como base para la evaluación oral del nivel de conocimientos después de aplicada la estrategia colaborativa implementada a través de AYLLU.

Por su parte, la etapa 3 consistió en la elaboración de un Informe por cada Equipo Temático y su socialización a todos los Equipos Base conformados. Luego se realizó una discusión a través de un foro en la plataforma Moodle por cada uno de los Equipos Base. Para ello se procedió a la elaboración de preguntas por parte de los integrantes del equipo; en ese momento se envió un mensaje reflejando el tema de

discusión o inquietud. Luego, la respuesta a la pregunta o inquietud correspondiente fue enviada por el estudiante "especializado en la temática" del mismo Equipo Base a quien realizó la pregunta. Se sugirió que cada estudiante generara al menos dos preguntas o comentarios en temáticas diferentes.

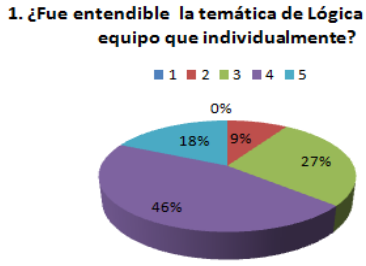
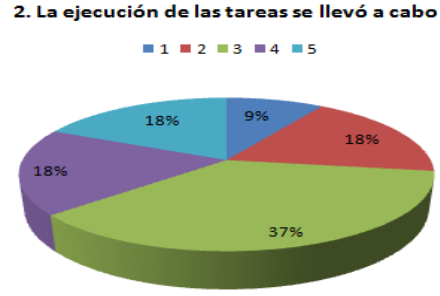
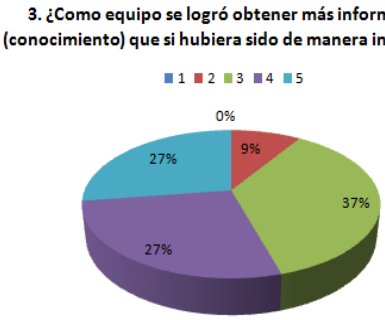
Finalmente, se llevó a cabo una evaluación oral por parte del profesor utilizando el banco de preguntas que se había generado, así como otras preguntas elaboradas por el profesor. Para ello se escogieron dos estudiantes de cada Equipo Base y se les formularon 3 preguntas, para las que tenían la posibilidad de apoyarse el uno con el otro. La nota que obtuvieron en esta evaluación oral se asignó a todo el Equipo Base. Es importante señalar que debido a la característica de repartición de temas y reagrupación temática posterior a la estrategia colaborativa JIGSAW, así como a la responsabilidad que conlleva la asignación de notas grupales, se puede concluir por parte del profesor que la adquisición de conocimientos y el desempeño de los estudiantes durante la fase de evaluación fueron satisfactorios. Finalmente, y teniendo en cuenta los informes temáticos realizados, los cuales fueron de buena calidad, muy completos, bien redactados y pertinentes para la temática abordada, se puede concluir que hubo buena repartición de roles y bastante coordinación de los integrantes dentro de los Equipos Temáticos. De esta forma, se alcanzaron las metas establecidas y, por lo tanto, se logró el aprendizaje a través de una estrategia colaborativa.

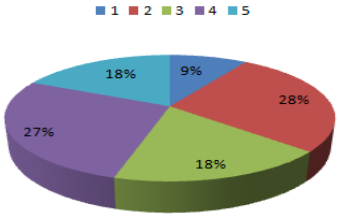
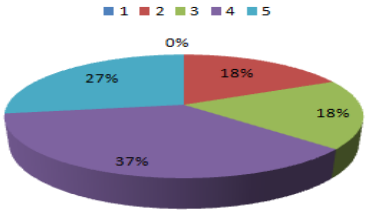
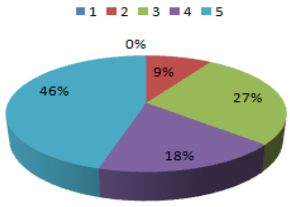
5. Resultados del preexperimento

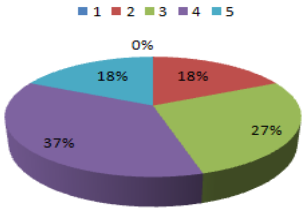
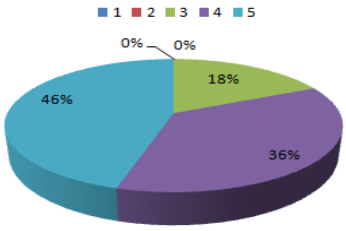
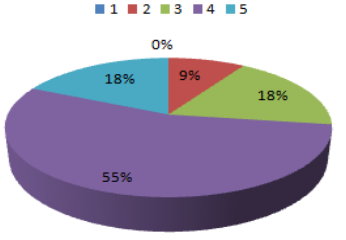
Los preexperimentos se definen como un proceso en el que se "administra un estímulo o tratamiento a un grupo y después se aplica una medición de una o más variables para observar cuál es el nivel del grupo en estas" (Hernández Sampieri *et al*., 2010). De esta forma, se realizó el estudio de un caso con una sola medición, ya que no se requirió de una referencia previa sobre cuál era el nivel que tenía el grupo de estudio (antes del estímulo) en las variables que se evaluaron. Esta simplificación del análisis es válida debido a que casi ninguno de los estudiantes participantes había tenido conocimiento previo de las temáticas. Se tomó este tipo de experimento como un primer acercamiento al problema de investigación en la realidad. Para esto, se realizó una encuesta a los estudiantes que participaron en el piloto de AYLLU. También se realizó la misma encuesta al profesor de la clase para conocer su punto de vista.

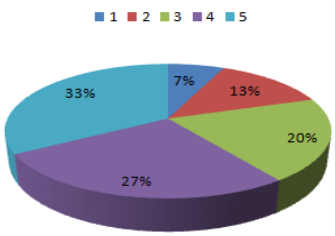
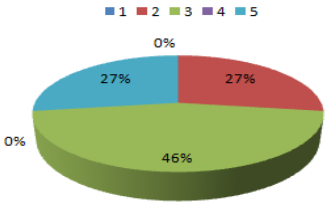
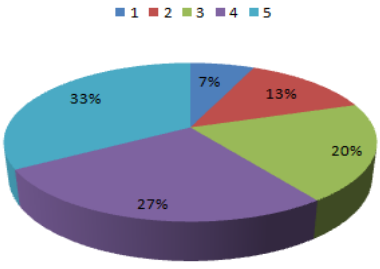
En la tabla 1 los resultados de cada una de las preguntas realizadas al profesor y a sus estudiantes dentro de la encuesta para evaluar el piloto de AYLLU y su impacto en aspectos de colaboración.

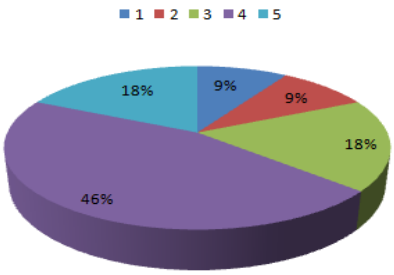
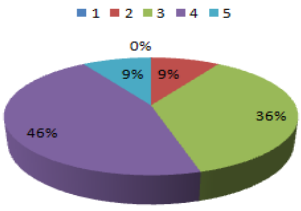
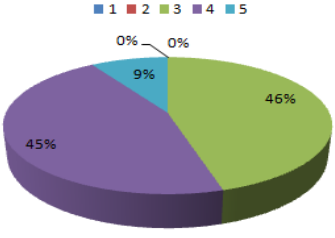
TABLA 1. RESULTADOS DE LA PRUEBA PILOTO

 Escala: (1) poco entendible, (5) muy entendible.	Se observa que la mayoría de los estudiantes (36 %) considera que es mucho más entendible la temática de clase cuando se trabaja en equipo. A la misma pregunta, el profesor respondió 4 en la escala, lo que llevaría a pensar que, por sus observaciones de todos los equipos de su clase, considera que el aprendizaje mayormente en equipo (no totalmente) permite un mejor entendimiento del tema. De esta manera estudiantes y profesor coinciden en la importancia del aprendizaje mayormente en equipo para el mejor entendimiento de un tema.
 Escala: (1) de manera totalmente individual, (5) de manera totalmente conjunta.	Se observa que la mayoría de los estudiantes (37 %) considera que la ejecución de tareas se llevó a cabo tanto de manera individual como conjunta. A la misma pregunta, el profesor respondió 4 en la escala, es decir, por sus observaciones de todos los equipos de su clase, considera que los estudiantes llevaron a cabo la ejecución de las tareas de manera conjunta mayormente (no totalmente). De esta manera se puede concluir que, aunque los estudiantes consideran que la ejecución de tareas se llevó a cabo de manera individual y en equipo, lo que percibió el profesor fue un trabajo mayormente colaborativo entre ellos.
 Escala: (1) no se logró en equipo, (5) sí se logró en equipo.	Se observa que la mayoría de los estudiantes (37 %) considera que se logró obtener información tanto de manera individual como conjunta. A la misma pregunta, el profesor respondió 3 en la escala, que tiene que ver con el tipo de tareas que asignó a sus estudiantes, tanto de tipo individual como en equipo, por lo que se obtuvo información de ambas maneras. De esta forma, estudiantes y profesor coinciden en que se logró obtener información tanto de manera individual como conjunta.

4. ¿Considera que la toma de decisiones fue hecha en equipo?  Escala: (1) totalmente en equipo, (5) totalmente individual.	Se observa que la mayoría de los estudiantes (27 %) considera que las decisiones se tomaron mayormente en equipo. A la misma pregunta, el profesor respondió 5 en la escala, lo que significa que observó que, de manera general en su clase, las decisiones se tomaron totalmente en equipo. En conclusión, se observa que efectivamente las decisiones fueron tomadas en equipo.
5. ¿Qué tan satisfecho se sintió con las decisiones tomadas en equipo?  Escala: (1) poco satisfecho, (5) totalmente satisfecho.	Se observa que la mayoría de los estudiantes (37 %) se encuentra satisfecho con las decisiones tomadas en equipo. A la misma pregunta, el profesor respondió 3, lo que significa que el profesor no quedó ni satisfecho ni insatisfecho con la toma de decisiones de los equipos. Con esta última respuesta no se puede concluir el grado de satisfacción de todos los usuarios.
6. ¿Qué tanta coordinación considera que hubo entre los miembros del equipo para realizar las tareas con el fin de alcanzar las metas establecidas?  Escala: (1) poca coordinación, (5) bastante coordinación.	Se observa que la mayoría de los estudiantes (46 %) considera que hubo bastante coordinación entre los miembros de su equipo. A la misma pregunta, el profesor respondió 5 en la escala, es decir, también considera que, de manera general, hubo bastante coordinación dentro de los equipos. Se puede concluir que, tanto estudiantes como profesor coinciden en un alto grado de coordinación que se tuvo para realizar las tareas con el fin de alcanzar las metas establecidas.

7. ¿Todas las personas trabajaron de la misma manera equilibrada y equitativa, de acuerdo a su rol, dentro del equipo?  Escala: (1) no, no todas las personas, (5) sí, todas las personas.	Se observa que la mayor parte de los estudiantes (37 %) considera que la mayoría de las personas trabajaron de manera equilibrada y equitativa de acuerdo con su rol. A la misma pregunta, el profesor respondió 4, lo que significa que el profesor observó lo mismo que sus estudiantes. Se puede concluir que, mayormente, hubo una asignación de trabajo equilibrado y equitativo al interior de los equipos.
8. ¿Las responsabilidades de cada uno de los miembros del equipo estaban bien definidas y delimitadas?  Escala: (1) no, (5) sí.	Se observa que la mayoría de los estudiantes (46%) considera que las responsabilidades fueron totalmente definidas y delimitadas. A la misma pregunta, el profesor respondió 5 en la escala, es decir, también considera que las responsabilidades fueron totalmente definidas y delimitadas. De esta manera se puede concluir que, tanto estudiantes como profesor coinciden en que las responsabilidades fueron totalmente bien definidas y delimitadas.
9. ¿Se lograron alcanzar las metas establecidas con los recursos asignados? (tiempo, personas, materiales, etc.)  Escala: (1) no se lograron totalmente, (5) sí se lograron totalmente.	Se observa que la mayoría de los estudiantes (55%) considera que lograron alcanzar mayormente las metas establecidas con los recursos asignados. A la misma pregunta, el profesor respondió 3 en la escala, es decir, se alcanzaron aceptablemente las metas establecidas, no mayormente. Se puede concluir que, aunque se hayan logrado las metas establecidas, las perspectivas de profesor y estudiantes discrepan.

10. ¿El trabajo realizado por cada persona contribuyó al trabajo del equipo?  Escala: (1) no contribuyó, (5) contribuyó bastante.	Se observa que la mayoría de los estudiantes (33 %) considera que el trabajo realizado por cada persona contribuyó bastante al trabajo en equipo. A la misma pregunta, el profesor respondió 4 en la escala, lo que significa que considera que el trabajo realizado por cada persona contribuyó al trabajo en equipo. De esta manera se observa que efectivamente el trabajo realizado por cada persona contribuyó al trabajo en equipo.
11. ¿Qué tanta participación (entendida como opinar, debatir y participar ecuánimamente) hubo dentro de su equipo de trabajo?  Escala: (1) muy poca participación, (5) bastante participación.	Se observa que la mayoría de los estudiantes (46 %) considera que hubo participación, pero su grado no fue significativo. A la misma pregunta, el profesor respondió 5, lo que significa que el profesor considera que hubo bastante participación de parte de los equipos de trabajo. De esta manera se observa que efectivamente hubo participación; sin embargo, estudiantes y profesor discrepan en el grado en que se dio.
12. ¿El rol que le fue asignado concuerda con su perfil?  Escala: (1) poco, (5) bastante.	Se observa que la mayoría de los estudiantes (33 %) considera que el rol que le fue asignado concuerda bastante con su perfil. A la misma pregunta, el profesor respondió 4, lo que significa que el profesor considera que, de manera general, los roles asignados concuerdan con los perfiles de los estudiantes. Así se observa que efectivamente se asignaron los roles de acuerdo con el perfil de los participantes.

13. ¿Se sintió a gusto con el rol que le fue asignado?  Escala: (1) poco a gusto, (5) muy a gusto.	Se observa que la mayoría de los estudiantes (46 %) se sintió a gusto con el rol asignado. A la misma pregunta, el profesor manifiesta que quedó satisfecho con la asignación de los roles "secretario" y "coordinador" en la mayoría de los equipos. Por tanto, se puede deducir que tanto estudiantes como profesor quedaron a gusto con los roles asignados.
14. ¿Considera que fueron bien asignados los roles a sus compañeros? (de acuerdo a las competencias y comportamientos de ellos)  Escala: (1) mal asignado, (5) bien asignado.	Se observa que la mayoría de los estudiantes (46 %) considera que los roles asignados a sus compañeros se hicieron de manera correcta. A la misma pregunta, el profesor respondió 4, lo que significa que el profesor considera que, de manera general, los roles asignados fueron bien asignados. Así se observa que efectivamente se asignaron los roles de acuerdo con las competencias y comportamientos de los estudiantes.
15. ¿Considera que sus compañeros se sintieron a gusto con los roles asignados?  Escala: (1) poco a gusto, (5) muy a gusto.	Se observa que la mayoría de los estudiantes (46 %) considera que sus compañeros se sintieron a gusto con los roles asignados. El profesor manifiesta su satisfacción con respecto a la asignación de los roles "secretario" y "coordinador" en la mayoría de los equipos. Por tanto, se puede deducir que tanto estudiantes como profesor quedaron a gusto con los roles asignados.

Fuente: elaboración propia.

Como se pudo ver, los encuestados consideraron que las decisiones fueron tomadas en equipo, todos los estudiantes participaron y su trabajo fue equilibrado y equitativo al interior de estos, se definieron y delimitaron claramente las responsabilidades, hubo un aprendizaje mayormente en equipo para el mejor entendimiento de un tema y se obtuvo nueva información tanto de manera individual como conjunta. Con esto, se intentó reducir el impacto de problemas tales como: a) falta de coordinación del trabajo, b) influencias inapropiadas de las dinámicas de equipo, c) tendencia a "descansar en otros", d) responsabilidades no claras, d) uso incompleto o inapropiado de la información.

6. Análisis y conclusiones

Los resultados obtenidos en el caso de estudio que se aplicó en el contexto educativo universitario (universidad pública y privada) refuerzan la tesis que establece lo siguiente: "el aprendizaje colaborativo es un acto social natural en el cual los participantes interactúan entre sí, y es a través de esta interacción que el aprendizaje se obtiene". En efecto, al aplicar la estrategia JIGSAW, las habilidades, destrezas y conocimientos de cada uno de los miembros del equipo se complementan para lograr una meta común, en este caso, el estudio de una temática, la adquisición de nuevos conceptos, la elaboración de preguntas temáticas motivadoras y sus correspondientes respuestas, así como la producción de documentos de calidad.

Es importante señalar que la estrategia JIGSAW presenta fortalezas debido a que los participantes de cada equipo (base y temático) asumen roles y responsabilidades que garantizan el éxito del aprendizaje colaborativo. La clave está en saber conformar los equipos, ya sean homogéneos u heterogéneos y asignarles los roles más acordes con su perfil, el cual está compuesto de competencias, conocimientos y habilidades de cada estudiante. La herramienta MATEO realizó una apropiada conformación de equipos, lo cual se confirmó a través del grado de satisfacción expresado por los estudiantes y el profesor en diferentes ítems de la encuesta realizada, relacionados con la generación de los equipos de trabajo.

A través de la aplicación del caso de estudio con la herramienta AYLLU bajo la estrategia JIGSAW se pudo comprobar que se llevaron a cabo, entre los equipos base y temáticos, los siguientes aspectos clave en aprendizaje colaborativo: (1) una interdependencia positiva, ya que los integrantes de los equipos debieron confiar en sus compañeros para lograr los objetivos de aprendizaje (ej., adquisición de nuevos conceptos, generación de preguntas motivadoras, elaboración de informes temáticos de calidad, etc.); (2) una responsabilidad individual, pues cada integrante

aportó desde su rol a las actividades grupales con el fin de lograr las metas comunes; (3) se promovió la interacción cara a cara, interactuando dentro de los equipos base y temáticos, discutiendo conclusiones y razonamientos, al igual que enseñando y apoyando a sus compañeros. La herramienta AYLLU propició diversos mecanismos de interacción (chat, correos, foros, etc.) facilitando la actividad del equipo; (4) se propició el uso de habilidades colaborativas, tales como el liderazgo, la toma de decisiones, la comunicación y la solución de conflictos, entre otros; (5) el mejoramiento continuo grupal, ya que los equipos base y temáticos pudieron establecer objetivos comunes que periódicamente evaluaron, identificaron dificultades y establecieron cambios que los llevaron a funcionar de una manera más eficiente en la etapa subsiguiente de la estrategia colaborativa.

Como trabajo futuro se puede señalar la implementación en la herramienta AYLLU de servicios colaborativos de *awareness* (Alcázar *et al*., 2013) tales como *participómetro* por equipo, grafo de interacción, participantes en línea, entre otros, los cuales pueden mejorar el desempeño de los equipos en la medida en que alertan en tiempo real a cada integrante del equipo sobre los eventos que se presentan en la evolución del trabajo colaborativo. Así, por ejemplo, estos servicios de *awareness* pueden indicar quiénes ya han contribuido con la actividad grupal, qué recursos están disponibles, cuál es el avance de la actividad colaborativa según el cronograma, quiénes están interactuando en un momento dado, entre otros, lo cual conlleva una conciencia individual de los eventos producidos y, por ende, a la motivación para contribuir con el trabajo colaborativo asignado.

Referencias

Alcázar H., Martínez, J., Pantoja, L., Collazos, C., Paz, A. (2013). Method for Incorporating Awareness Mechanisms in Driving Simulation Environments. *Proceedings of the 2013 Chilean Conference on Human - Computer Interaction (ChileCHI 2013)*, 30-35.

Arias-Báez, M. P., Pavlich-Mariscal, J. y Carrillo Ramos, Á. (2013). Forming Adapted Teams Oriented to Collaboration: Detailed Design and Case Study. *Revista DYNA*, *180*(80), 87-95.

Arias-Báez, M., Torres-Ribero, L., Carrillo, A., Pomares, A., González, E. (2012). AYLLU: Architecture for Collaboration in a Distributed Environment. *Proc. Collaborative Technologies (CT 2012) - IADIS Multi Conf. on Computer Science and Information System*s. Lisboa.

Hernández Sampieri, R., Fernández Collado, C. y Baptista Lucio, P. (2010). *Metodología de la Investigación*. 5ta edición. Mc Graw Hill.

Herrmann-Nehdi, A. (2015). The Best of Both Worlds - Making Blended Learning Really Work by Engaging the Whole Brain. *Herrmann International*. Disponible en http://www.hbdi.com/Resources/whitepapers/index.cfm?viewall=1.

JIGSAW. (s. f.). Jigsaw Classroom. Disponible en http://www.jigsaw.org/.

Yen, J. (1999). Fuzzy Logic-A Modern Perspective. *IEEE Trans. On Knowledge and Data Engineering*, *11*(1), 153-165.

LOS AUTORES

Pontificia Universidad Javeriana
Facultad de Educación
Mónica Brijaldo Rodríguez
Martha Leonor Sabogal Modera
Facultad de Ingeniería – Departamento de Ingeniería de Sistemas
Ángela Carrillo Ramos
Enrique González Guerrero
Alexandra Pomares Quimbaya
Luis Guillermo Torres Ribero
María Paula Arias Báez
Juan Pablo Garzón

Universidad Nacional de Colombia - Sede Medellín
Demetrio Arturo Ovalle Carranza
Julián Moreno Cadavid
Francisco Arias Sánchez

Universidad Federal del Río Grande del Sur, PortoAlegre, Brasil
Rosa María Vicari

Laboratorio de Informática de Grenoble, Francia
Hervé Martin

Este libro fue realizado en caracteres
TwCenMTCondensed y
AmeriGarmnd BT e impreso en
xxxx
en el mes de xxxx del xxxx,
en Bogotá, D. C., Colombia.