



ESCUELA POLITÉCNICA SUPERIOR
Departamento de Ingeniería Informática



Propuesta de una Metodología
para la Evaluación de Cursos
Hipermedia Adaptativos

T E S I S D O C T O R A L

Javier BRAVO AGAPITO

Madrid, mayo de 2010

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

T E S I S D O C T O R A L

Propuesta de una Metodología
para la Evaluación de Cursos
Hipermedia Adaptativos

Autor:

Javier BRAVO AGAPITO

Director:

D. Alvaro ORTIGOSA

Madrid, mayo de 2010

TÍTULO : Propuesta de una Metodología
para la Evaluación de Cursos Hi-
permedia Adaptativos
AUTOR : Javier Bravo Agapito
DIRECTOR : Alvaro Ortigosa
DTO. : Dpto. de Ingeniería Informática

Miembros del Tribunal :

Presidente :

Secretario :

Vocal 1 :

Vocal 2 :

Vocal 3 :

Fecha de lectura :

Calificación :

Agradecimientos

Esta tesis refleja el trabajo de 5 años de investigación que no hubiera sido posible sin la ayuda de mis padres, mis hermanos, mi director, mis amigos, mis familiares y las personas que he conocido en este período.

En primer lugar, quiero dar las gracias a mi director, Alvaro, por haber confiado en mí desde los comienzos de esta investigación, y haberme guiado de forma satisfactoria en este largo y arduo proceso. Sé por experiencia propia que el tiempo es finito, y muy valioso, por eso quiero dar sinceramente las gracias a Alvaro por cederme parte de su tiempo, y dedicarme tantas horas que difícilmente le podré compensar. También quiero agradecer a Pilar por los necesarios informes que me ha ido haciendo durante este período, y su buena disposición para cualquier problema que surgiera. Gracias a Roberto Moriyón por leer la tesis y sugerirme modificaciones para mejorar este documento.

Mis padres, Gonzalo y Nines, han sido mi fundamental apoyo para iniciar y terminar esta tesis. Ellos me han apoyado de manera incondicional en los momentos más difíciles de esta investigación, por eso quiero darles las gracias, si bien escribir gracias en cada una de las páginas de esta tesis no sería suficiente para agradecer toda la ayuda que he recibido. También mis hermanos, Cristina y “Juanjo”, Carlos y Jorge, me han ayudado a su manera, aunque no entendieran el contenido de esta tesis, así que muchas gracias. Igualmente quiero agradecer a mis tías y tíos, y especialmente a Manoli y Nili, y mi abuelo Gonzalo Bravo, por confiar siempre en mis posibilidades. Asimismo, mis numerosos primos, y en especial, José, Francisco y Jose Ignacio se han interesado siempre por esta investigación y me han ofrecido consejos muy útiles. También quiero acordarme de mis tíos Ramón y Tere, y mis primos, que están pasando por momentos difíciles. Espero que todos ellos puedan ver pronto esta tesis. Por último, no me puedo olvidar de mis abuelos, Andrés Agapito y Lourdes Serrano, y mi abuela Teresa Castañeda, que ya no están en este mundo.

Tampoco puedo olvidarme de mis fieles amigos: de la universidad, la empresa, Guadarrama, Francia, USA, Japón, Méjico y Perú. Gracias a Julián, Chema y Manuel por estar ahí cuando lo necesité, y por su comprensión y apoyo. Gracias a Jose Manuel, Jose Daniel, Quelo y Txabi por los buenos momentos que he pasado con ellos. Gracias a Roberto, Mari Paz, Eduardo, César y Carlos por ayudarme y motivarme. En especial, Carlos fue un apoyo muy importante en los momentos más difíciles, muchas gracias por todo. Gracias a la gente de *Reciclaje*, y especialmente a Javi, Marisa, Juanjo y Araceli, por vivir unos momentos inolvidables en este bar que espero siga siempre en su sitio. Gracias a todas las personas de la lista *SpS*, y en especial a Samu, Emilio, Jose Antonio, Kike, Julen, Fran, Alberto, Silvia, María, Neus, Fátima, Jesús, Isma, Gatusso, Olivier y Nadime, por haber pasado momentos inolvidables en *Brest*, *Rennes*, *Nantes*, *Bonn*, Colonia, y las reuniones posteriores en Barcelona y Madrid. Gracias a Elisabeth, Nori, Ivan, Kodgi y toda la comunidad asiática por ayudarme a integrarme durante mi estancia en *Pittsburgh*. En

especial, gracias a Elisabeth por prestarme su ayuda incondicional en los buenos y no tan buenos momentos durante mi estancia en *Pittsburgh*, y desde la distancia (en Estados Unidos) en estos últimos años. Gracias también a Abraham con quien he compartido mi afición al fútbol hablando del eterno rival, y espero verle pronto en España cuando vuelva. Por supuesto, no me quiero olvidar de mi buen amigo peruano César, pues pasé grandes momentos con él y su familia y amigos en Lima e Ica. También como buen amigo me dió buenos consejos.

No puedo olvidarme de mis fieles compañeros del laboratorio B207 y Odisea, algunos de la segunda planta, y aquellos que ya no están porque o bien promocionaron al siguiente piso, o buscaron suerte en otras universidades. En especial gracias a Manu Freire, Pablo, Pedro, Leila, Germán, Manu Herranz, Miguel Mora, Rosa y Estefanía por haber convivido conmigo durante estos años. Con ellos he pasado gran parte de mi tiempo, en las comidas, fiestas del laboratorio, amigos invisibles, cenas, etc. Gracias también a Alejandro y Fernando del laboratorio de al lado, por los partidos de baloncesto y las interesantes conversaciones sobre el sistema universitario y la investigación en los servicios. Tampoco puedo olvidarme de mis colegas de Inteligencia Artificial: Alberto, Fernando, Ruth y Arturo. Gracias especialmente a Alberto Suárez por aconsejarme en cuestiones sobre matemáticas.

Merci beaucoup à Serge Garlatti et à son équipe de recherche, et à tout la gens que j'ai connu en la École Nationale Supérieure de Télécommunications Bretagne (actuellement Télécom Bretagne). I would like to thank to Peter Brusilovsky and his team PAWS for allowing me to analyze the interaction data of QuizGuide. I have collaborated with Peter and this research team, and they also gave me some suggestions to improve this research.

Gran parte de este trabajo ha sido posible gracias a la beca Predoctoral de *FPI* (con referencia BES-2005-8178) asociada al proyecto de investigación *U-CAT* (TIN2004-03140) que me fue concedida por el Ministerio de Educación y Ciencia, cofinanciada por el Fondo Social Europeo (BOE del 29 de julio 2005). Además, quiero dar las gracias también por la financiación recibida por el proyecto *HADA* (TIN2007-64718), que fue imprescindible para presentar los resultados obtenidos a la comunidad científica.

Resumen

La utilización de los sistemas *e-Learning* ha experimentado un gran incremento en los últimos años debido sobre todo a las numerosas posibilidades que nos ofrece su tecnología para el aprendizaje. De hecho, hoy en día es una práctica habitual utilizarlos para complementar la enseñanza tradicional aprovechando las nuevas posibilidades que nos aportan. Además, estos sistemas se están empezando a utilizar fuera del ámbito de las universidades. Así, podemos encontrar ejemplos en otros ámbitos como la enseñanza secundaria, musical y los cursos de formación de las empresas.

La principal característica que aportan estos sistemas es proporcionar tanto a profesores como estudiantes nuevas formas de transmitir, organizar, presentar y almacenar los contenidos educacionales. Sin embargo, esta mayor interactividad y flexibilidad, que constituyen ventajas a la hora de intentar mejorar el proceso de aprendizaje, se convierten en grandes dificultades al realizar las tareas de diseño y evaluación.

Los responsables de estas tareas son los profesores, por ser quienes poseen los conocimientos y experiencias necesarios. La tarea de diseñar un curso es compleja, pues no sólo se deben diseñar sus contenidos, sino también el orden en el que deben ser presentados para su máximo aprovechamiento y cómo debe ser su presentación. En el ámbito de los sistemas *e-Learning* a esta complejidad hay que añadir la dificultad de que el diseño debe seguir las especificaciones impuestas por cada sistema. Afortunadamente, los profesores cuentan, por lo general, con la ayuda de herramientas de autor, incluidas en muchos de estos sistemas. Sin embargo, son pocos los sistemas que incluyen métodos o herramientas de evaluación, a pesar de la importancia que este proceso de su evaluación tiene, para mejorar un curso *e-Learning*. Este proceso trata de responder a las preguntas sobre si los contenidos proporcionados por el curso fueron adecuados para todos los estudiantes, o si el orden de presentación fue adecuado, etc. En resumen, esta evaluación consiste en valorar la eficacia del curso, en el sentido de medir su utilidad para el estudiante.

Uno de los principales problemas a los que se enfrentan los profesores al llevar a cabo esta tarea es que mientras en la enseñanza tradicional el profesor puede observar en todo momento las reacciones y comportamientos de los estudiantes, en el caso de estos sistemas no es posible obtener esta información de la misma manera. Esto es así, pues los estudiantes pueden realizar sus actividades educativas sin que sea necesaria la presencia física del profesor. No obstante, es posible conocer esta información de otra forma, pues la mayoría de estos sistemas almacenan las interacciones de los estudiantes en logs. Por tanto, el análisis de estos logs puede ofrecer al profesor información sobre los comportamientos y reacciones de los estudiantes.

Un problema en este análisis de logs es que por su tamaño plantean dos grandes retos: ¿Cómo se pueden analizar grandes volúmenes de datos? y ¿Cómo seleccionar la información relevante? Esta tesis presenta un método que resuelve estas dos

cuestiones. El método propuesto, cuyo nombre es *GeSES*, utiliza la técnica *reglas de decisión* y un modelo de selección matemático que extrae la información relevante para los profesores. Dicha información se presenta como una lista de signos que pueden indicar baja eficacia en el sistema. Así, el método ofrece al profesor información, de manera entendible, sobre aquellos elementos del curso que requieren su atención. Con esta información el profesor puede revisar, modificar, o incluso añadir nuevos elementos que mejoren la eficacia del curso, y consecuentemente el proceso de aprendizaje. El método *GeSES* fue probado con tres conjuntos distintos de datos pertenecientes a dos universidades (Universidad Autónoma de Madrid y *University of Pittsburgh*) obteniendo resultados satisfactorios. Asimismo, se probó la eficacia del método mediante técnicas de simulación donde se comprobó que el tamaño y proporción de los datos que constituyen un signo afectan a la calidad de resultados obtenidos.

Además, esta tesis incluye el desarrollo de dos herramientas: *SimuLog* y *ASquare*. La primera es un simulador que genera logs de estudiantes de distintos perfiles, incluyendo supuestos signos de baja eficacia relacionados con el bajo rendimiento académico. *ASquare* es la herramienta que da soporte al método *GeSES*, ofreciendo al profesor una más fácil utilización del método.

Abstract

The utilization of e-Learning systems has been increased in the last few years. This increase is due in part to the availability of technology resources that improve the learning process. Currently, it is standard or the norm for traditional teaching methods to be complemented by a e-Learning system. The utilization of this technology is not isolated to just academia, it is also utilized in for profit Industry and the arts.

The main advantage offered by these systems is to provide for teachers and students a new ways to transmit, organize, present, and store the educational contents. This high interactivity and flexibility, which are advantages for improving the learning process, is becoming a great obstacle in the designing and evaluation process. In regards to the use of e-Learning systems in academic environments, the task of designing the e-Learning course falls upon the shoulders of the teacher or instructor. Design process is a complex task, because teachers not only have to design the contents, but also the overall layout. Moreover, this complexity of e-Learning systems can be challenging as the design process is limited to the specific rules and requirements of the system. Fortunately, teachers generally can use authoring tools, which are included in many systems. Nonetheless, there are a few systems that provide evaluation methods and tools, in spite of the importance of evaluation process to improve an e-Learning course. This process tries to answer questions on if course contents were suitable for the students, or if the order of presentation was adequate, etc. Summarizing, this evaluation consists of assessing the course efficacy, in the sense of measuring the utility for the student.

One of the inherent problems of the evaluation of e-Learning systems is that unlike traditional classrooms, teachers are unable to observe student reactions and behavior. The students can carry out their educational activities without the physical presence of the teacher. However, it is possible to obtain this information in other way, since most of these systems store student interactions in logs. A solution around the obstacle of lack of student observation, is the analysis of logs of student activity and behavior in a e-Learning course.

One of the main challenges of analyzing the logs is the volume of size which poses two problems: How is it possible to analyze large volumes of data? And how the relevant information can be selected? This thesis presents a method that solves these questions. The proposed method, called *GeSES*, utilizes decision rules technique and a mathematical model to extract the relevant information for teachers. This information is presented in a list of signs that can indicate low efficacy of the system. Thus, the method offers to teachers an easily understandable information about course elements that require his/her attention. With this information the teacher can review, modify, or even add new elements that improve the course efficacy. In turn, the learning process is improved. The *GeSES* method was tested successfully with three different data sets of two universities (*Universidad Autónoma de Madrid*

and University of Pittsburgh). Also, their efficacy was tested by using simulation techniques. These results showed that size and proportion of data which generate a sign, affect to quality of the results.

In addition, this thesis includes the development of two tools: *SimuLog* and *ASquare*. The first tool is a simulator able to generate students' logs of different profiles, including synthetic signs of low efficacy related to low student performance. *ASquare* is the tool which implements the *GeSES* method, offering to teachers an easily way to apply the method.

Índice general

1. Introducción	7
1.1. Contexto	7
1.1.1. Sistemas <i>e-Learning</i>	9
1.1.2. Sistemas Tutores Inteligentes	10
1.1.3. Sistemas <i>Hypermedia</i> Adaptativos Orientados a la Enseñanza	11
1.2. Motivación	13
1.3. Solución propuesta	17
1.4. Objetivos de la tesis	18
1.5. Método de trabajo	19
1.6. Contribuciones de la tesis	20
1.7. Publicaciones a las que ha dado lugar	21
1.8. Estructura del documento	25
2. Estado de la Cuestión	29
2.1. Introducción	29
2.2. Evaluación de Sistemas Interactivos	30
2.3. Evaluación de Sistemas <i>E-Learning</i>	31
2.4. Evaluación de Sistemas <i>Hypermedia</i> Adaptativos	33
2.4.1. Modelos de evaluación	33
2.4.2. Evaluación de Sistemas <i>Hypermedia</i> Adaptativos Orientados a la Enseñanza	37
2.5. Contexto de la presente propuesta	54
3. Minería de Datos	57
3.1. Introducción	57
3.2. Reglas de asociación	60
3.3. Árboles de decisión	61
3.4. Herramientas de aplicación utilizadas	63
3.4.1. <i>Weka</i>	64
3.4.2. Paquete de programas <i>C4.5</i>	65
4. Signos de disminución de la eficacia en sistemas o cursos <i>e-Learning</i>	75
4.1. Introducción	75
4.2. Signos y diagnóstico	77
4.3. Tipos de signos	80
4.4. Ontología de signos	82
4.5. Conclusiones	85

5. Métodos para detectar signos de baja eficacia	87
5.1. Introducción	87
5.2. Una primera aproximación: el método <i>Key-node</i>	88
5.2.1. Descripción del método	88
5.2.2. Experimentos	90
5.3. El método <i>GeSES</i>	97
5.3.1. Descripción del método	98
5.3.2. Exposición gráfica del método	113
5.3.3. Experimentos realizados	116
5.4. Conclusiones	142
6. Aplicación y Evaluación de la propuesta	145
6.1. Introducción	145
6.2. Aplicación de la propuesta	146
6.2.1. Análisis inicial de los datos	146
6.2.2. Análisis mediante el método <i>GeSES</i>	147
6.3. Procedimiento de evaluación	154
6.4. Introducción a <i>SimuLog</i>	155
6.5. Evaluación	155
6.5.1. Preparación de los datos (logs)	156
6.5.2. Análisis de los datos mediante el método <i>GeSES</i>	157
6.6. Conclusiones	168
7. Herramientas para la Evaluación	171
7.1. Introducción	171
7.2. Ciclo de creación y mantenimiento de cursos <i>e-Learning</i>	172
7.3. <i>SimuLog</i>	173
7.3.1. Arquitectura	174
7.3.2. Entradas de simulación	174
7.3.3. Motor de simulación	178
7.3.4. Procesador de logs	186
7.4. <i>ASquare</i>	189
7.4.1. Elementos de <i>ASquare</i>	189
7.4.2. Interfaz de <i>ASquare</i>	190
7.5. Conclusiones	192
8. Conclusiones y trabajo futuro	195
8.1. Conclusiones	195
8.2. Limitaciones	200
8.2.1. Consideraciones respecto al método <i>GeSES</i>	200
8.2.2. Consideraciones respecto a <i>SimuLog</i>	201
8.2.3. Consideraciones respecto al proceso de evaluación del método	202
8.2.4. Consideraciones respecto a <i>ASquare</i>	202
8.3. Trabajo futuro	202

8.4.	Publicaciones a las que ha dado lugar este trabajo	204
8.4.1.	Publicaciones en revistas con índice de impacto	205
8.4.2.	Publicaciones en congresos internacionales	205
8.4.3.	Publicaciones en congresos nacionales	206
8.4.4.	Capítulos de libro	207
8.4.5.	Trabajo de Iniciación a la Investigación	207

A.	Apéndice	209
A.1.	Ficheros de datos	210
A.1.1.	Fichero <i>weather.csv</i>	210
A.1.2.	Fichero <i>weather.names</i>	211
A.1.3.	Fichero <i>weather.data</i>	211
A.2.	Reglas de asociación	211
A.2.1.	Reglas de asociación de los datos de la tabla 3.1 (página 58)	211
A.3.	Árboles de decisión	214
A.3.1.	Árbol de decisión de los datos de la tabla 3.1, página 58 (algoritmo <i>J48</i>)	214
A.3.2.	Árbol de decisión de los datos de la tabla 3.1, página 58 (algoritmo <i>C4.5</i>)	215
A.3.3.	Reglas de decisión de los datos de la tabla 3.1, página 58	216
A.4.	Sistemas <i>e-Learning</i> evaluados	217
A.4.1.	<i>Wotan</i>	217
A.4.2.	<i>QuizGuide</i> y <i>QuizPACK</i>	218
A.4.3.	<i>CoMoLE</i>	220
A.5.	Ficheros de logs	222
A.5.1.	Curso de “Introducción a la programación en el lenguaje C” en los sistemas <i>QuizGuide</i> y <i>QuizPACK</i>	222
A.5.2.	Curso de “Sistemas Operativos I” en el sistema <i>CoMoLE</i>	229
A.5.3.	Curso de “Estructura de Datos y de la Información I” en el sistema <i>CoMoLE</i>	238
A.6.	Reglas de decisión	242
A.6.1.	Reglas de decisión para el curso “Introducción a la programación en el lenguaje C” en los sistemas <i>QuizGuide</i> y <i>QuizPACK</i>	243
A.6.2.	Reglas de decisión para el curso “Sistemas Operativos I” en el sistema <i>CoMoLE</i>	246
A.6.3.	Reglas de decisión para el curso “Estructura de Datos y de la Información I” en el sistema <i>CoMoLE</i>	258
A.7.	Método <i>GeSES</i> para distintos tamaños de datos	272
A.7.1.	Resultados para $N = 20$	273
A.7.2.	Resultados para $N = 50$	273
A.7.3.	Resultados para $N = 80$	273
A.7.4.	Resultados para $N = 100$	274
A.7.5.	Resultados para $N = 150$	274
A.7.6.	Resultados para $N = 300$	275

A.7.7. Resultados para $N = 600$	275
A.7.8. Resultados para $N = 900$	276
A.7.9. Resultados para $N = 1200$	277
A.7.10. Resultados para $N = 1500$	278
A.7.11. Resultados para $N = 1800$	278
A.7.12. Resultados para $N = 2100$	279
A.7.13. Resultados para $N = 2400$	280
A.7.14. Resultados para $N = 2700$	281
A.7.15. Resultados para $N = 3100$	282
A.7.16. Resultados para $N = 3500$	283
A.7.17. Resultados para $N = 4000$	284
A.7.18. Resultados para $N = 5000$	285
A.7.19. Resultados para $N = 6000$	286
Bibliografía	287
Lista de abreviaturas	303

Índice de figuras

1.1. Disciplinas relacionadas con los Sistemas Tutores Inteligentes	11
1.2. Contexto de los Sistemas <i>Hypermedia</i> Adaptativos Orientados a la Enseñanza	13
1.3. Distribución de los logs en el cajón de la actividad <i>act1</i>	15
1.4. Distribución de los logs en el cajón de la actividad <i>act1</i> después de aplicar <i>DM</i>	16
3.1. Árbol de decisión de los datos de la tabla 3.1	60
3.2. Interfaz de <i>Weka</i> : módulo de preprocesamiento de datos	64
3.3. Interfaz de <i>Weka</i> : módulo de técnicas de clasificación	65
3.4. Interfaz de <i>Weka</i> : módulo de reglas de asociación	66
3.5. Parte superior de la salida de un ejemplo de ejecución del programa <i>c4.5</i>	67
3.6. Parte central e inferior de la salida de un ejemplo de ejecución del programa <i>c4.5</i>	68
4.1. Proceso general de diagnóstico y tratamiento de una enfermedad	77
4.2. Cuadro de mandos de un automóvil	78
4.3. Evaluación de un sistema o curso <i>e-Learning</i>	79
4.4. Ontología de síntomas para cursos <i>e-Learning</i>	84
5.1. Detección de signos mediante el método <i>Key-node</i>	90
5.2. Árbol de decisión generado en el primer experimento	94
5.3. Árbol de decisión generado en el segundo experimento	96
5.4. Pesos otorgados por la frontera <i>F2</i> a las reglas para el concepto <i>used</i>	102
5.5. Pesos graduados otorgados por la frontera <i>F2</i> a las reglas para el concepto <i>used</i>	104
5.6. Función sigmoideal de parámetro <i>y</i>	105
5.7. Función sigmoideal desplazada para valores positivos	108
5.8. Función sigmoideal desplazada para valores positivos con el parámetro <i>k</i>	109
5.9. Funciones sigmoideales para distintos valores de la <i>k</i>	109
5.10. Fronteras de selección <i>F1</i> y <i>F2</i> en las observaciones del concepto <i>used</i>	111
5.11. Detección de signos mediante el método <i>GeSES</i>	114
5.12. Fase 2: Reglas de decisión generadas con el algoritmo <i>C4.5</i>	115
5.13. <i>Ranking</i> de reglas de decisión	115
5.14. Fase 3: <i>DR</i> seleccionadas por clase	116
5.15. Funciones sigmoideales para distintos valores de la <i>k</i> para los datos de <i>QuizGuide</i>	124
5.16. Pesos graduados otorgados por la frontera <i>F2</i> a las reglas para el parámetro <i>used</i>	137

5.17. Funciones sigmoideas para distintos valores de la k para los datos de <i>CoMoLE</i>	139
6.1. Procedimiento de evaluación de la propuesta	155
6.2. <i>SimuLog</i> generando logs sintéticos	158
7.1. Ciclo de creación y mantenimiento de cursos <i>e-Learning</i>	172
7.2. Arquitectura de <i>SimuLog</i>	175
7.3. Interfaz de <i>SimuLog</i> : el editor de dimensiones	176
7.4. Interfaz de <i>SimuLog</i> : el editor de signos en el estado inicial	178
7.5. Interfaz de <i>SimuLog</i> : el editor de signos con la descripción de un signo	179
7.6. Interfaz de <i>SimuLog</i> : el editor de signos con la modificación de un signo	179
7.7. Interfaz de <i>SimuLog</i> después de haber añadido un signo	179
7.8. Traslación de una distribución <i>Normal</i> d unidades hacia la izquierda	182
7.9. Intervalos de la distribución <i>Normal</i> para tres categorías	183
7.10. Arquitectura de <i>ASquare</i>	190
7.11. <i>ASquare</i>	191
A.1. Interfaz de <i>QuizGuide</i>	219
A.2. Ejemplo de página <i>web</i> generada por <i>CoMoLE</i>	221

Índice de tablas

2.1. Técnicas utilizadas en las distintas aplicaciones de la Minería de Uso de <i>Web</i>	42
3.1. Datos del ejemplo weather	58
3.2. Coste de codificación de subconjuntos de <i>DR</i>	71
3.3. <i>Ranking</i> de <i>DR</i>	73
3.4. Matriz de confusión de <i>DR</i>	73
5.1. Descripción de los atributos de un registro de log en el sistema <i>TAN-GOW</i>	92
5.2. Perfiles de estudiantes simulados	93
5.3. Pesos otorgados por las fronteras <i>F1</i> y <i>F2</i>	101
5.4. Pesos graduados otorgados por las fronteras <i>F1</i> y <i>F2</i>	103
5.5. Ejemplos de logs en <i>QuizGuide</i>	118
5.6. Distribución de los datos de análisis de <i>QuizGuide</i>	119
5.7. <i>Ranking</i> de <i>DR</i> de los datos de análisis de <i>QuizGuide</i>	120
5.8. Reglas de decisión seleccionadas del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>QuizGuide</i>	120
5.9. Fronteras de selección para los conceptos del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>QuizGuide</i>	121
5.10. Pesos otorgados por las fronteras para los conceptos del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>QuizGuide</i>	122
5.11. Pesos graduados otorgados por las fronteras para los conceptos del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>QuizGuide</i>	123
5.12. Pesos otorgados por la función sigmoideal modificada para los datos de análisis de <i>QuizGuide</i>	125
5.13. Reglas más relevantes respecto a la característica de evaluación para los datos de análisis de <i>QuizGuide</i>	126
5.14. Reglas cercanas a las más relevantes para los datos de análisis de <i>QuizGuide</i>	127
5.15. Signos detectados para los datos de análisis de <i>QuizGuide</i>	127
5.16. Ejemplos de logs en <i>CoMoLE</i> (1/2)	130
5.17. Ejemplos de logs en <i>CoMoLE</i> (2/2)	130
5.18. Distribución de los datos de análisis de los logs de <i>CoMoLE</i>	131
5.19. <i>Ranking</i> de <i>DR</i> de los datos de análisis de <i>CoMoLE</i>	132
5.20. Reglas de decisión seleccionadas del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>CoMoLE</i>	133
5.21. Fronteras de selección para los conceptos del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>CoMoLE</i>	134

5.22. Pesos otorgados por las fronteras para los conceptos del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>CoMoLE</i>	135
5.23. Pesos graduados otorgados por las fronteras para los conceptos del <i>ranking</i> de <i>DR</i> de los datos de análisis de <i>CoMoLE</i>	136
5.24. Pesos otorgados por la función sigmoideal modificada para los datos de análisis de <i>CoMoLE</i>	139
5.25. Reglas más relevantes respecto a la característica de evaluación para los datos de análisis de <i>CoMoLE</i>	140
5.26. Reglas cercanas a las más relevantes para los datos de análisis de <i>CoMoLE</i>	141
5.27. Signos detectados para los datos de análisis de <i>CoMoLE</i>	142
6.1. Distribución de los datos de análisis de los logs de <i>CoMoLE</i> en el curso de <i>EDI1</i>	148
6.2. Reglas de decisión seleccionadas de los datos del curso de <i>EDI1</i> en <i>CoMoLE</i>	149
6.3. Reglas y valores de la frontera de selección para los datos del curso de <i>EDI1</i> en <i>CoMoLE</i>	150
6.4. Pesos totales para las reglas de los datos del curso de <i>EDI1</i> en <i>CoMoLE</i>	152
6.5. Reglas más relevantes respecto a la característica de evaluación para los datos de análisis de <i>EDI1</i> en <i>CoMoLE</i>	153
6.6. Signos detectados para los datos de análisis del curso <i>EDI1</i> en <i>CoMoLE</i>	154
6.7. Resumen de la aplicación del método <i>GeSES</i> para distintos tamaños de datos	159
6.8. Reglas obtenidas en la fase 3 para $N = 900$	161
6.9. <i>Ranking</i> de reglas en la fase 5 para $N = 900$	162
6.10. Estadísticas de los perfiles generados para $N = 900$	162
6.11. Reglas obtenidas en la fase 3 para $N = 2100$	164
6.12. <i>Ranking</i> de reglas en la fase 5 para $N = 2100$	165
6.13. Estadísticas de los perfiles generados para $N = 2100$	165
6.14. Reglas obtenidas en la fase 3 para $N = 6000$	167
6.15. <i>Ranking</i> de reglas en la fase 5 para $N = 6000$	167
6.16. Estadísticas de los perfiles generados para $N = 6000$	168
A.1. Reglas obtenidas en la fase 3 para $N = 20$	273
A.2. Reglas obtenidas en la fase 3 para $N = 50$	273
A.3. Reglas obtenidas en la fase 3 para $N = 80$	273
A.4. Reglas obtenidas en la fase 3 para $N = 100$	274
A.5. Reglas obtenidas en la fase 3 para $N = 150$	274
A.6. Reglas obtenidas en la fase 3 para $N = 300$	275
A.7. Reglas obtenidas en la fase 3 para $N = 600$	275
A.8. Reglas obtenidas en la fase 3 para $N = 900$	276
A.9. <i>Ranking</i> de reglas en la fase 5 para $N = 900$	276
A.10. Reglas obtenidas en la fase 3 para $N = 1200$	277

A.11. <i>Ranking</i> de reglas en la fase 5 para $N = 1200$	277
A.12. Reglas obtenidas en la fase 3 para $N = 1500$	278
A.13. Reglas obtenidas en la fase 3 para $N = 1800$	278
A.14. Reglas obtenidas en la fase 3 para $N = 2100$	279
A.15. <i>Ranking</i> de reglas en la fase 5 para $N = 2100$	279
A.16. Reglas obtenidas en la fase 3 para $N = 2400$	280
A.17. <i>Ranking</i> de reglas en la fase 5 para $N = 2400$	280
A.18. Reglas obtenidas en la fase 3 para $N = 2700$	281
A.19. <i>Ranking</i> de reglas en la fase 5 para $N = 2700$	281
A.20. Reglas obtenidas en la fase 3 para $N = 3100$	282
A.21. <i>Ranking</i> de reglas en la fase 5 para $N = 3100$	282
A.22. Reglas obtenidas en la fase 3 para $N = 3500$	283
A.23. <i>Ranking</i> de reglas en la fase 5 para $N = 3500$	283
A.24. Reglas obtenidas en la fase 3 para $N = 4000$	284
A.25. <i>Ranking</i> de reglas en la fase 5 para $N = 4000$	284
A.26. Reglas obtenidas en la fase 3 para $N = 5000$	285
A.27. <i>Ranking</i> de reglas en la fase 5 para $N = 5000$	285
A.28. Reglas obtenidas en la fase 3 para $N = 6000$	286
A.29. <i>Ranking</i> de reglas en la fase 5 para $N = 6000$	286

**Propuesta de una Metodología para la Evaluación de Cursos
Hipermedia Adaptativos**

Introducción

Índice de contenidos

1.1. Contexto	7
1.1.1. Sistemas <i>e-Learning</i>	9
1.1.2. Sistemas Tutores Inteligentes	10
1.1.3. Sistemas <i>Hypermedia</i> Adaptativos Orientados a la Enseñanza	11
1.2. Motivación	13
1.3. Solución propuesta	17
1.4. Objetivos de la tesis	18
1.5. Método de trabajo	19
1.6. Contribuciones de la tesis	20
1.7. Publicaciones a las que ha dado lugar	21
1.8. Estructura del documento	25

1.1. Contexto

El uso y aparición de nuevos sistemas *e-Learning* se ha incrementado de forma exponencial en los últimos años, y hoy en día es una práctica habitual complementar la enseñanza tradicional aprovechando las posibilidades que nos ofrecen. La principal característica de estos sistemas es proporcionar tanto a estudiantes como a profesores nuevos medios para transmitir, organizar y almacenar los contenidos educativos. Por ejemplo se pueden transmitir por cualquier dispositivo electrónico (*PC*¹, *Netbooks*, *PDA*s, *Smartphones*, pizarras inteligentes, etc.), se pueden organizar y almacenar en sistemas de administración de contenidos y laboratorios virtuales, entre otros. Además, permiten la utilización de diferentes protocolos de comunicación (*SMS*, *Bluetooth*, entre otros) [Chen 2000].

Estas nuevas tecnologías han facilitado la amplia aceptación de los sistemas *e-Learning* en la última década. Por un lado, estos sistemas ofrecen mayor interactividad y mayor flexibilidad, pues se pueden utilizar en distintos dispositivos, distintas localizaciones y en cualquier momento, lo que puede proporcionar mejor experiencia de aprendizaje en los estudiantes; pero, por otro lado, el mantenimiento de estos

¹Se utilizan las abreviaturas definidas en las páginas finales de esta tesis, consúltense para mayor información.

sistemas representa un importante obstáculo para su utilización por parte de los profesores. En general, los profesores son los encargados de realizar la compleja tarea del diseño de un curso. Ellos diseñan los contenidos educativos que quieren transmitir a los estudiantes, ya que por sus conocimientos y su experiencia son las personas más adecuadas. El diseño de un curso *e-Learning* es más complicado aún, pues hay que realizarlo de una manera determinada por la plataforma del sistema. En este sentido, los profesores tienen que aprender a diseñar cursos de forma que sigan las especificaciones impuestas por el sistema. En muchas ocasiones esta tarea tiene cierta complejidad pues, los profesores deben conocer: a) el formato del curso (normalmente dependiente de cada sistema), por ejemplo, un curso puede estar definido en *XML*, a través de una base de datos relacional, etc.; b) definición de los distintos elementos que componen un curso, es decir, cómo se le indica al sistema el tipo de actividad, si es una actividad teórica o práctica, si la actividad está compuesta por otras, etc.; c) formas de presentar los contenidos ofrecidos por el sistema, es decir, cómo indicar si los contenidos deben ser presentados mediante texto, imágenes, audio o vídeo. . a manejar las herramientas de diseño proporcionadas por estos sistemas, que en muchas ocasiones su uso es poco intuitivo. Por tanto, la tarea de diseñar un curso *e-Learning* es compleja, pues requiere de tiempo para aprender el manejo del sistema y experiencia en su uso. Asimismo, los profesores realizan la tarea de evaluar un curso, es decir, conocer si el curso fue útil para los estudiantes, y de qué manera se podría mejorar. El principal obstáculo al que se enfrentan los profesores al llevar a cabo esta tarea es que, mientras en la enseñanza tradicional el profesor percibe mediante observación directa las reacciones y comportamientos de los estudiantes, pudiendo adaptar la forma en la que transmite los conocimientos en función de lo observado, en el caso de estos sistemas el profesor no puede obtener este tipo información de la misma forma, ya que generalmente no está físicamente presente en el momento en el que los estudiantes siguen un curso *e-Learning*. Por esta razón, la evaluación de un curso es también una tarea compleja.

Ambas tareas, el diseño y la evaluación, constituyen campos de investigación cada vez más demandados en la actualidad, por la inclusión de las nuevas tecnologías en las clases tradicionales. Respecto al diseño, existe un amplio desarrollo en el campo de las herramientas de autor, que proporcionan la ayuda necesaria para que los profesores puedan diseñar un curso *e-Learning*. De hecho, la mayoría de estos sistemas incluyen una herramienta de autor, que transforma el diseño realizado en el formato requerido por el sistema. Sin embargo, en el caso de la evaluación, según lo analizado en la literatura, estos sistemas no incluyen una herramienta específica que ayude a efectuar esta tarea.

Es práctica común que los sistemas *e-Learning* almacenen en bases de datos relacionales o ficheros las interacciones de los estudiantes, que por simplificación en este trabajo se denominarán **logs**. Estas interacciones reflejan el comportamiento que tuvieron los estudiantes, ya que contienen información sobre sus acciones, progresos y resultados obtenidos en cada una de las actividades realizadas. Por tanto, el profesor dispone de esta información, aunque está oculta en los logs, pero existe la posibilidad de obtenerla mediante el análisis de estos logs. No obstante, normal-

mente el tamaño de éstos es suficientemente grande para que sea inviable su análisis sin la ayuda de métodos y herramientas adecuadas para esta tarea.

La *Minería de Datos* es la disciplina que es capaz de extraer información implícita contenida en los datos, que previamente es desconocida, y además es potencialmente útil [Witten 2000, p. XIX], Esta disciplina proporciona un conjunto de métodos y técnicas que han demostrado su capacidad para analizar grandes cantidades de datos con éxito en ámbitos como el comercio electrónico. Debido a que los logs contienen grandes cantidades de datos, una posibilidad para conseguir analizarlos consiste en que el profesor utilice las técnicas mencionadas. No obstante, los resultados obtenidos de su aplicación para el análisis de logs generalmente no son fácilmente entendibles, ya que el profesor no suele ser un experto en estas técnicas. Esta tesis propone un método de evaluación de cursos *e-Learning* que ayude a realizar el análisis de logs y que los resultados de este análisis sean entendibles. Dicho en otras palabras, este método permite al profesor evaluar el curso *e-Learning*, de forma que los resultados sean fácilmente entendibles.

Las siguientes subsecciones muestran una breve descripción de las tecnologías que se encuentran dentro del ámbito de aplicación de la propuesta de evaluación contenida en la presente tesis.

1.1.1. Sistemas *e-Learning*

En el sentido literal *e-Learning* significa *aprendizaje electrónico*, es decir, el aprendizaje producido a través de un medio tecnológico-digital. Rosenberg define *e-Learning* como el uso de las tecnologías basadas en *Internet* para proporcionar un amplio despliegue de soluciones que impactan en el aprendizaje y el rendimiento [Rosenberg 2005, Rosenberg 2001, p. 1]. Otros autores como Clark y Mayer definen este concepto como la enseñanza transmitida desde un ordenador a través de los medios *DVD-ROM*, *Internet*, o *intranet* con las siguientes características [Clark 2008, p. 10]:

- Incluye contenido relevante para alcanzar los objetivos de aprendizaje.
- Utiliza métodos de enseñanza como ejemplos y ejercicios para ayudar en el aprendizaje.
- Utiliza elementos de comunicación como palabras e imágenes para transmitir el contenido y los métodos.
- Puede ser enseñanza dirigida por el profesor (*e-Learning* síncrono) o diseñada para auto-aprendizaje (*e-Learning* asíncrono).
- Sirve de ayuda a los estudiantes para conseguir sus objetivos educacionales.

Por tanto, un **curso *e-Learning*** incluye contenidos educativos, utiliza métodos de enseñanza, es ofrecido por ordenadores o por cualquier dispositivo electrónico como *PDA*, *Smartphone*, etc., y su fin último es ayudar en el aprendizaje. Generalmente los contenidos educativos se dividen en actividades, que pueden ser de diversos

tipos dependiendo de los objetivos educativos del curso. Por ejemplo, un curso puede estar formado por actividades de tipo teórico, para transmitir determinados conceptos, y por actividades de tipo práctico, para comprobar si se han aprendido tales conceptos. Los **Sistemas e-Learning** son los encargados de ofrecer, gestionar y evaluar las actividades que componen un curso *e-Learning*. Actualmente, estos sistemas están siendo implantados en áreas de la enseñanza universitaria y secundaria, aunque empiezan a tener gran aceptación en las empresas y otros ámbitos, especialmente en el campo de cursos de formación para los empleados. Un ejemplo de la aceptación de estos sistemas en el área de la enseñanza es el estudio realizado por Sancerni y Villar sobre la implantación del sistema *Elluminate Live!*² en la Universidad de Valencia [Sancerni 2008], en el que la mayoría de profesores consideraron adecuada la implantación de dicho sistema como apoyo a sus actividades profesionales.

Además, el *Plan Bolonia* firmado el 19 de Junio de 1999 [EHEA 1999] exige un proceso de adaptación de las universidades europeas, también conocido como *Proceso de Bolonia* [EHEA 2010]. Este proceso ha llevado a las universidades a realizar programas de innovación docente, cuyo objetivo es estimular a los profesores para que desarrollen nuevas tecnologías de enseñanza y aprendizaje [UC 2010]. En este sentido, los sistemas *e-Learning* serán parte importante de la enseñanza universitaria en un futuro próximo, ya que ayudan a cumplir gran parte de los objetivos fijados en el proceso de adaptación al *Plan Bolonia*. Por ejemplo, en la Universidad Autónoma de Madrid se está implantando el sistema *e-Learning Moodle* [Moodle-Community 2010] como plataforma de gestión, administración y ofrecimientos de cursos *e-Learning*.

1.1.2. Sistemas Tutores Inteligentes

Los Sistemas Tutores Inteligentes (*Intelligent Tutoring Systems*, en adelante: *ITS*) surgieron en la década de 1970, y resurgieron en la década de 1990 debido a la fuerte expansión de los ordenadores. El primer sistema tutor inteligente basado en un sistema experto fue *GUIDON* [Clancey 1979, Clancey 1987]. Este sistema fue diseñado para enseñar en el ámbito de la medicina a los estudiantes a identificar enfermedades infecciosas como la meningitis.

La figura 1.1 muestra las tres disciplinas en las que se basan los *ITS*. En la figura citada se refleja que algunos métodos y herramientas de las disciplinas Ingeniería Informática (*Computer Science*), Psicología y Educación son complementarios, y colectivamente cubren los campos de la Inteligencia Artificial y la Educación. Así, la **Inteligencia Artificial** (*Artificial Intelligence*, en adelante: *AI*), campo perteneciente a la Ingeniería Informática, se complementa con la **Ciencia Cognitiva**, campo de la Psicología, estudia la forma en la que las personas piensan y aprenden; y la **Educación** se enfoca en cómo proporcionar la mejor enseñanza [Woolf 2009, pp. 42-45]. Por lo anteriormente explicado, se puede decir que un **Sistema Tutor Inteligente** es aquel que aplicando métodos de la Inteligencia Artificial apoyados

²<http://www.illuminate.com>

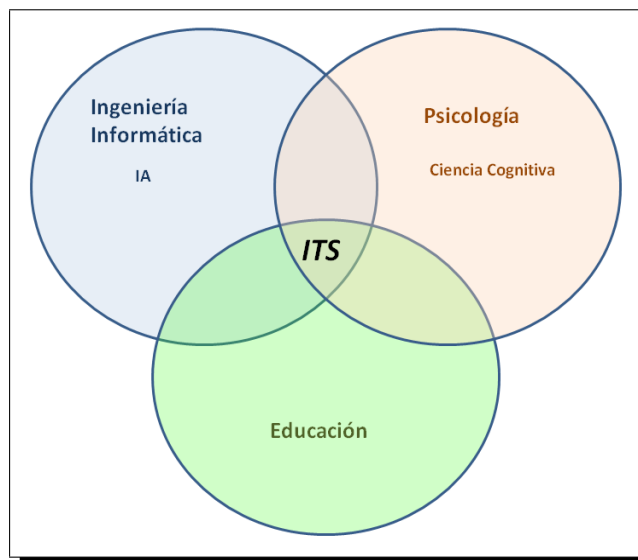


Figura 1.1: Disciplinas relacionadas con los Sistemas Tutores Inteligentes. Fuente: figura 2.10 *The field of artificial intelligence and education is grounded in three disciplines: computer science, psychology, and education* en [Woolf 2009]

con métodos de la Psicología es capaz de mejorar el aprendizaje. En este sentido, [Freedman 2000] define a un *ITS* de manera amplia como cualquier programa de ordenador más o menos inteligente que se utiliza para el aprendizaje. Dicho en otras palabras, una de las características que distinguen a los *ITS* de los otros sistemas es su ámbito de aplicación, i.e., la enseñanza. Consecuentemente, se puede considerar que los *ITS* pertenecen al conjunto de los Sistemas *e-Learning*.

1.1.3. Sistemas *Hypermedia* Adaptativos Orientados a la Enseñanza

El término **hypermedia** se utiliza para designar a documentos que integran información en distintos tipos de formato: texto, imágenes, sonido y vídeo. También, *Hypermedia* es un subconjunto de la multimedia interactiva, y se utiliza para denominar a todo sistema multimedia que tiene conectados elementos de información y los presenta de forma conjunta [Woodhead 1991]. El concepto **adaptación** se refiere a la forma en la que la aplicación cambia su comportamiento (e.g. presentación de contenidos adecuados, navegación en los contenidos) según las necesidades o preferencias de los usuarios.

El área *Adaptive Hypermedia* (*AH*) nació en los comienzos de 1990 como la unión de dos grandes áreas: *Hypertext* y *User Modeling*. Una limitación de los sistemas *hypermedia* tradicionales es que proporcionan los mismos contenidos y conjuntos de enlaces a otros contenidos a todos los usuarios. Si la población de usuarios es relativamente diversa, estos sistemas no satisfarán los objetivos de todos los usuarios. Los

Sistemas Hypermedia Adaptativos (*Adaptive Hypermedia System*, en adelante: *AHS*) tratan de dar respuesta a este problema por medio de la personalización³ de la información ofrecida en función de las necesidades o preferencias de los usuarios [Brusilovsky 1996]. Brusilovsky define un *AHS* como todo sistema *Hypermedia* que almacena alguna de las características del usuario en el modelo de usuario y utiliza este modelo para adaptar varios aspectos visibles del sistema [Brusilovsky 2001]. En otras palabras, este sistema debería satisfacer tres premisas:

- Debe ser un sistema *Hypermedia*.
- Debe tener un modelo de usuario.
- Debe ser capaz de adaptar la navegación entre contenidos y la forma de presentarlos utilizando este modelo.

En el caso de los **Sistemas Hypermedia Adaptativos Orientados a la Enseñanza** (*Adaptive Educational Hypermedia Systems*, en adelante: *AEHS*) su objetivo es similar a los anteriores, pero están orientados mejorar el proceso de aprendizaje [Brusilovsky 2001]. En este sentido, en el ámbito de los sistemas *e-Learning* y en el caso particular de los sistemas *AEH* hablaremos de **estudiantes** en lugar de usuarios, pues éstos serán los usuarios finales de estos sistemas.

Los sistemas *AEH* toman las decisiones de adaptación en función de un **modelo de estudiante** que almacena los posibles parámetros de personalización. Este modelo generalmente contiene las características del estudiante y su contexto. Por ejemplo, el modelo de estudiante puede incluir aspectos como estilo de aprendizaje, conocimiento previo, localización, dispositivo utilizado, etc.

Como se observa en la figura 1.2 los *AEHS* se encuentran en la intersección de los sistemas *e-Learning* y los *AHS*. Hay que indicar que los sistemas *e-Learning* pueden no ser adaptativos y los sistemas hypermedia adaptativos pueden no estar orientados a la enseñanza. Respecto a la diferencia entre *ITS* y *AEHS*, Weber y Brusilovsky citan que a principios de 1995 muchos de los *ITS* existentes se transformaron para adaptarlos a la *Web* formando los primeros *AEHS* [Weber 2001]. Por tanto, se puede considerar que no todos los *AEHS* son *ITS*, ya que existen *ITS* que no son sistemas *Web* como *GUIDON*.

Un ejemplo actual de *AEHS* lo constituye el sistema *MetaTutor* [Azevedo 2008]. Su objetivo es fomentar el estudio *on-line* sobre los aparatos del cuerpo humano: aparato circulatorio, aparato digestivo, sistema nervioso. Ejemplos del éxito de utilización de estos sistemas en el ámbito universitario son: *TANGOW* [Carro 1999], *ELM-ART* [Weber 2001], *AHA!* [de Bra 2002b], *QuizGuide* [Brusilovsky 2004] y *CoMoLE* [Martín 2008]. Además, en otros ámbitos como la educación musical también se utilizan los *AEHS*. Un ejemplo de aplicación es el sistema *ProLobe* [Eckhardt 2009], que ofrece actividades para desarrollar una destreza auditiva. El objetivo de este sistema es que el estudiante consiga desarrollar lo que se denomina un oído absoluto⁴.

³Proceso de presentar la información de una manera adecuada a cada usuario.

⁴Reconocer el sonido de una nota musical

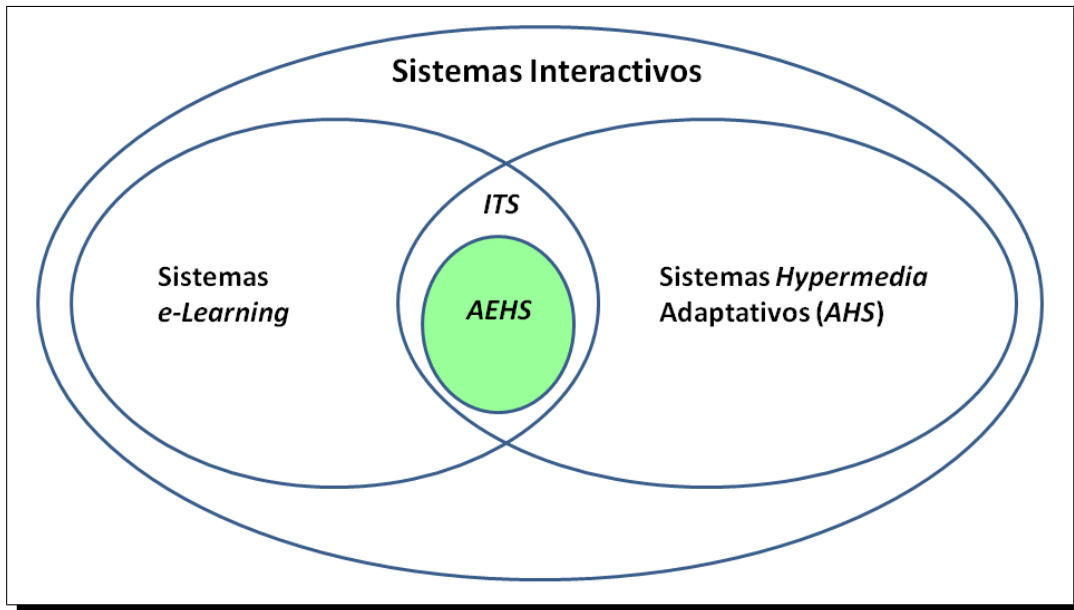


Figura 1.2: Contexto de los Sistemas *Hypermedia* Adaptativos Orientados a la Enseñanza

1.2. Motivación

La tarea de evaluar un curso *e-Learning* es compleja y en muchas ocasiones inabordable para los profesores. Por este motivo deciden no realizarla, o bien la realizan de forma inadecuada, pues no disponen de métodos o herramientas que les asistan en esta tarea.

La principal dificultad a la que se enfrentan los profesores es el hecho de no conocer de forma directa las reacciones y comportamientos de los estudiantes, ya que en los sistemas *e-Learning* el profesor no suele estar físicamente presente en el momento en el que éstos realizan el curso. Esta dificultad se deriva de las nuevas posibilidades que ofrecen estos sistemas permitiendo su uso en distintos dispositivos y/o localizaciones. Por ejemplo, el estudiante Carlos puede estar siguiendo el curso *e-Learning* en una *PDA*, mientras que otro estudiante, Jorge, lo está siguiendo en su ordenador portátil, y Gonzalo, otro estudiante, realiza las actividades de este curso en el ordenador personal de su casa. Es evidente que el profesor no conocerá las reacciones y comportamientos de los tres estudiantes, ya que no puede estar físicamente presente en las tres localizaciones. No obstante, la mayoría de estos sistemas almacenan las interacciones de los estudiantes en logs (bases de datos relacionales o ficheros). Estos logs contienen entre otras cosas las acciones y resultados de los estudiantes en el curso. Usualmente cada entrada de los logs refleja la acción de un determinado estudiante en un determinado espacio de tiempo, en consecuencia, el análisis de estos logs puede ofrecer al profesor las reacciones y comportamientos de los estudiantes.

Los logs pueden contener datos sobre las respuestas de los estudiantes en las actividades, el recorrido de éstas que siguieron, el número de veces que se consultó la ayuda o se volvió al índice de actividades, número de veces que se les proporcionan consejos para resolver las actividades, número de respuestas correctas o incorrectas, tiempo empleado en resolver una actividad, etc. Como puede observarse, la información anterior, que puede ser extraída a través del *análisis de logs*, puede ser muy valiosa para los profesores, debido a que refleja el comportamiento de los estudiantes. Por tanto, si los profesores fueran capaces de extraer esta información podrían modificar el curso, al igual que lo hacen en las clases tradicionales modificando sus explicaciones en función de la observación directa de los estudiantes.

El *análisis de logs* plantea nuevos retos para los profesores o evaluadores de cursos debido a dos obstáculos principalmente: el tamaño de los logs y la dificultad de utilizar métodos y herramientas para su análisis⁵. Para entender por qué el tamaño es un obstáculo se presenta el siguiente ejemplo. Supongamos que un curso está compuesto por 16 actividades, 8 de tipo teórico y 8 de tipo práctico, las actividades deben ser realizadas en el mismo orden por todos los estudiantes, y nuestro sistema sólo almacena las acciones de comenzar (caso de las actividades de tipo teórico) o resolver una actividad (caso de las actividades de tipo práctico). Cada estudiante generará en media 16 líneas y consecuentemente 100 estudiantes en torno a 1600 líneas. El análisis de un fichero de 1600 líneas, aún utilizando una hoja de cálculo es una tarea engorrosa (en el sentido de que consume tiempo) para los profesores. Si además el sistema del ejemplo ofreciera la oportunidad de repetir aquellas actividades que no fueron resueltas de forma correcta hasta 10 veces, un estudiante podría generar más de 40 líneas y consecuentemente 100 estudiantes generarán en torno a 4000 líneas. Si el sistema anterior además adaptara los contenidos según el nivel de conocimientos adquirido por los estudiantes (e.g. novato, avanzado y experto) un estudiante no aumentaría el número de líneas generadas en gran medida, pero sí la complejidad de analizarlas, pues los logs serán muy heterogéneos en el sentido de que no todos siguieron el mismo recorrido de actividades. Hay que indicar que este último caso puede corresponder a un *AEHS* o un *ITS*. Por tanto, el análisis de los logs de un sistema *e-Learning* es complejo y costoso. De hecho, dicha complejidad aumenta a medida que éste ofrece mayores posibilidades de adaptación.

Llegados a este punto, es claro que los profesores se tienen que enfrentar al análisis de grandes volúmenes de datos. Ahora la dificultad reside en la manera de analizar estos datos. Dicho de otro modo, se deben seleccionar los métodos y técnicas que puedan ser útiles para realizar esta tarea. En las últimas décadas la **Minería de Datos** ha demostrado en ámbitos como el comercio electrónico que posee métodos y técnicas para analizar con éxito grandes cantidades de datos. No es fácil comprender la dificultad a la que se enfrenta el profesor y la ayuda que pueden obtener de *DM* sin un ejemplo gráfico. Se puede entender que los logs están almacenados en una cajonera sin ningún orden, donde cada cajón contiene los logs

⁵Generalmente los profesores no suelen tener conocimientos suficientes para utilizar y entender los resultados de métodos y herramientas que puedan analizar los logs. Por esta razón, se considera un obstáculo para el análisis.

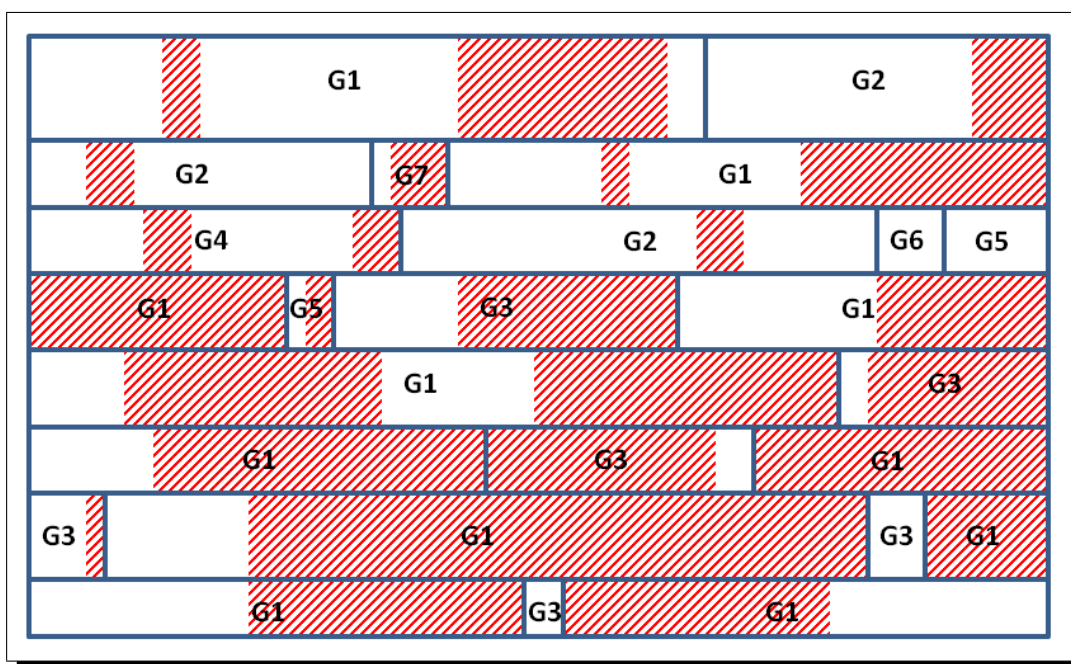


Figura 1.3: Distribución de los logs en el cajón de la actividad *act1*

de los estudiantes que realizaron una determinada actividad. Por tanto, la cajonera tendrá tantos cajones como actividades existan en el curso. El problema al que se enfrenta el profesor es cómo buscar un determinado tipo de información dentro de cada “cajón desordenado”. La figura 1.3 muestra la posible distribución de los logs dentro del cajón correspondiente a la actividad *act1*. En ésta se representan los resultados obtenidos por grupos de estudiantes (grupos en la figura: *G1*, *G2*, *G3*, *G4*, *G5*, *G6* y *G7*). Cada uno de los rectángulos con borde grueso representan los logs de un grupo de estudiantes. Se indica dentro de éstos a los estudiantes que no resolvieron correctamente la actividad con un rectángulo sombreado en rojo, y a los que la resolvieron de forma correcta con un rectángulo sin sombreado. Observando la figura no es fácil distinguir si un grupo presentó más dificultades que otro, ya que los logs de los grupos se encuentran en distintas partes del cajón. Sin embargo, unos datos mejor organizados pueden ayudar en esta tarea. Esta es la situación que se muestra en la figura 1.4, donde los logs están dispuestos de forma ordenada. De esta forma, es fácilmente observable que el grupo *G1* tuvo un gran número de respuestas incorrectas, y que los grupos minoritarios como el *G3* y *G7* mostraron también dificultades, hechos que no eran fácilmente observables en la figura anterior. En definitiva, la idea que se quiere transmitir con este ejemplo es que determinados métodos de *DM* (las técnicas de clasificación) establecen un orden en los datos. Dicho de otro modo, ordenan cada uno de los cajones de forma que se facilite la búsqueda o extracción de determinado tipo de información.

En este punto, se considera que dos son los problemas a los que se enfrentan

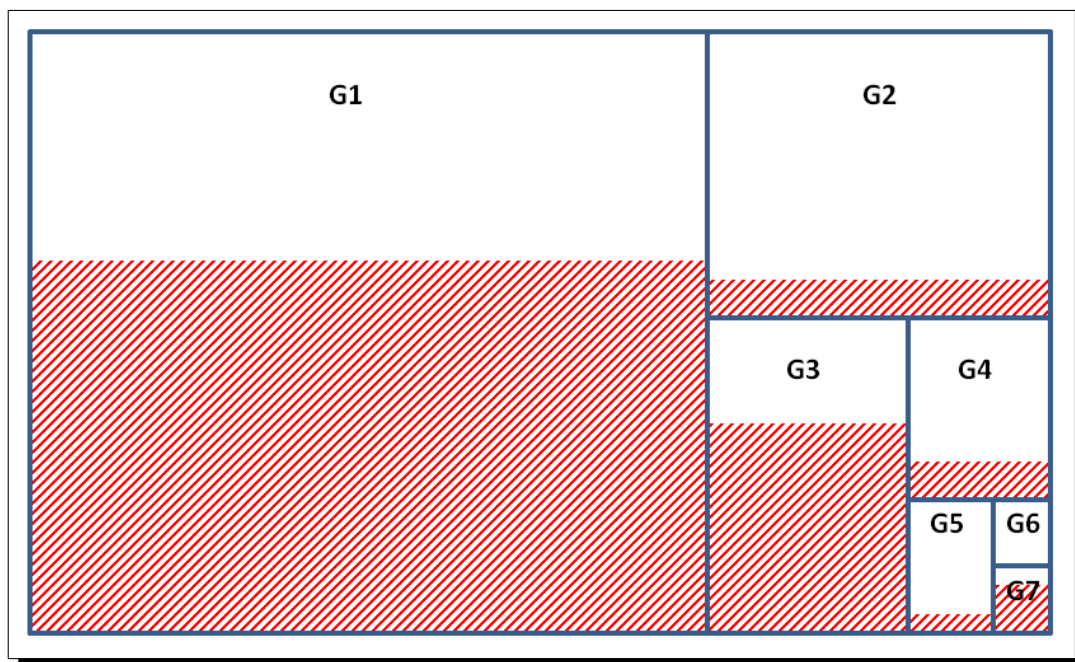


Figura 1.4: Distribución de los logs en el cajón de la actividad *act1* después de aplicar *DM*

los profesores que utilicen métodos de *DM* para analizar los logs: cómo extraer la información y entender el orden establecido. En primer lugar, aunque es cierto que se da un gran paso en el análisis de logs al incorporar métodos de *DM*, sigue siendo complicado extraer la información relevante de entre todos los cajones. En segundo lugar, el orden establecido (forma de clasificar los datos) por métodos de *DM* puede no ser entendible por los profesores, ya que generalmente no poseen los conocimientos necesarios en esta disciplina.

Esta tesis, entre otras cosas, propone un método basado en técnicas de *DM*, capaz de proporcionar la información más relevante de los logs a los profesores. Dicha información suministrada es útil para advertir al profesor sobre aquellos elementos del curso que requieren mayor atención para mejorar el proceso de aprendizaje de los estudiantes. Dicho de otro modo, se informa al profesor sobre los problemas mostrados por determinados grupos de estudiantes, seleccionando aquellos que son más relevantes, y que por tanto requieren mayor atención. Es importante destacar que el método de detección de problemas que se propone en este estudio presenta la información anterior de forma que pueda ser entendible por un profesor. Con esto se consiguen dos objetivos: informar sobre los principales problemas en el curso, y que dicha información sea comprensible por personas con poco o ningún conocimiento en *DM*.

1.3. Solución propuesta

En la sección anterior se han explicado las dificultades que entraña la evaluación de un curso *e-Learning* para los profesores. Hay que hacer especial hincapié en que ésta no es igual a la evaluación de los estudiantes, debido a que el objetivo de la primera es comprobar si la estructura y contenidos del curso son adecuados para el aprendizaje de todos los estudiantes, mientras que la segunda se refiere a la manera de evaluar la adquisición de conocimientos.

Un curso *e-Learning* es un conjunto de actividades educativas, las cuales incluyen los contenidos educativos que se quieren transmitir. Es el sistema *e-Learning* el que debe tomar ciertas decisiones para el mayor aprovechamiento de los estudiantes como los contenidos que se deben presentar, la forma de presentarlos, o cuándo sugerir recomendaciones o ayudas. En este trabajo, entre otras cosas, se pretende evaluar el conocimiento implícito en el curso y las decisiones de adaptación tomadas por el sistema. En el caso de sistemas no adaptativos la evaluación de un curso es equivalente a evaluar cómo han sido transmitidos los contenidos sin tener en cuenta las decisiones de adaptación. No obstante, en el caso de los cursos *AEH* es necesario evaluar cómo han sido transmitidos los contenidos y si se tomaron las decisiones de adaptación correctas. Estos cursos tienen la particularidad de que algunos contienen en su propia estructura las decisiones de adaptación y otros no las contienen porque de manera externa es el sistema en que las contiene. En el caso de estos últimos no sólo se evaluará el curso, sino también las decisiones de adaptación tomadas por el sistema. Por eso, en los casos en que el sistema no sea un simple contenedor interesa evaluar el sistema, sino normalmente hablaremos de evaluar cursos. Por supuesto, no se olvida que la evaluación también incluye evaluar la usabilidad y cada uno de los elementos que componen un sistema. No obstante, este trabajo está enfocado a la evaluación del funcionamiento del sistema, sin prestar especial atención a la usabilidad, la evaluación del sistema de administración de cursos y sistema de administración de usuarios⁶.

Un medio para la evaluación de estos sistemas consiste en analizar las interacciones de los estudiantes, es decir, llevar a cabo el análisis de los logs. Dicho análisis plantea dos nuevas dificultades, por un lado el tamaño de los logs de estos sistemas suele ser demasiado grande para intentar su análisis por los métodos estadísticos tradicionales. Por otro, debe decidirse la manera de analizar estos datos, es decir, seleccionar el tipo de información relevante para los docentes y el modo de buscarla. Esta tesis entre otras cosas propone utilizar técnicas y métodos de *DM* para el análisis de los logs.

En esta línea se presenta la propuesta de un nuevo método capaz de extraer la información que puede ser útil para los profesores. Este método, denominado *GeSES*, se apoya en la técnica *reglas de decisión* (*decision rules*, en adelante: *DR*) y un modelo matemático que permite la selección de este tipo de información. El proceso consiste en detectar signos que pueden indicar una disminución de la eficacia

⁶Existen trabajos en la literatura que ofrecen soluciones a estos tipos de evaluación como el realizado por [Weibelzahl 2002a].

en el sistema. En este sentido, estos signos ponen en evidencia problemas que pueden ser la causa de la disminución de la eficacia. Éstos son definidos como aquellas situaciones consideradas problemáticas en función de la característica que se quiere evaluar. Por ejemplo, si el profesor decide que la característica a evaluar son las puntuaciones de los estudiantes en las actividades, un posible signo sería que un grupo de estudiantes obtenga puntuaciones más bajas que la media, o bien si la característica es el tiempo empleado en resolver las actividades, otro signo podría ser que un grupo de estudiantes dedique más tiempo que la media. Éstos son ejemplos del tipo de situaciones que se pueden detectar con el método *GeSES*. Además, este método suministra de forma resumida y fácilmente entendible una selección de los signos que se consideran más importantes. De esta forma, se posibilita que un profesor, generalmente sin grandes conocimientos en *DM*, pueda ser informado sobre las posibles mejoras y/o modificaciones que se necesitan realizar en el sistema para su máximo aprovechamiento.

1.4. Objetivos de la tesis

En las secciones anteriores se han explicado: el contexto en el que se desarrolla esta investigación, las razones que impulsaron a realizarla y, a grandes rasgos, la solución propuesta para resolver los problemas planteados. En esta sección se exhiben los objetivos que se han seguido a lo largo de esta tesis. El principal objetivo es:

- **Facilitar la evaluación de un curso o sistema *e-Learning*.**

Este objetivo primordial, en torno al cual gira esta investigación, se refiere a ayudar a los profesores en la tarea de evaluación de un curso *e-Learning*, ya que como se ha explicado es una tarea compleja y difícil de realizar. Dicho en otras palabras, se pretende proporcionar ayuda, de manera que los docentes sean capaces de realizar el análisis de las interacciones de los estudiantes. Este objetivo no puede ser satisfecho sin haber completado los siguientes subobjetivos:

1. *Analizar y seleccionar las técnicas que permitan el análisis de grandes volúmenes de datos.* Este subobjetivo comprende la revisión y análisis de trabajos relacionados con la evaluación de sistemas y la búsqueda y selección de las técnicas de *DM* más apropiadas para el análisis de los datos.
2. *Definir la información que se desea conocer.* Este subobjetivo se refiere a definir el tipo de situaciones que pueden ser la causa de una reducción de la eficacia de un sistema *e-Learning*. La consecución de este punto es fundamental para desarrollar un método capaz de detectar estas situaciones.
3. *Desarrollar un método de evaluación a partir de la consecución de los subobjetivos 1 y 2.* Este subobjetivo incluye que el método desarrollado sea capaz de detectar la información definida en el subobjetivo anterior.

4. *Presentar la información extraída con el método de forma que sea entendible por un profesor.* Este punto es necesario para que los resultados obtenidos por el método sean usables. En este sentido, se añade una nueva característica al método, i.e., seleccionar la información más relevante.
5. *Evaluar la eficacia del método.* Aquí lo que se busca es comprobar si el método consigue su objetivo. Además, este punto es de vital importancia para conocer las limitaciones propias del método y medir su fiabilidad bajo determinadas condiciones.
6. *Implementar una herramienta de evaluación con la que se pueda aplicar el método.* Este punto se refiere a desarrollar una herramienta que sea capaz de analizar los logs y de extraer la información relevante para los profesores. De esta forma, esta herramienta informa a los profesores sobre los elementos del curso que necesitan ser revisados, o que requieren una supervisión por parte de los profesores.

1.5. Método de trabajo

Esta tesis refleja el trabajo de cinco años de investigación en la cual se han seguido siete fases para satisfacer los objetivos propuestos. Por supuesto, la octava fase, no incluida en la siguiente enumeración, consistió en la presentación de resultados mediante la redacción del presente documento. Hay que indicar que en cada una de las fases también se realizó la presentación de resultados mediante otras publicaciones, que se exponen en la sección 1.7. Consecuentemente el método de trabajo se dividió en las siguientes fases:

- **Fase 1. Revisión de los trabajos relacionados.** Esta fase comprendió la revisión y análisis de los trabajos en las áreas de evaluación de sistemas y aplicación de la *Minería de Datos* a los sistemas *e-Learning*. Los resultados de esta fase se exponen en el capítulo 2, *Estado de la Cuestión*.
- **Fase 2. Análisis y búsqueda de técnicas que sean capaces de analizar grandes volúmenes de datos.** En esta fase se analizaron distintas técnicas de *DM*, seleccionando aquellas que se consideraron más adecuadas para la presente investigación. En el capítulo 3, *Minería de Datos*, se exponen los principales conceptos de las dos técnicas seleccionadas: las reglas de asociación y los árboles de decisión.
- **Fase 3. Diseño de una arquitectura de evaluación para los cursos e-Learning.** El objetivo de esta fase fue incluir en el mantenimiento de un curso *e-Learning* el proceso de evaluación, así como la validación de éste mediante técnicas de simulación. Por este motivo, parte de esta fase es el diseño e implementación de *SimuLog*, una herramienta de simulación de logs. El capítulo 7, *Herramientas para la Evaluación*, contiene la descripción de la arquitectura y una extensa explicación sobre la herramienta de simulación.

- **Fase 4. Establecimiento de la información que se desea extraer del análisis de logs.** Esta fase comprendió la definición de los elementos que podrían ser signos de baja eficacia en el curso. El capítulo 4, *Signos de disminución de la eficacia en sistemas o cursos e-Learning*, incluye la definición de estos elementos que en esta tesis los denominamos *signos de baja eficacia*, así como una clasificación de éstos.
- **Fase 5. Diseño de un método para detectar signos de baja eficacia en el curso.** Esta fase consistió en el diseño inicial de un método que se mejoró progresivamente mediante distintas pruebas hasta llegar a obtener la propuesta final. El capítulo 5, *Métodos para detectar signos*, explica el proceso llevado a cabo para la obtención del método **GeSES**, propuesta final del método de detección de signos.
- **Fase 6. Valorar la eficacia del método *GeSES*.** Esta fase fue de vital importancia para conocer los límites propios del método. El capítulo 6, *Aplicación y Evaluación de la propuesta*, expone el proceso de validación que se realizó en el que se utilizó la herramienta de simulación *SimuLog*.
- **Fase 7. Implementación de un prototipo que aplique el método *GeSES*.** El resultado de esta fase fue la implementación de **ASquare**, la herramienta que ayuda a los profesores a evaluar un curso *e-Learning* mediante la aplicación del método *GeSES*. El capítulo 7 incluye una detallada descripción de *ASquare*.

1.6. Contribuciones de la tesis

La presente tesis propone un **nuevo método** para la evaluación de cursos *e-Learning*. Aunque es cierto que existen otros modelos de evaluación como los que se pueden encontrar en el estudio realizado por Rubio [Rubio 2003] y los modelos propuestos por Buendía y Hervás [Buendía 2006b], estos modelos realizan un análisis superficial del funcionamiento de los sistemas *e-Learning*. En otras palabras, indican una serie de pautas o pasos generales, pero no especifican de forma concreta cómo realizar estos pasos. Por ejemplo, no dicen qué tecnología o técnicas de análisis son las más adecuadas. El método presentado en esta tesis consta de una serie de pasos, especificados de forma concreta y precisa, con el fin de que un profesor que los siga sea capaz de evaluar la eficacia de un curso *e-Learning*. Ésta es la principal y más importante contribución de esta tesis, proporcionar un método para la evaluación que pueda ser utilizado por un profesor. Las principales contribuciones que esta tesis realiza a la comunidad científica se pueden resumir en:

1. *Evaluación de un curso e-Learning mediante el análisis de logs.* En este trabajo se utiliza el análisis de logs para detectar aquellas situaciones que necesitan ser revisadas por el profesor. Por tanto, el método presentado está orientado a ayudar a profesores, y este objetivo difiere significativamente al de otros

autores como Zaïane [Zaïane 2006] y Merceron et al. [Merceron 2003], cuyo objetivo reside en proporcionar ayuda a los estudiantes.

2. *Aplicación de las reglas de decisión para la detección de signos de baja eficacia.* El método propuesto en esta tesis se apoya fuertemente en las *DR*, obtenidas a partir de un árbol de decisión, para descubrir las situaciones conflictivas en el curso. En este sentido, se utilizan los árboles de decisión no como un modelo predictivo sino como manera de representar los datos.
3. *Selección de los signos de baja eficacia más relevantes para los profesores.* El método *GeSES* selecciona la información que pueda resultar más útil para los profesores, informando de un conjunto reducido de los puntos débiles del curso. Con esta información el profesor puede corregirlos o mejorarlos, evitando que éste consuma tiempo resolviendo otros problemas de menor prioridad.
4. *Propuesta de valoración de la eficacia de una herramienta o método de evaluación.* Esta tesis presenta un modo de evaluar la eficacia del propio método o herramienta de evaluación mediante técnicas de simulación de perfiles de estudiantes y signos sintéticos. Esta propuesta constituye una posible forma de valorar la eficacia de la herramienta que evalúa la eficacia de un curso. Hay que destacar que valorar la eficacia del propio método o herramienta de evaluación también es necesario e importante, pues dicha evaluación servirá para conocer si el método consigue su objetivo. Adicionalmente, en esta evaluación se comprueba también la fiabilidad del método propuesto.

1.7. Publicaciones a las que ha dado lugar

Este trabajo ha dado lugar a **2 artículos en revistas** con índice de impacto, **9 artículos en congresos internacionales** entre los que cabe destacar los congresos *EDM* y *E-Learn*, **un capítulo de libro** y el **trabajo de iniciación a la investigación** (consúltese la dirección <http://publicationslist.org/javier.bravo> para ver un listado de las publicaciones). Hay que señalar que *EDM* agrupa a los mayores expertos en el área de *DM* aplicada a sistemas *e-Learning*. Por tanto, los trabajos más importantes y novedosos en este área se presentan en este congreso. *E-Learn* es uno de los congresos más importantes de *USA* en el área *e-Learning*; en él se presentan y discuten modelos de enseñanza utilizando las nuevas tecnologías. A continuación se exponen las publicaciones ordenadas en orden cronológico:

- Javier Bravo y Alvaro Ortigosa. *Validating the Evaluation of Adaptive Systems by User Profile Simulation*. En Proceedings of Fifth Workshop on User-Centred Design and Evaluation of Adaptive Systems held at the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), páginas 479-483, National College of Irland, Dublín, Irlanda, junio 2006. (ISSN: 1649-8623) [Bravo 2006c]

Este trabajo constituye la primera toma de contacto del presente autor de esta tesis con la investigación en el área de la Evaluación de sistemas *AEH*. Este artículo sienta las bases para diseñar y utilizar una herramienta de simulación para valorar la eficacia de una herramienta de evaluación. El resultado de este estudio fue la presentación de la primera versión de *SimuLog*.

- Javier Bravo y Alvaro Ortigosa. *Integración y Prueba de Herramientas de Evaluación en un Entorno Hipermedia Adaptativo*. En Proceedings of 8th International Symposium on Computers in Education (SIIE2006), páginas 253-261, Universidad de León, León, España, octubre 2006. (ISBN: 84-9773-302-9) [Bravo 2006b]

La necesidad de diseñar métodos de evaluación eficientes para evaluar el funcionamiento de los sistemas *AEH* es el núcleo principal de este trabajo. Se propone además una arquitectura de evaluación en la que se incluye la herramienta de evaluación en el contexto de los *AEHS*.

- Javier Bravo. Mecanismos de Integración y Prueba de Herramientas Automáticas de Evaluación de Calidad de Cursos Adaptativos. *Trabajo de Iniciación a la Investigación*, Escuela Politécnica Superior, UAM, Madrid, España, septiembre 2006. [Bravo 2006a]

Este trabajo presenta una arquitectura en la que se integran las herramientas de evaluación para los sistemas *AEH*, y se propone una forma de validar sus resultados con métodos de simulación. La herramienta de simulación que se describe de forma amplia es *SimuLog*, simulador de logs de estudiantes.

- César Vialardi, Javier Bravo y Alvaro Ortigosa. *Empowering AEH Authors Using Data Mining Techniques*. En Proceedings of Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H2007) held at the 11th International Conference on User Modeling (UM2007), páginas 33-43, Corfu, Grecia, junio 2007. [Vialardi 2007]

Este artículo expresa los motivos por los que es importante desarrollar métodos y herramientas que ayuden a los profesores y diseñadores de cursos a mantener sistemas *AEH*. Se propone analizar las interacciones de los estudiantes mediante el uso de *DM* utilizando dos de sus técnicas, los árboles de decisión y las reglas de asociación. Además, se presenta el diseño de la primera versión de *ASquare*, herramienta de evaluación que proporciona ayuda a los profesores en la tarea de evaluar un sistema de este tipo. Una extensión de este artículo se encuentra publicado en la revista *J.UCS* 14(17).

- Javier Bravo, César Vialardi y Alvaro Ortigosa. *A Problem-Oriented Method for Supporting AEH Authors through Data Mining*. En Proceedings of International Workshop on Applying Data Mining in e-Learning (ADML07) held at the Second European Conference on

Technology Enhanced Learning (EC-TEL2007), páginas 53-62, Creta, Grecia, septiembre 2007. (ISSN: 1613-0073) [Bravo 2007]

Este estudio expone los principales problemas al valorar si las decisiones de adaptación tomadas por los sistemas *AEH* son beneficiosas para todos los estudiantes o existe algún grupo que no le beneficiaron. En este sentido, el artículo propone utilizar técnicas de *DM* para detectar potenciales problemas de adaptación en los *AEHS*. En consecuencia, se presenta un método capaz de detectar estos problemas basado en los árboles de decisión.

- Javier Bravo, César Vialardi y Alvaro Ortigosa. *ASquare: A Powerful Evaluation Tool for Adaptive Hypermedia Course System*. En Proceedings of Hypertext 2008 Conference, páginas 219-220, University of Pittsburgh, Pittsburgh, USA, junio 2008. (ISBN: 978-1-59593-998-2) [Bravo 2008a]

Este artículo presenta la necesidad de utilizar métodos de *DM* para ayudar en las tareas de diseño y evaluación de cursos adaptativos. Además, se muestran las posibilidades de la versión mejorada de *ASquare*.

- Javier Bravo, César Vialardi y Alvaro Ortigosa. *Using Decision Trees for Discovering Problems on Adaptive Courses*. En Proceedings of E-Learn 2008: World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, páginas 268-277, Las Vegas, USA, noviembre 2008. (ISBN: 1-880094-66-5) [Bravo 2008b]

Este artículo muestra la necesidad de proveer a los profesores de métodos para que puedan realizar la tarea de evaluar un curso *AEH* para así poder mejorar su eficiencia. En concreto, se presenta un método que utiliza la técnica de los árboles de decisión para detectar problemas de adaptación, y dos experimentos con logs reales utilizando dicho método.

- César Vialardi, Javier Bravo y Alvaro Ortigosa. Improving AEH Courses through Log Analysis. *Journal of Universal Computer Science - J.UCS* 14(17), páginas 2777-2798, febrero 2009. [Vialardi 2009a]

El artículo publicado en esta revista presenta de forma detallada cómo realizar el análisis de logs mediante el uso de técnicas de *DM*: árboles de decisión y reglas de asociación. El objetivo de este análisis es proporcionar ayuda a los profesores en las tareas de diseño y evaluación de sistemas *AEH*. Además, se muestra de forma amplia a *ASquare* así como un ejemplo de su validación mediante *SimuLog*.

- Javier Bravo y Alvaro Ortigosa. *Detecting Symptoms of Low Performance Using Production Rules*. En Proceedings of Second Educational Data Mining conference, editado por: T. Barnes; M.

Desmarais; C. Romero; S. Ventura, páginas 31-40, Universidad de Córdoba, Córdoba, España, julio 2009. (ISBN: 978-84-613-2308-1) [Bravo 2009b]

Este artículo presentado en el congreso *EDM*, congreso de especial relevancia en el área de esta investigación, plantea la conveniencia de desarrollar métodos que ayuden a los profesores a evaluar sistemas *e-Learning*. Estos métodos deben ser capaces de procesar grandes cantidades de datos, ya que estos sistemas almacenan las interacciones de los estudiantes en forma de logs, que tienen un tamaño suficientemente grande para que sea difícil su análisis mediante las técnicas tradicionales. En este sentido, se propone realizar la tarea de evaluación por medio de la detección de potenciales síntomas de bajo rendimiento en los cursos *e-Learning* mediante el análisis de logs. Con este objetivo se presenta un método de detección de estos síntomas, utilizando las reglas de decisión *C4.5* y un filtrado que selecciona las reglas más relevantes. Además, este enfoque se probó con los logs de los estudiantes de la *University of Pittsburgh*. Hay que indicar que el método propuesto constituye la base sobre la que se desarrolló el método *GeSES*.

- César Vialardi, Javier Bravo, Leila Shafti y Alvaro Ortigosa. *Recommendation in Higher Education Using Data Mining Techniques*. En Proceedings of Second Educational Data Mining conference, editado por: T. Barnes; M. Desmarais; C. Romero; S. Ventura, páginas 190-199, Universidad de Córdoba, Córdoba, España, julio 2009. (ISBN: 978-84-613-2308-1) [Vialardi 2009b]

Este artículo plantea el problema de recomendar asignaturas a los estudiantes con el fin de disminuir el fracaso en los centros educativos. En este sentido, este trabajo propone utilizar las reglas de decisión para predecir si un estudiante superaría con éxito o no una determinada asignatura teniendo en cuenta los resultados obtenidos por otros estudiantes de similar perfil.

- Javier Bravo, Estefanía Martín, Alvaro Ortigosa y Rosa M. Carro. *Checking the Reliability of GeSES: Method for Detecting Symptoms of Low Performance*. En Proceedings of workshop on Educational Data Mining held at the 9th International Conference on Intelligent System Design and Applications (ISDA09), páginas 1108-1113, Pisa, Italia. IEEE press. (ISBN: 978-1-4244-4735-0) [Bravo 2009a]

El artículo presentado en este congreso plantea que es importante incluir el proceso de evaluación en los sistemas *AEH* con el fin de mejorar sus rendimientos. Con este propósito se presenta la primera propuesta del método *GeSES*. Este método basado en el análisis de logs a través de las reglas de decisión *C4.5* es capaz de detectar síntomas de bajo rendimiento en entornos *AEH*. En este trabajo por un lado se comprueba la fiabilidad del método en entornos

reales, y por otro se consiguen detectar síntomas de bajo rendimiento en el entorno de aprendizaje *CoMoLE*.

- David Camacho, Estrella Pulido, Maria D. Rodríguez-Moreno, Rosa M. Carro, Alvaro Ortigosa y Javier Bravo. Automatic Course Redesign: global vs. individual adaptation. *International Journal of Engineering Education* 25(6), páginas 1270-1282, diciembre 2009. (ISSN: 0949-149X/91) [Camacho 2009]

Este artículo plantea añadir a los sistemas *AEH* los elementos de planificación y programación (*planning and scheduling*) con el fin de mejorar el proceso educativo. Este nuevo enfoque permitiría a los profesores detectar problemas en el curso *AEH*. Se añadió el módulo *IPSS (AI planning/scheduling system)* al sistema *TANGOW*. Este módulo permite detectar inconsistencias en el diseño del curso y propone modificaciones en el diseño para mejorar la calidad del curso. Con el objetivo de comprobar el rendimiento de este nuevo módulo se generaron logs sintéticos con *SimuLog*, y posteriormente estos logs se analizaron en el módulo *IPSS* obteniéndose sugerencias que los profesores consideraron útiles.

- Javier Bravo, César Vialardi y Alvaro Ortigosa. Using decision trees for improving AEH courses, capítulo 26. *Handbook of Educational Data Mining*. Editorial Taylor & Francis, 2010 (octubre). [Bravo 2010]

En este capítulo de libro se introducen diversos conceptos sobre la evaluación y se explican las ventajas de añadir el proceso de evaluación a los sistemas *AEH*. De este modo, se propone un modelo en espiral para la construcción y mantenimiento de cursos que gira en torno al análisis de logs mediante el uso de técnicas de *DM*. Se presenta el método *Key-node*, cuya característica es detectar potenciales problemas en un sistema *AEH*. Este método consiste en utilizar los árboles de decisión para analizar los logs y seleccionar aquellas ramas del árbol que contengan la información de potenciales problemas. La eficacia del método se comprobó mediante dos conjuntos de datos sintéticos con distintos parámetros de simulación, siendo un conjunto más afectado por el efecto aleatorio en las respuestas de los estudiantes que el otro. Estos datos contenían además una serie de problemas generados de forma artificial por *SimuLog*. Se concluyó que el método fue capaz de detectar los problemas generados con éxito en ambos conjuntos.

1.8. Estructura del documento

Esta tesis se ha dividido en 8 capítulos y un apéndice donde se explican con mayor detalle determinados elementos de los capítulos que por su extensión se ha decidido incluirlos en el apéndice. Al principio de cada capítulo se incluye su índice de contenidos, de esta forma se facilita que el lector pueda acceder de forma más

rápida a una sección o subsección determinada, evitando la consulta al índice general. Además, inmediatamente después del índice de contenidos cada capítulo contiene una breve descripción de lo que contiene, de esta manera se consigue preparar al lector para la lectura del capítulo.

También se incluyen al principio de este documento los índices de tablas y figuras, ordenadas por capítulos. Además, cabe señalar que en esta tesis se han utilizado abreviaturas o siglas, según la lista que figura al final, si bien la primera mención de ellas suele estar precedida tanto de su traducción al castellano como de su desarrollo en su lengua original.

Respecto a las citas bibliográficas cabe comentar que se ha elegido el formato:

[<primer-apellido-de-primer-autor><año-publicación><letra>]

Por ejemplo la cita bibliográfica [*Baker 2009*]⁷ indica que el apellido del primer autor es Baker y que el año de publicación es 2009, y la cita [*Zañane 2001b*] indica que es la segunda referencia⁸ de este autor en el año 2001. Cabe reseñar que la bibliografía está ordenada en orden alfabético y después por año de publicación, por tanto la referencia de la cita [*Zañane 2001a*] aparecerá por encima de la de [*Zañane 2001b*], y éstas aparecerán por debajo de la de [*Baker 2009*]; además se han incluido (si existen) en la bibliografía las *URL* y *DOI*, y las páginas donde son fueron citadas las referencias. Hay que indicar que el documento electrónico permite navegar de forma rápida por éste, ya que cualquier cita, página (incluidas las páginas donde fueron citadas las referencias bibliográficas), o referencia a una figura, tabla o cualquier parte del documento, es un enlace. Por tanto, el lector accede al elemento referenciado con sólo pulsar en el enlace. La distribución de los capítulos se puede resumir en los siguientes puntos:

Capítulo 1 Contiene la descripción del contexto en el que se desarrolla la presente investigación, las razones que fueron el impulso para su realización donde se exponen los problemas planteados, la solución propuesta a estos problemas, los objetivos que se persigue satisfacer, el método de trabajo seguido, las contribuciones que aporta esta investigación a la comunidad científica, las publicaciones a las que ha dado lugar este trabajo y la organización de este documento.

Capítulo 2 Se realiza una revisión de los modelos existentes en la evaluación de Sistemas Interactivos, *E-Learning* y Sistemas *Hypermedia* Adaptativos. Además, se exponen los principales trabajos en el área de *análisis de logs* relacionándolos en cada caso con el contexto de esta investigación.

Capítulo 3 Se introducen algunos conceptos básicos sobre la Minería de Datos y se explican las técnicas reglas de asociación, árboles de decisión y reglas de

⁷Las citas bibliográficas se presentan en color rojo en la versión electrónica, mientras que se utiliza el gris para la versión impresa. Esta decisión responde a facilitar la lectura de las citas.

⁸Las letras indican el orden de las publicaciones en el mismo año por el mismo autor. De esta forma, una referencia con la letra *b* indica que es la segunda publicación de este autor en este mismo año, y la letra *a* que es la primera publicación.

decisión, siendo la explicación de estas dos últimas más amplia. Además, se muestran las posibilidades que ofrecen las dos herramientas utilizadas: *Weka* y el paquete de programas *C4.5*.

Capítulo 4 Se definen los elementos que pueden causar disminución de la eficacia en los sistemas *e-Learning*, a estos elementos se les denomina *signos de baja eficacia*, cuya definición se incluye en este capítulo. Además, se presenta una propuesta de clasificación de signos.

Capítulo 5 Se desarrolla la metodología conducente a la obtención de métodos para detectar los signos. En primer lugar se expone una primera aproximación a la solución de los problemas planteados, el método *Key-node*. Este método sirve como base para el desarrollo de la propuesta final. En segundo lugar se incluye el proceso llevado a cabo para obtener el método *GeSES*, así como las pruebas realizadas.

Capítulo 6 Se muestra la aplicación del método *GeSES* y se expone el procedimiento seguido para valorar su eficacia. Dicho procedimiento consiste en emplear técnicas de simulación, en concreto utilizando *SimuLog*, para evaluar la eficacia del método.

Capítulo 7 Se expone una propuesta de arquitectura para crear y mantener cursos *e-Learning* que consiste en añadir en el ciclo de creación y mantenimiento de estos sistemas el método de evaluación propuesto. También se muestra de forma detallada la arquitectura, interfaz y elementos que componen *SimuLog*. Asimismo se presenta el prototipo de *ASquare*, herramienta de evaluación que aplica el método *GeSES*.

Capítulo 8 Se presentan las conclusiones obtenidas en esta investigación, y se mencionan posibles futuras líneas de investigación.

Estado de la Cuestión

Índice de contenidos

2.1. Introducción	29
2.2. Evaluación de Sistemas Interactivos	30
2.3. Evaluación de Sistemas <i>E-Learning</i>	31
2.4. Evaluación de Sistemas <i>Hypermedia</i> Adaptativos	33
2.4.1. Modelos de evaluación	33
2.4.2. Evaluación de Sistemas <i>Hypermedia</i> Adaptativos Orientados a la Enseñanza	37
2.5. Contexto de la presente propuesta	54

*Este capítulo muestra el contexto del presente trabajo de investigación. Se exponen distintas metodologías para evaluar distintos sistemas, haciendo especial hincapié en los sistemas *e-Learning* y adaptativos orientados a la enseñanza.*

2.1. Introducción

El uso y aparición de nuevos Sistemas Interactivos se ha incrementado de forma exponencial en los últimos años. Sin embargo, aunque estos sistemas ofrecen mayor interactividad y mayor flexibilidad, su mantenimiento y evaluación siguen siendo tareas complejas que requieren tiempo y experiencia, que en muchos casos los diseñadores no disponen. Especialmente en el caso de los sistemas *e-Learning*, los profesores suelen ser los propios diseñadores de: i) los contenidos educativos; ii) la secuencia de éstos que deben seguir los estudiantes para su máximo aprovechamiento; iii) la forma de presentarlos a los estudiantes. Es decir, diseñan un curso, pues son ellos los que poseen el conocimiento necesario, y en el caso de los sistemas adaptativos una visión global del curso.

Además, los profesores actúan también como evaluadores, normalmente con dos objetivos:

- Conocer el nivel de conocimiento alcanzado por cada estudiante en el curso, muchas veces con el objetivo de asignar una calificación.
- Conocer si el curso fue útil para los estudiantes, y de qué manera se podría mejorar.

Ambas tareas, el diseño y la evaluación, constituyen campos de investigación cada vez más demandados en la actualidad, por la inclusión de las nuevas tecnologías en las clases tradicionales. En especial, la evaluación de estos sistemas es compleja y costosa. No obstante, en las últimas décadas han existido intentos por establecer metodologías que ayuden a realizar la tarea de evaluación de los sistemas interactivos. Las siguientes secciones muestran las propuestas existentes en la literatura sobre evaluación de sistemas interactivos, sistemas *e-Learning*, y sistemas *Hypermedia* Adaptativos.

2.2. Evaluación de Sistemas Interactivos

El término evaluación es definido según la Real Academia Española como “estimación, apreciación, y cálculo del valor de algo”. **Worthen** et al. [**Worthen 1996**, p. 5] proponen una definición más compleja en la que definen la evaluación como “el proceso de identificación, aclaración y aplicación de criterios para determinar el valor, calidad, utilidad, efectividad o importancia del objeto evaluado en relación con los anteriores criterios”. **Karat** afirma que en todos los casos de evaluación hay un objeto que es evaluado (e.g., grabación musical, comida, personas) y un proceso en el que uno o más atributos de ese objeto son valorados (e.g., si el sabor de la comida es bueno, si la grabación es de alta calidad) [**Karat 1997**, p. 689]. Siguiendo esta idea, la evaluación de un sistema interactivo debe asegurar que el comportamiento del sistema es el esperado por su diseñador, y que el sistema satisface los requisitos de los usuarios [**Dix 2004**, p. 319]. De acuerdo con esto, **Dix** et al. proponen tres objetivos a satisfacer en la evaluación de un Sistema Interactivo [**Dix 2004**, pp. 319-320]:

1. Evaluar la funcionalidad del sistema: valorar si los requisitos de los usuarios están recogidos en el sistema, valorar qué funcionalidades ofrece el sistema a los usuarios. La evaluación en este nivel puede incluir medir el rendimiento del usuario con el sistema, para evaluar la efectividad del sistema.
2. Evaluar la experiencia del usuario en la interacción con el sistema: este objetivo está relacionado con aspectos de la usabilidad, es decir, valorar si es fácil de aprender el uso del sistema, y si el usuario está satisfecho con éste.
3. Identificar cualquier problema específico con el sistema: esta fase se refiere a identificar problemas en el diseño del sistema. Estos problemas pueden indicar la existencia de comportamientos inesperados por el sistema, o bien pueden ser la causa de posibles incongruencias que surgen entre las expectativas del usuario y el diseñador del sistema.

El campo de los Sistemas Interactivos es demasiado amplio para abarcarlo en esta tesis, por lo tanto el presente trabajo se enfoca en la evaluación de los Sistemas *e-Learning* y Sistemas *Hypermedia* Adaptativos. De esta forma, de acuerdo con el objetivo principal presentado en el capítulo 1, facilitar la tarea de evaluar un sistema

e-Learning a los profesores, buscaremos satisfacer las tres premisas necesarias para evaluar un Sistema Interactivo adaptadas al contexto de los Sistemas *e-Learning*. En particular, esta propuesta está principalmente centrada en satisfacer el primer punto (evaluar la funcionalidad del sistema) y el último (identificar cualquier problema específico con el sistema). Si bien, el segundo punto (evaluar la experiencia del usuario) se puede satisfacer de forma indirecta, pues uno de los objetivos es la detección de signos de disminución en la eficiencia del sistema, y en el caso particular de los signos asociados al rendimiento académico pueden ser indicadores de problemas de usabilidad.

2.3. Evaluación de Sistemas *E-Learning*

La evaluación de los Sistemas *E-Learning* plantea dos importantes problemas, cómo realizar la evaluación y qué evaluar. Respecto al cómo realizar la evaluación se pueden seguir las indicaciones de **Totterdell** y **Boyle** [Totterdell 1990a] que permiten la evaluación del software:

- Identificación de los propósitos u objetivos de la evaluación: la especificación de los criterios de evaluación es el principal objetivo de esta fase.
- Especificación del diseño de los experimentos: en esta fase se especifican los métodos, medidas y técnicas que más de adecúan a los criterios de evaluación.
- Recogida de resultados: dependiendo del método los resultados pueden conocerse a través de los ficheros de logs, observaciones del comportamiento, y cuestionarios a los usuarios.
- Análisis de los datos: tanto los análisis cuantitativos y como cualitativos pueden ser aplicados. El resultado de esta fase debería mostrar problemas específicos del sistema.
- Extraer conclusiones: esta fase consiste en interpretar los resultados del análisis para ofrecer posibles mejoras o modificaciones en el sistema.

La segunda pregunta pregunta no es fácil de responder, pues existen dos tendencias de evaluación del *e-Learning*, según señala **Rubio** en el estudio que realiza sobre los diferentes modelos de evaluación del *e-Learning* [Rubio 2003]. Las dos tendencias se basan en la evaluación mediante el enfoque parcial, o mediante el enfoque global.

- Evaluación mediante el enfoque parcial: está centrado principalmente en alguno de los siguientes aspectos:
 - Evaluación de la actividad formativa: es el proceso orientado a evaluar una acción concreta de formación, un curso *on-line* por ejemplo. Este tipo de evaluación persigue tres objetivos: comprobar el nivel de cumplimiento de los objetivos educativos, mejorar la acción formativa y determinar el retorno de la inversión realizada.

- Evaluación de los materiales de formación: es el proceso relativo a evaluar los materiales utilizados en el *e-Learning*, pueden ser textuales, hipertextuales o multimedia. La evaluación de materiales multimedia está centrada en los siguientes parámetros: evaluación de necesidades, del input, del proceso, del producto y de los resultados.
 - Evaluación de las plataformas tecnológicas: está orientada a valorar la calidad del entorno virtual en el que se desempeña el *e-Learning*.
 - Evaluación de la relación coste/beneficio: este tipo de evaluación está relacionada con factores de medición económicos. En este proceso cobra especial importancia el concepto de retorno de inversión (*Return On Investment*, en adelante: *ROI*) cuya fórmula es *beneficios/costes*. De esta forma el *ROI* puede justificar la importante inversión inicial para implantar un sistema *e-Learning*.
- Evaluación mediante el enfoque global: está centrado en el conjunto total de elementos que intervienen en un sistema *e-Learning*.
- Sistemas de evaluación centrados en modelos y/o normas de calidad estándar y calidad total. Consiste en aplicar el conjunto de normas de la calidad al ámbito *e-Learning*.
 - Sistemas basados en la práctica del *benchmarking*¹.

Respecto a la práctica del *benchmarking*, **Buendía** y **Hervás** proponen un modelo de evaluación que utiliza el *benchmarking* [Buendía 2006a, Buendía 2006b]. Este modelo está basado en tres áreas principales del modelo *Learning Platform Evaluation Model* desarrollado por *Quality Improvement Agency* [Ferl 2009]: Contenido (*Content*), Interacción (*Interaction*) y Gestión (*Management*). Cada una de las áreas contiene los siguientes elementos:

- Contenido: incluye las herramientas y servicios que permiten diseñar y presentar los contenidos de aprendizaje (e.g. editor de contenidos, fuentes externas, contenido multimedia, organización de contenido).
- Interacción: contiene las herramientas que permiten la interacción entre el profesor y los estudiantes, o entre grupos de estudiantes (e.g. e-mail, foros, chat).
- Gestión: incluye las herramientas que ayudan a valorar el proceso de aprendizaje de los estudiantes (e.g. evaluación, puntuaciones obtenidas).

Los autores proponen auditar cada uno de los elementos anteriores mediante el uso del *benchmarking*. Además, señalan que este modelo permite:

¹Benchmarking es el proceso que permite a un centro u organización compararse con otro que obtiene resultados excelentes de calidad, con el fin de emularlo. En el ámbito del *e-Learning* este proceso pretende mejorar los sistemas mediante herramientas e indicaciones basadas en buenas prácticas.

- Identificar las necesidades y requerimientos en el proceso de aprendizaje.
- Realizar una auditoría de las herramientas disponibles.
- Detectar si estas herramientas cubren las necesidades anteriores.

Los modelos de evaluación de Sistemas *E-Learning* realizan un análisis muy superficial del funcionamiento de éstos, es decir, indican una serie de pautas o pasos generales que se deben seguir en la evaluación, pero no especifican cómo realizar estos pasos, i.e., qué tecnología se puede utilizar o técnicas de análisis adecuadas. Debido a la amplitud de la tarea de evaluar un sistema *e-Learning*, esta tesis se centra en establecer un método que permita evaluar la funcionalidad de un sistema *e-Learning*.

2.4. Evaluación de Sistemas *Hypermedia* Adaptativos

La evaluación de Sistemas *Hypermedia* Adaptativos es compleja y costosa como ya se ha indicado anteriormente. Por eso, diferentes modelos de evaluación han sido propuestos en la literatura con el objetivo de facilitar esta tarea. Esta sección presenta los modelos de evaluación más relevantes, así como el caso particular de la evaluación de estos sistemas orientados a la enseñanza.

2.4.1. Modelos de evaluación

Esta sección presenta dos modelos de evaluación de sistemas adaptativos existentes en la literatura. La propuesta de **Weibelzahl** está centrada en la evaluación del sistema mediante la evaluación por capas, mientras que la propuesta de **Paramythhis** es relativa a la evaluación de la interfaz de un sistema adaptativo mediante un enfoque modular. Ambas propuestas se muestran en los siguientes apartados.

2.4.1.1. Propuesta de Weibelzahl

Los métodos empíricos son utilizados generalmente en algunas áreas de la Inteligencia Artificial. Dos ejemplos de su aplicación son: valorar la efectividad de los algoritmos de búsqueda en dominios estandarizados y evaluar los algoritmos de aprendizaje con conjuntos de datos reales. Además, Weibelzahl indica que la evaluación empírica es absolutamente necesaria para una estimación de la efectividad, eficiencia y usabilidad de un sistema *e-Learning* [Weibelzahl 2002a], pues ciertos tipos de problemas no serían descubiertos sin la utilización del enfoque empírico. Sin embargo, según señala Weibelzahl la evaluación empírica tiene ciertos límites o desventajas [Weibelzahl 2002b]. Por ejemplo, la evaluación empírica obtiene buenos resultados al identificar errores de diseño, pero sin embargo, no sugiere nuevas teorías. Este mismo autor propone una estructura para evaluar cualquier sistema adaptativo, también llamada “evaluación por capas” o “layered evaluation” [Weibelzahl 2002a, pp. 50-76]. La evaluación por capas se divide en cuatro fases, y es necesario haber finalizado la fase anterior antes de comenzar la siguiente:

- **Evaluación de los datos de entrada en el sistema.** Los datos de entrada son los datos de los usuarios adquiridos por el sistema. Se dividen en datos estáticos (perfiles de los usuarios: preferencias, datos de identificación, datos biológicos, etc.) y datos dinámicos (interacciones de los usuarios con el sistema). Debido a que estos datos son la base de futuras inferencias es importante comprobar su fiabilidad, y en especial comprobar que los datos dinámicos son fiables. Dicho de otro modo, el ruido existente en estos datos debe ser el mínimo posible, de otra manera las inferencias obtenidas no serían robustas. Otro aspecto que es importante evaluar es la validez de los datos de entrada, es decir, comprobar si los datos reflejan los comportamientos reales de los usuarios. Por ejemplo, muchos sistemas adaptativos almacenan el número de páginas web visitadas por los usuarios, éste indicador puede ser erróneo, ya que pueden existir usuarios que sólo visitaron parcialmente estas páginas. Por eso, para comprobar la fiabilidad y validez de los datos de entrada, el autor recomienda realizar cuestionarios o tests sobre los contenidos ofrecidos a los usuarios. De esta forma, se puede comprobar que los usuarios conocen los contenidos, y su interés se mantiene estable. Por ejemplo, se puede realizar un cuestionario sobre el contenido de la página que ha leído el usuario.
- **Evaluación del mecanismo de inferencia.** Las decisiones de adaptación tomadas por el sistema se sustentan por el mecanismo de inferencia. Por esta razón, es necesario comprobar la validez de la inferencia, es decir, analizar si las suposiciones o inferencias que realiza el sistema sobre determinadas propiedades de los usuarios son correctas o no. El autor propone utilizar valoraciones de expertos o, tests externos que puedan revelar si el sistema realiza falsas suposiciones, para analizar la validez de las inferencias que realiza el sistema. Por ejemplo, un sistema puede realizar las decisiones de adaptación en base al conocimiento del usuario inferido por el sistema, se puede valorar si esta inferencia es correcta mediante un test de conocimientos al usuario.
- **Evaluación de las decisiones de adaptación.** El propósito de esta fase es analizar si las decisiones de adaptación tomadas por el sistema fueron óptimas, pues en la fase anterior se comprobó que la inferencia se hizo correctamente.
- **Evaluación de la interacción total.** En esta última fase se evalúan el comportamiento del sistema y de los usuarios.
 - Comportamiento del sistema: varias dimensiones relativas al comportamiento del sistema pueden ser evaluadas, como el tiempo de computación, tiempo de reacción y estabilidad. Sin embargo, la dimensión más importante es la frecuencia con que el sistema realiza las adaptaciones. En concreto, es importante analizar la frecuencia de determinados tipos de adaptación.
 - Comportamiento de los usuarios: la adaptación producida por un sistema tuvo éxito sólo si los usuarios alcanzaron sus objetivos y estuvieron

satisfechos con la interacción con el sistema. Por este motivo, esta última fase se puede dividir en la evaluación del rendimiento de los usuarios y la evaluación de la usabilidad. La primera se mide en términos de eficiencia y efectividad. Así, el éxito de un sistema adaptativo orientado a la enseñanza depende de la ganancia del aprendizaje de sus usuarios. La segunda puede ser medida mediante una serie de parámetros específicos. Los más usados son la duración de la interacción y cuestionarios no estandarizados para evaluar el grado de satisfacción del usuario.

2.4.1.2. Propuesta de Paramythis

Paramythis et al. dividen el proceso de adaptación en ocho etapas: monitorización de la interacción, interpretación/inferencias, conocimiento provisto explícitamente, creación de modelos a partir de los datos de las dos fases anteriores, toma de decisiones de adaptación, aplicación de las adaptaciones, transparencia en modelos y fundamentos de la adaptación, y evaluación automática de la adaptación ². El trabajo [Paramythis 2001] presenta una propuesta para evaluar la interfaz de un sistema adaptativo mediante la evaluación modular. Dicha evaluación divide el proceso de evaluación en siete módulos:

- Módulo A1. Este módulo está relacionado con las etapas monitorización de la interacción, interpretación/inferencias y creación de modelos. El objetivo de este módulo es asegurar que los modelos deducidos por el sistema a través de la evaluación de las interacciones dinámicas son óptimos. Se debe entender esta condición de optimalidad respecto a los siguientes criterios de evaluación: corrección (*correctness*) de las interpretaciones/inferencias, dimensión (*comprehensiveness*) del modelo, redundancia (*redundancy*) del modelo, precisión (*precision*) del modelo, sensibilidad (*sensitivity*) del modelo, etc.
- Módulo A2. Las etapas relacionadas con este módulo son: conocimiento provisto explícitamente y creación de modelos. El propósito de este módulo es muy parecido al del anterior, pero con distintos criterios de evaluación. Así, los criterios destinados a evaluar la interacción y las interpretaciones/inferencias no tienen sentido en este módulo. Adicionalmente se utilizan criterios como la transparencia (*transparency*) del proceso, o el tiempo (*overhead*) que puede ser impuesto en las principales tareas de interacción por el suministro explícito de conocimiento.
- Módulo B. La toma de decisiones es la única etapa que está relacionada con este módulo. Su objetivo es asegurar que las decisiones de adaptación realizadas fueron correctas. Para conocer en qué medida fueron correctas se utilizan

²Las ocho etapas presentadas en esta sección son una traducción personal de las etapas expuestas en [Paramythis 2001, pp. 13-14]. El texto original contiene las siguientes etapas: *interaction monitoring, interpretation/inferences, explicitly provided knowledge, modeling, adaptation decision making, applying adaptations, transparent models & adaptation "rationale", automatic adaptation assessment*.

los siguientes criterios: evaluar la necesidad (*necessity*), el grado de adecuación (*appropriate*), el grado de aceptación (*acceptance*) de la adaptación, etc.

- Módulo C. Este módulo comprende la etapa de aplicación de adaptaciones. Este módulo complementa al Módulo B. Los criterios que se tienen en cuenta en la evaluación son: tiempo necesario (*timeliness*) para realizar la adaptación, impacto (*obtrusiveness*) que produce la adaptación en el usuario, nivel de control del usuario (*user control*) para anular, retirar o ignorar la adaptación ofrecida por el sistema, etc.
- Módulo D1. Corresponde a evaluar las etapas de creación de modelos y transparencia de modelos. El objetivo es asegurar que la percepción de los usuarios, recogida en los modelos, concuerda con el estado actual de los modelos. Los criterios de evaluación que usan en este módulo son respecto a la presentación de los contenidos: completitud (*completeness*), coherencia (*coherence*), racionalidad (*rationality*), etc.
- Módulo D2. Comprende las etapas de toma de decisiones de adaptación y fundamentos de la adaptación. Este módulo es similar al anterior, aunque se utilizan criterios respecto a cómo se produce la adaptación. De esta forma, los criterios que pueden utilizarse son: coherencia (*coherence*) de la adaptación, causalidad (*causality*) del fundamento, etc.
- Módulo E. Comprende la última etapa, evaluación automática de la adaptación. El principal objetivo de este módulo es asegurar que el sistema comparte el mismo punto de vista que los usuarios con respecto al éxito o fracaso de las adaptaciones. Es decir, es necesario comparar las opiniones de los usuarios respecto a adaptaciones específicas y los efectos que producen con la información que recoge el sistema sobre estos tipos de adaptaciones.

Adicionalmente, Paramythis y Weibelzahl proponen un modelo mejorado sobre la evaluación por capas, que une los dos modelos anteriores. Estos autores proponen dividir el proceso de adaptación en diferentes etapas. Para ello examinaron las propiedades comunes de los modelos y arquitecturas existentes de sistemas adaptativos interactivos (*Interactive Adaptive Systems*, en adelante: *IAS*), como *AHAM* [de Bra 1999], *The Munich Reference Model* [Koch 2002], *Flexcel (Flexible Excel)* [Oppermann 1994] y el enfoque presentado por [Totterdell 1990b]. Como resultado, estos autores descomponen el proceso de adaptación en cinco etapas principales: *recogida de datos de entrada*, *interpretación de los datos recogidos*, *modelado del estado actual del “mundo”*, *decisiones sobre la adaptación*, y *aplicación de las decisiones de adaptación* (más información sobre este modelo conjunto puede encontrarse en [Paramythis 2005]).

2.4.2. Evaluación de Sistemas *Hypermedia* Adaptativos Orientados a la Enseñanza

Los modelos anteriores sobre la evaluación de Sistemas *Hypermedia* Adaptativos podrían ser aplicados, en particular, al subconjunto de los *AEHS*, ya que éstos últimos están incluidos en los Sistemas *Hypermedia* Adaptativos. En concreto, la propuesta de Weibelzahl, presentada en el apartado anterior, divide el proceso de evaluación en cuatro fases. En la primera fase, se propone realizar cuestionarios o tests sobre los contenidos educativos ofrecidos a los estudiantes, con el fin de comprobar la fiabilidad y validez de los datos de entrada del sistema. El autor señala que el ruido existente en los datos de entrada debe ser el mínimo posible, sin embargo sólo nos ofrece una metodología para detectar el ruido, consistente en uso de cuestionarios o tests de conocimientos. El problema de validar los datos de entrada es un problema muy complejo, que escapa a los intereses de esta tesis, y que pensamos que es demasiado ambicioso resolverlo mediante cuestionarios. De hecho, la información que nos pueden aportar no es lo suficientemente significativa para poder afirmar que los datos de un sistema son válidos, pues estos resultados no dejan de ser subjetivos, pues dependen de la buena voluntad de los encuestados. Del mismo modo, Weibelzahl propone utilizar tests externos o validaciones por expertos para evaluar el mecanismo de inferencia. Sin embargo, el autor no indica la forma de analizar estos tests, y teniendo en cuenta que estos sistemas contienen un alto número de usuarios, el análisis de estos tests es complejo y costoso. Igualmente, estas pruebas destinadas a evaluar el mecanismo de inferencia no recogen determinados aspectos del comportamiento de los estudiantes, que, sin embargo, están recogidos en las interacciones de los estudiantes o logs. Las dos fases siguientes, la evaluación de las decisiones de adaptación y la evaluación de la interacción total, pretenden comprobar si la adaptación fue óptima, y evaluar tanto el comportamiento del sistema como de los estudiantes. La propuesta de Paramythis presenta una evaluación modular dividida en siete módulos. Los módulos A1 y A2 son similares a la segunda fase del modelo anterior, pues tratan de establecer que los modelos deducidos por el sistema a partir de los datos dinámicos son óptimos. En este sentido, Paramythis expone una serie de métricas que pueden ser usadas para medir esta optimalidad. El siguiente módulo se encarga de asegurar que las decisiones de adaptación fueron las correctas, que coincide con el objetivo de la fase 3 de la propuesta de Weibelzahl. Sin embargo, en este punto Paramythis nos sugiere el uso de métricas de tipo cualitativo, como medir el grado de adecuación, grado de aceptación y necesidad de adaptación. Del mismo modo, que sucedía en el modelo anterior los resultados son subjetivos y dependen de la buena voluntad de los usuarios. El módulo C complementa al anterior, con más métricas cualitativas como medir el impacto que produce la adaptación, el nivel de control del usuario, aunque aporta una métrica cuantitativa, tiempo necesario para realizar la adaptación. El módulo D1 es el encargado de asegurar que los modelos inferidos por el sistema son coherentes con la percepción de los usuarios. Nuevamente, se vuelven a utilizar métricas cualitativas, como medida de la completitud, la coherencia, y racionalidad de los contenidos, lo que produce subjetividad en las

conclusiones obtenidas. El módulo D2 comprende las etapas de toma de decisiones y fundamentos de la adaptación. De esta forma, los objetivos de ambos módulos son similares a la tercera etapa en la propuesta de Weibelzahl. El último módulo de la propuesta de Paramythis persigue asegurar que el sistema comparte el mismo punto de vista que los usuarios respecto al éxito o fracaso de las adaptaciones.

También podrían ser aplicados los diferentes modelos de evaluación del *e-Learning* [Rubio 2003]. La característica principal que subyace en estos modelos es que realizan un análisis muy superficial del sistema, basado fundamentalmente en analizar si éste cumple determinadas normas de calidad, si el sistema cumple una serie de objetivos educativos, o si los contenidos cubrían las necesidades de los estudiantes, o bien comparar el sistema con otros. Es importante destacar que estos modelos son interesantes desde el punto de vista de la usabilidad y la calidad, pero no lo son para evaluar la funcionalidad o rendimiento de un sistema *AEH*.

El **análisis de logs** o, interacciones de los usuarios, es el proceso de extraer información útil, “escondida” en los logs, mediante el uso de técnicas automáticas o semi-automáticas. Distintas técnicas pueden ser utilizadas, desde las técnicas clásicas de la estadística inferencial hasta las avanzadas técnicas de la minería de datos. De hecho, los logs contienen el comportamiento de los estudiantes (usuarios), y además es posible inferir a partir de la información contenida en los logs el comportamiento del sistema, y por tanto, se puede analizar si la adaptación del sistema fue la correcta. Por todas estas razones, pensamos que el análisis de logs es útil para evaluar la adaptación, el rendimiento de los estudiantes, y el rendimiento de un curso o sistema *AEH*.

El análisis de logs nos proporciona resultados cuantitativos, si bien no exentos de ruido, de las interacciones de los estudiantes. Los modelos anteriores apenas proporcionan información a cerca de cómo interactuaron los estudiantes con el sistema, basada en lo que realmente realizaron los estudiantes: si tuvieron o no problemas en el desarrollo del curso, o si tuvieron éxito o fracaso en las actividades. Por eso, creemos que el análisis de logs puede aportar este tipo de información, útil para evaluar la funcionalidad de un sistema *hypermedia* adaptativo orientado a la enseñanza. Hay que destacar, que el método utilizado en la literatura hasta el alcance de nuestros conocimientos es el análisis de logs, y en relación con lo anterior, el análisis de logs es útil para detectar posibles irregularidades o mejoras en un sistema. En este sentido, los trabajos que se exponen a continuación emplean distintas técnicas para el análisis de logs, pero su objetivo no es el propio análisis, sino la construcción de herramientas que ayuden en el proceso de aprendizaje de los estudiantes.

2.4.2.1. Evaluación basada en análisis de logs

Como se indicó anteriormente, el *análisis de logs* es especialmente útil tanto para evaluar el rendimiento del sistema como de los usuarios, así como para añadir mejoras en un sistema. Por ejemplo, **Ueno** propone un método para la detección de caídas en el rendimiento de determinados estudiantes, o en otras palabras, detección de *outliers* [Barnett 1994] en los procesos de aprendizaje (procesos irregulares de

aprendizaje) de los estudiantes mediante el análisis de logs [Ueno 2006]. Estos logs contienen información sobre la identificación del contenido (*Contents ID*), identificación del estudiante (*Learner ID*), el número de conceptos completados (*number of topics*), el tiempo de inicio (*Time ID*) y el tiempo empleado. Este método se presenta como una alternativa al clásico diagrama número de conceptos aprendidos frente al tiempo empleado o tiempo de respuesta, con el que se pueden descubrir procesos irregulares de aprendizaje observando aquellos en los que el tiempo fue muy superior a la media. La idea que presenta Ueno es predecir un nuevo proceso de aprendizaje mediante una distribución bayesiana basada en el tiempo de respuesta. De esta forma, el autor relaciona dicha distribución con la distribución t de *Student*, y como la representación de esta distribución es la *curva t*, por esta razón este método recibe el nombre de *curva de detección outlier*. Así, se utiliza esta curva para detectar procesos de aprendizaje irregulares mediante el siguiente método: i) se calcula el nuevo proceso de aprendizaje, ii) se calcula el valor de la t correspondiente a ese nuevo proceso, iii) si el valor calculado de la t es superior o inferior a la curva de la t con nivel de significación α el proceso de aprendizaje es detectado como un *outlier*. Ueno evaluó este método con sus estudiantes presentándoles contenidos, y después de cada contenido una pregunta sobre si han entendido el contenido presentado. Las respuestas posibles a esta pregunta son tres: *sí*, *no*, *no contesta*. De esta forma, es posible saber si el método detecta de forma eficaz si un contenido es aprendido. Como resultado de esta evaluación, el autor concluye que este método detecta eficazmente los procesos irregulares de aprendizaje, aunque el método es relativamente estricto, ya que la probabilidad de falso positivo en el caso de respuesta sí es de 0,24. El método de Ueno es especialmente útil en el caso de cursos que ofrezcan mayor contenido teórico que práctico, pues este método no se tiene en cuenta los resultados de los estudiantes en las actividades prácticas, puesto que la variable principal de estudio es el tiempo empleado en completar una actividad. En este sentido, la presente tesis también pretende detectar procesos irregulares de aprendizaje mediante el análisis de logs, pero considerando como variable de estudio las puntuaciones recibidas por los estudiantes en las actividades.

En general, realizar el *análisis de logs* mediante el uso de las técnicas tradicionales de análisis es complejo y costoso, pues estas técnicas no son adecuadas para analizar grandes volúmenes de datos, por lo que es necesario emplear otro tipo de técnicas. Debido a que la *DM* obtiene buenos resultados al analizar grandes cantidades de datos [Fayyad 1997], en los últimos años esta disciplina se ha convertido en una de las técnicas más utilizadas no sólo en el área de investigación de mercados [Mobasher 2000], sino también en el área de los sistemas *e-Learning* [Romero 2005]. Prueba de este crecimiento son los numerosos *workshops* realizados (*Fifth Workshop on User-Centred Design and Evaluation of Adaptive Systems* [Weibelzahl 2006], *International Workshop on Applying Data Mining in E-Learning* [Romero 2008]) en congresos internacionales como *Adaptive Hypermedia* (en adelante: *AH*) y *European Conference on Technology Enhanced Learning* (en adelante: *EC-TEL*), que dieron lugar a la serie de conferencias internacionales de la Minería de Datos aplicada en los sistemas *e-Learning* (*Educational Data Mining*, en ade-

lante: *EDM*) [Baker 2008, Romero 2009, edm 2010]. Otras evidencias del creciente interés de la utilización de esta disciplina en el área de los sistemas orientados a la enseñanza a través de Internet son, el completo estudio desarrollado por **Romero** y **Ventura** [Romero 2007], así como la creación de un repositorio de datos (*open data repository*) desarrollado por el *Pittsburgh Science of Learning Center* (en adelante: *PSLC*), *PSLC DataShop* [Koedinger 2008]. En el estudio realizado por estos dos autores se muestran los principales trabajos, desarrollados entre los años 1995 a 2005, que aplican técnicas de la Minería de Datos en el área del *e-Learning*. Debido a la importancia que suscita este tema se explicará de forma más detallada en la siguiente sección.

2.4.2.2. Aplicación de la Minería de Datos al análisis de logs en sistemas educativos

Según el estudio de Romero et al. [Romero 2007] las técnicas más usadas son los algoritmos de clasificación [Arabie 1996] y las reglas de asociación [Agrawal 1993]. Además, esta tendencia se sigue manteniendo según el reciente estudio realizado por **Baker** y **Yacef** [Baker 2009] en el que se comparan el número de artículos de cada categoría publicados en las conferencias *Educational Data Mining* de 2008 y 2009; sin embargo, las técnicas de predicción, donde estos autores encuadran a las técnicas de clasificación y de regresión, están siendo más aplicadas en la actualidad que las técnicas de asociación y *clustering*³. Respecto al repositorio *PSLC DataShop*, cabe destacar que almacena actualmente más de 110 conjuntos de datos, los cuales incluyen más de 18 millones de interacciones de los usuarios, siendo este repositorio una gran iniciativa para poder realizar una amplia batería de pruebas.

La mayor parte de los sistemas *on-line* educativos son sistemas *Web*, por lo que sería más preciso hablar de Minería *Web* (*Web Mining*). En concreto, nos centraremos en la Minería de Uso de *Web* (*Web Usage Mining*), que es el campo de la Minería *Web* encargado del análisis de los datos de interacción. **Srivastava** et al. [Srivastava 2000] definen a la Minería de Uso de *Web* como el proceso de aplicar técnicas de la Minería de Datos al descubrimiento de patrones de uso en los datos *Web* (*Web data*). Este proceso se divide en tres etapas principales: preprocesamiento (*pre-processing*), descubrimiento de patrones (*pattern discovery*) y análisis de patrones (*pattern analysis*).

- **Preprocesamiento:** es la etapa más difícil del proceso, debido a que los datos disponibles pueden ser incompletos. La tarea de identificar usuarios y sesiones es compleja, pues en muchas ocasiones es difícil saber cuando un usuario abandona un sitio *Web* (se suele utilizar como medida orientativa 30 minutos para el tiempo de expiración de sesión).

³Técnica que consiste en agrupar objetos basándose únicamente en la información encontrada en los datos que describe los objetos y sus relaciones. El objetivo es conseguir que los objetos dentro de un mismo grupo sean similares y diferentes de los objetos de otros grupos. Cuanto mayor sea la similitud dentro del grupo y mayor sea la diferencia entre grupos mejor es el *clustering* [Tan 2006, p. 490].

- **Descubrimiento de patrones:** en esta etapa se utilizan métodos y algoritmos desarrollados en diferentes disciplinas como la estadística, la minería de datos, *machine learning*, y descubrimiento de patrones.
 - Análisis Estadístico: es el método más común para extraer conocimiento de los usuarios de un sistema. Por ejemplo, estos autores indican que se pueden utilizar diferentes medidas (media, mediana, frecuencia, etc.) sobre distintas variables relacionadas con el número de páginas visitadas, tiempo empleado en cada una, y longitud de secuencia de páginas recorridas. Este tipo de análisis es potencialmente útil para mejorar el rendimiento del sistema.
 - Reglas de Asociación: se pueden usar para determinar la relación existente entre las páginas más frecuentemente utilizadas, es decir, páginas que son accedidas en conjunto con una probabilidad dada.
 - *Clustering*: es una técnica que agrupa un conjunto de objetos con similares características. Se puede utilizar para descubrir grupos de usuarios que presentan comportamientos muy similares.
 - Clasificación: es la tarea de asignar objetos a varias clases predefinidas. Puede ser realizada mediante algoritmos supervisados como: árboles de decisión, Naive Bayes, K vecinos cercanos, etc.
 - Patrones Secuenciales: se usan para encontrar patrones dentro de las sesiones de los usuarios. Esta técnica se puede usar para predecir futuros patrones de visita que serán útiles para ofrecer sugerencias a determinados grupos de usuarios.
 - *Dependency Modelling*: consiste en desarrollar un modelo capaz de representar dependencias significativas entre varias variables del dominio *Web*.
- **Análisis de patrones:** el principal objetivo de esta etapa es filtrar las reglas o patrones no interesantes del conjunto encontrado en la etapa anterior, i.e., seleccionar aquellos que son relevantes para el análisis.

Hay que señalar que los trabajos existentes en la Literatura realizan en menor o mayor medida cada una de estas etapas en el análisis de logs. Asimismo, la propuesta de esta tesis realiza las etapas presentadas por Srivastava et al. Las aplicaciones de la Minería de Uso de *Web* son muy diversas en los sistemas *hypermedia* orientados a la enseñanza. Según apuntan Romero et al. las principales aplicaciones de esta disciplina residen en: creación de sistemas de personalización [Mobasher 2000], sistemas de **recomendación** [Li 2004, Zaiane 2006, Talavera 2004, Ueno 2006, Romero 2006b, Khribi 2008], sistemas de **detección** de *outliers* y de situaciones problemáticas [Baker 2004a, Romero 2004], y sistemas de **modificación** [Perkowitz 1998, Su 2006]. La tabla 2.1 muestra los principales trabajos realizados en la aplicación de la Minería de Uso de *Web* y las técnicas utilizadas.

Así, por ejemplo [Merceron 2003] se sirve de análisis estadísticos y reglas de asociación para construir un sistema de detección. Los trabajos presentados en esta tabla serán explicados con más detalle a continuación.

Técnicas	Sistemas		
	Recomendación	Detección	Modificación
Análisis Estadístico		[Merceron 2003], [Ueno 2006]	
Reglas de Asociación		[Merceron 2003], [Ben-Naim 2009]	
<i>Clustering</i>	[Talavera 2004], [Zaiane 2006], [Khribi 2008]		[Su 2006]
Clasificación	[Ueno 2006] (<i>DT</i>)	[Baker 2004a] (<i>LRM</i>)	[Su 2006] (<i>DT</i>)
Patrones Secuenciales	[Romero 2006b] (<i>AprioriAll</i> , <i>GSP</i> , <i>PrefixSpan</i>)		[Su 2006] (<i>GSP</i>)
Algoritmos Evolutivos		[Romero 2004] (<i>GA</i>)	

Tabla 2.1: Técnicas utilizadas en las distintas aplicaciones de la Minería de Uso de *Web*. Leyenda: *DT*: *Decision Trees*, *LRM*: *Latent Response Model*, *GA*: *Genetic Algorithms*, *GSP*: *Generalized Sequential Pattern*

Autores como **Talavera** y **Gaudioso** se sirven de *clustering* para analizar el comportamiento de los estudiantes al realizar actividades colaborativas en entornos de comunidades virtuales (*virtual learning community*) [Talavera 2004]. Estas actividades colaborativas son elementos de un curso sobre el uso de Internet en la educación desarrollado bajo el sistema *ACS*⁴ (*ArsDigita Community System*). Este curso contiene elementos como foros, noticias, *chat*, espacio de almacenamiento para ficheros, y páginas web personales. Al principio del curso se presenta a los estudiantes un cuestionario sobre los conocimientos previos en el software educativo y el uso de Internet. Adicionalmente, se les presenta otro cuestionario donde se pueden indicar sus intereses. Por el lado de los profesores, éstos controlan las respuestas de los estudiantes y las dificultades que tienen, proponiendo actividades colaborativas y visitando los *chats* para comprobar si los estudiantes tuvieron dificultades en el curso. El objetivo del trabajo de estos autores es detectar patrones de comportamiento y relacionarlos con el rendimiento del estudiante. Talavera y Gaudioso señalan que

⁴Sistema de código abierto que contiene un conjunto de aplicaciones para el soporte de comunidades web. El sistema ofrece una colección de módulos para administración de usuarios/grupos, administración de contenidos, noticias, *FAQs* (preguntas frecuentes), foros, chat, etc. En la actualidad este sistema ha derivado a *OpenACS* (*Open Architecture Community System*) [OpenACS.org 2009].

la fase de preprocesamiento (comprende la selección y limpieza de los datos) fue compleja, ya que los datos estaban almacenados en una base de datos relacional (con más de 50 tablas), y fueron necesarias varias consultas *SQL* para obtener el conjunto de logs final. De esta forma, los logs almacenan cuatro tipos de datos: datos demográficos, datos de los cuestionarios de conocimiento previo, datos de los cuestionarios de intereses, y datos de interacción. A su vez, los datos de interacción contienen parámetros relacionados con el número de sesiones empezadas, número de ficheros en el espacio de almacenamiento, presentaciones publicadas, marcadores (*bookmarks*) añadidos, *e-mails* enviados, registro en otros cursos, número de ficheros en otras áreas de almacenamiento, número de páginas visitadas del curso, *chat* (número de entradas en el *chat*, número de mensajes enviados al *chat*, número de entradas en otros *chats*, número de mensajes enviados a otros *chats*), con el foro (número de mensajes enviados al foro, alertas activadas en el foro, número de hilos creados en el foro, foros creados, número de mensajes enviados a otros foros, alertas activadas en otros foros, número de hilos respondidos a otros estudiantes en el foro, número de hilos finalizados en el foro). Además, las características numéricas han sido discretizadas manualmente por los profesores, observando los histogramas de cada una, en los valores: bajo, medio y alto. Por este motivo, los autores utilizan un *clustering* para datos continuos y discretos basado en el modelo de Naive Bayes. Los primeros resultados del análisis *cluster* no fueron significativos para el estudio, ya que las características principales para formar los *clusters* fueron las respuestas a los cuestionarios, no apareciendo ninguno de los elementos de los datos de interacción. Los siguientes experimentos mostraron correlación entre los grupos y el rendimiento, obteniendo seis grupos o *clusters*. Del análisis realizado por los autores, se pueden destacar los siguientes *clusters*:

- *Cluster 1*: estudiantes altamente colaborativos, crean nuevos foros, publican presentaciones, participan en foros externos. La mayoría de los estudiantes pertenecientes a este grupo aprobaron el curso.
- *Cluster 4*: estudiantes con baja participación en el *chat*, foro, no interactúan demasiado. La mayoría de éstos suspendieron el curso.
- *Cluster 6*: estudiantes con grado medio de interacción en *chat* y foro, activaron alertas en el foro, usaron el espacio de almacenamiento. La mayoría de los pertenecientes a este grupo aprobaron el curso.

Como conclusión, Talavera y Gaudioso señalan que estos resultados no son suficientes para concluir que existe correlación entre colaboración y rendimiento de los estudiantes. Además, hacen especial énfasis en que las técnicas de Minería de Datos, como el *clustering*, pueden obtener buenos resultados aplicadas al campo de los sistemas *e-Learning*, pero la calidad de éstos depende en gran medida del diseño previo de los logs, y que en éstos se encuentre toda la información útil posible.

Zaïane [Zaïane 2006] propone una arquitectura para desarrollar un sistema recomendador no intrusivo en el campo del *e-Learning* basado en el análisis de logs.

Muchos sistemas recomendadores desarrollados en el ámbito del comercio electrónico son intrusivos, en el sentido de que utilizan las valoraciones que realizan los usuarios sobre los elementos que visitan para realizar sus recomendaciones. Por ejemplo, muchos sistemas de compra a través de Internet piden que los usuarios valoren la utilidad del objeto comprado, por ejemplo, un libro. Este autor considera que las valoraciones realizadas por los usuarios no deben ser utilizadas en el campo del *e-Learning*, pues las recomendaciones basadas en las actividades realizadas (se encuentran en los logs) reflejan las elecciones reales de los usuarios. Una recomendación de un sistema *e-Learning* consiste en ofrecer una lista de recursos (páginas web, *applets*, imágenes, vídeos, etc) que permite a los estudiantes saltar directamente a los contenidos deseados, evitando buscar en el laberinto de enlaces que muchas veces existe en un sistema *e-Learning*. Siguiendo esta idea, un sistema recomendador sugiere recursos y acciones. Por ejemplo, un sistema recomendador puede sugerir enlaces a partir de los enlaces visitados por otros estudiantes, o puede sugerir que el estudiante realice una determinada acción, e.g. visitar un contenido teórico, antes de realizar otra, e.g. un test, pues otros estudiantes con perfil similar tuvieron éxito en esta acción realizando primero la otra. Zaïane nos indica que existen dos tendencias en el diseño de sistemas recomendadores: los que están basados en reglas de asociación y los basados en *clustering* [Zaïane 2001b]. Los primeros utilizan los consecuentes de las reglas que se satisfagan para obtener las recomendaciones [Zaïane 2001a], mientras que los segundos realizan sus recomendaciones en base al cluster seleccionado. En estos últimos los *clusters* están formados normalmente por enlaces (páginas que contienen los contenidos), y el sistema busca la página actual, en la que se encuentra el estudiante, en los *clusters*, y realiza las recomendaciones en base a los enlaces más cercanos dentro de ese *cluster*. La arquitectura que propone este autor divide el sistema en dos módulos: módulo *off-line* y módulo *on-line* [Li 2004]. El módulo *off-line* es el encargado del preprocesamiento y análisis de los logs, mientras que la tarea del módulo *on-line* es establecer las recomendaciones (motor de recomendaciones). Los logs son analizados extrayendo los patrones de navegación mediante dos análisis *clusters*, en los que primero se extraen las sesiones, de éstas se extraen las páginas con sus contenidos, después se identifican sub-sesiones que formarán mediante el segundo análisis los *clusters* de los patrones de navegación. Estos últimos *clusters* son los empleados por el motor de recomendaciones para establecer las recomendaciones. El funcionamiento es el siguiente: cuando un estudiante empieza un curso, el sistema identifica su patrón de navegación después de haber visitado varias actividades, busca entre los *clusters* de patrones de navegación este patrón, y en caso de alguno coincida se recomiendan las páginas más relevantes de ese cluster. Además, Zaïane compara su sistema con otro basado en reglas de asociación, y concluye que al menos su sistema es un 30 % más consistente que el otro.

Otras técnicas como las *reglas de asociación* y los *árboles de decisión* sirven para añadir mejoras en los sistemas. Por ejemplo, Khribi et al. proponen un sistema recomendador compuesto por dos módulos de análisis: módulo *off-line* y módulo *on-line* [Khribi 2008]. El primero preprocesa y analiza los logs para construir los

perfiles de las sesiones de usuario y contenidos, mientras que el cometido del segundo es construir la lista de recomendaciones. Los perfiles de las sesiones de los usuarios (conjunto de *URLs* visitadas) se construyen a partir de los logs utilizando *clustering*, y los perfiles de contenidos se construyen a través de un motor de búsqueda indexando los contenidos educativos de acuerdo a las normas y estándares usados en *e-Learning*. Las recomendaciones se generan en base a los enlaces recomendados por el motor de búsqueda en los perfiles de contenidos, y las extraídas por la aplicación de las reglas de asociación de las *URLs* visitadas. Igualmente, **Ueno** propone mejorar el proceso de aprendizaje en el sistema *Samurai*⁵ mediante un agente animado que muestra mensajes de motivación a los estudiantes que tuvieron dificultades [Ueno 2005]. Además, dicho agente sirve de ayuda para que los profesores en el caso de que la *curva de detección outlier* (método desarrollado por este autor en [Ueno 2006] y explicado al inicio de esta sección) detecte procesos irregulares de aprendizaje no se saturen al enviar *mails* con mensajes de motivación para los estudiantes detectados. Los mensajes de este tipo que produce este agente recomendador son: “¿Evitaste los contenidos 21, 22, 23 y 29? Inténtalo de nuevo.”, “Tardaste demasiado tiempo en los contenidos 41, 42 y 51. Si sigues teniendo problemas, quizás debas preguntar al profesor.”. Además, este agente animado produce mensajes de motivación diferentes a los anteriores basados en la predicción de la situación del estudiante. La situación en la que puede estar un estudiantes puede ser: *Failed* (calificación en el examen final por debajo de 60), *Abandon* (el estudiante no se presenta al examen final), *Successful* (calificación en el examen final entre 60 y 80), y *Excellent* (calificación en el examen final por encima de 80). El método escogido consiste en aplicar la técnica de árboles de decisión a los logs, teniendo en cuenta las siguientes variables:

1. Número de temas que el estudiante aprendió.
2. Número de veces que el estudiante accedió al sistema.
3. Media del número de veces que el estudiante completó cada tema.
4. Media del tiempo en cada lección, que consiste en varios tipos de contenidos.
5. Media de valoraciones sobre el entendimiento del tema.
6. Media del tiempo empleado en cada curso (cada curso está compuesto por 15 lecciones).
7. Media del número de veces que el estudiante cambió su respuesta en las cuestiones preguntadas.
8. Número de veces que el estudiante escribió opiniones en la pizarra de trabajo.

⁵El sistema *e-Learning* *Samurai* proporciona más de 78 cursos *e-Learning*. Está formado por un módulo de presentación de contenidos, una base de datos de contenidos, una base de datos de los logs, y un sistema de minería de datos (*Data Mining System*, en adelante: *DMS*). Más información puede encontrarse en [Ueno 2002, Ueno 2003]

9. Media del tiempo empleado por cada tema.

Estas variables son utilizadas para generar los árboles de decisión. Debido a que los logs se van incrementando gradualmente cada semana, y la duración de cada curso es de 15 semanas, se generaron 15 árboles de decisión. Para la elección del mensaje para el agente animado Ueno propone seguir los siguientes pasos:

- El sistema predice la situación del estudiante y su probabilidad mediante el árbol de decisión correspondiente.
- Si la situación es *Excellent*, entonces el mensaje es del tipo “¡Muy bien!”, “Continúa haciéndolo así”, o “La probabilidad de éxito es xx %”. Si por el contrario es otro valor, el sistema busca en el árbol de decisión la hoja *Excellent* más próxima, y determina las operaciones para conseguir cambiar este valor a *Excellent*. Estas operaciones se traducen en mensajes de motivación correspondientes a cada una de las 9 variables anteriores. Por ejemplo, para la variable 4 el mensaje de motivación correspondiente es “Parece que estás empleando poco tiempo en las lecciones. Por favor, emplea más tiempo en cada lección.”

Ueno concluye que este agente animado recomendador es suficientemente efectivo en su sistema, ya que su utilización redujo en un tercio el abandono en el sistema *e-Learning* de sus estudiantes [Ueno 2005].

Romero et al. proponen aplicar la Minería de Datos para descubrir información útil que ayude a los profesores en la tarea de mejorar un curso. Para conseguir este objetivo proponen una metodología de construcción de cursos adaptativos formada por las siguientes etapas [Romero 2002]:

1. **Construcción del curso:** el profesor elabora el curso usando la información del modelo de dominio y el modelo pedagógico. Generalmente, para esta tarea utiliza una herramienta de autor. Una vez el curso ha sido elaborado se puede publicar en un servidor web.
2. **Ejecución del curso:** el curso es ofrecido a los estudiantes y éstos realizan las actividades ofrecidas. Al mismo tiempo el sistema almacena, de forma transparente, las interacciones producidas por éstos en forma de logs.
3. **Aplicación de la Minería de Datos:** el profesor aplica algoritmos de la Minería de Datos sobre los logs con el objetivo de obtener importantes relaciones existentes en los datos, información útil. Es útil en esta fase utilizar herramientas de minería de datos, o herramientas personalizadas para este análisis.
4. **Mejorar el curso:** la información obtenida en el paso 3 sirve de ayuda para que el profesor lleve a cabo las modificaciones apropiadas para mejorar el rendimiento del curso.

A continuación, cabe destacar que el enfoque seguido en esta tesis es muy similar y sigue la metodología de construcción de cursos adaptativos presentada

anteriormente. Romero et al. plantean utilizar como técnica de minería de datos (en el paso 3) los algoritmos evolutivos, y en particular los algoritmos genéticos [Romero 2003, Romero 2004]. El algoritmo genético lo aplican en cuatro etapas:

- *Inicialización*: generación de un grupo inicial de reglas a partir de los criterios de selección de datos indicados por el profesor o diseñador del curso.
- *Evaluación*: cálculo de la función de idoneidad (*fitness*) de las reglas generadas en la fase de inicialización. El objetivo es seleccionar las mejores reglas de acuerdo a la función de idoneidad.
- *Selección*: selección de dos reglas del conjunto de las seleccionadas en el paso anterior y el resto de reglas no seleccionadas. El objetivo es convertir las dos reglas en padres, que serán utilizadas para el proceso de reproducción.
- *Reproducción*: creación de nuevas reglas mediante la mutación o cruce de las reglas anteriores.
- *Finalización*: en esta etapa se establece el número de pasos o generaciones necesario para obtener un número de reglas fiable.

Con el objetivo de facilitar el proceso de descubrimiento de reglas a los profesores o diseñadores de cursos, los autores desarrollaron la herramienta *EPRules* (*Education Prediction Rules*). Dicha herramienta está orientada a los profesores o los diseñadores de cursos, y permite de forma sencilla e intuitiva el descubrimiento de reglas a través de distintos algoritmos: *ID3*⁶ [Quinlan 1986], *Apriori*⁷, *PRISM*⁸ [Cendrowska 1987], y el algoritmo genético.

El sistema propuesto fue probado en dos experimentos. El primer experimento consistió en la prueba del sistema, sin la herramienta *EPRules*, en un curso adaptativo en el ámbito de la medicina, en concreto, el curso versaba sobre estudios de la reumatología, y fue seguido por 30 estudiantes [Romero 2002]. Los logs analizados contienen información de los usuarios y sus interacciones, estando definidos por las variables: identificador, rendimiento, rendimiento medio, tiempo empleado en la actividad, puntuación en la actividad, número de accesos a la aplicación, y puntuación en las cuestiones. Además, los valores continuos fueron transformados en discretos, los tres primeros (rendimiento, rendimiento medio, tiempo) fueron transformados en variables discretas con los valores: muy alto, alto, medio, bajo, y muy bajo. Los valores discretos fijados para las variables relacionadas con las puntuaciones fueron: éxito en el primer intento, éxito en el segundo intento, éxito en el tercer intento, y éxito en el cuarto intento. Aplicando la metodología propuesta por estos autores se obtuvo un conjunto de reglas que expresaban relaciones interesantes, del que se puede señalar la regla que indica: *si la puntuación en la actividad 2 fue éxito en el primer intento y la puntuación de la actividad 4 fue éxito en el primer intento*

⁶Es un algoritmo desarrollado por J. Ross Quinlan para construir árboles de decisión.

⁷Algoritmo más utilizado para construir reglas de asociación.

⁸Algoritmo basado en el ID3, pero utiliza diferente estrategia para inducir las reglas.

entonces la puntuación de la actividad 1 será éxito en el primer intento con una probabilidad de 0,73.

El segundo experimento, realizado con la herramienta *EPRules*, trató de demostrar la efectividad de la metodología frente a otros algoritmos. Para ello, se analizaron los logs de un curso sobre el sistema operativo Linux desarrollado en el sistema *AHA!* [de Bra 2001, de Bra 2002b, de Bra 2002a], y seguido por 50 estudiantes [Romero 2004, Romero 2003]. Los algoritmos con los que se comparó el algoritmo genético fueron: *Apriori*, *ID3*, y *PRISM*. Los resultados mostraron que estos algoritmos generan un conjunto mayor de reglas, y son más rápidos que el algoritmo genético. Pero, desde el punto de vista de los autores la obtención de un conjunto de reglas reducido es más beneficioso para los profesores, pues éstos se pueden ver saturados si el conjunto de reglas es grande. El conjunto de reglas obtenido mostró relaciones en tres aspectos: tiempo, éxito, y nivel. Por ejemplo, una de las reglas obtenida indicaba que *la mayoría de estudiantes tuvieron problemas (tardaron más de 60 minutos y fallaron) con la cuestión 3 del test inicial en el capítulo 3*. Esta regla puede indicar al profesor que debería modificar esta cuestión para mejorar el rendimiento del curso.

Los resultados en ambos experimentos fueron satisfactorios, pues se obtuvieron reglas importantes para la posible mejora de un curso adaptativo. En este sentido, esta tesis utiliza un enfoque similar al seguido por Romero et al., pero la metodología empleada es diferente.

Adicionalmente, Romero et al. desarrollaron una herramienta de recomendación mediante el análisis de secuencia de patrones en los logs [Romero 2006b]. El objetivo de esta herramienta es crear recomendaciones (lista de enlaces a páginas web) a partir de las secuencias descubiertas para que los estudiantes las puedan aprovechar mientras interactúan con un sistema *e-Learning*. La herramienta fue integrada en el sistema *e-Learning AHA!*, y permite a los profesores realizar las siguientes tareas:

- Seleccionar los datos de los que se quieren obtener recomendaciones (selección de cursos, de estudiantes, de tiempo límite).
- Ejecutar uno de los algoritmos de descubrimiento de patrones: *AprioriAll* [Agrawal 1995], *GSP* [Srikant 1996], *PrefixSpan* [Pei 2001].
- Presentación de las secuencias descubiertas.
- Selección de las secuencias que se utilizarán en las recomendaciones. Cada una de las secuencias está formada por dos partes: antecedente y consecuente. El antecedente representa la página en la que la recomendación será mostrada, mientras que la recomendación se halla en el consecuente.
- Guardar las recomendaciones en el sistema *AHA!*.

Su et al. proponen una forma de analizar las interacciones de los estudiantes para que el profesor pueda comprender las razones por las que un estudiante obtuvo una puntuación alta o baja. En concreto, su propuesta se centra en la generación

automática de un árbol de actividades personalizado para cada estudiante en cursos desarrollados bajo el estándar *SCORM*⁹. El método presentado se sirve de técnicas como análisis de patrones secuenciales, *clustering*, y árboles de decisión [Su 2006]. Se divide en cuatro fases:

- **Definición del modelo de usuario:** en esta fase se define el perfil del estudiante como una estructura formada por: identificador del estudiante, características de aprendizaje del estudiante, secuencia de aprendizaje del estudiante. A su vez, las características de aprendizaje son: género (masculino, femenino), edad (baja, media, alta), nivel educativo (elemental, junior, senior, bachiller, universitario), experiencia previa con ordenadores (sí, no), motivación de aprendizaje (baja, media, alta), estilo cognitivo (*Field Dependence*, *Field Independence*), estilo de aprendizaje (*Doing*, *Watching*, *Thinking*, *Feeling*)¹⁰ [Kolb 2006]). La secuencia de aprendizaje está formada por las actividades visitadas por el estudiante.
- **Extracción del patrón de aprendizaje:** en esta fase se extraen los patrones más frecuentes de aprendizaje de los estudiantes con alto rendimiento (se utiliza el algoritmo modificado *GSP* [Srikant 1996]). Una vez extraídos, se codifica una tabla donde las columnas son los patrones y las filas son los identificadores de los estudiantes. Los valores que pueden tener las celdas son binarios, 1 (si el patrón está contenido en la secuencia de aprendizaje del estudiante) ó 0 (si el patrón no está contenido). Se aplica *clustering* a la tabla anterior para agrupar a los estudiantes según su forma de aprendizaje.
- **Construcción del árbol de decisión:** se forma una tabla con las características de aprendizaje de cada estudiante y el *cluster* al que pertenece. Se construye el árbol de decisión (mediante el algoritmo ID3 [Quinlan 1986]) tomando como conjunto de entrenamiento 2/3 de los datos de la tabla, y el resto para *testing*.
- **Generación del árbol de actividades:** se crea un árbol de actividades para cada uno de los *clusters*, transformando cada patrón de aprendizaje en reglas de secuenciación y navegación. De esta forma, cuando un nuevo estudiante entra en el sistema se predice en qué cluster de aprendizaje encaja (mediante el árbol de decisión), y a partir de éste se genera el árbol de actividades personalizado.

⁹*Sharable Content Object Reference Model (SCORM)* es el estándar más popular para la creación de contenidos educativos. Fue creado en 1997 por el *Department of Defense* y la *White House Office of Science and Technology Policy* bajo la iniciativa de *ADL (Advanced Distributed Learning)* para que los cursos creados tuvieran las siguientes características: reusabilidad, accesibilidad, interoperabilidad, y durabilidad [ADL 2004].

¹⁰El modelo de estilos de aprendizaje presentado por David Kolb divide los estilos de aprendizaje en un eje de coordenadas, donde el eje de las abscisas está formado por las variables opuestas *doing* y *watching*, y el eje de ordenadas por las variables *feeling* y *thinking*.

Dicho sistema fue implementado y probado con 90 estudiantes en el curso “*Equation of a Straight Line*”. Se dividió a los estudiantes en dos grupos: grupo de control (todos los estudiantes reciben el mismo árbol de actividades) y grupo experimental (los estudiantes reciben un árbol de actividades personalizado). Estos autores demostraron mediante el *test de la t de Student* que el grupo experimental obtuvo mejores puntuaciones que el grupo de control, de hecho esta diferencia fue significativa. Por tanto, los autores demostraron que esta creación de un curso personalizado produjo una mejora significativa en el rendimiento de los estudiantes. El sistema desarrollado por Su et al. es sumamente interesante, ya que este sistema consigue la creación de diferentes estructuras del curso de forma automática, tarea que es especialmente complicada en el caso de los sistemas *AEH*. Por esta razón, este sistema se puede encuadrar en los sistemas modificadores, ya que se modifica la estructura del curso. Sin embargo, los autores no indican de qué manera se podría evaluar si las decisiones tomadas de manera automática por sistema son las adecuadas para el conjunto de estudiantes, o si determinado grupo de estudiantes mostró problemas en algunas actividades, pues podría darse el caso de que los estudiantes de referencia que utiliza el sistema no son en realidad tan “buenos” como el sistema cree. Por tanto, este sistema depende en gran medida de la calidad de los cuestionarios y de la selección de los estudiantes de referencia o estudiantes con alto rendimiento. En este sentido, esta tesis plantea resolver el problema de la evaluación del rendimiento de estos sistemas mediante la detección de patrones, que indican que un grupo de estudiantes tuvieron dificultades en algunas de las partes del curso.

Merceron et al. [Merceron 2003] proponen una metodología para realizar el análisis de logs que permita extraer la información útil de las respuestas de los estudiantes con el fin de mejorar el proceso de enseñanza. El sistema que utilizan estos autores es *Logic-ITA* [Lesta 2002] (*Intelligent Teaching Assistant of logic exercises*), e incluye el presentador de cursos *Logic Tutor* [Abraham 2001]. Dicho sistema realiza las siguientes funciones: propone ejercicios a los estudiantes, monitoriza las respuestas de los ejercicios resueltos por los estudiantes, y muestra mensajes de error y sugerencias en el caso de que las respuestas sean incorrectas. Un ejercicio está formado por un conjunto de fórmulas, compuesto por una serie de premisas y una conclusión. De esta forma, resolver el ejercicio consiste en derivar la conclusión de las premisas utilizando reglas lógicas en un número determinado de pasos. Los ejercicios se presentan a los estudiantes en forma de tablas, donde cada fila representa a un paso o línea, y las columnas indican: las premisas utilizadas (premisas), la línea de derivación (número), la fórmula derivada (fórmula), la regla lógica utilizada (justificación), y las líneas en las que la regla lógica es aplicada (referencias). Además, el sistema permite cinco niveles distintos de progreso, así si el estudiante completa de forma correcta una serie de ejercicios se le sube de nivel (el nivel más bajo es el 1 y el más alto el 5). Los logs almacenan toda la información anterior y la información relacionada con los fallos. Por ejemplo, si uno de los pasos de un determinado ejercicio fue incorrecto porque el estudiante aplicó la regla *Modus Ponens* (se representa como *MP*) en vez de *Modus Tollens* (*MT*), se muestra al estudiante el mensaje “la regla *Modus Ponens* no puede ser aplicada, inténtalo con *Modus To-*

llens”, y se almacena en los logs la siguiente línea “uso incorrecto de Modus Ponens”. El análisis que plantean estos autores se divide en dos etapas: **análisis de estadísticos descriptivos** y **análisis de asociación de fallos**. El primero consiste en analizar los aspectos relacionados con:

- **Análisis de los fallos más comunes:** se computan los fallos más comunes hechos por los estudiantes, número de fallos por regla lógica, porcentaje de uso incorrecto de cada regla.
- **Análisis de ejercicios:** se comprueban los ejercicios que causaron la mayor parte de los errores, media del rendimiento de todos los estudiantes en una cuestión determinada.
- **Nivel de progreso de los estudiantes:** se analiza el número de estudiantes por cada nivel, estudiantes que estuvieron estancados en el nivel 1 en más de 10 ejercicios.

La segunda etapa consiste en encontrar las asociaciones existentes entre los fallos producidos. Se divide en dos fases:

- **Obtener las transacciones de cada estudiante,** donde una transacción está formada por la lista de fallos cometidos por un estudiante en un determinado ejercicio.
- **Obtener las reglas de asociación del conjunto de transacciones.**

Los autores aplicaron este tipo de análisis al conjunto de los logs generados por 279 estudiantes al realizar un curso sobre lógica proposicional en el sistema *Logic-ITA*. Los principales resultados que obtuvieron se pueden resumir en:

- **Fase de estadísticos descriptivos:** se encontraron un total de 2746 fallos, de los cuales el más frecuente fue usar la premisa incorrecta con las reglas *Indirect Proof* y *Conditional Proof*. El mayor número de fallos se registró en la regla *Modus Ponens*. De hecho, el 30 % de las veces que fue aplicada fue de forma incorrecta. En el análisis de los ejercicios se detectó que los ejercicios que presentaban mayor número de fallos están relacionados con las anteriores reglas. El análisis de estudiantes obtuvo que más de la mitad de los estudiantes estuvieron en el nivel 1, el 20 % de éstos progresó hacia el siguiente nivel, y tan sólo el 10 % llegaron a uno de los tres últimos niveles (niveles 3, 4 y 5).
- **Fase de asociación de fallos:** se obtuvieron un número considerable de reglas, pero las más representativas fueron:
 1. La regla puede ser aplicada, pero la deducción es incorrecta → premisas incorrectas
 2. Premisas incorrectas → número incorrecto de referencias
 3. La regla puede ser aplicada, pero la deducción es incorrecta → número incorrecto de referencias

Las reglas 1 y 3 demuestran que el uso incorrecto de las reglas está relacionada con las premisas incorrectas y el número incorrecto de las reglas. Esta información corrobora las observaciones realizadas por los profesores, de las que se deduce que los estudiantes a menudo tienen dificultades en comprender los conceptos y detalles requeridos en un ejercicio. En concreto, los resultados demostraron que los ejercicios resultaron complejos para ellos, pues un ejercicio exige especificar los detalles de aplicación de una regla (premisas, referencias, fórmula resultante) y no basta con especificar sólo la regla lógica. Además, estos resultados indicaron que los estudiantes no comprenden la diferencia de aplicar una u otra regla, por lo que se debería sugerir a los profesores que explicaran con más detalle los distintos tipos de reglas.

El análisis de logs planteado por Merceron et al. demuestra que su aplicación puede ser muy útil para que los profesores puedan conocer mejor a sus estudiantes, es decir, quiénes presentaron más dificultades, cuáles fueron los fallos más comunes, etc. Sin embargo, pretender que los profesores o diseñadores de cursos puedan realizar consultas a una base de datos puede ser demasiado ambicioso, si tenemos en cuenta que gran parte de ellos no tienen conocimientos suficientes para realizar esta tarea. Por eso, es necesaria una herramienta que les ayude en esta tarea, algo que Merceron et al. plantean como trabajo futuro. Además, el análisis propuesto está demasiado ajustado al sistema *Logic-ITA*, por lo que sería difícilmente aplicable a otros sistemas.

Baker et al. [Baker 2004a] presentan un modelo de predicción que ayude a identificar si los estudiantes están haciendo mal uso de un *ITS*. Estos autores identifican el mal uso con el hecho de “jugar en el sistema”. Así, en [Baker 2004b] identificaron que los estudiantes que “jugaron en el sistema” aprendieron $2/3$ de lo que aprendieron los que lo utilizaron de forma correcta. Definen el término “jugar en el sistema” como el comportamiento que tiene un estudiante de realizar correctamente las actividades del sistema aprovechándose masivamente de las ayudas y sugerencias propuestas por éste. Se puede dividir este tipo de comportamiento en dos tipos: abuso en el uso de la ayuda, y resolución por el método de prueba y error. Los autores indican que este tipo de comportamiento es peligroso, ya que la frecuencia con la que un estudiante juega está relacionada con un aprendizaje bajo. Una posible solución para evitarlo puede consistir en anular las sugerencias y ayudas que ofrece el sistema. Aunque es cierto que esta solución reduciría el juego, el coste sería excesivamente alto, debido a que al mismo tiempo se puede aumentar la frustración en otros estudiantes. Por esta razón, Baker et al. proponen como solución la detección automática de este tipo de comportamiento mediante un modelo de predicción. Para la construcción de este modelo, se analizaron los logs que generaron 70 estudiantes mientras realizaban un curso sobre generación de diagramas de dispersión. Además, se añadió la información proporcionada por dos fuentes más de datos: tiempo empleado por cada estudiante en cada actividad (medido por observación directa por los profesores), y resultados de cada estudiante en las actividades. Los logs almacenan las acciones realizadas por los estudiantes (cada estudiante genera de 71 a 478 acciones) en su interacción con el sistema. De esta manera se almacenan características relacionadas con:

- La evaluación de la acción (acción correcta, incorrecta, uso de ayuda).
- Tipo de interfaz utilizada (introducción de texto, número, uso de menú, dibujo de un punto, selección de un *checkbox*).
- Probabilidad de que un estudiante conozca la habilidad requerida en una acción (variable $pKnow$).
- Primer intento de solución en un paso de un problema, o de consulta de la ayuda.
- Medida para controlar el primer intento y los sucesivos (variable $pKnow-direct$).
- Tiempo (medido en segundos) que el estudiante necesitó para realizar la acción.
- Tiempo empleado en las últimas tres o cinco acciones.
- Número de veces que el estudiante realizó incorrectamente un paso específico de un problema.
- Número de veces que el estudiante seleccionó la ayuda.
- Número de errores que el estudiante cometió en las últimas cinco acciones.

Se dividió a los estudiantes en tres grupos: estudiantes que no juegan (53 estudiantes), estudiantes que juegan, pero no afecta a su rendimiento (grupo llamado *GAMED-NOT-HURT* formado por 9 estudiantes), y los que juegan y afecta a su rendimiento (grupo llamado *GAMED-HURT* formado por 8 estudiantes). El modelo de predicción que se utilizó consistió en aplicar el algoritmo *forward-selection* (algoritmo perteneciente a *LRM* [Maris 1995]) a las tres fuentes de datos anteriores, y desarrollar un clasificador. El objetivo de este clasificador es identificar los estudiantes que juegan de los que no juegan, mediante el establecimiento de un límite en el modelo, a partir del cual se considera que un estudiante juega. Las pruebas que se realizaron obtuvieron que la fiabilidad¹¹ del clasificador es del 82 %. Además, los autores demostraron que el 88 % de los estudiantes del grupo *GAMED-HURT* fueron clasificados correctamente, y el 15 % de los estudiantes del grupo no juegan fueron clasificados como juegan. Así, Baker et al. indican que el modelo presentado no simplemente identifica a todos los estudiantes que juegan, ni identifica a todos los estudiantes que tuvieron peor rendimiento, sino que identifica a los estudiantes que juegan y mostraron bajo rendimiento, es decir, los estudiantes pertenecientes al grupo *GAMED-HURT*.

Otros trabajos no sólo proponen utilizar técnicas de la Minería de Datos, sino que proponen medidas alternativas para valorar la efectividad de éstas. Por ejemplo, **Merceron** y **Yacef** proponen utilizar unas medidas alternativas para medir

¹¹Probabilidad de que el modelo, dados dos estudiantes, uno que juega y otro que no lo hace, los identifique correctamente.

la relevancia de las reglas de asociación en lugar de los criterios *confidence* y *support* [Merceron 2008]. De esta forma, estas nuevas medidas son útiles para filtrar las reglas de asociación de baja relevancia.

Entre los trabajos más recientes, **Ben-Naim** et al. han desarrollado una herramienta que proporciona ayuda a los profesores en el proceso de evaluar el rendimiento que tienen los estudiantes en el sistema adaptativo *AeLP* (*Adaptive eLearning Platform*) [Ben-Naim 2009]. Este sistema adaptativo, desarrollado en 2006 por la *University of New South Wales*, incluye herramientas de autor para desarrollar los contenidos educativos, un módulo que presenta los contenidos, y herramientas de análisis y control del progreso de los estudiantes [Ben-Naim 2007]. La herramienta de análisis, también llamada *Interactive Solution Trace Graph*, proporciona a los profesores un gráfico de transiciones en el que pueden observar los intentos de los estudiantes al resolver los ejercicios. De esta manera, observando el gráfico los profesores pueden detectar dos tipos de comportamientos anómalos entre los estudiantes: reiteración de errores por una adaptación no efectiva y reiteración de errores por confusión de conceptos. El primer tipo de detección consiste en buscar bucles en el gráfico de transiciones, mientras que el otro tipo consiste en analizar patrones de comportamiento entre los estudiantes. Ambos tipos no son fáciles de llevar a cabo por los profesores, y por este motivo estos autores proponen detectar el primer tipo con una búsqueda exhaustiva de bucles en relación con el número de intentos, y el segundo tipo mediante el uso de reglas de asociación. Además, respecto al segundo tipo se establece una clasificación de advertencias o avisos, basado en la cobertura y la confianza de las reglas, mostrado al profesor en forma de tabla. De esta forma, el profesor puede descartar, o bien prestar atención a las advertencias mostradas por la herramienta de análisis.

2.5. Contexto de la presente propuesta

La presente propuesta se enmarca en el campo de los sistemas de detección (columna **Detección** en la tabla 2.1) mediante el uso técnicas de **clasificación** y **reglas de asociación** (filas 5 y 2 de la tabla 2.1). La tabla muestra que existen diversos trabajos relacionados con los sistemas de detección. En concreto, Merceron et al. utilizan análisis estadístico y reglas de asociación para detectar los conceptos en los que los estudiantes mostraron problemas. En este sentido, la presente propuesta persigue un objetivo similar, pues se presenta una metodología para detectar el perfil de estudiantes que tuvieron problemas, así como las actividades en las que mostraron un número de problemas significativo. Esta metodología está basada fundamentalmente en la aplicación de técnicas de clasificación. La propuesta expuesta por Merceron et al. plantea que los profesores o diseñadores de cursos sean capaces de realizar tareas relacionadas con la gestión de bases de datos, así como entender la información suministrada por las reglas de asociación. En este sentido, la metodología que se exhibe en la presente tesis tiene entre sus objetivos proporcionar facilidad en su uso e interpretación de resultados a las personas que no tienen conocimientos

en el área de Minería de Datos. Ueno, sin embargo, propone un método para la detección de procesos irregulares de aprendizaje mediante el análisis estadístico. En particular, su propuesta es útil para evaluar las actividades de contenido teórico, pues la variable principal de este estudio es el tiempo empleado en completar una actividad. En este sentido, la metodología que esta tesis presenta se puede utilizar para detectar procesos irregulares de aprendizaje, pero en el ámbito de cursos *on-line* con mayor contenido práctico que teórico.

La propuesta que plantea Romero et al. se basa en la aplicación de algoritmos genéticos para descubrir información útil que ayude a los profesores o diseñadores de cursos en la tarea de mejorar un curso. Estos autores amplían el ciclo de construcción de un curso adaptativo añadiendo dos nuevas etapas: la aplicación de la Minería de Datos, y mejorar el curso a partir de la información obtenida de la etapa anterior. De esta forma, la presente metodología es útil bajo la hipótesis de este nuevo ciclo de construcción, y la información que se proveerá a los profesores o diseñadores de cursos será útil para mejorar o corregir los puntos débiles del curso.

Minería de Datos

Índice de contenidos

3.1. Introducción	57
3.2. Reglas de asociación	60
3.3. Árboles de decisión	61
3.4. Herramientas de aplicación utilizadas	63
3.4.1. <i>Weka</i>	64
3.4.2. Paquete de programas <i>C4.5</i>	65

Este capítulo expone los conceptos básicos sobre la Minería de Datos, y en especial se explican en líneas generales las técnicas de reglas de asociación y árboles de decisión, que constituyen la base de la propuesta de esta tesis. Adicionalmente, al final del capítulo se muestran dos de las herramientas que se utilizan en el presente trabajo. El objetivo de este capítulo es exponer los conceptos necesarios para comprender los capítulos 5 y 6.

3.1. Introducción

La gran capacidad de almacenamiento de los servidores nos permite guardar grandes cantidades de datos a bajo coste, muchas veces con la idea de extraer información útil que se encuentre contenida en éstos. Normalmente, cada decisión que tomamos mientras interactuamos con un sistema *Web* es guardada en una base de datos, por ejemplo nuestras compras en un supermercado, nuestras operaciones financieras, etc. El almacenamiento de datos no es costoso en la actualidad, el valor en el mercado de los discos duros de varios Terabytes es poco más de 200 Euros. Esta facilidad de poder guardar datos causa que mientras el volumen de los datos aumenta inexorablemente, la proporción de personas que son capaces de extraer la información que está contenida en los datos, potencialmente útil, disminuye. Con esta idea nació la **Minería de Datos**, como la disciplina que nos facilita la búsqueda de patrones en los datos, es decir, la obtención de información a partir de los datos. Según la definición que ofrecen Witten y Frank [Witten 2000, p. XIX], “la *Minería de Datos* es la extracción de información implícita, previamente desconocida y potencialmente útil contenida en los datos”. Desde un punto de vista operacional, la *Minería de Datos* se define como el proceso automático o semi-automático de

descubrir patrones en grandes cantidades de datos, bajo la premisa de que los patrones descubiertos deben ser útiles [Witten 2000, pp. 3-6]. Otros autores, como Tan et al., definen la *Minería de Datos* como la tecnología que combina las técnicas y métodos tradicionales de análisis con sofisticados algoritmos para procesar grandes cantidades de datos [Tan 2006, p. 1]. Por tanto, se puede definir a la *Minería de Datos* como la tecnología que utilizando las técnicas tradicionales de análisis con sofisticados algoritmos permite descubrir, de manera automática o semi-automática, patrones útiles en grandes volúmenes de datos. Las técnicas de la *Minería de Datos* son muy numerosas, pero el presente trabajo está centrado en dos: reglas de asociación y árboles de decisión. Estas dos técnicas se caracterizan por generar resultados bastante entendibles por las personas, una de las razones por las que han sido elegidas en el presente trabajo.

caso	outlook	temperature	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

Tabla 3.1: Datos del ejemplo weather. Conjunto de datos típico para mostrar las virtudes de las técnicas de la *Minería de Datos*. Fuente: tabla 1.2 de [Witten 2000, p. 9]

Para entender qué nos proporcionan estas dos técnicas, veremos un ejemplo sencillo. Se trata de un ejemplo muy difundido en la literatura del tema, que describe las condiciones meteorológicas necesarias para practicar un determinado deporte (ver tabla 3.1). Hay que señalar que aunque este ejemplo es extremadamente sencillo, sirve para capturar la esencia del problema y entender qué pueden ofrecernos las técnicas reglas de asociación y árboles de decisión. Sin embargo, no hay que olvidar que los problemas reales en el que se aplican las técnicas de la *Minería de Datos* contienen miles, cientos de miles, o incluso millones de casos.

Un **caso o instancia** es una fila de la tabla de datos, caracterizada por los valores de las **variables o atributos** (columnas de la tabla). Por tanto, en el ejemplo

mostrado en la tabla 3.1 14 casos componen la tabla de datos. Esta tabla muestra cuatro atributos: *outlook*, *temperature*, *humidity* y *windy*, y un atributo especial, *play*, sobre si se debe jugar o no. Este atributo especial en muchas ocasiones se le denomina **clase o atributo de clasificación**. La primera técnica a analizar se compone de reglas que expresan relaciones entre los valores de los atributos, por eso a este tipo de reglas se las denomina *reglas de asociación*. Algunos ejemplos son:

- R1: **Si** *temperature* = “cool” **entonces** *humidity* = “normal”
- R2: **Si** *humidity* = “normal” **y** *windy* = “false” **entonces** *play* = “yes”
- R3: **Si** *outlook* = “sunny” **y** *play* = “no” **entonces** *humidity* = “high”
- R4: **Si** *windy* = “false” **y** *play* = “no” **entonces** *outlook* = “sunny” **y** *humidity* = “high”

Es fácilmente observable que la regla R1 se puede aplicar a los casos 5, 6, 7 y 8, ya que en estos casos cuando la variable *temperature* es *cool*, la variable *humidity* tiene el valor de *normal*. Además, la regla R2 es aplicable a los casos 5, 9, 10 y 13, pues en estos casos la humedad es normal, no hay viento, y el valor de *play* es siempre *yes*. Finalmente, la regla R3 se puede aplicar a los casos 1,2,8, mientras que la regla R4 al caso 8. Sin embargo, este conjunto de reglas contiene una pequeña porción de las relaciones existentes entre atributos. Para describir la mayoría de estas relaciones sería necesario generar al menos 100 reglas con este conjunto de datos. El conjunto de las primeras 100 reglas de asociación generadas pueden verse en el apéndice A.2.1.

La segunda técnica, *árbol de decisión* (*decision tree*, en adelante: *DT*), es una técnica de clasificación o predicción. La figura 3.1 muestra el *DT* para el ejemplo anterior¹. Podemos observar en la imagen que en un *DT* los nodos contienen los atributos, las aristas los valores de los atributos, y las hojas los valores de la clase o atributo de clasificación. Por ejemplo, el nodo raíz contiene al atributo *outlook*, mientras que los otros nodos contienen a los atributos *humidity* y *windy*. La arista *sunny* contiene un valor del atributo *outlook*, y conecta los nodos *outlook* y *humidity*. Las hojas pueden tener los valores de la clase *play*, *yes* o *no*. Una forma de entender el árbol es la siguiente: la variable *play* tomará el valor de *no* si *outlook* es *sunny* y *humidity* es *high*, o si *outlook* es *rainny* y *windy* es *true*.

El árbol de la figura 3.1 es fácilmente interpretable por una persona, sin embargo no lo es para una máquina. Como las máquinas procesan de forma muy rápida cualquier tipo de reglas, se suele utilizar un conjunto de reglas de clasificación deducidas a partir del árbol de decisión. Estas reglas reciben el nombre de *reglas de decisión* (*decision rules*, en adelante: *DR*) o *lista de decisión*, pues el orden en el que aparecen es relevante. Las siguientes reglas han sido deducidas² del árbol de decisión:

- R1: **Si** *outlook* = “sunny” **y** *humidity* = “high” **entonces** *play* = “no”

¹La salida de generación del *DT* del ejemplo se puede ver en el apéndice A.3.1 y A.3.2.

²Las reglas de decisión generadas por el programa *cl5rules* se pueden ver en el apéndice A.3.3.

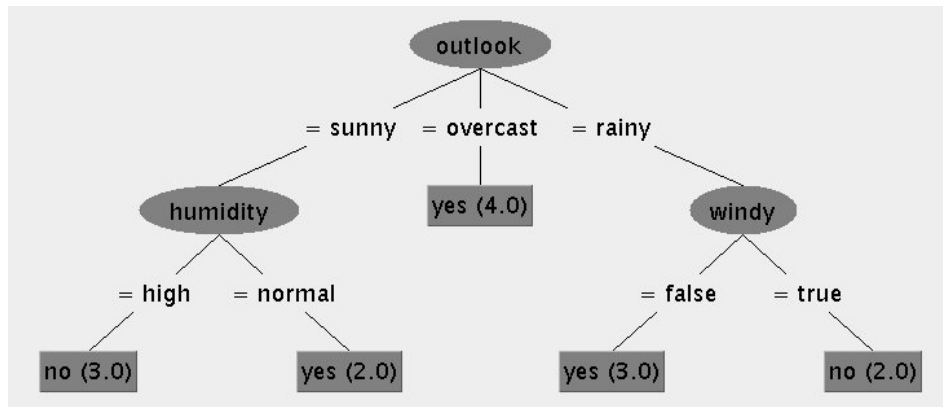


Figura 3.1: Árbol de decisión de los datos de la tabla 3.1. Generado con el programa *Weka* (versión 3.6.0) con las opciones por omisión.

- R2: Si outlook = “rainy” y windy = “true” entonces play = “no”
- R3: Si outlook = “overcast” entonces play = “yes”
- R4: Si humidity = “normal” entonces play = “yes”
- R5: Si outlook = “rainy” y windy = “false” entonces play = “yes”
- R6: En cualquier otro caso play = “yes”

Las *DR* deben ser interpretadas en orden. Por ejemplo, si un caso no encaja con la primera regla, habría que mirar si encaja con la segunda, si no encaja con la segunda, habría que probar con la siguiente, y así sucesivamente hasta que no queden más reglas. En caso de no encajar con ninguna, se aplicaría la última regla, R6, regla por omisión.

A continuación se formalizan algunos de estos conceptos.

3.2. Reglas de asociación

Las reglas de asociación permiten descubrir las relaciones existentes entre atributos. Por tanto, pueden ser utilizadas para predecir cualquier atributo o combinación de atributos.

Definición 3.2.1 Sean X e Y conjuntos de pares atributo-valor, se define una **regla de asociación** como una implicación del tipo $X \rightarrow Y$, donde X e Y son conjuntos disjuntos, i.e., $X \cap Y = \emptyset$. [Tan 2006, p. 329]

Ya que muchas reglas de asociación pueden ser derivadas, incluso de pequeños conjuntos de datos, se utilizan los conceptos cobertura (*support*) y precisión (*confidence*) para seleccionar las mejores. La **cobertura** determina cuántas veces una regla

se puede aplicar a un determinado conjunto de datos, es decir, cuántos casos contienen a X . El concepto *precisión* es la proporción de casos que la regla predice correctamente, es decir, la proporción de casos que contienen a X e Y . Por ejemplo, la siguiente regla extraída del apéndice A.2.1 tiene un valor de support de 4, ya que 4 casos (caso 3, 7, 12 y 13 de la tabla 3.1) contienen el valor de *outlook* = *overcast*.

1. outlook=overcast 4 ==> play=yes 4 **conf:(1)**

Por otro lado, se observa que el valor de la precisión es 1 (100 %), ya que de los 4 casos que contienen la parte derecha de la regla, los cuatro contienen que el valor del atributo *play* = *yes*.

3.3. Árboles de decisión

Clasificación es la tarea de asignar objetos a una de las categorías predefinidas. Los datos de entrada que recibe una técnica de clasificación son una colección de instancias o casos, cada uno de los cuales contiene los valores de los atributos y el valor de un atributo especial, conocido también como clase o variable de clase. En este sentido, una técnica de clasificación es aquella que utiliza una función objetivo o modelo de clasificación para mapear cada atributo o conjunto de atributos a una de las clases. Un modelo de clasificación se puede usar como un modelo descriptivo, para distinguir objetos de clases diferentes; o como un modelo predictivo, para predecir la clase de nuevas instancias [Tan 2006, pp. 145-148]. Los árboles de decisión son técnicas de clasificación.

Definición 3.3.1 *Un árbol de decisión es una estructura formada por un nodo raíz, nodos de decisión, hojas y aristas. Donde cada una de las partes está caracterizada por:*

- **nodo raíz:** no recibe aristas entrantes, tiene cero o más aristas salientes, y contiene uno de los atributos.
- **nodo de decisión:** recibe únicamente una arista entrante, tiene una o varias aristas salientes, y contiene uno de los atributos.
- **hoja:** nodo terminal que sólo recibe una arista entrante, no tiene aristas salientes, y contiene la clase.
- **arista:** une dos nodos (nodo inicial y nodo final) o un nodo con una hoja, y contiene un valor del atributo del nodo inicial.

La figura 3.1 muestra el *DT* para los datos de la tabla 3.1. Los nodos están dibujados en el árbol mediante elipses, y las hojas usando cuadrados. El nodo raíz corresponde al atributo *outlook*, los nodos de decisión son: *humidity* y *windy*. Cada una de las aristas contiene un valor del nodo precedente, por ejemplo, las aristas *high* y *normal* son los valores posibles del atributo *humidity*. El árbol contiene cinco hojas, con tres clases de *yes* y dos para la clase *no*. Cada una de las hojas contiene el valor

de la clase seguido de un número entre paréntesis. Este número indica el número de instancias que, siguiendo el camino desde el nodo raíz hasta la hoja, contienen los mismos valores de los atributos de ese camino. En algunos casos puede aparecer un segundo número en la hoja, que indica el número de instancias incorrectamente clasificadas en la hoja. Por ejemplo, el camino: `outlook=sunny → humidity=high → no (3.0)` indica que hay 3 instancias que contengan estos valores en los atributos (casos 1, 2 y 8 de la tabla 3.1) y en este caso están correctamente clasificadas.

Definición 3.3.2 Una *hoja* de un árbol de decisión está formada por los siguientes elementos: $\langle \text{valor de la clase} \rangle (N|E)$. Sea un camino formado por $R \xrightarrow{v_1} A_1 \xrightarrow{v_2} \dots \xrightarrow{v_{m-1}} A_m \xrightarrow{v_m} C_1$, donde R es el nodo raíz, el conjunto de los nodos de decisión es $\{A_1, A_2, \dots, A_m\}$, y las aristas son los $v_i, \forall i \in [1, m]$, se definen:

- **N**: número de instancias que contienen a los atributos $v_i, \forall i \in [1, m]$.
- **E**: número de instancias que contienen a los atributos $v_i, \forall i \in [1, m]$ con clase distinta a C_1 .

La hoja **no (3.0)** del ejemplo anterior muestra un único valor correspondiente al parámetro **N**. No se muestra el valor del parámetro **E**, que es 0, ya que las tres instancias fueron clasificadas correctamente³.

Generalmente la construcción de un *DT* se realiza a partir de un subconjunto de los datos. Este subconjunto suele estar formado por el 80 % de los datos, y recibe el nombre de datos de entrenamiento (*training data*). El 20 % restante se utiliza para medir la precisión del modelo, y recibe el nombre de datos no vistos (*unseen data*). Debido a la forma en la que se genera un *DT*, los errores producidos en éstos pueden ser de dos tipos: errores de entrenamiento y errores de generalización. Un error de entrenamiento es el número de instancias erróneamente clasificadas del conjunto de datos de entrenamiento, mientras que el otro tipo de error corresponde al error esperado en la clasificación de los datos no vistos [Tan 2006, p. 172]. Por tanto, la situación ideal es que un *DT* tenga bajos ambos tipos de errores. Sin embargo, en muchas ocasiones un árbol que esté excesivamente ajustado a los datos de entrenamiento (error de entrenamiento bajo) puede tener un error de generalización más alto que otro que se ajuste menos a los datos de entrenamiento (error de entrenamiento alto). Tal situación es conocida con el nombre de sobre-ajuste (*overfitting*). Dicho de otro modo, las predicciones obtenidas a partir del árbol son correctas para el conjunto de datos de entrenamiento, pero sin embargo, pueden ser erróneas para un conjunto de datos nuevo. Con el objetivo de evitar este tipo de situaciones se utiliza la poda (*prunning*) en el árbol. La poda se realiza en función de un parámetro llamado *confidence factor* (*CF*), valores pequeños de *CF* producen mayor poda, mientras que la poda es menor con valores altos de este parámetro. En muchas ocasiones, se puede tomar como valor de referencia el nivel del 25 % ($CF = 0.25$); de hecho tanto *Weka* como el programa *C4.5* lo utilizan como valor de referencia si no es especificado un valor determinado para el parámetro *CF*.

³Más información sobre los árboles de decisión se puede encontrar en [Tan 2006, capítulo 4] y [Witten 2000, capítulo 6]

Aunque la poda en los árboles de decisión genera árboles más compactos que los no podados, siguen siendo incómodos y complejos de manejar por dos razones:

- Los árboles de decisión son difíciles de interpretar, pues cada nodo de decisión tiene un contexto específico establecido por los resultados de los tests de los nodos antecedentes.
- La propia estructura de un árbol puede causar que los subconceptos aparezcan fragmentados. Esto quiere decir que pueden aparecer subárboles idénticos en distintas ramas, lo que origina que el árbol sea difícil de interpretar.

Las reglas de decisión pretenden resolver estos dos problemas. Las siguientes definiciones determinan la estructura de una regla de decisión.

Definición 3.3.3 *Una regla de decisión es una implicación del tipo $L \rightarrow R$, en el que la parte izquierda, L , es una conjunción de tests de atributos (condiciones), y la parte derecha, R , es la clase.*[Quinlan 1993, p. 9]

Definición 3.3.4 *Se define **precisión** de una regla como el número de veces, en promedio, que la regla clasifica correctamente los casos no vistos que satisfacen la parte izquierda de la regla.*[Quinlan 1993, p. 11]

Por ejemplo, la regla 1, mostrada a continuación, está formada por una parte izquierda, y una parte derecha con la precisión delimitada entre corchetes. L está formado por la conjunción de las condiciones *outlook = sunny* y *humidity = high*. R es la clase *no*, y la precisión de la regla es del 63 %. Esto quiere decir que la regla del ejemplo clasifica correctamente el 63 % de los casos del conjunto de los no vistos. Hay que señalar que este valor es una aproximación del error (100 - precisión) que tendría la regla en nuevos casos.

Rule 1:

```
outlook = sunny
humidity = high
->class no [63.0 %]
```

3.4. Herramientas de aplicación utilizadas

Esta sección muestra dos de las herramientas más utilizadas en la actualidad para generar árboles de decisión y reglas de asociación, y que han sido usadas en el presente trabajo. No obstante, en la actualidad existen multitud de herramientas de código abierto tal y como se señala en el estudio realizado por Zupan y Demsar [Zupan 2008], en el que analizan una muestra de las herramientas más relevantes, de las que podemos destacar, a parte de las nombradas en esta sección, **R**⁴, **Orange**⁵ y **RapidMiner**⁶. Más información relativa a herramientas de la *Minería de*

⁴<http://www.r-project.org/>

⁵<http://www.ailab.si/orange/>

⁶Conjunto de herramientas anteriormente conocido como *YALE* (*Yet Another Learning Environment*). La web actual de RapidMiner se puede encontrar en <http://rapid-i.com/>

Datos, tanto de código abierto como comerciales, se puede encontrar en *KDnuggets* [KDnuggets 2009].

3.4.1. *Weka*

Weka es una colección de algoritmos de minería de datos y herramientas para el preprocesamiento de los datos. Fue desarrollado por la *University of Waikato* (Nueva Zelanda), y el nombre proviene de *Waikato Environment for Knowledge Analysis*. Esta aplicación fue desarrollada en *JAVA* y es distribuida bajo la licencia *GNU General Public License*⁷.

Weka proporciona las implementaciones de algoritmos, como técnicas de clasificación, reglas de asociación, y análisis *cluster*, que fácilmente se pueden aplicar a un conjunto de datos. Además incluye una variedad de herramientas para realizar el preprocesado de los datos, necesario en las técnicas de la *DM*: transformaciones, discretizaciones, filtros, análisis estadísticos descriptivos, visualización, etc. (Más información sobre cómo utilizar *Weka* se puede encontrar en [Witten 2005].)

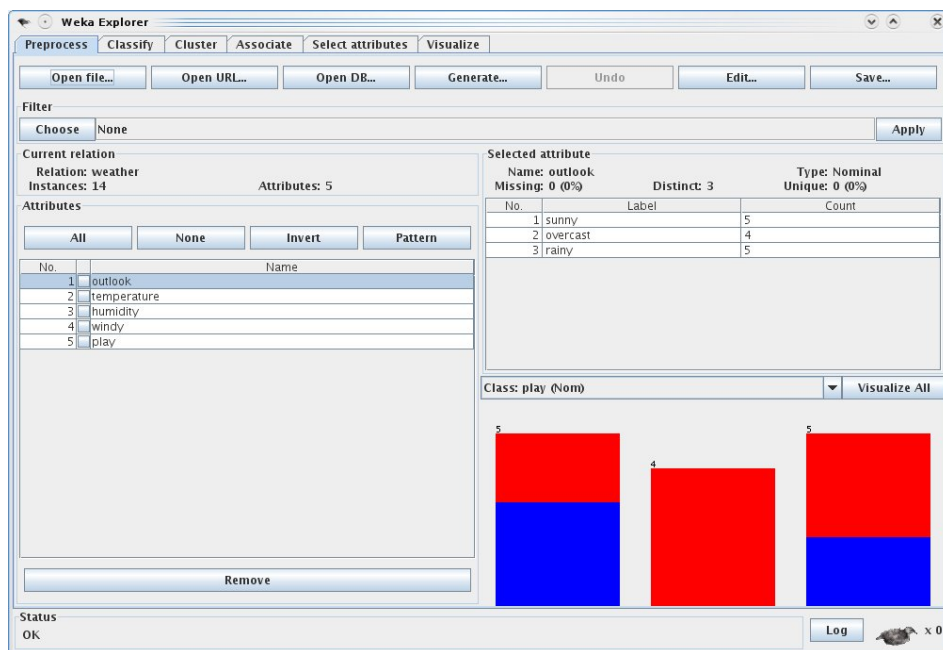


Figura 3.2: Interfaz de *Weka*: módulo de preprocesamiento de datos

Las figuras 3.2, 3.3 y 3.4 muestran las distintas interfaces correspondientes a una de las aplicaciones de *Weka*, el *Explorer*. La figura 3.2 muestra la interfaz de preprocesamiento de datos que proporciona *Weka*. En este caso se analizan los datos del ejemplo anterior (ver apéndice A.1). La parte izquierda de la figura muestra los atributos, mientras que la parte derecha muestra estadísticas (número de instancias

⁷ *Weka* está disponible en <http://www.cs.waikato.ac.nz/ml/weka>.

por cada valor del atributo) sobre el atributo seleccionado (*outlook* en este caso), variable de clase (*play*), y visualización de los datos respecto a la variable de clase. Este módulo es especialmente útil para un análisis preliminar de los datos.

La figura 3.3 muestra la interfaz correspondiente al módulo de técnicas de clasificación, en el que se puede observar el tipo de técnica elegida (algoritmo *J48*⁸), opciones de ejecución del algoritmo, tipos de test de evaluación, y salida de resultados del algoritmo.

La interfaz correspondiente al módulo de reglas de asociación muestra información relativa al tipo de algoritmo, sus parámetros de ejecución, y la salida de resultados (ver figura 3.4).

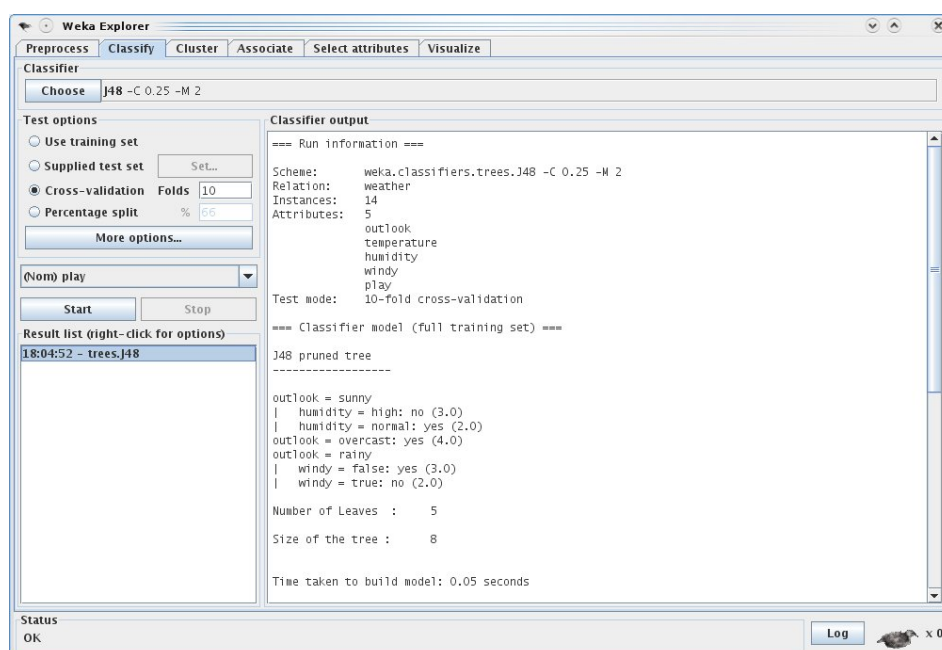


Figura 3.3: Interfaz de *Weka*: módulo de técnicas de clasificación

3.4.2. Paquete de programas *C4.5*

Los programas *c4.5* y *c4.5rules*⁹ implementan el algoritmo *C4.5*, desarrollado por J. Ross Quinlan en 1993 [Quinlan 1993], y están incluidos en el paquete de programas *C4.5*. *C4.5* es el nombre de un conjunto de programas que construyen modelos de clasificación mediante el análisis y descubrimiento de patrones encontrados en los datos. Estos programas requieren que el fichero de datos a analizar sea de tipo plano con las siguientes restricciones:

⁸Técnica de clasificación que genera un *DT* utilizando el algoritmo *C4.5*.

⁹Sitio web donde se puede descargar el programa *c4.5* y *c4.5rules*: <http://www.rulequest.com/Personal/>.

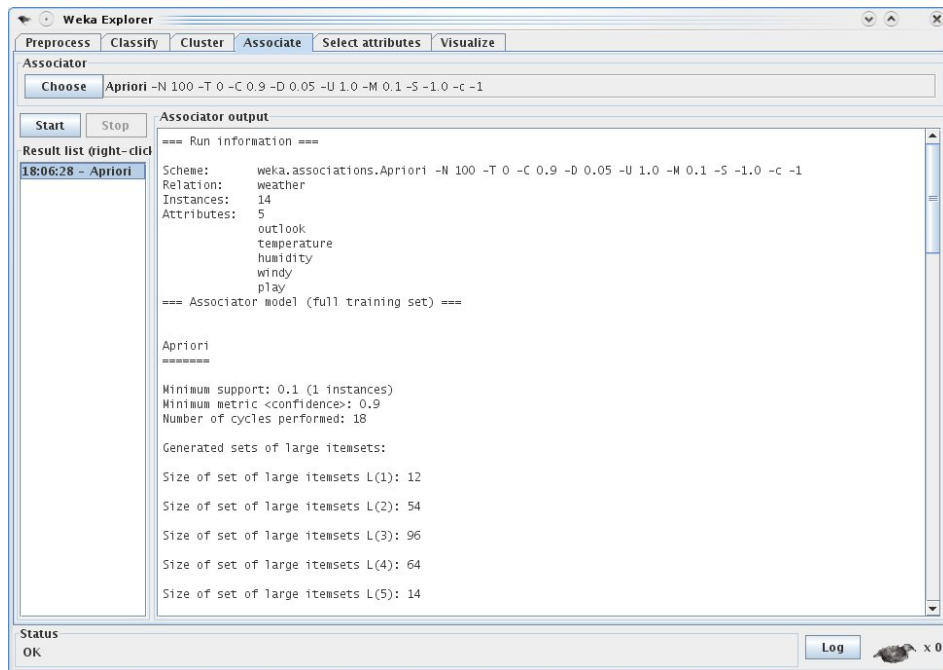


Figura 3.4: Interfaz de *Weka*: módulo de reglas de asociación

- Todos los casos deben estar descritos por los mismos atributos (no con los mismos valores), y los valores de los atributos pueden ser de tipo numérico o discreto.
- Las categorías o clases tienen que ser definidas a priori, por tanto esta técnica es de tipo supervisada.
- Las clases tienen que ser discretas, y en caso de tener una clase continua es necesario discretizarla.
- La cantidad de datos a analizar debe ser suficientemente grande, cientos o miles de datos de entrenamiento, para construir modelos fiables.

3.4.2.1. Programa *c4.5*

Es el principal programa del paquete *C4.5*. Genera un árbol de decisión siguiendo el algoritmo *C4.5*. A continuación se muestra en la figura 3.5 la parte superior de la salida de la ejecución del programa *c4.5* con los datos del ejemplo anterior.

Se puede observar que se muestran las opciones de ejecución que se han solicitado, así como la información del número de instancias en los datos (14 en este ejemplo) y número de atributos (cuatro en este ejemplo).

La parte central muestra el árbol de decisión obtenido, y en la parte final se muestran estadísticas sobre el modelo de clasificación antes de realizar la poda y después de realizar la poda (ver figura 3.6). La columna *size* indica el tamaño del

```
C4.5 [release 8] decision tree generator Wed Sep  2 21:15:20 2009
-----
```

```
Options:
File stem <weather>

Read 14 cases (4 attributes) from weather.data
```

Figura 3.5: Parte superior de la salida de un ejemplo de ejecución del programa *c4.5*

DT, es decir, la suma de nodos y hojas (3 nodos y 5 hojas). La siguiente columna, *errors*, contiene los errores producidos con el conjunto de datos de entrenamiento. Estas dos primeras columnas corresponden a parámetros del *DT* antes de la poda (más adelante se explica de forma más detallada este concepto), mientras que las tres siguientes son parámetros del árbol después de realizar la poda. La última columna, *estimate*, ofrece la estimación del error después de la poda. La salida completa del programa con las opciones por omisión se puede observar en el apéndice A.3.2 (información sobre las opciones de ejecución se puede encontrar en [Quinlan 1993, pp. 81-87]).

Una opción interesante que nos proporciona este programa, y que no está disponible en el algoritmo *J48* de *Weka*, es poder agrupar los atributos con valores discretos. Esta opción es especialmente útil para reducir el tamaño del *DT* resultante.

Otros aspectos a decidir son cómo y cuándo realizar la poda, cuyo objetivo principal es disminuir el sobre-ajuste. Quinlan propone utilizar la tasa de error (*error rate*) como indicador de poda mediante el método de *poda pesimista*. Este método consiste en estimar la tasa de error para cada rama (suma de las tasas de error de las hojas que derivan de la rama) y la tasa de error para la hoja que sustituiría a la rama. Si la tasa de error estimada para la nueva hoja es menor que la de la rama, se sustituye la rama por la nueva hoja. Bajo el supuesto de que la tasa de error es E/N , siendo E el número de sucesos y N el número de intentos, se puede decir que la probabilidad de que se produzca un suceso sigue una distribución Binomial $B(N,p)$. Por tanto, E es una variable aleatoria binomial y su probabilidad puede ser estimada mediante un intervalo de confianza.

Definición 3.4.1 *Se define tasa de error estimado como el límite superior del intervalo de confianza de nivel CF para una distribución Binomial $B(N,p)$. Se denota a esta tasa como $U_{CF}(E, N)$, siendo CF el confidence factor, E número de instancias mal clasificadas (errores), y N número de instancias totales en la hoja (errores+aciertos). [Quinlan 1993, p. 41]*

Como es sabido, la probabilidad de una distribución Binomial $B(n,p)$ se obtiene mediante la siguiente fórmula (siendo k el número de sucesos y N el número de

Decision Tree:

```

outlook = overcast: yes (4.0)
outlook = sunny:
|  humidity = high: no (3.0)
|  humidity = normal: yes (2.0)
outlook = rainy:
|  windy = false: yes (3.0)
|  windy = true: no (2.0)

```

Tree saved

Evaluation on training data (14 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
8	0(0.0%)	8	0(0.0%)	(38.5%) <<

Figura 3.6: Parte central e inferior de la salida de un ejemplo de ejecución del programa *c4.5*

intentos):

$$P(X = k) = \binom{N}{k} p^k (1-p)^{N-k} \quad (3.1)$$

Por tanto, el intervalo superior de confianza de p se puede obtener despejando p_U de la siguiente ecuación (siendo α el nivel de significación):

$$\sum_{i=0}^k P(X = i) = \frac{\alpha}{2} \Rightarrow \sum_{i=0}^k \binom{N}{i} p_U^i (1-p_U)^{N-i} = \frac{\alpha}{2} \quad (3.2)$$

Como el nivel de confianza debe ser CF , entonces $\alpha/2 = CF$. Por ejemplo, para una hoja en la que el número de casos totales fue 16, y un caso fue mal clasificado, la tasa de error estimada es $p_U = U_{25\%}(1, 16) = 0,1428$.

Definición 3.4.2 Sea N el número de instancias totales en una hoja y E el número de instancias incorrectamente clasificadas, se define el **error estimado** \hat{E} en una hoja como $N \times U_{CF}(E, N)$. [Quinlan 1993, p. 41]

Aplicando esta definición, el error estimado para la hoja del ejemplo anterior sería $\hat{E} = 16 \times 0,1428 = 2,5536$. Como se puede apreciar la poda pesimista toma siempre

el peor valor, pues utiliza el intervalo superior de confianza de la probabilidad del error de clasificación.

3.4.2.2. Programa *c4.5rules*

Quinlan propone una serie de pasos para generar las DR^{10} a partir del árbol de decisión.

Paso 1: Partiendo del DT sin poda el primer paso natural es transformar cada una de las ramas en reglas, empezando por una hoja y subiendo por el árbol hasta alcanzar el nodo raíz. De esta forma, se generarán tantas reglas como hojas tenga el árbol. Sin embargo, las reglas obtenidas no resuelven los problemas anteriores, ya que siguen siendo difíciles de interpretar (hay una regla por cada hoja), y pueden aparecer condiciones irrelevantes que no afectan a la precisión. Por este motivo, es necesario un segundo paso en el que se eliminen las condiciones irrelevantes.

Paso 2: Quinlan propone utilizar la “tasa de error pesimista” (*Pessimistic Error Rate*, en adelante: PER) para determinar qué condiciones afectan en menor medida a la precisión de la regla.

Definición 3.4.3 Se define *tasa de error pesimista* de una regla como $PER = U_{CF}(E, N)$.

En concreto, este autor propone seguir el siguiente procedimiento:

1. Calcular el PER de la regla sin eliminar ninguna condición, se obtiene PER_R .
2. Para cada condición, C_i , calcular el PER de la regla eliminando C_i , se obtiene para cada C_i el PER_{R-C_i} .
3. Obtener $PER_{R-C_m} = \min \{PER_{R-C_i}\}$. Si $PER_{R-C_m} < PER_R$ ir a 4, sino terminar.
4. Eliminar la condición C_m , y volver a 1 con $R = R - C_m$.

Por ejemplo, supongamos que una regla está formada por tres condiciones ($C1$, $C2$, $C3$), y clasifica incorrectamente un caso ($E = 1$) de tres ($N = 3$). Por tanto, su PER es de 0,6737, i.e. 67%. Imaginemos los siguientes supuestos:

- Eliminando $C1$ el número de casos incorrectamente clasificados sigue siendo 1, pero N aumenta a 4.
- Eliminando $C2$ el número de casos incorrectamente clasificados sigue siendo 1, pero N aumenta a 7.
- Eliminando $C3$ tanto N como E permanecen constantes.

Calculando los PER :

- $PER_{R-C1} = U_{25\%}(1, 4) = 0,5437 \rightarrow 54\%$

¹⁰Quinlan denomina a estas reglas *production rules*.

- $PER_{R-C2} = U_{25\%}(1, 7) = 0,3407 \rightarrow 34\%$
- $PER_{R-C3} = U_{25\%}(1, 3) = 0,6737 \rightarrow 67\%$

Es claro que la condición que menor PER tiene es C2, por lo que se eliminaría aplicando de nuevo el procedimiento a la regla menos la condición C2.

El problema que tienen las reglas obtenidas en este paso es que algunas de ellas pueden contener una tasa de error muy alta. Por eso, el siguiente paso que propone este autor es obtener un subconjunto de reglas, S , que minimice los falsos positivos y los falsos negativos.

Definición 3.4.4 Sea S el subconjunto de reglas que cubren una particular clase C , se definen:

- **Falsos positivos** como el conjunto de casos cubiertos por S , pero que no pertenecen a la clase C .
- **Falsos negativos** como el conjunto de casos pertenecientes a la clase C que no fueron cubiertos por el subconjunto S .

[Quinlan 1993, p. 51]

Paso 3: Quinlan propone utilizar el principio de *Minimum Description Length* (MDL) para seleccionar el mejor subconjunto S . Este principio consiste en que el mensaje óptimo para enviar entre un emisor y un receptor es aquel que minimice el número de bits requerido para codificar el mensaje. Supongamos que emisor y receptor tienen el mismo conjunto de entrenamiento, pero tan sólo el receptor conoce la clase de cada caso de entrenamiento. El emisor debe enviar esta información al receptor por medio de un mensaje. Este mensaje consiste en una teoría de clasificación con la que el receptor pueda clasificar cada uno de sus casos, y los casos que no pueden ser clasificados por esa teoría, o también llamados excepciones. En el caso que nos ocupa, la teoría de clasificación sería el subconjunto S , y las excepciones, son los casos cubiertos por las reglas que son falsos positivos y los casos no cubiertos que son falsos negativos. Teniendo esto en cuenta, como una teoría es un subconjunto de reglas, es necesario hallar el coste de codificar una regla.

Definición 3.4.5 Sea R una regla formada por n condiciones, se define el **coste de codificar una regla** como $C_R = \log_2(n!)$. [Quinlan 1993, p. 52]

Definición 3.4.6 Sea S un subconjunto de N reglas, se define el **coste de codificar S** o **coste de codificar una teoría** como $C_t = \sum_{i=1}^N C_{Ri} - \log_2(N!)$. [Quinlan 1993, p. 52]

Definición 3.4.7 Sea r el número de casos cubiertos por S de n casos de entrenamiento, fp el número de falsos positivos y fn el número de falsos negativos, se define el **coste de codificar las excepciones** como $C_e = \log_2\left(\binom{r}{fp}\right) + \log_2\left(\binom{n-r}{fn}\right)$. [Quinlan 1993, p. 52]

Definición 3.4.8 Se define *coste total de codificación* como $C_T = C_e + W \times C_t$. Siendo, $W \in [0, 1]$. [Quinlan 1993, p. 52]

Por tanto, el subconjunto S que minimice el número de falsos positivos y falsos negativos es aquel que tenga el mínimo coste total de codificación.

A continuación se muestra en la tabla 3.2 los subconjuntos de DR posibles de las obtenidas de 2514 casos de entrenamiento de un ejemplo tomado del libro de Quinlan, que muestra los datos de hipertiroidismo suministrados por el *Garvan Institute of Medical Research* de Sidney [Quinlan 1993, p. 54], así como el cálculo de los costes de teoría, de excepción y total. Por ejemplo, el coste de codificar la teoría del subconjunto de reglas $\{R4, R5, R7\}$ es $17,1 + 19,8 + 15,7 - \log_2(3!) = 50,015$, que redondeando es 50,0. Este mismo conjunto cubrió 66 casos ($r=66$), con lo que su coste de codificar las excepciones es $\log_2(\binom{66}{4}) + \log_2(\binom{2514-66}{2}) = 40,97$, y redondeando es 41,0. Por último, si tomamos $W = 0,5$ el coste total para este subconjunto es $40,97 + 0,5 \times 50,015 = 65,977$ (66,0 redondeando). El mínimo valor del coste total se obtiene con el subconjunto $\{R5, R7\}$, que es el subconjunto que sería seleccionado en este paso.

S	C_t	fp	fn	C_e	C_T
{R4}	17.1	2	12	116.8	125.4
{R5}	19.8	1	6	63.9	73.8
{R7}	15.7	1	61	411.8	419.7
{R4,R5}	35.9	3	5	64.6	82.6
{R4,R7}	31.8	3	9	97.8	113.7
{R5,R7}	34.5	2	3	42.1	59.4
{R4,R5,R7}	50.0	4	2	41.0	66.0

Tabla 3.2: Coste de codificación de subconjuntos de DR . Fuente: datos de hipertiroidismo suministrados por el *Garvan Institute of Medical Research* de Sidney. [Quinlan 1993, p. 54]

Como se ha visto, en este último paso se logra obtener un subconjunto de DR para cada una de las clases. Sin embargo, en muchas ocasiones el orden en que aparezca cada subconjunto puede producir que el número de falsos positivos no sea el mínimo. Por eso, el último paso que propone Quinlan es optimizar el rendimiento de los subconjuntos obtenidos, es decir, minimizar el número de falsos positivos.

Paso 4: Este proceso se divide en tres fases. La primera fase consiste en **establecer un orden en los subconjuntos de reglas**, es decir, establecer un orden de los subconjuntos de clase que minimice el número de falsos positivos. Con este propósito, cada uno de los subconjuntos de clase son examinados, y se selecciona aquel que tenga el menor número de falsos positivos. Este subconjunto seleccionado corresponde a la primera clase. De nuevo se calculan los falsos positivos y se vuelve a seleccionar otra clase, y así sucesivamente. Respecto a la segunda fase, este autor indica que es necesario **establecer una clase por defecto**, i.e. aquella que clasifica

los casos no cubiertos por las reglas anteriores. Se elige la clase que clasifique a la mayoría de casos de entrenamiento no cubiertos por las reglas. La última fase consiste en **refinar** el conjunto obtenido en las dos fases anteriores, eliminando aquellas reglas cuya omisión reduzca el número de errores de clasificación.

Como resultado de los cuatro pasos anteriores se obtienen las *DR* optimizadas del *DT*. Cabe destacar, que el proceso explicado de forma detallada en esta sección muestra al lector la complejidad de reducir un *DT* a una lista de *DR*.

El programa *c4.5rules* genera el listado de *DR* a partir de un *DT* sin poda (información a cerca de las opciones de ejecución del programa *c4.5rules* se puede encontrar en [Quinlan 1993, pp. 87-88]). La salida de la ejecución de este programa se muestra en el apéndice A.3.3. Al igual que en el programa *c4.5*, en la parte superior se muestran las opciones de ejecución e información sobre los datos. La parte central ofrece el conjunto de *DR*, y la parte final presenta un *ranking* de las reglas y un análisis de la precisión del modelo de clasificación. Veremos ahora como interpretar las reglas, por ejemplo la siguiente regla extraída del apéndice A.3.3:

Rule 4:

```
outlook = rainy
windy = false
->class yes [63.0 %]
```

Se puede observar como esta regla está formada por dos condiciones o parte izquierda, es decir, *outlook* es *rainy* y *windy* es *false*; y por un consecuente (clase) o parte derecha (*class yes*). El número que aparece entre corchetes es la precisión estimada de la regla, i.e. en media cuántas veces la regla es cierta (ver definición 3.3.4 para mayor información). En este caso, la regla se puede interpretar como: si el atributo *outlook* es igual a *rainy* y el atributo *windy* es igual a *false*, entonces la instancia será clasificada correctamente con el valor *yes* un 63% de las veces. Además, se puede notar que después de las reglas se establece la clase por defecto, que en el ejemplo actual es *yes*.

A continuación se presenta el *ranking* de las reglas en la tabla 3.3 y la descripción de esta tabla es la siguiente:

Rule Identificador de la regla.

Size Número de condiciones o tests que posee la regla.

Error Tasa de error estimado al utilizar la regla.

Used Número de casos del conjunto de entrenamiento cubiertos por la regla, es decir, aquellos que cumplen las condiciones de la parte izquierda de la regla de decisión. Dicho de otro modo, número de veces que la regla se usó para clasificar los casos del conjunto de entrenamiento.

Wrong Número de casos del conjunto de entrenamiento incorrectamente clasificados por la regla. Entre paréntesis se indica el porcentaje de errores.

Advantage Esta columna indica qué hubiera pasado si la regla se hubiera omitido. Está expresada en la forma $a (b|c)$, indicando que si la regla se hubiera omitido b casos clasificados correctamente por la regla serían incorrectamente clasificados, y c casos clasificados incorrectamente por esta regla serían clasificados correctamente si se omitiera esta regla; el parámetro a indica la ganancia de retener esta regla, i.e. $a = b - c$.

Class Clase de la regla, es decir, valor de la clase al que sería clasificado un caso cubierto por la regla.

Rule	Size	Error	Used	Wrong	Advantage	Class
1	2	37.0 %	3	0 (0.0 %)	3 (3 0)	no
5	2	50.0 %	2	0 (0.0 %)	2 (2 0)	no
3	1	29.3 %	4	0 (0.0 %)	0 (0 0)	yes
2	1	33.8 %	4	0 (0.0 %)	0 (0 0)	yes
4	2	37.0 %	1	0 (0.0 %)	0 (0 0)	yes

Tabla 3.3: *Ranking* de *DR*. Tabla extraída del apéndice A.3.3

Asimismo, se puede observar en la tabla 3.3 que en primer lugar aparecen las reglas de una clase, y en segundo lugar aparecen las reglas de la otra clase. Este resultado es producto del resultado de la primera fase (establecer un orden en los subconjuntos de reglas) del cuarto paso. Por ejemplo, las reglas 1 y 5, correspondientes a la clase *no*, aparecen en la parte superior de la tabla 3.3. Además, estas reglas están ordenadas de menor a mayor tasa de error de estimación. Igualmente, las siguientes tres reglas (reglas 3, 2 y 4) corresponden a la clase *yes*, y están ordenadas de menor a mayor tasa de error de estimación, siendo la regla 3 la que menor tasa de error presenta (29.3 %). Un aspecto importante a destacar en esta tabla es que todas las reglas clasifican correctamente todos los casos del conjunto de entrenamiento, por eso la columna *wrong* es 0.0 en todas las reglas. Este hecho, se debe principalmente a que el conjunto de datos es muy pequeño, y el modelo de reglas de decisión está fuertemente ajustado a los datos de entrenamiento.

(a)	(b)	<-classified as
5		(a): class no
	9	(b): class yes

Tabla 3.4: Matriz de confusión de *DR*. Tabla extraída del apéndice A.3.3

Por último, la parte final de la salida generada por el programa *c4.5rules* muestra la matriz de confusión (*confusion matrix*). Esta matriz es útil para detectar errores de clasificación en el conjunto de datos de entrenamiento. Su objetivo es mostrar la clasificación dada al conjunto de entrenamiento por el algoritmo frente al valor

real. Dicha matriz permite la detección de falsos positivos y falsos negativos. Por ejemplo, la tabla 3.4 muestra la matriz de confusión para el modelo de clasificación obtenido de los datos del ejemplo (datos de la tabla 3.1). En ésta se puede observar que 5 casos fueron clasificados por el modelo como *play = no*, y los cinco contenían la clase *play = no*. Para la clase *play = yes* ocurre exactamente lo mismo, 9 casos que contenían la clase *yes* de *play* fueron clasificados por el modelo con la clase *yes*. Por lo tanto, la matriz indica que el número de falsos positivos y de falsos negativos fue nulo. De nuevo, esta consecuencia es debida a la pequeña cantidad de datos del conjunto de entrenamiento.

Signos de disminución de la eficacia en sistemas o cursos *e-Learning*

Índice de contenidos

4.1. Introducción	75
4.2. Signos y diagnóstico	77
4.3. Tipos de signos	80
4.4. Ontología de signos	82
4.5. Conclusiones	85

Este capítulo presenta las definiciones sobre los términos síntoma, signo, diagnóstico y tratamiento adaptados desde los ámbitos de la Medicina y la Mecánica hacia los sistemas e-Learning. Asimismo, se plantea una clasificación de síntomas o signos en el contexto de estos sistemas.

4.1. Introducción

En esta tesis se van a extrapolar conceptos de otros ámbitos con el fin de establecer las bases conceptuales que se seguirán en esta investigación. Por este motivo, a continuación se exponen las definiciones de tales conceptos, así como la adaptación de éstas al contexto de los sistemas *e-Learning*.

El diccionario de la Real Academia Española (RAE) [RAE 2001] expone dos definiciones para el término **síntoma**: “fenómeno revelador de una enfermedad” y “señal, indicio de algo que está sucediendo o va a suceder”. Estas dos acepciones nos indican que en este término están relacionados los indicadores, señales y fenómenos con la presencia, por ejemplo, de enfermedades en el contexto de la Medicina, crisis en el contexto de la Economía, problemas mecánicos en el contexto de la Mecánica, y problemas de aprendizaje en el contexto de los sistemas educativos. Una cuestión que merece la pena señalar es que el concepto de síntoma se refiere únicamente a los datos subjetivos de la enfermedad, los que percibe el propio paciente, y se opone hasta cierto punto al de **signo**, basado en los datos que pueda percibir objetivamente el médico. Hay que indicar que las diferencias no son siempre claras y a menudo un síntoma puede ser observado también de forma objetiva, por ejemplo, la fiebre. Por

tanto, la diferencia entre síntoma y signo reside en la forma en que fue observado e interpretado el fenómeno.

Diagnóstico se define como “identificación de la naturaleza de una enfermedad mediante la observación de sus signos y síntomas característicos” [RAE 2001]. Por otro lado, la *Gran Enciclopedia Larousse* [lar 1977, tomo 3, p. 847] nos propone una definición más completa: “parte de la labor médica consistente en determinar la naturaleza de una enfermedad y clasificarla en un cuadro nosológico¹”. Por tanto, el diagnóstico es uno de los actos médicos más importantes, puesto que de él depende un **tratamiento** adecuado.

La figura 4.1 muestra el proceso general de diagnóstico de una enfermedad. En ésta, se puede observar a un paciente experimentando los síntomas: dolor de cabeza, coriza² y escalofríos. Estos síntomas son transmitidos al médico por el paciente, y seguidamente el médico realiza una exploración del paciente para detectar los signos (astenia³, coriza y fiebre). En este caso, se puede observar que algunos de los síntomas coinciden con los signos, pero otros no. El médico trata de identificar o de realizar un diagnóstico de la enfermedad a través de estos signos, pero el problema que se le presenta es que tres enfermedades contienen estos signos: virus de la gripe estacional, virus de la gripe porcina (*H1N1*), y virus de la gripe aviar (*H5N1*) [Min 2009]. Sin embargo, utilizando el contexto del entorno el médico puede averiguar con cual de los tres virus ha sido infectado el paciente. Por ejemplo, teniendo en cuenta que la temperatura ambiente no es demasiado baja, y que el paciente no tuvo un trato continuado con aves, parece que el diagnóstico más fiable es el de gripe porcina (también conocida como Gripe A). De acuerdo con estos datos, el médico le puede administrar un tratamiento al paciente. Esta figura evidencia que el contexto juega un papel muy importante en el diagnóstico en el ámbito de la medicina general. En el siguiente caso, veremos cómo el contexto no es tan relevante.

La figura 4.2 muestra el cuadro de mandos de un automóvil. En él se observa que algunos testigos están encendidos, lo cual puede ser signo de un problema. En concreto, dos de estos testigos corresponden a las pastillas de freno y sistema hidráulico de frenos, por lo tanto indican que puede existir un problema con el sistema de frenos. El diagnóstico en este caso, pudiera ser que el nivel del líquido de freno está por debajo del nivel mínimo establecido por el fabricante. Este ejemplo nos muestra que en este contexto tan controlado, como es una máquina, los síntomas y signos son equivalentes. Además, en este ámbito el contexto no aporta información relevante para realizar el diagnóstico. De igual modo, los síntomas o signos no sólo son indicadores de problemas, sino que también pueden servir para prevenirlos. Por ejemplo, el cuadro de mandos de esta figura incluye una serie de indicadores para alertar al conductor sobre el tiempo que falta para realizar el mantenimiento del vehículo en la revisión oficial del fabricante (indicador de intervalos de servicio en

¹La nosología es la parte de la medicina que tiene por objeto describir, diferenciar y clasificar las enfermedades.

²Inflamación de la mucosa nasal, generalmente acompañada de secreción mucosa.

³Respuesta insuficiente del organismo a un estímulo, dicho de otro modo, falta o pérdida de fuerza.

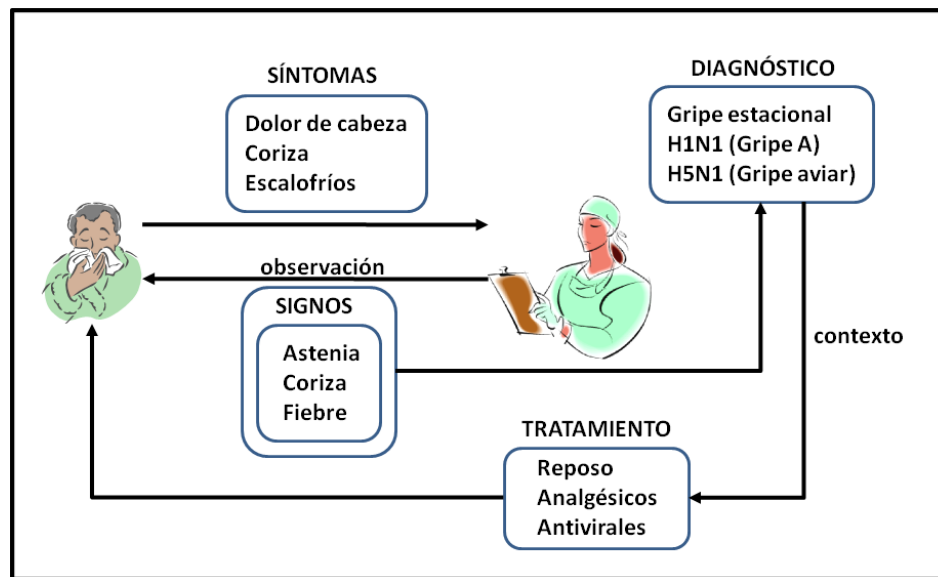


Figura 4.1: Proceso general de diagnóstico y tratamiento de una enfermedad

la figura 4.2). En el caso particular de este automóvil, cada diodo luminoso (*Light-Emitting Diode*, en adelante: *LED*) verde indica un período de 2 meses antes de la revisión, un *LED* amarillo indica que la revisión está próxima y el *LED* rojo indica la revisión inmediata del automóvil. Se puede observar en la figura que tres diodos luminosos verdes están encendidos (próximo mantenimiento del automóvil se debe realizar en 6 meses), lo que es un signo de prevención de problemas generales en el automóvil [BMW-AG 1987].

Los dos casos anteriores han mostrado que los signos pueden indicar problemas o prevenir la aparición de éstos. Además, se ha demostrado que las diferencias entre síntoma y signo dependen fundamentalmente del contexto, siendo esta diferencia poco significativa si se trata de contextos controlados, como el que existe en una máquina.

La siguiente sección expone los signos, síntomas, y diagnóstico desde el contexto de los sistemas *e-Learning*, así como el proceso de detección en el que se basa la presente tesis.

4.2. Signos y diagnóstico

Definición 4.2.1 Se define *signo* en el ámbito de los sistemas *e-Learning* como conjunto de factores que indican irregularidades en el proceso de aprendizaje de un estudiante o grupo de estudiantes.

Definición 4.2.2 Se define *diagnóstico* en el ámbito de los sistemas *e-Learning* como identificación y clasificación de las causas responsables de irregularidades en el proceso de aprendizaje de un estudiante o grupo de estudiantes.



Figura 4.2: Cuadro de mandos de un automóvil. Fuente: BMW E30 316i

En el caso del *análisis de logs* de los sistemas *e-Learning* los términos **signos** y **síntomas** son equivalentes, pues los últimos se obtienen de datos objetivos. Esto es así, ya que se analizan las interacciones reales de los estudiantes con el sistema, y éstos son datos objetivos, pues no son posibles diferentes interpretaciones sobre los mismos. Sin embargo, en el ámbito de estos sistemas esta equivalencia no es cierta, pues pueden almacenar datos subjetivos como son las encuestas. El *análisis de logs* se puede utilizar para descubrir los signos o síntomas indicadores de procesos irregulares de aprendizaje en los estudiantes. Ueno [Ueno 2006] define los procesos irregulares de aprendizaje en función del tiempo de respuesta de los estudiantes como aquel proceso que necesita más o menos tiempo del tiempo normal (tiempo medio), sin embargo, esta irregularidad también expresarse mediante un rendimiento académico bajo de los estudiantes o del sistema. Consecuentemente, se puede decir que existe un proceso irregular de aprendizaje cuando un estudiante o grupo de estudiantes obtienen un rendimiento académico más bajo que el rendimiento medio. Por ejemplo, si un grupo de estudiantes presentan dificultades en una determinada actividad de un curso, este grupo probablemente disminuirá su rendimiento respecto de otros grupos, y consecuentemente el promedio del rendimiento del conjunto total de estudiantes disminuirá respecto al esperado por el diseñador del curso. Dicho de otro modo, la eficacia del curso está relacionada con el rendimiento académico de los estudiantes. Por tanto, en este ejemplo el signo detectado sería “rendimiento bajo del grupo de estudiantes G en la actividad X”, y el diagnóstico podría ser “la actividad X no es adecuada para los estudiantes del grupo G”.

La figura 4.3 muestra la propuesta de evaluación de un sistema *e-Learning* que se realiza en el presente trabajo. Esta figura presenta un proceso muy similar al expuesto en la sección anterior. Como se puede observar, en este caso el paciente es el propio sistema como en el ejemplo del automóvil, y el médico o mecánico es la figura del profesor o diseñador del curso. Es claro que el sistema no le transmite los síntomas de la manera que lo realiza un paciente a un médico, sino que lo hace de forma similar a como lo hace un automóvil. El sistema almacena en los logs toda la información relativa a los estudiantes. De esta forma, el profesor mediante el análisis de logs es capaz de detectar los signos de baja eficacia del sistema. A partir de éstos, debería ser capaz de diagnosticar el problema existente, y finalmente aplicar un tratamiento. Hay que destacar que, debido a la ingente cantidad de datos que almacena un sistema de este tipo, es conveniente una metodología que permita extraer los signos al profesor, siendo éste el propósito principal de esta tesis. Además, es útil disponer de una herramienta automática o semi-automática que sugiera diagnósticos al diseñador y sus posibles tratamientos. Sin embargo, en esta tesis no se diseñan herramientas para proponer diagnósticos, aunque sí se dan indicaciones de cómo podría realizarse el proceso de diagnóstico. Este estudio está muy enfocado a detectar signos que provocan baja eficacia del sistema o curso, ya que como se expuso en el comienzo de esta tesis el proceso de detección de signos es arduo y complejo para los profesores, y es imprescindible para un correcto diagnóstico. La asistencia en este proceso es necesaria para la mejora y aprovechamiento de los sistemas *e-Learning*.

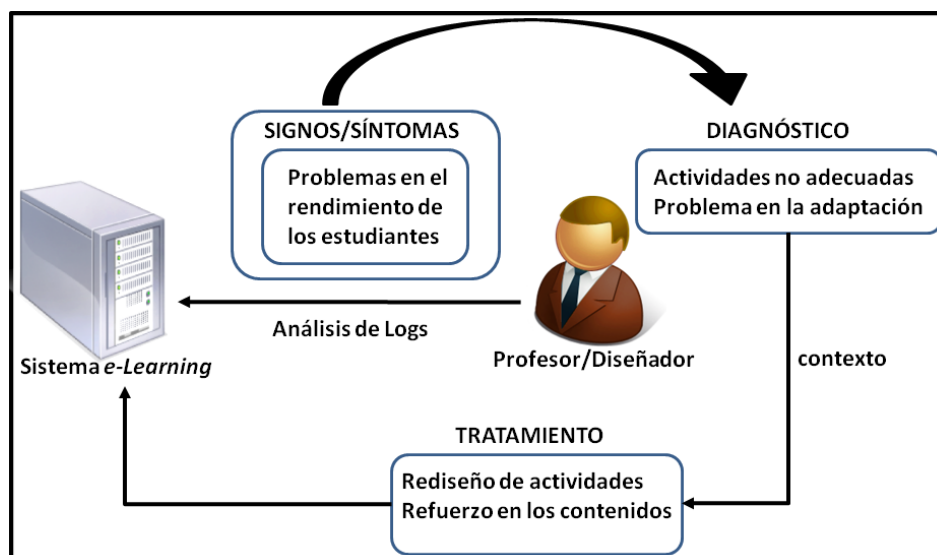


Figura 4.3: Evaluación de un sistema o curso *e-Learning*

4.3. Tipos de signos

En el apartado anterior se ha mostrado que los signos o síntomas son indicadores de problemas en el proceso de aprendizaje o en el rendimiento del sistema *e-Learning*. Este trabajo expone una clasificación de los distintos tipos de signos que pueden ser indicadores de bajo rendimiento en estudiantes y en el sistema. Distintas variables pueden servir para distinguir los signos, en concreto, se han utilizado para esta clasificación tres: resultados de los estudiantes en las actividades, tiempo empleado en resolver las actividades y consulta de ayuda o del índice de actividades. Es importante señalar que, las respuestas introducidas por los estudiantes en las actividades prácticas generalmente son evaluadas de forma automática por el propio sistema, obteniendo cada estudiante una puntuación en cada ejercicio. Es tarea del profesor o diseñador del curso decidir el rango de puntuaciones para considerar que el estudiante no resolvió la actividad de manera correcta, i.e., un fallo. A continuación se expone la clasificación dividida en tres criterios: **frecuencia**, **porcentaje** y **tiempo empleado**.

- **Frecuencia** Distintos criterios se pueden aplicar a la frecuencia, pero este trabajo considera que los más relevantes son dos: frecuencia en el uso de la ayuda y frecuencia entre fallos.
 - Frecuencia de uso de la ayuda o consulta del índice del curso: muchos sistemas almacenan datos sobre el número de veces que cada estudiante recurrió a la ayuda o a la consulta del índice; sin embargo, otros no lo hacen. Si esta información está disponible, un excesivo uso de ayuda en una determinada actividad puede ser un signo que revela dificultades en el proceso de aprendizaje. Tal y como apuntan Baker et al. [Baker 2004a], los estudiantes que utilizan frecuentemente la ayuda suelen obtener un bajo rendimiento en las actividades. Este perfil de estudiantes, como se vio en el capítulo 2, es característico por “jugar” con el sistema, y se le denomina *GAMED-HURT*.
 - Frecuencia entre fallos: la mayoría de los sistemas *e-Learning* almacenan la puntuación obtenida de cada ejercicio resuelto. Según lo expuesto anteriormente, un fallo está determinado por la puntuación obtenida en una actividad práctica. Así, una sucesión de fallos es un signo de que el estudiante tuvo dificultades en una serie de actividades prácticas, y consecuentemente obtuvo un bajo rendimiento.
- **Porcentaje** Este criterio es muy similar al anterior, pero se puede aplicar a distintas variables relacionadas con las actividades y puntuaciones obtenidas en éstas.
 - Porcentaje de actividad completada: en muchas ocasiones una actividad está formada por subactividades, en este sentido es útil analizar si los estudiantes completaron las actividades, o si hubo algunas que

no fueron completadas. Por ejemplo, si hubo un grupo de estudiantes que no completaron una determinada actividad, éste es un signo de un problema en el proceso de aprendizaje.

- **Porcentaje de fallos:** de la misma forma que la frecuencia entre fallos, es útil conocer el porcentaje de fallos en las actividades, en el curso en general, o para un conjunto de actividades. Éstos pueden ser signos de bajo rendimiento en el curso. Además, también es útil saber el porcentaje de fallos que tuvo un estudiante o un grupo de ellos, pues puede ser un signo de bajo rendimiento, o bien puede indicar falta de interés en el curso. Por ejemplo, pueden ser signos de bajo rendimiento:
 - Un grupo de estudiantes presenta un porcentaje de fallos muy superior en una determinada actividad que otros grupos.
 - Un grupo de estudiantes presenta un porcentaje de fallos alto en la mayoría de actividades.
 - Una actividad presenta un porcentaje de fallos alto para la mayoría de estudiantes.
 - Una actividad o grupo de actividades presenta un porcentaje de fallos alto para un grupo de estudiantes.
- **Tiempo empleado:** el tiempo puede ser una variable importante, siempre y cuando su medición sea consistente. Muchos sistemas, tal y como indican Srivastava et. al [Srivastava 2000], utilizan un tiempo máximo de expiración de sesión, generalmente se suele fijar en 30 minutos. Aceptando la hipótesis de que los estudiantes utilizan los sistemas *e-Learning* para aprender o reforzar sus conocimientos, el tiempo puede indicar procesos irregulares de aprendizaje.
 - Tiempo en completar una actividad: muchos sistemas registran el tiempo al iniciar una actividad y al terminarla, incluso pueden almacenar datos relacionados con la sesión. Si estos datos son consistentes, un ejemplo de signo puede ser un grupo de estudiantes que necesita un tiempo mayor que el resto para realizar una determinada actividad o un grupo de actividades. En este sentido, Ueno [Ueno 2006] realiza un amplio estudio en el que relaciona los procesos de aprendizaje irregulares con el tiempo empleado por los estudiantes.
 - Tiempo empleado en actividades con fallos: examinar si hay correlación entre el tiempo necesario para realizar una actividad y si ésta fue resuelta de forma correcta, puede ser útil para detectar procesos irregulares de aprendizaje.
 - Tiempo empleado en consulta de ayuda o índice de actividades: el tiempo que un estudiante dedica en consultar la ayuda o revisar el índice de actividades no es fácil de obtener. Por un lado, muchos sistemas guían al estudiante en el recorrido del curso, imposibilitando que éste pueda elegir una determinada actividad, otros sistemas no

registran si el estudiante consultó la ayuda. Por otro lado, existen sistemas que no permiten acceder libremente al estudiante a cualquier contenido del curso, a no ser que el estudiante haya demostrado tener ciertas habilidades adquiridas o ciertos conocimientos necesarios. Sin embargo, un tiempo excesivo de consulta de la ayuda por un grupo de estudiantes puede ser un síntoma de que este grupo tiene un problema, bien ha perdido el interés en el curso, o bien está desorientado.

4.4. Ontología de signos

Esta sección presenta una ontología inicial de síntomas o signos, desarrollada por el autor bajo la supervisión del *Dr. Serge Garlatti*. La figura 4.4 (página 84) muestra el diagrama de clases de la ontología (los ficheros *RDFS* y *RDF*⁴ que definen la estructura e instancias de la ontología se pueden ver en el apéndice). Esta ontología está formada por cinco clases: *Activity*, *Dimension*, *Property*, *Concept*, *Criterion* y *Symptom*.

- *Activity*: contiene el identificador y el nombre de la actividad de un curso.
- *Dimension*: contiene el nombre e identificador de una dimensión del perfil del estudiante. Estas dimensiones pueden ser relativas a estilos de aprendizaje (*dimension_1*), conocimiento previo (*dimension_2*), edad (*dimension_3*), etc.
- *Property*: hace referencia a la propiedad que contiene un síntoma. Puede ser de tres tipos: de igualdad, de no igualdad (menor que, mayor que) e indeterminada.
- *Concept*: hace referencia al concepto involucrado en un criterio. Por ejemplo, son instancias de esta clase: “todos”, “*dimension_1* igual a 2”, “*dimension_2* igual a 3”.
- *Criterion*: contiene el criterio o los sujetos a los que afecta un síntoma. Puede ser un criterio complejo, que está formado por uno o varios conceptos y un identificador (e.g. “fallos en las actividades 2 y 3”, “*dimension_1* igual a 2” y “*dimension_3* igual a 3”); o simple (e.g. “porcentaje de errores”, “porcentaje de actividades completadas”).
- *Symptom*: representa el signo o síntoma. Está formado por:

$$\langle \text{sujeito} \rangle \text{ tiene } \langle \text{criterio} \rangle \langle \text{propiedad} \rangle \langle \text{valor} \rangle \text{ en } \langle \text{complemento} \rangle \quad (4.1)$$

⁴ *Resource Description Framework (RDF)* es un estándar (*W3C Recommendation*) desarrollado por el *World Wide Web Consortium (W3C)* para representación de ontologías [W3C 2004b]. *RDF Schema (RDFS)* describe el vocabulario para construir una ontología bajo el estándar *RDF* [W3C 2004a].

- *< sujeto >*: contiene el perfil que experimenta el síntoma. Es una instancia de la clase *Criterion*.
- *< criterio >*: contiene el criterio que es evaluado por el síntoma. Es una instancia de la clase *Criterion*.
- *< propiedad >*: contiene la propiedad por la que es evaluado el criterio. Es una instancia de la clase *Property*.
- *< valor >*: contiene el número evaluado por la propiedad.
- *< complemento >*: representa la actividad o conjunto de actividades que cumplen el criterio bajo la propiedad y el valor. Son instancias de la clase *Activity*.

Los siguientes ejemplos de síntomas se pueden construir con esta ontología:

- “La mayoría de los estudiantes tienen una frecuencia de uso de ayuda mayor que 75”
 - *< sujeto >*: Todos (indica la mayoría y todos)
 - *< criterio >*: frecuencia de uso de ayuda
 - *< propiedad >*: mayor que
 - *< valor >*: 75
 - *< complemento >*: vacío
- “Los estudiantes de perfil con *dimension_1* igual a 2⁵ y *dimension_3* igual a 3⁶ tienen un porcentaje de fallos mayor que 80 % en las actividades 2 y 4”
 - *< sujeto >*: {*dimension_1* igual a 2, *dimension_3* igual a 3}
 - *< criterio >*: porcentaje de fallos
 - *< propiedad >*: mayor que
 - *< valor >*: 80
 - *< complemento >*: {actividad 2, actividad 4}
- “Los estudiantes de perfil con *dimension_3* igual a 1⁷ tienen un porcentaje de actividad completada menor que 20 % en la actividad 4”
 - *< sujeto >*: *dimension_3* igual a 1
 - *< criterio >*: porcentaje de actividad completada
 - *< propiedad >*: menor que
 - *< valor >*: 20
 - *< complemento >*: actividad 4

⁵Si esta variable representara estilos de aprendizaje, el número 2 indica que se selecciona la segunda dimensión del estilo correspondiente.

⁶Si esta variable almacenara la edad, dividida en: baja, media y alta. *Dimension_3* = 3 equivale a seleccionar la edad alta.

⁷Selección de edad = baja.

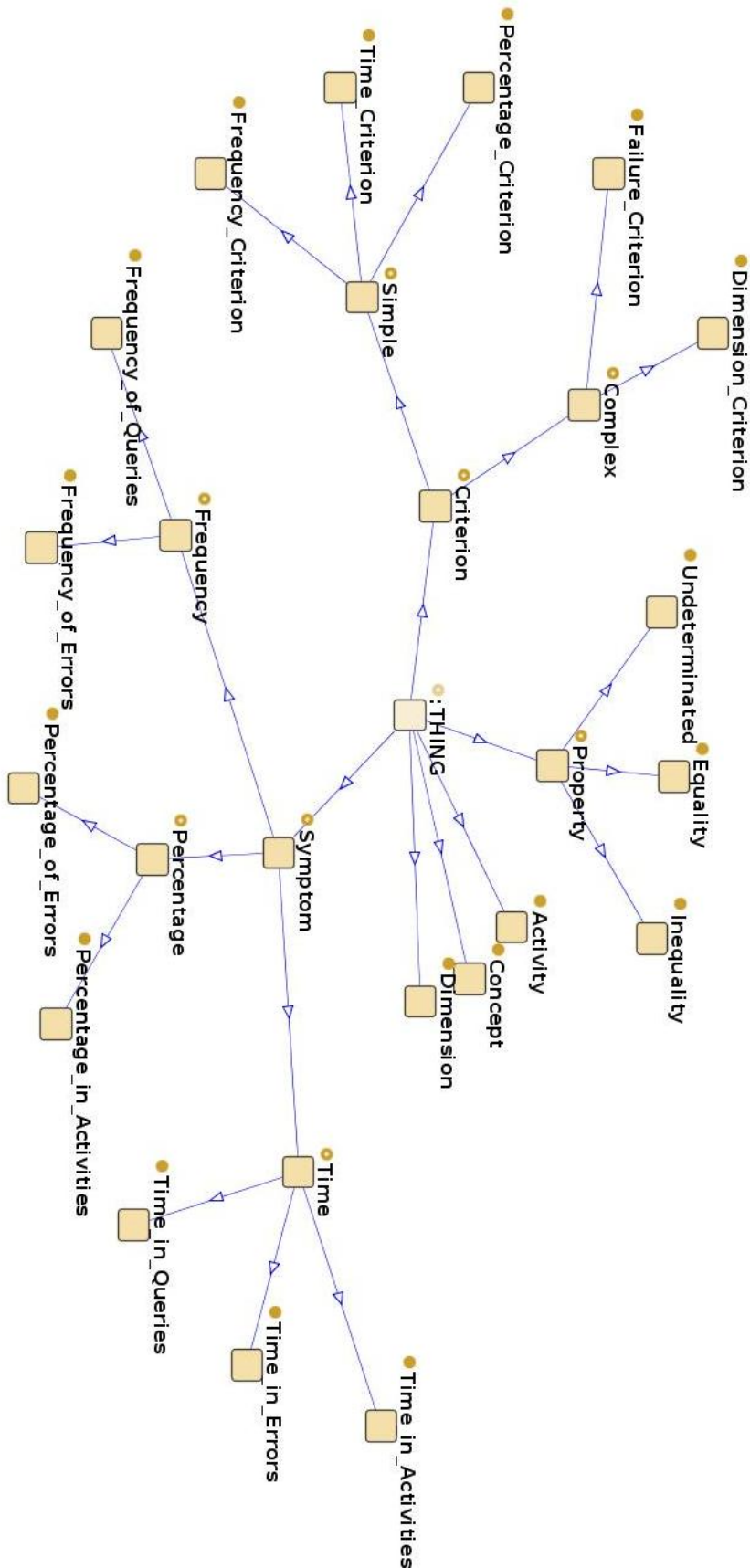


Figura 4.4: Ontología de síntomas para cursos *e-Learning*. Cada flecha indica la relación *es hijo de* o *es subclase de*. La ontología fue creada mediante el sistema *Protégé* v.3.4.1 [Sta 2009].

4.5. Conclusiones

Este capítulo ha presentado las diferencias entre signos, síntomas, diagnóstico y tratamiento, así como la forma en la que están relacionados en diferentes ámbitos. En este sentido, se ha visto que los signos y síntomas en entornos controlados, como son las máquinas son términos equivalentes. También, se ha demostrado que ambos términos son equivalentes en el campo de los sistemas *e-Learning*. Además, se ha expuesto una forma de evaluar un sistema o curso basado en el proceso general de diagnóstico y tratamiento de una enfermedad, basándose en el hecho de que si un sistema presenta signos de bajo rendimiento, dicho sistema “padece una enfermedad”. Finalmente, este capítulo exhibe una clasificación de signos que son indicadores de problemas en un curso *e-Learning* y una ontología inicial que sirve para construir signos o síntomas.

El siguiente capítulo expone la propuesta del método que es capaz de detectar *signos de baja eficacia* en los cursos *e-Learning*. Hay que indicar que en este capítulo no sólo se presenta el método sino que se exhibe el desarrollo seguido a lo largo de esta investigación para conseguir llegar a obtener la propuesta final. Como adelanto para el lector este método se apoya en las reglas de decisión y un modelo matemático para decidir qué información extraída de éstas es más importante para el profesor. Como se indicó en el inicio de este documento uno de los objetivos consiste en conseguir que dicha información sea entendible por cualquier profesor, por esta razón es necesaria la selección de la información. Además, en los siguientes capítulos se muestra la aplicación del método, la valoración de su eficacia (i.e. evaluación) y las herramientas desarrolladas en el transcurso de esta investigación.

Métodos para detectar signos de baja eficacia

Índice de contenidos

5.1. Introducción	87
5.2. Una primera aproximación: el método <i>Key-node</i>	88
5.2.1. Descripción del método	88
5.2.2. Experimentos	90
5.3. El método <i>GeSES</i>	97
5.3.1. Descripción del método	98
5.3.2. Exposición gráfica del método	113
5.3.3. Experimentos realizados	116
5.4. Conclusiones	142

*Este capítulo presenta un método capaz de detectar signos de baja eficacia; así como el proceso que ha llevado a su obtención. En primer lugar se presenta el método *Key-node* como una primera aproximación para detectar este tipo de signos, y su aplicación para analizar los logs del sistema TANGOW. La siguiente parte de este capítulo expone el método propuesto, cuyo nombre es *GeSES*, y su perfeccionamiento mediante su aplicación en distintos sistemas *e-Learning*.*

5.1. Introducción

Un **detector** es “cualquier aparato utilizado para detectar, descubrir o poner de manifiesto la presencia de un cuerpo o de un fenómeno oculto”. Siguiendo esta idea, el término **detectar** es “poner de manifiesto por un procedimiento físico algo que no puede observarse directamente” [lar 1977, tomo 3, p. 829]. En este sentido, este capítulo expone un método que permite detectar signos de situaciones problemáticas en los estudiantes y el curso. Dicho de otro modo, los sistemas *e-Learning* producen gran cantidad de logs y la información contenida en éstos está “oculta” para los profesores, pues su análisis es complejo y costoso. El método que se expone en este capítulo es capaz de descubrir esta “información oculta”, que en el contexto de esta tesis se identifica con los signos de baja eficacia, como por ejemplo los relacionados con el bajo rendimiento académico.

En las siguientes secciones se expone el proceso llevado a cabo para obtener un método que sea adecuado para detectar este tipo de signos. La siguiente sección expone los primeros pasos en el desarrollo de este método que condujeron al desarrollo del método *Key-node*. Este método constituye una primera aproximación para la detección de signos, realiza el análisis de logs a través de los árboles de decisión, y en concreto utiliza el algoritmo *C4.5* [Quinlan 1993].

5.2. Una primera aproximación: el método *Key-node*

Este método se diseñó con el objetivo de detectar problemas en las reglas de adaptación en los sistemas *AEH*. Se sustenta bajo la suposición que la existencia de ciertos signos en las interacciones de los estudiantes, y en particular los relacionados con el bajo rendimiento académico, puede ser una señal de problemas en las reglas de adaptación o en los contenidos del curso, y consecuentemente puede indicar que la adaptación no es adecuada y necesita ser mejorada. Por lo tanto, una manera de detectar problemas en la adaptación es buscar los anteriores signos.

Se asume que la siguiente estructura de logs: cada entrada (instancia) corresponde a la ejecución de una determinada actividad por un estudiante. De esta forma, los logs contienen las acciones realizadas por los estudiantes en las actividades.

5.2.1. Descripción del método

El método *Key-node* utiliza los árboles de decisión, en particular se emplea el algoritmo *C4.5*, para detectar estos signos. Se eligió esta técnica porque proporciona una descripción explícita de las propiedades de los datos. Recibe este nombre porque su objetivo radica en encontrar el nodo o nodos relevantes del árbol de decisión que son indicadores de los signos de problemas en la adaptación. El método *Key-node* se divide en tres fases: **fase de limpieza**, **fase de aplicación del algoritmo *C4.5*** y **fase de análisis**. Hay que indicar que estas tres fases coinciden con las etapas de aplicación de Minería *Web* expuestas por Srivastava et al. [Srivastava 2000]. De esta forma, la primera fase de este método se refiere a la etapa de preprocesamiento, la fase de aplicación del algoritmo *C4.5* corresponde con la etapa de descubrimiento de patrones y la tercera fase con el análisis de patrones.

- (1) **Fase de limpieza (*cleanning*)** Consiste en seleccionar los datos de interacción que hagan referencia a actividades prácticas. Se eligen este tipo de actividades porque son las que contienen algún tipo de valoración de los resultados de los estudiantes. Por tanto, cualquier tipo de actividad que puede ser evaluada también se puede añadir en esta fase. Es importante que estos datos contengan una variable, que puede ser sintética, y que indique éxito o fallo para cada actividad respecto a la característica de evaluación. Por ejemplo, son características de evaluación la puntuación que reciben los estudiantes, el tiempo empleado por éstos, si la actividad se completó o no, etc. Esta variable

recibe el nombre de *variable de evaluación*. Para el ejemplo en el que la característica de evaluación sea la puntuación, la *variable de evaluación* puede mostrar si un estudiante supera con éxito o no una actividad práctica. Dos valores son posibles para esta variable: *sí* (puntuación del estudiante es igual o superior al mínimo establecido por el profesor) o *no* (puntuación del estudiante es inferior al mínimo establecido por el profesor). El resultado de esta fase es una tabla con los perfiles de los estudiantes, sus datos de interacción y la *variable de evaluación*.

- (2) **Fase de aplicación del algoritmo C4.5** Consiste en la aplicación del algoritmo C4.5 con los siguientes parámetros:
 - Atributos: dimensiones del perfil del estudiante y el nombre de la actividad.
 - Variable de clasificación o clase: *variable de evaluación*.

El resultado de esta fase es un árbol de decisión que generalmente contiene un nodo para cada atributo¹ y sus hojas contienen los valores de la *variable de evaluación*. Es decir, el árbol está compuesto por los nodos de las dimensiones del perfil de los estudiantes, los nodos correspondientes a las actividades realizadas y las hojas indicando el éxito o fallo en estas actividades.

- (3) **Fase de análisis** En esta última fase se extrae la información relevante del árbol de decisión con el objetivo de encontrar los signos. El proceso consiste en:
 - Seleccionar las hojas en las que el valor de la *variable de evaluación* es *no*. Desde el punto de vista de detección de problemas, estas hojas son más importantes que las de la otra clase, pues indican fallos en un cierto número de estudiantes que realizaron las actividades.
 - Analizar cada camino desde las hojas seleccionadas en el punto anterior hasta la raíz del árbol. Para cada camino se necesita realizar dos pasos:
 - Encontrar en el camino el nodo correspondiente a la actividad.
 - Encontrar en el camino los valores del perfil del estudiante.

El resultado de esta fase es una lista de pares “{actividad, perfil}”. Cada uno de estos pares indica que un determinado perfil tuvo problemas al resolver una determinada actividad. Estos pares son potenciales signos de una inadecuada adaptación en el curso.

Con el propósito de mejorar la comprensión del método por parte del lector se presenta este método de forma gráfica en la figura 5.1 (página 90). Esta figura muestra las tres fases del método, señaladas con los números (1), (2) y (3), y en

¹Pueden existir atributos que no estén representados en el árbol, pues los datos asociados a éstos pueden no tener suficiente variabilidad para conseguir formar un nodo en el árbol.

color negro; en color azul se muestran los resultados y datos de entrada para cada fase. En primer lugar, se realiza la fase de limpieza sobre los logs, en la que se eliminan los registros no relevantes para el análisis, y se seleccionan los relevantes. Además, en esta fase se codifica la *variable de evaluación*. El resultado de esta primera fase es un conjunto de registros que forman los *datos de análisis*. Estos datos son analizados aplicando el algoritmo *C4.5*, obteniéndose el árbol de decisión resultante. En este árbol se puede observar que las hojas han sido coloreadas en verde y rojo. Las hojas verdes corresponden al valor *sí* de la *variable de evaluación* y las rojas corresponden al valor *no*. La tercera fase, fase de análisis, consiste en analizar cada uno de los recorridos o caminos del árbol que terminen en una hoja roja. De esta forma, se extrae la información relativa a los perfiles y actividad, contenidos dentro del camino. En la figura, se puede observar que las aristas *E*, *D* y *A* corresponden a distintas dimensiones de un perfil de estudiante, mientras que la arista *A1* indica la actividad que se realizó. El resultado de esta fase son los signos detectados, i.e., un conjunto de perfiles y actividades.

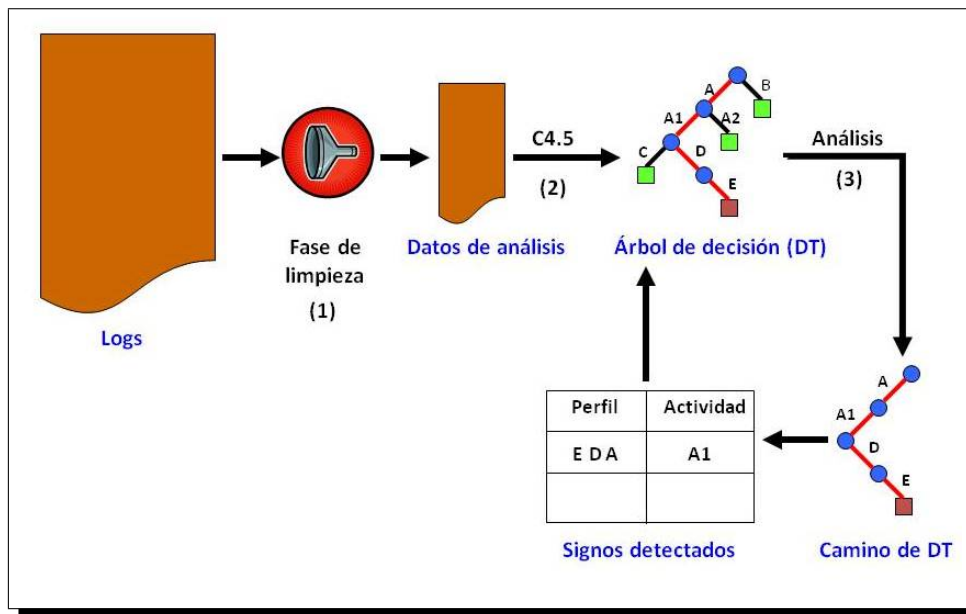


Figura 5.1: Detección de signos mediante el método *Key-node*

5.2.2. Experimentos

Con el objetivo de probar la eficacia de este método se realizaron dos experimentos. Para ello se utilizaron dos herramientas: *SimuLog* y *Weka* (ver capítulo 3, sección 3.4, página 64). *SimuLog* fue diseñada por el autor para generar interacciones sintéticas de los estudiantes de acuerdo a una serie de parámetros indicados por el diseñador (*SimuLog* se explica con más detalle en el capítulo 7). Hay que indicar que, la característica que hace útil a *SimuLog* en estos experimentos es su capacidad

para incluir fallos en las actividades realizadas por los estudiantes simulados. De esta forma, es posible indicar que *SimuLog* genere un determinado tipo de fallo en una actividad o grupo de actividades cometido por un determinado perfil de estudiantes, y en una determinada proporción.

Las interacciones generadas corresponden a un curso sobre normas viales ofrecido por el sistema de cursos adaptativos *TANGOW* [Carro 1999]. Un registro de log en el sistema *TANGOW* está compuesto por los siguientes atributos: $\langle user-id, profile, activity, complete, grade, numVisit, action, activityType, syntheticTime \rangle$. La tabla 5.1 muestra el significado de cada uno de estos atributos, así como los valores posibles que pueden tener. Cada registro contiene la acción que realiza un estudiante en un instante de tiempo, siendo las acciones posibles: *START-SESSION*, *FIRSTVISIT*, *REVISIT*, *LEAVE-COMPOSITE* y *LEAVE-ATOMIC*, tal y como se indica en la tabla 5.1. El atributo *profile* está compuesto por las dimensiones: idioma del curso, conocimiento previo y edad. Por ejemplo, el perfil {"español"; "novato"; "joven"} indica que un estudiante de este perfil ha de seguir la versión del curso en el idioma español, debe ser novato y es joven.

Un ejemplo de los datos almacenados en los logs se presenta a continuación. En éste se muestran dos registros de log del estudiante e78, en los cuales dicho estudiante realiza dos acciones: visita por primera vez la actividad y seguidamente la abandona.

- $\langle e78, joven, español, novato, S_Ag_Exer, 0,0, 0,0, 1, FIRSTVISIT, P \rangle$
- $\langle e78, joven, español, novato, S_Ag_Exer, 0,0, 0,0, 1, LEAVE-ATOMIC, P \rangle$

Los dos registros contienen la información del perfil del estudiante, es decir, el perfil del estudiante es {"español"; "novato"; "joven"}. En el primer registro se indica que el estudiante realizó por vez primera (*FIRSTVISIT*) la actividad *S_Ag_Exer* (ejercicios relacionados con las señales de un agente de tráfico). Ya que es el primer intento en esta actividad su puntuación es 0,0, y no fue completada (0,0 en la variable *complete*). El tipo "P" indica que la actividad realizada fue de tipo práctico. El segundo registro muestra que el estudiante abandonó (*LEAVE-ATOMIC*) esta actividad sin completarla (*complete* = 0,0) y obteniendo una puntuación insuficiente para superar la actividad (*grade* = 0,0). El tiempo generado de forma sintética no se muestra en estos registros por razones de legibilidad.

5.2.2.1. Primer experimento

Para este experimento se generaron las interacciones de 240 estudiantes simulados por *SimuLog*. La tabla 5.2 muestra los parámetros de simulación para este experimento. Así, se indica que se quiere simular perfiles de estudiantes en los que el 35 % sigan un curso en idioma español, y el resto repartido por igual entre los idiomas inglés y alemán. Estos perfiles deben tener igual proporción en el conocimiento previo, es decir, 50 % serán novatos y el otro 50 % avanzados. Además, se indica al simulador que se generen las mismas proporciones por edades, siendo estos valores:

Atributos	Descripción
<i>Activity</i>	Identificador de la actividad.
<i>Complete</i>	Representa el nivel de compleción de una actividad. Si la actividad es compuesta, para el cálculo de este nivel se tiene en cuenta el nivel de compleción de todas las subactividades de la que está compuesta. Es un parámetro numérico con rango de 0 a 1. El valor 0 indica que la actividad no fue completada, mientras que el valor 1 indica que la actividad fue completada.
Grade	Puntuación obtenida por el estudiante en la actividad. En el caso de actividades prácticas se calcula generalmente mediante una fórmula proporcionada por el profesor. En particular, para las actividades compuestas, este atributo se calcula como la media aritmética de las puntuaciones de las subactividades. Es un parámetro numérico, que puede tomar valores de 0 a 1. El valor 1 indica que la actividad fue terminada con éxito, y el valor 0 indica lo contrario.
<i>NumVisit</i>	Número de veces que el estudiante realizó la actividad.
<i>Action</i>	Contiene la acción ejecutada por el estudiante. Puede ser de los siguientes tipos: “ <i>START-SESSION</i> ”: comienzo del curso o sesión. “ <i>FIRSTVISIT</i> ”: primer intento de realización de la actividad. “ <i>REVISIT</i> ”: después del primer intento, siguientes intentos de realización de la actividad. “ <i>LEAVE-COMPOSITE</i> ”: el estudiante finaliza o abandona una actividad compuesta. “ <i>LEAVE-ATOMIC</i> ”: el estudiante finaliza o abandona una actividad atómica.
<i>ActivityType</i>	Tipo de actividad realizada: teórica (T), práctica (P), ejemplo (E).
<i>SyntheticTime</i>	Marca de tiempo generado por <i>SimuLog</i> cuando el estudiante realiza una acción.

Tabla 5.1: Descripción de los atributos de un registro de log en el sistema *TANGOW*

50 % para jóvenes y 50 % para mayores. Por ejemplo, el perfil {“español”; “novato”; “joven”} indica que un estudiante de este perfil ha de seguir la versión del curso en el idioma español, debe ser novato y es joven.

Además, se generaron los siguientes signos de bajo rendimiento académico:

- Perfil: *Idioma del curso*: español, *conocimiento previo*: novato y *edad*: joven.

Dimensión	Valores
Idioma del curso	Español (35%), Inglés (32,5%) y Alemán (32,5%)
Conocimiento previo	Novato (50%) y Avanzado (50%)
Edad	Joven (50%) y Mayor (50%)

Tabla 5.2: Perfiles de estudiantes simulados

- Actividad: *S_Ag_Exer* (ejercicios relacionados con las señales de un agente de tráfico).
- Tipo de signo: proporción de fallos.
- Proporción: 70%.

Este signo de baja eficacia representa que el 70% de los estudiantes con perfil {"español"; "novato"; "joven"} cometieron fallos en la actividad práctica de ejercicios relacionados con las señales de un agente de tráfico.

Fase de limpieza. Esta fase consistió en eliminar los registros que no son necesarios para la fase de aplicación del algoritmo *C4.5*. El proceso consistió en seleccionar los registros que contienen actividades prácticas (*activityType* = P) y que la acción realizada por el estudiante es "*LEAVE-ATOMIC*". Debido al funcionamiento del sistema *TANGOW*, sólo estos registros tienen puntuaciones fiables de las actividades, ya que *TANGOW* actualiza la puntuación de una actividad realizada por el estudiante solamente cuando se ha finalizado o se cambia a otra actividad. En este caso, el sistema genera un registro en los logs con: el identificador del estudiante, su perfil, actividad realizada, número de intentos, puntuación en la actividad, tipo de actividad = P, acción = *LEAVE-ATOMIC* e instante de tiempo. Estos registros son los que nos pueden indicar si los estudiantes tuvieron dificultades o no; por eso, todos los registros que cumplan alguna de las siguientes restricciones fueron eliminados:

- Acción distinta de *LEAVE-ATOMIC*.
- Tipo de actividad distinto de P.

Como resultado de esta operación se seleccionaron 960 registros. Después de esta fase de selección es necesario añadir la *variable de evaluación*; en este caso, como el objetivo es detectar fallos en las actividades se generó una variable adicional, *success*, que indica si el estudiante realizó la actividad con éxito o no. Por tanto, esta variable puede tener dos valores posibles "sí" y "no". El valor "sí" se obtiene cuando la variable *grade* sea mayor o igual que 0,5 y menor o igual que 1, en el resto de los casos el valor de esta variable es "no". Los registros anteriores con la *variable de evaluación* forman los *datos de análisis*.

Fase de aplicación del algoritmo *C4.5*. Esta fase consistió en aplicar el algoritmo *C4.5* a los datos de análisis, siendo los atributos las variables del perfil del estudiante y el nombre de la actividad, y la variable de clasificación la *variable de*

evaluación. La figura 5.2 muestra el árbol de decisión generado; se puede observar que el tamaño del árbol es 12, y está compuesto por 4 nodos y 8 hojas. Los nodos *idioma*, *conocimiento previo* y *edad* corresponden a dimensiones del perfil del estudiante, y el nodo *actividad* al nombre de la actividad realizada. Se puede observar que en la figura 5.2 las hojas con el valor *sí* están coloreadas en verde para indicar que la actividad se realizó con éxito, y las hojas rojas corresponden al valor *no*. En este ejemplo, sólo se obtiene una hoja *no*.

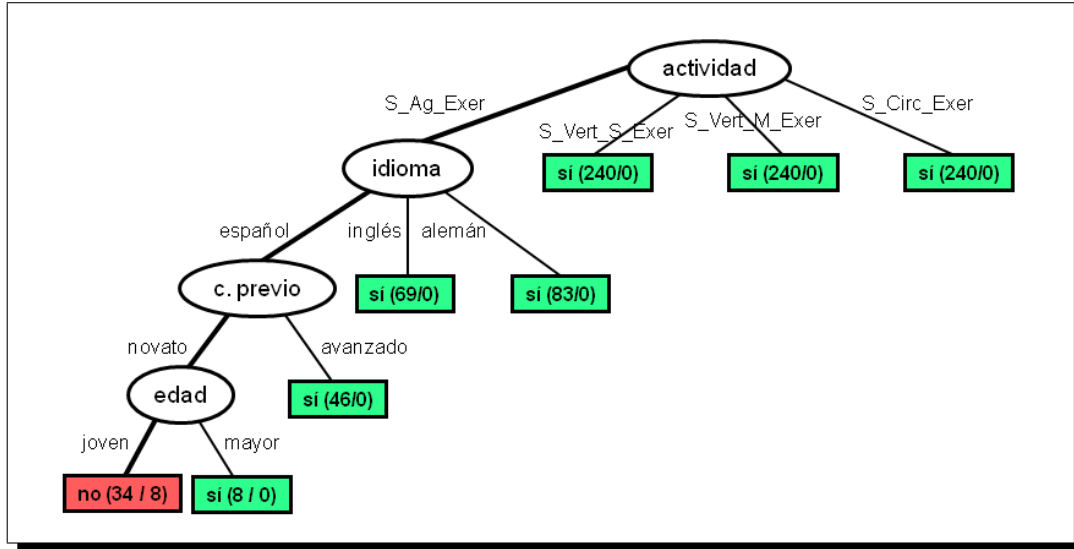


Figura 5.2: Árbol de decisión generado en el primer experimento. Se generó utilizando el clasificador *J48* de *Weka*. Nota: el árbol mostrado fue rediseñado por cuestiones de claridad, pero contiene los mismos elementos que el generado en *Weka*.

Fase de análisis. Consiste en extraer la información relevante del árbol de decisión con el objetivo de detectar signos de fallos. El primer paso en esta fase es seleccionar las hojas en las que el valor de la *variable de evaluación* es *no*. El árbol sólo presenta una hoja con el valor *no*. Esta hoja señala que el 77% ($26/34^2$) de los estudiantes jóvenes, novatos y que siguieron el curso en español, no realizaron la actividad *S_Ag_Exer* correctamente. El siguiente paso es analizar la rama desde el nodo raíz hasta la hoja. Este análisis extrae que la actividad correspondiente a la rama seleccionada es *S_Ag_Exer*, y que el perfil de los estudiantes es: estudiantes jóvenes, novatos, que siguieron el curso en español. Por tanto, el signo encontrado señala que la mayoría de estos estudiantes tuvieron problemas al realizar los ejercicios relacionados con las señales de un agente de tráfico.

²En la figura 5.2 el nodo *no* contiene la siguiente información: (34/8). El primer número indica el total de instancias cuyos valores coinciden con los de la rama desde el nodo *edad* hasta el nodo raíz. El segundo número indica las instancias que no fueron clasificadas de forma correcta. Por tanto, el número de instancias clasificadas correctamente fueron 26, i.e., $34 - 8$, y la proporción de instancias correctamente clasificadas es $26/34$.

Es importante señalar que en este experimento el *DT* tiene un alto porcentaje de instancias bien clasificadas. Esto se debe a la ausencia de aleatoriedad en la variable *grade*, que otorga puntuaciones máximas a los estudiantes que no encajan en el signo de bajo rendimiento generado de forma sintética. El siguiente experimento tiene en cuenta el factor de la aleatoriedad, y se genera más de un signo de bajo rendimiento académico.

5.2.2.2. Segundo experimento

Para este experimento se generaron las interacciones de 480 estudiantes simulados por *SimuLog* con dos signos de bajo rendimiento y añadiendo el factor de aleatoriedad en la variable *grade*. Por tanto, en este caso existen dos fuentes de ruido: el número de signos y el efecto aleatorio. La proporción de perfiles generada es la misma que en el experimento anterior (ver tabla 5.2, página 93). Se generaron dos signos de baja eficacia, relacionados con el rendimiento académico, definidos por:

- Signo 1: “60 % de estudiantes con perfil {“español”; “novato”; “joven”} tuvieron fallos en la actividad *S_Ag_Exer*”, que puede dividirse de la siguiente forma:
 - Perfil: *Idioma del curso*: español, *conocimiento previo*: novato y *edad*: joven.
 - Actividad: *S_Ag_Exer*.
 - Tipo de signo: proporción de fallos.
 - Proporción: 60 %.
- Signo 2: “60 % de estudiantes con perfil {“inglés”; “novato”; “joven”} tuvieron fallos en la actividad *S_Circ_Exer*”, que se puede dividir en:
 - Perfil: *Idioma del curso*: inglés, *conocimiento previo*: novato y *edad*: joven.
 - Actividad: *S_Circ_Exer* (ejercicios de señales de tráfico circulares).
 - Tipo de signo: proporción de fallos.
 - Proporción: 60 %.

Fase de limpieza. Esta fase se realizó de la misma forma que en el primer experimento. Como resultado se obtuvo un conjunto de datos con 1920 registros después de haber eliminado los registros con acción distinta de *LEAVE-ATOMIC* y tipo de actividad distinto de P. Después de esta fase de selección, del mismo modo que en el experimento anterior, se añadió la variable *success* (*variable de evaluación*). El conjunto anterior junto con la *variable de evaluación* formaron los *datos de análisis*.

Fase de aplicación del algoritmo C4.5. Esta fase consistió en aplicar el algoritmo *C4.5* a los datos de análisis, siendo los atributos las variables del perfil del estudiante y el nombre de la actividad, y la variable de clasificación la *variable de evaluación*. La figura 5.3 muestra el árbol de decisión generado. El tamaño de dicho árbol es 17, y está formado por 6 nodos y 11 hojas. Los nodos *idioma*, *conocimiento*

previo y *edad* corresponden a dimensiones del perfil del estudiante, y el nodo *actividad* al nombre de la actividad realizada. Examinando este árbol se aprecia que es más complejo que el de la figura 5.2 (página 94), ya que el tamaño del árbol es mayor y tiene dos nodos para el *idioma* y dos para el *conocimiento previo*. En este caso, la edad es el nodo raíz. Además, es importante notar que en la figura 5.3 las hojas con el valor *sí* están coloreadas en verde, y las de valor *no* en rojo.

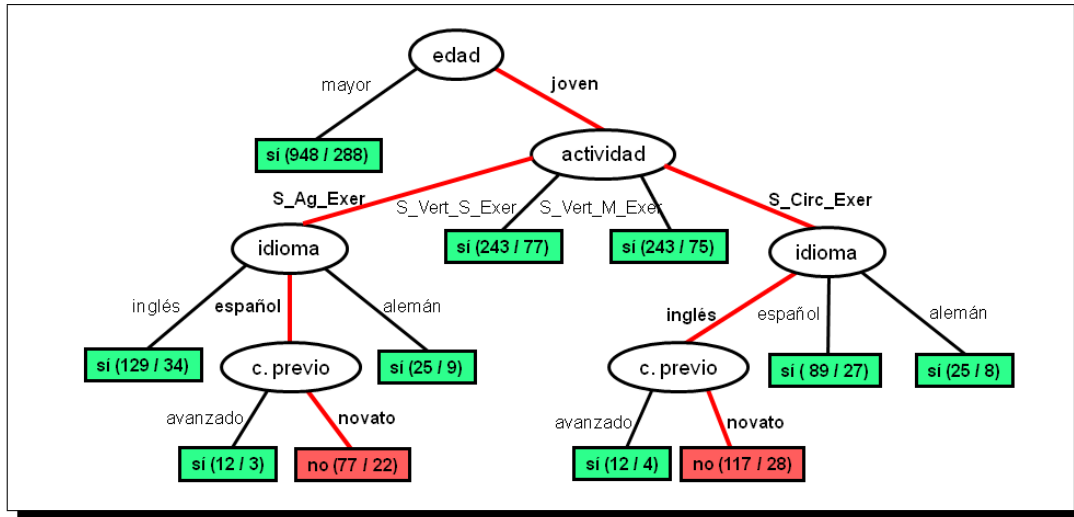


Figura 5.3: Árbol de decisión generado en el segundo experimento. Se generó utilizando el clasificador *J48* de *Weka*. Nota: el árbol mostrado fue rediseñado para facilitar su comprensión, pero contiene los mismos elementos que el generado en *Weka*.

Fase de análisis. Consiste en extraer la información relevante del *DT* con el objetivo de detectar signos. El primer paso en esta fase es seleccionar las hojas en las que el valor de la *variable de evaluación* es *no*. El árbol presenta dos hojas con el valor *no*, y adicionalmente se han resaltado en color rojo las ramas correspondientes a estas hojas. El siguiente paso es analizar las dos ramas desde el nodo raíz (*edad*) hasta la hoja.

- Análisis de la primera hoja: el análisis obtiene el siguiente perfil y actividad relacionados con el signo de bajo rendimiento académico:
 - Perfil: {"español"; "novato"; "joven"}
 - Actividad: *S_Ag_Exer*

El árbol de decisión muestra que más del 70% ($55/77^3$) de los estudian-

³En la figura 5.3 un nodo *no* seleccionado contiene los números (77/22). El primer número indica el total de instancias cuyos valores coinciden con los de la rama desde el nodo *conocimiento previo* hasta el nodo raíz. El segundo número indica las instancias que no fueron clasificadas de forma correcta. Por tanto, el número de instancias clasificadas correctamente fueron 55, i.e., $77 - 22$, y la proporción de instancias correctamente clasificadas es $55/77$.

tes jóvenes, novatos, que siguieron el curso en español fallaron la actividad S_Ag_Exer .

- Análisis de la segunda hoja: el análisis obtiene el siguiente perfil y actividad relacionados con el signo de bajo rendimiento académico:
 - Perfil: {"inglés"; "novato"; "joven"}
 - Actividad: S_Circ_Exer

El árbol de decisión muestra que más del 70 % (89/117) de los estudiantes jóvenes, novatos, que siguieron el curso en inglés tuvieron dificultades en la actividad S_Circ_Exer .

El método *Key-node* consigue detectar dos signos de baja eficacia correspondientes a la categoría de bajo rendimiento académico: "más del 70 % de estudiantes jóvenes, novatos, que siguieron el curso en español mostraron dificultades en los ejercicios relacionados con las señales de un agente de tráfico" y "más del 70 % de estudiantes jóvenes, novatos, que siguieron el curso en inglés mostraron dificultades en los ejercicios de señales de tráfico circulares".

Estos dos experimentos han demostrado que el método *Key-node* funciona razonablemente bien. En el primer experimento se generaron signos de forma artificial, y se comprobó que este método consiguió detectar los fallos incluidos en los logs. El segundo experimento incluyó dos fuentes de ruido en los logs, con el objetivo de simular logs más aproximados a la realidad. Este experimento demostró que el método es eficaz también bajo estas dos fuentes de ruido, pues se consiguieron detectar los dos signos insertados de forma artificial en los logs. No obstante, el método *Key-node* puede presentar problemas relacionados con la computación y complejidad si el tamaño del árbol es muy grande (e.g. superior a 200 nodos) y posee un gran número de hojas a analizar. Por otro lado, este método no proporciona ninguna clasificación por orden de relevancia de los signos encontrados, lo que facilitaría la labor del profesor en la mejora del curso. La siguiente sección expone el método que basándose en el método *Key-node* soluciona los dos problemas comentados anteriormente.

5.3. El método *GeSES*

El método *Key-node* puede no ser eficaz cuando el *DT* es grande. Los árboles de decisión grandes son difíciles de entender porque cada nodo tiene un contexto específico establecido por los resultados obtenidos en los tests de los nodos precedentes. Por ejemplo, el árbol de decisión de la figura 5.3 (página 96) muestra que si el *conocimiento previo* es *avanzado* la variable *success* toma el valor *sí*. Esto no necesariamente sugiere que cualquier estudiante con *conocimiento previo* avanzado vaya a tener éxito, sino que debe ser entendido en conjunto con los resultados de los anteriores tests de los nodos precedentes. En este caso, los resultados de los tests nos indican que el *idioma* es español, la *actividad* es S_Ag_Exer y la *edad* es joven. Estos resultados también forman parte del contexto del nodo *conocimiento previo*.

Debido a esta característica, cada test de cada nodo tiene un único contexto que es crucial conocer para el entendimiento del *DT*. Por lo tanto, árboles de decisión frondosos pueden ser difíciles de entender, pues el contexto va cambiando continuamente de un nodo a otro. Otro de los problemas que presentan los árboles de decisión es que pueden aparecer partes del árbol (subárboles) replicadas en el mismo árbol, esto produce que la interpretación del árbol sea también compleja. Como forma de solucionar estos problemas se presenta el **método *GeSES***.

5.3.1. Descripción del método

El método *GeSES* se basa en el método *Key-node*, los pasos propuestos son muy similares a los seguidos en el método anterior. Sin embargo, la técnica utilizada es diferente, así como los resultados mostrados por el método. La tercera fase del método anterior analizaba cada rama o camino desde la hoja seleccionada hasta el nodo raíz. En este sentido, este análisis es muy similar al paso 1 del procedimiento para transformar un árbol de decisión en un conjunto de reglas de decisión (ver capítulo 3, apartado 3.4.2.2, página 69). Este paso consiste en transformar cada una de las ramas en reglas, que es muy similar a la tercera fase del método *Key-node*, si bien, en este último sólo se transforman las ramas seleccionadas.

5.3.1.1. Propuesta inicial del método *GeSES*

El método *GeSES* se sirve de una técnica que mejora los resultados de un *DT*, las *reglas de decisión*, y en particular las del algoritmo *C4.5*. La propuesta inicial de este método contiene las siguientes etapas:

- (1) **Fase de limpieza.** Consiste en seleccionar los datos de interacción que hagan referencia a actividades prácticas, ya que generalmente en los sistemas *e-Learning* este tipo de actividades contienen información sobre puntuaciones en los ejercicios, número de intentos, tiempo empleado, etc. Es importante que estos datos contengan una variable, que puede ser generada a partir de otras variables, y que indique si se cumple o no la característica de evaluación que se quiere analizar. Por ejemplo, son características de evaluación la puntuación que reciben los estudiantes, el tiempo empleado por éstos, si la actividad se completó o no, etc. Esta variable recibe el nombre de *variable de evaluación*. En el caso de que la característica de evaluación sea la puntuación, la *variable de evaluación* puede mostrar si un estudiante supera con éxito o no una actividad práctica. Es el profesor quién puede fijar los valores posibles para esta variable. En el caso de las puntuaciones, se propone que esta variable tome dos valores posibles: *sí* (puntuación del estudiante es igual o superior al mínimo establecido por el profesor) o *no* (puntuación del estudiante es inferior al mínimo establecido por el profesor). El resultado de esta fase es una tabla con los perfiles de los estudiantes, sus datos de interacción y la *variable de evaluación*. Esta tabla recibe el nombre de *datos de análisis*.

- (2) **Generación de las reglas de decisión.** Consiste en la aplicación del algoritmo *C4.5* a los *datos de análisis* para generar las *DR*, siguiendo los pasos de transformación de un *DT* a reglas de decisión descritos en el capítulo 3 (página 69). Tal y como se explica en el capítulo sobre Minería de Datos, es necesario construir en primer lugar el árbol de decisión. Este árbol debe ser construido con los siguientes parámetros:
 - Atributos: dimensiones del perfil del estudiante, nombre de la actividad y contexto de la actividad (si es una actividad compuesta o no, el padre de la actividad, conceptos relacionados con la actividad, etc).
 - Variable de clasificación o clase: *variable de evaluación*.

Una vez construido el *DT* se pueden generar las reglas de decisión a partir de éste. El resultado de esta fase es un conjunto de *DR*, cada una con su correspondiente precisión, y un *ranking* de reglas con los siguientes conceptos: *size*, *error*, *used*, *wrong*, *advantage* y *class*⁴. Hay que indicar que para evitar la sobrecarga de información es necesario seleccionar las reglas más relevantes. Esta selección se realiza en las siguientes fases mediante un proceso de asignación de pesos a partir de un punto de corte (frontera).

- (3) **Seleccionar las reglas relevantes para la característica de evaluación.** Consiste en seleccionar del *ranking* de reglas aquellas en las que la *variable de evaluación* cumpla la característica que se quiere evaluar. Por ejemplo, en el caso de que la característica de evaluación esté relacionada con las puntuaciones en las actividades prácticas, si el objetivo es detectar problemas en el rendimiento académico de los estudiantes, en esta fase se seleccionarían las reglas cuya clase (columna *class*) indique un fallo en la actividad.
- (4) **Establecer la frontera de selección.** Esta frontera o punto a partir del cual se realiza la selección de las reglas, se establece en función de los conceptos del *ranking* de reglas. Su función es hacer posible la selección de las reglas más relevantes respecto a los conceptos del *ranking*. Se proponen las siguientes fronteras:
 - Frontera 1: *diez por ciento* respecto a los conceptos: *used*, *size* y *error*.

$$filter = \frac{\sum_{i=primera-regla}^{ultima-regla} parametro_i}{10} \quad (5.1)$$

- Frontera 2: *tercer cuartil*, Q_3 , respecto de los conceptos: *used*, *size* y *error*. Se calcula de la siguiente forma [OpenOffice.org 2009]⁵:

⁴La descripción de estos conceptos se puede encontrar en la página 72, capítulo 3.

⁵Para conocer información más detallada sobre el cálculo de los percentiles, cuartiles o deciles véase [García Ferrando 1995, p. 60-62].

1. Siendo N el número total de elementos en los datos, se calcula el rango, r , del cuartil mediante:

$$r = \frac{3}{4} \cdot (N - 1) \quad (5.2)$$

2. Comprobar si $|r|$ es entero. Siendo $f(x)$ la función de la distribución de los datos ordenados de menor a mayor respecto al *parámetro* seleccionado, y siendo x un valor entre 0 y $N-1$ (i.e. $f(5)$ devuelve el dato que ocupe la posición 5), se calcula el tercer cuartil de la siguiente manera:

$$Q_3 = f(r) \quad (5.3)$$

3. En caso de que $|r|$ no sea entero, el tercer cuartil se calcula mediante una ponderación entre el valor correspondiente al rango inmediatamente anterior, $f(r_{ant})$, y el correspondiente al rango del cuartil, $f(r)$. La siguiente fórmula realiza el cálculo del *tercer cuartil* o *percentil 75*:

$$Q_3 = f(r_{ant}) + (f(r) - f(r_{ant})) \cdot \frac{1}{4} \quad (5.4)$$

- (5) **Seleccionar las mejores reglas.** Esta fase está dividida en dos etapas: asignación de pesos a las observaciones (valores de los conceptos de la regla = observaciones) y selección de las reglas.

- (5.1) Asignación de pesos a las reglas. Se asigna peso máximo (número entre 0 y 1) si la observación supera la *frontera de selección* establecida para cada uno de los conceptos (*used*, *size* y *error*). En caso contrario, se asigna a esta observación peso 0. El *peso total* de cada regla se obtiene sumando los pesos de los conceptos, ponderados de acuerdo a la importancia de cada uno, siendo esta ponderación un número entre 0 y 1.

$$W_r = \sum_{j \in \{1,2\}} w_{used}^{Fj} * s_{used} + w_{size}^{Fj} * s_{size} + w_{error}^{Fj} * s_{error} \quad (5.5)$$

Donde, W_r es el peso total de la regla r , w_c^{Fj} es el peso que otorga la frontera j respecto al concepto c ($c \in \{used, size, error\}$), su valor está entre 0 y el peso máximo ($w_c^{max} \in (0, 1]$), y s_c es la importancia que tiene el concepto c ($s_c \in [0, 1]$).

- (5.2) Selección de las reglas. Se ordenan las reglas respecto al *peso total* y se seleccionan aquellas que mayores pesos han obtenido.

5.3.1.2. Análisis de la propuesta inicial del método

Los experimentos realizados (ver sección 5.3.3) mostraron que el método de selección, en concreto la fase 5, presenta algunos problemas si las observaciones⁶

⁶En el contexto en que nos encontramos una observación es el valor de un concepto respecto de la regla de decisión. Por ejemplo, los valores 17,6, 18,2 y 20,9 son observaciones del concepto error

estaban cercanas a la *frontera de selección*, ya que el peso asignado a éstas fue el mismo que las que estuvieron más alejadas de la *frontera de selección*. La tabla 5.3 muestra las *DR* seleccionadas del *ranking* que proporciona el programa *C4.5rules* (paso 3 del método *GeSES*) en el experimento mostrado en la sección 5.3.3.1. Teniendo en cuenta que se ha fijado el peso máximo en el valor 1 ($w_c^{max} = 1, \forall c \in \{used, size, error\}$), y que cada uno de los conceptos tiene la misma importancia, i.e, $s_{used} = s_{size} = s_{error} = 1$, esta tabla muestra los pesos obtenidos por cada concepto respecto a la frontera 1 (*F1*) y la 2 (*F2*), y el peso total (W_r). En esta tabla se puede observar que las tres reglas de decisión que mayor peso total obtienen son *DR23*, *DR19* y *DR12*. No obstante, las dos últimas tienen el mismo peso total (peso total = 4), con lo que es difícil discernir cuál es la mejor de las dos. Además, hay observaciones muy cercanas a las *fronteras de selección* que obtienen el mismo peso que las más lejanas, es decir, aquellas que están situadas por debajo de la frontera, aunque estén cercanas a ésta, obtienen peso con valor 0. Por ejemplo, la regla *DR6* tiene una cobertura de 2508 instancias, que está por debajo de *F2*, 2533 instancias, pero muy cercana a dicha frontera. El método de selección actual le otorga peso 0, pues este valor es inferior al de *F2*.

Rule	Size	Error (%)	Used	w_{used}^{F1}	w_{used}^{F2}	w_{size}^{F1}	w_{size}^{F2}	w_{error}^{F1}	w_{error}^{F2}	W_r
DR19	2	17,6	71	0	0	1	1	1	1	4
DR23	2	18,2	2608	0	1	1	1	1	1	5
DR8	2	20,9	242	0	0	1	1	1	0	3
DR33	1	29,8	1771	0	0	0	0	0	0	0
DR7	2	32,1	344	0	0	1	1	0	0	2
DR6	2	32,4	2508	0	0	1	1	0	0	2
DR12	2	34,2	20419	1	1	1	1	0	0	4
DR14	2	39,7	253	0	0	1	1	0	0	2
F1	2	22 ^a	2822							
F2	2	20,23 ^b	2533							
Suma	15	224,9	28216							
σ	0,33	7,66	6460,38							

^a En el caso del concepto error, obtienen peso máximo las *DR* que tengan un error inferior o igual a *F1*.

^b En el caso del concepto error, la variable *F2* corresponde al primer cuartil.

Tabla 5.3: Pesos otorgados por las fronteras *F1* y *F2*

El gráfico de la figura 5.4 muestra los pesos otorgados por la frontera *F2* a las observaciones para el concepto *used*. En esta figura se puede observar que la función que asigna los pesos (señalada en la figura con una línea de puntos) es dicotómica entre 0 y 1. Esto produce un efecto no deseado, ya que existe una observación (2508)

en la tabla 5.3.

muy cerca de la frontera que recibe peso 0. Por tanto, parece lógico pensar que esta valoración dicotómica de los pesos es incompleta, y es más adecuada una valoración gradual. De esta forma, observaciones cercanas a la frontera obtendrían un peso proporcional a su distancia con la frontera.

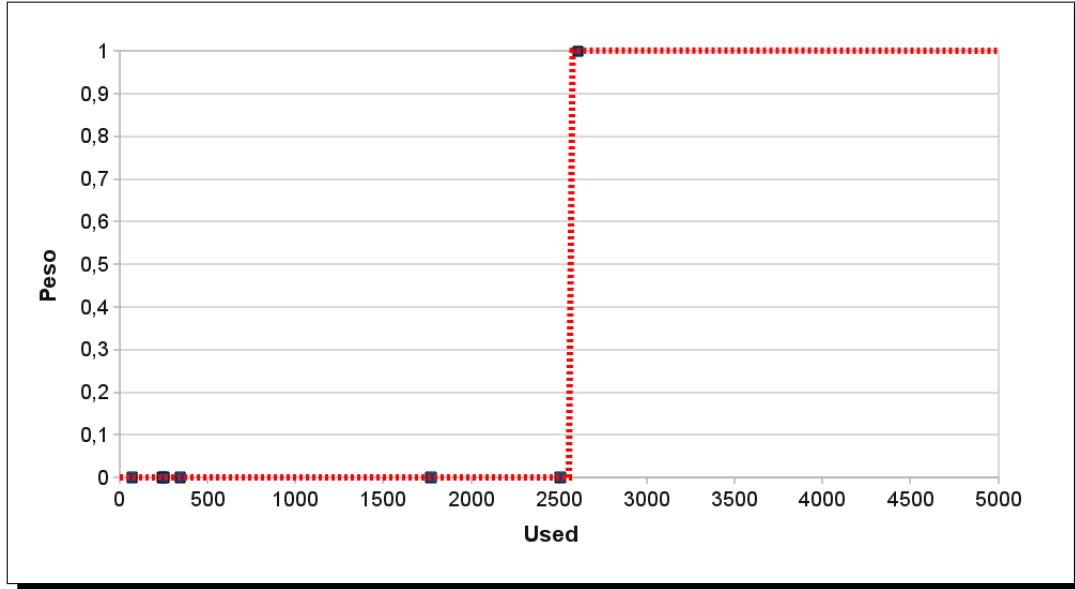


Figura 5.4: Pesos otorgados por la frontera $F2$ a las reglas para el concepto *used*

5.3.1.3. Asignación de pesos gradual

La asignación gradual de los pesos debe conseguir que observaciones cercanas a la frontera obtengan un peso inferior al peso máximo, pero superior al de otras observaciones que están más alejadas por debajo. Con el objetivo de medir la distancia entre las observaciones y la frontera se utilizó la desviación típica de las observaciones, σ . De esta forma, el peso asignado dependerá si la observación se encuentra a σ , 2σ , 3σ , etc. Sea F la *frontera de selección*, c el concepto, x la observación y n un número natural, el peso se calcula mediante las siguientes fórmulas:

$$w_c^F = 1, \quad \text{si } x \geq F \quad (5.6)$$

$$w_c^F = \frac{1}{2^n}, \quad \text{si } F - n \cdot \sigma \leq x < F - (n - 1) \cdot \sigma \quad (5.7)$$

Por ejemplo, si x estuviera a σ de distancia ($n = 1$) de F , su peso sería de $1/2$. Para el caso de los pesos de *used* puede producirse un problema cuando σ sea mayor que $F1$ y $F2$. En este tipo de situaciones es necesario reducir *sigma*. La posibilidad que se ha elegido para resolver este problema es eliminar la observación que más lejos está de $F1$ y $F2$, y realizar de nuevo el cálculo de *sigma* con el resto de observaciones. En el experimento de *QuizGuide*, presentado más adelante, la observación *20419* es

la más alejada, eliminándola la nueva σ obtenida es 1088,63, inferior tanto a $F1$ como a $F2$. La tabla 5.4 muestra los pesos otorgados de forma gradual mediante la fórmula anterior utilizando la nueva σ . En esta tabla se observa que las reglas con mayor peso total son las mismas que utilizando la función de pesos dicotómica, pero en esta ocasión no existen empates en los pesos totales. Por tanto, se puede afirmar que la mejor regla es la *DR23*, la segunda mejor es la *DR12* y la siguiente mejor la *DR19*. Además, se observa que el peso total que obtiene la regla *DR6* está más cerca del peso total de la *DR19*, mientras que en el caso anterior parecía que existía gran diferencia entre estos pesos.

Rule	w_{used}^{F1}	w_{used}^{F2}	w_{size}^{F1}	w_{size}^{F2}	w_{error}^{F1}	w_{error}^{F2}	W_r
DR19	0,125	0,125	1	1	1	1	4,25
DR23	0,5	1	1	1	1	1	5,5
DR8	0,125	0,125	1	1	1	0,5	3,75
DR33	0,5	0,5	0	0	0,25	0,25	1,5
DR7	0,125	0,125	1	1	0,25	0,25	2,75
DR6	0,5	0,5	1	1	0,25	0,25	3,5
DR12	1	1	1	1	0,25	0,25	4,5
DR14	0,125	0,125	1	1	0,125	0,125	2,5

Tabla 5.4: Pesos graduados otorgados por las fronteras $F1$ y $F2$

5.3.1.4. Asignación de pesos mediante la función *sigmoidal*

La figura 5.5 muestra los pesos asignados por la frontera $F2$ mediante la fórmula anterior. Se puede observar que se asignan diferentes pesos a cada observación, y éstos son proporcionales a la distancia (medida en σ) entre las observaciones y la *frontera de selección*. Analizando el gráfico, se puede notar que la observación con valor 2608 obtiene el mismo peso que la observación con valor 20419. Tal situación, se debe a que la asignación es sólo gradual por debajo de la frontera, ya que por encima de la frontera es siempre 1. En el caso expuesto, parece más adecuado asignar distintos pesos a estas dos observaciones, pues la distancia existente entre ellas es muy amplia. Una función que asigna pesos graduales por debajo y por encima de un punto determinado, frontera en nuestro caso, es la función *sigmoidal* (*sigmoid function*). Dicha función es ampliamente utilizada en el campo de las redes neuronales, y su fórmula general es la siguiente [Mitchell 1997, pp. 96-97]:

$$f(y) = \sigma(y) = \frac{1}{1 + e^{-y}} \quad (5.8)$$

En la figura 5.6 se puede ver la forma de la función *sigmoidal* para los valores de y entre -6 y $+6$. Como se ve en la figura, esta función asigna valores por debajo de 0,5 para las observaciones de y menores que 0, y valores por encima de 0,5 para las observaciones de y mayores que 0. En el caso que nos ocupa, no existen observaciones menores que 0, ya que la cobertura de cualquier regla es siempre positiva, el error

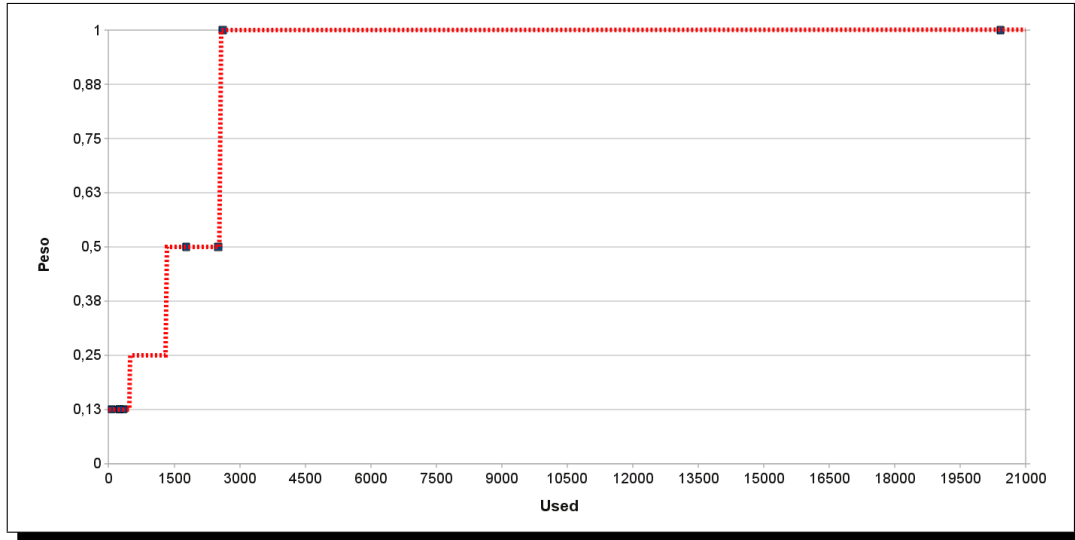


Figura 5.5: Pesos graduados otorgados por la frontera $F2$ a las reglas para el concepto *used*

o precisión no puede ser negativo, y el tamaño de una regla es siempre mayor que 0. Por lo tanto, es necesario hacer una traslación de la función *sigmoidal*, de forma que su dominio esté formado por números positivos reales. Una manera de conseguir este objetivo, en este ejemplo, es trasladar el origen de la función del punto $(-6,0)$ al punto $(0,0)$. Además, como los pesos deben ser proporcionales a la distancia entre la frontera y la observación, la frontera es el punto en el que la función debe cambiar de convexa a cóncava. Teniendo todo lo anterior en cuenta, el parámetro y de la función *sigmoidal* queda definido de la siguiente forma:

$$y = \frac{x - F}{a} \quad (5.9)$$

En la ecuación 5.9, el parámetro x corresponde al valor de la observación, F es la *frontera de selección* y a es un indicador del grado de inclinación⁷ de la función *sigmoidal*. Por tanto, la función anterior (véase la fórmula 5.8) queda reformulada de la siguiente forma:

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-\frac{x - F}{a}}} \quad (5.10)$$

Se adjunta en la figura 5.7 la representación gráfica de la función anterior (véase la fórmula 5.10) siendo el parámetro a igual a 1 y la frontera igual a 6. Se puede observar como se ha desplazado hacia la derecha la función *sigmoidal* general (véase fórmula 5.8). El análisis de esta figura muestra que las observaciones inferiores a la

⁷En [Mitchell 1997] se indica que e^{-y} puede ser sustituido en muchas ocasiones por $e^{-k \cdot y}$. El parámetro k , que modifica el valor de la y , se denomina grado de inclinación de la función.

frontera pierden al menos el 50 % del peso máximo⁸. Tal situación, no es demasiado preocupante en el caso de que las observaciones penalizadas estén lejos de la frontera, pero sí lo es para las que estén cercanas. Después de este análisis, es claro que la penalización inicial que se propuso mediante la fórmula 5.10 para las observaciones cercanas a la frontera sigue siendo muy elevada, ya que los pesos asignados a estas observaciones son al menos un 50 % inferiores que las que están por encima de la *frontera de selección*. Los experimentos realizados en la sección 5.3.3 muestran que es razonable que al menos se penalice a estas observaciones con un 20 %. Por tanto, su peso será como máximo un 80 % del peso máximo. Teniendo esta modificación en cuenta, la función se modifica de la siguiente forma:

$$\sigma(x) = \frac{1}{1 + k \cdot e^{-\frac{x - F}{a}}} \quad , \quad k \in (0, 1] \quad (5.11)$$

En esta ecuación, el parámetro k indica la penalización que sufrirán las observaciones que estén situadas por debajo de la frontera. El valor de $k = 1/4$ garantiza que una observación por debajo de la frontera obtenga un peso inferior al 80 % del peso máximo⁹. En este caso, como el peso máximo es 1 estas observaciones obtendrían un peso inferior a 0,8. Tal situación se puede observar en la figura 5.8, pues observaciones inferiores a 6 obtienen pesos inferiores a 0,8.

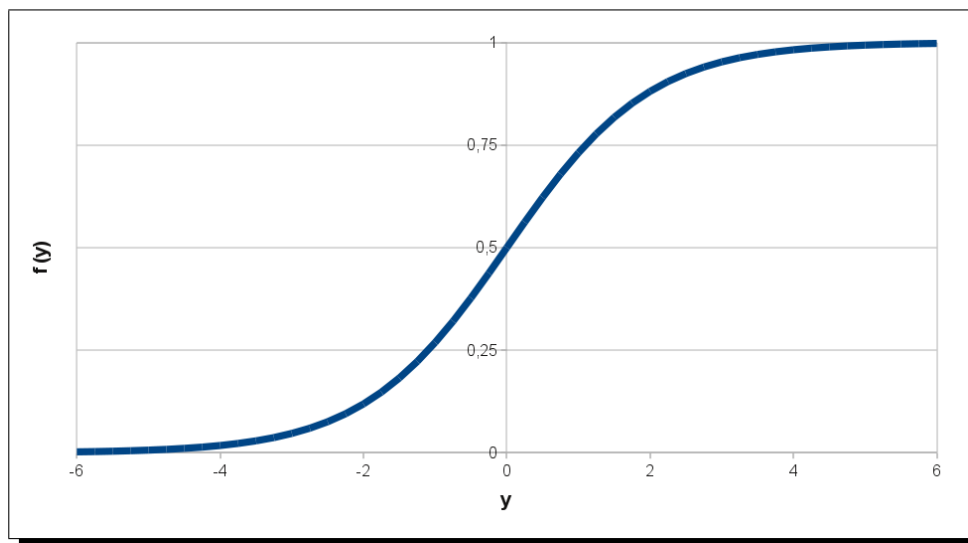


Figura 5.6: Función sigmoide de parámetro y

Anteriormente, se mencionó al parámetro indicador del grado de inclinación, a , pero no se expuso ninguna estimación sobre su cálculo, y se consideró el valor 1 como

⁸El peso máximo en este ejemplo está fijado en 1.

⁹El valor de $k = 1/4$ se obtiene de resolver la ecuación $1/(1 + k \cdot e^{-\frac{x - F}{a}}) = 0,8$.

valor por omisión. Debido a que este parámetro es el responsable de la inclinación de la función *sigmoidal*, o dicho de otro modo, la rapidez con la que crecen los pesos, es necesario establecer un valor adecuado para este parámetro. En las siguientes líneas se explica cómo se consiguió un valor adecuado para este parámetro. La idea que subyace en este punto, es que las observaciones máximas obtengan pesos cercanos a 1, y las mínimas a 0. Por tanto, si se quiere conseguir que la observación máxima obtenga un peso cercano a 1 hay que despejar el parámetro a de la siguiente fórmula:

$$\sigma(x) = \frac{1}{1 + k \cdot e^{-\frac{x_{max} - F}{a}}} = 1 \quad , \quad k \in (0, 1] \quad (5.12)$$

Por tanto, la ecuación 5.12 se puede reducir a:

$$\begin{aligned} 1 + k \cdot e^{-\frac{x_{max} - F}{a}} = 1 &\Rightarrow e^{-\frac{x_{max} - F}{a}} = 0 \Rightarrow \\ -\frac{x_{max} - F}{a} = \ln(0) &\Rightarrow a = \frac{x_{max} - F}{\infty} \Rightarrow a = 0 \end{aligned} \quad (5.13)$$

La solución matemática que se obtiene de resolver la ecuación 5.12 se muestra en la ecuación 5.13. Esta solución no es viable desde el punto de vista computacional, por lo que es necesario hallar una solución de manera computacional. La idea es encontrar una solución, tal que e^{-b} sea aproximado a cero con una cierta precisión. Teniendo esto en cuenta, los siguientes valores de la función exponencial garantizan las siguientes precisiones:

- $e^{-16} = 1,1250 \cdot 10^{-7}$ con una precisión de 7 dígitos.
- $e^{-32} = 1,2664 \cdot 10^{-14}$ con una precisión de 14 dígitos.
- $e^{-46} = 1,0531 \cdot 10^{-20}$ con una precisión de 20 dígitos.
- $e^{-64} = 1,6038 \cdot 10^{-28}$ con una precisión de 28 dígitos.

Entonces, se obtiene el valor de a mediante la fórmula 5.14, siendo b el exponente al que se eleva la función exponencial, y x_{max} la observación más alejada de la frontera:

$$a_b = \frac{x_{max} - F}{b} \quad , \quad b \in \{16, 32, 46, 64\} \quad (5.14)$$

Por otro lado, otra idea para obtener un valor para a es utilizar una medida de dispersión, como el recorrido intercuartílico (*interquartile range*, en adelante: IQ) por medio de las fórmulas 5.15 y 5.16:

$$a_{IQ} = Q_3 - Q_1 \quad (5.15)$$

$$a_{IQ2} = \frac{Q_3 - Q_1}{2} \quad (5.16)$$

La figura 5.9 presenta la gráfica de la función *sigmoideal* de la ecuación 5.11 para los valores del parámetro a : a_{16} , a_{32} , a_{46} , a_{64} , a_{IQ} y a_{IQ2} . En esta figura se exhiben seis funciones *sigmoideales* respecto a los anteriores valores mencionados para el parámetro a . De esta forma, la leyenda que se muestra junto al gráfico, contiene seis valores posibles: *sigmoideal-a16*, *sigmoideal-a32*, *sigmoideal-a46*, *sigmoideal-a64*, *sigmoideal-aIQ* y *sigmoideal-aIQ2*. Cada uno de los campos de esta leyenda corresponde a la función *sigmoideal* para los distintos valores de a . Por ejemplo, el campo de la leyenda *sigmoideal-a16* corresponde a la función *sigmoideal* calculada con $a = a_{16}$, mientras que, *sigmoideal-aIQ2* hace referencia a la función *sigmoideal* para $a = a_{IQ2}$. La *frontera de selección*, F , corresponde al valor 99 (Q_3), por este motivo las seis funciones se cruzan en este punto, ya que es el punto a partir del cual cambian de curvatura de convexa a cóncava.

La función *sigmoideal-a16* de la figura 5.9 (color azul claro) muestra que el mínimo valor que toma es 0,49, alcanzando valores cercanos a 1 para valores de x por encima de 400. Claramente, ésta es la función que más lentamente crece, aunque su valor base es muy alto (0,49) para el propósito que se busca. Por tanto, esta función no parece adecuada, ya que las observaciones más alejadas por debajo de F obtienen al menos un peso cercano a 0,5. La siguiente función, *sigmoideal-aIQ* (color marrón), obtiene un valor mínimo más bajo que la anterior, pero aún demasiado alto para el propósito buscado, pues las observaciones más alejadas obtendrían un peso cercano a 0,3; esta función tampoco parece adecuada. De igual forma, la función *sigmoideal-a32* (color amarillo) tampoco es adecuada, ya que observaciones alejadas por debajo de F obtendrían un peso mínimo por encima de 0,20. Las *sigmoideales* para los valores de a : a_{64} , a_{IQ2} y a_{46} son las funciones más adecuadas desde el punto de vista de las observaciones más alejadas por debajo de F , pues asignan pesos suficientemente bajos, por debajo de 0,10. Sin embargo, sí existen diferencias para las observaciones por encima de la frontera, pues unas crecen de forma más rápida que las otras. La función *sigmoideal-a64* (color azul oscuro) es la que más rápido crece, y por tanto, observaciones por encima de la frontera y cercanas a ésta obtendrán pesos demasiado altos, muy cercanos a los pesos obtenidos por las observaciones más alejadas por encima de F ; esto tampoco es deseable. El nivel de crecimiento de las dos funciones restantes es razonable por encima de la frontera, por lo que la elección está entre estas dos funciones. Si observamos detenidamente ambas gráficas, veremos como la *sigmoideal-aIQ2* (coloreada en verde en la figura 5.9) crece ligeramente más rápido que la *sigmoideal-a46* (color rojo en la figura 5.9) por encima de la frontera. Sin embargo, la función *sigmoideal-aIQ2* asigna pesos ligeramente más bajos que la otra función por debajo de la frontera. Es por esta razón, por la que la función *sigmoideal-aIQ2* es la más adecuada. Por tanto, el valor de $a = a_{IQ2}$, y la fórmula 5.11 queda modificada de la siguiente forma:

$$f(x) = \frac{1}{1 + (1/4) \cdot e^{-\frac{x - F}{\frac{Q_3 - Q_1}{2}}}} \quad (5.17)$$

Si observamos la fórmula 5.17, detectaremos que el denominador de la expresión $\frac{x - F}{Q_3 - Q_1}$ puede ser 0 cuando $Q_3 = Q_1$. En estos casos, no tiene sentido asignar pesos de forma gradual, pues si ambos cuartiles son iguales están indicando que prácticamente los datos son iguales. Dicho de otra forma, hay una variación tendente a 0. Esta situación puede darse en el concepto tamaño de la regla, y es más probable cuando las *DR* obtenidas tienen tamaños pequeños (por debajo de 2). En estas situaciones, se optará por una asignación de pesos dicotómica, de tal forma, que si la observación supera a la *frontera de selección* obtendrá peso máximo, y en caso contrario 0.

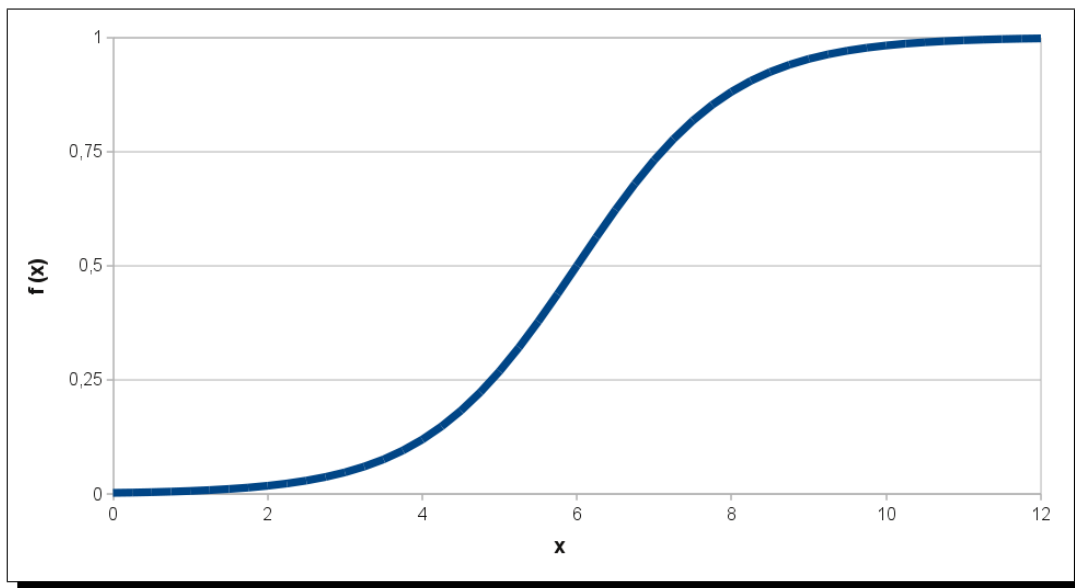


Figura 5.7: Función sigmoide desplazada para valores positivos

Otro análisis que merece especial atención, es el relativo a las *fronteras de selección*. La pregunta que a estas alturas el lector se formulará, es si disponiendo ahora de una función continua de asignación de pesos son necesarias dos *fronteras de selección*, o qué beneficios aporta una frontera adicional. Para ilustrar lo que significa disponer de dos fronteras, la figura 5.10 presenta una interpretación gráfica de las *fronteras de selección*. En ésta, se puede observar que la frontera *F1* está por encima de la *F2*, lo que implica que observaciones que estén por encima de ambas serán doblemente puntuadas. La frontera *F1* les asignará un peso que se sumará al peso que les asigne la frontera *F2*. Por tanto, la frontera adicional refuerza el peso de un determinado concepto para las observaciones más alejadas que estén por encima de dicha frontera. Esto se puede observar también en la tabla 5.3 (p. 101), ya que la función de la frontera *F1* añade peso a las observaciones que estén por encima de ésta. Por ejemplo, la observación 20419 se encuentra situada por encima

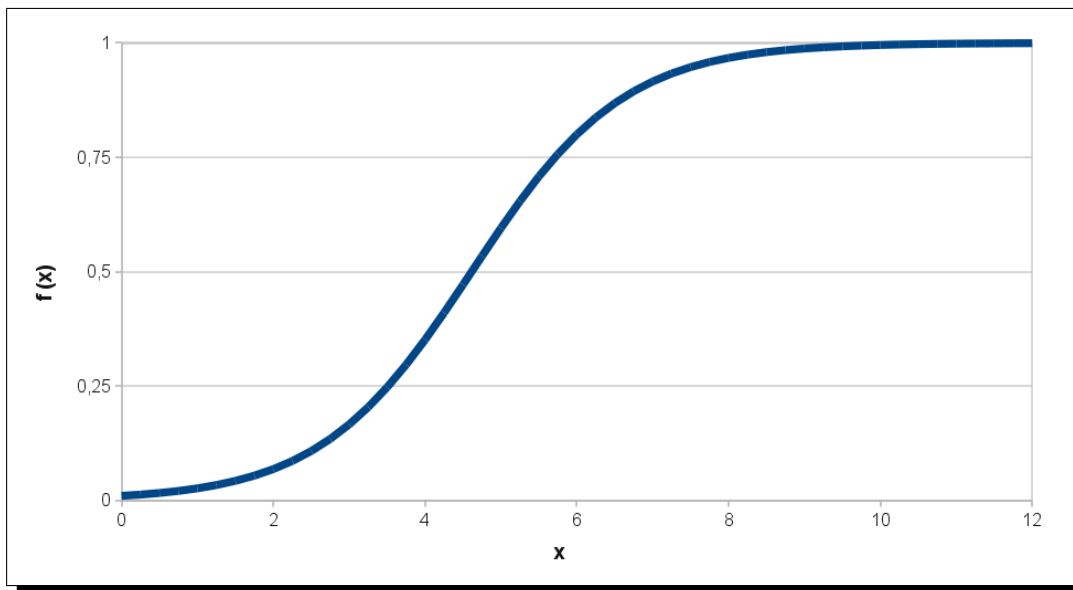


Figura 5.8: Función sigmoide desplazada para valores positivos con el parámetro k

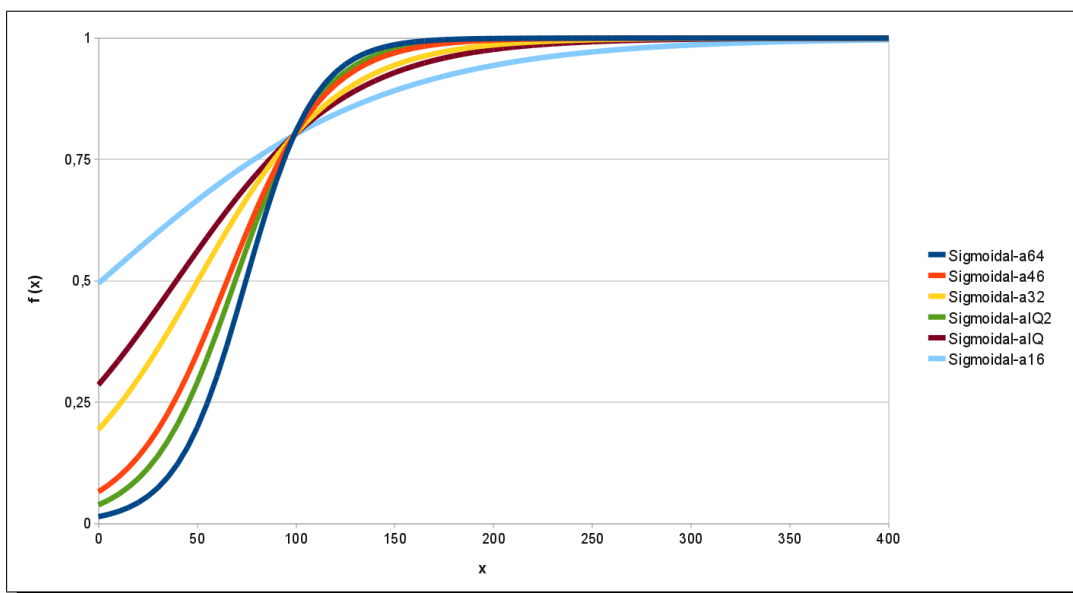


Figura 5.9: Funciones sigmoideas para distintos valores de la k . Las funciones sigmoideas han sido calculadas con los datos de las reglas de decisión obtenidas de los logs del sistema *CoMoLE* (véase sección de experimentos)

de $F1$ y $F2$ en la figura 5.10. Esta observación recibe un peso de 2^{10} para el concepto *used*, como se puede observar en la tabla 5.3. Existe una observación que está entre las fronteras $F2$ y $F1$, ésta es la observación 2608. Se puede observar en la tabla anterior (fila 2) que el concepto *used* otorga a esta observación un peso de 1. Como se explicó anteriormente, no es adecuado que las observaciones más alejadas tengan peso doble frente a otras que sí han superado alguna de las fronteras, aunque sí es cierto que debería existir una diferencia de peso entre éstas. Volviendo a la función *sigmoidal*, ésta garantiza la diferencia de peso, por lo que es innecesario la existencia de dos fronteras. Más aún, la frontera $F1$ (diez por ciento) se diseñó originariamente al principio de esta investigación, y después de realizar una serie de pruebas se constató que dicha frontera era demasiado restrictiva. De ahí partió la necesidad de establecer una nueva frontera menos restrictiva como es el tercer cuartil. El uso de la función *sigmoidal* para asignación de pesos hace redundante la existencia de la frontera $F1$, por lo que la única frontera útil es $F2$, que a partir de ahora tomará el nombre de F o Q_3 . Como consecuencia de este análisis, la función *sigmoidal* modificada de la fórmula 5.17 se modifica ligeramente a:

$$f(x) = \frac{1}{1 + (1/4) \cdot e^{-\frac{x - Q_3}{Q_3 - Q_1} \cdot 2}} \quad (5.18)$$

5.3.1.5. Propuesta final del método *GeSES*

Viendo las dificultades encontradas en la propuesta inicial del método *GeSES*, que han sido expuestas en este apartado con los anteriores análisis, dicho método ha sido mejorado modificándose las fases 4 y 5. Así, la propuesta final del método *GeSES* se resume en las siguientes etapas:

- (1) **Fase de limpieza.** Seleccionar los datos de interacción que hagan referencia a actividades prácticas, y establecer la *variable de evaluación*. El resultado de esta fase es una tabla con los perfiles de los estudiantes, sus datos de interacción y la *variable de evaluación*. Esta tabla constituye los *datos de análisis*.
- (2) **Generación de las reglas de decisión.** Consiste en la aplicación del algoritmo *C4.5* a los *datos de análisis* para generar las reglas de decisión. Los parámetros para este algoritmo son:
 - Atributos: dimensiones del perfil del estudiante, nombre de la actividad y contexto de la actividad (si es una actividad compuesta o no, el padre de la actividad, conceptos relacionados con la actividad, etc).
 - Variable de clasificación o clase: *variable de evaluación*.

¹⁰Los pesos para esta observación se pueden encontrar en la fila 7 de la tabla 5.3, correspondientes a la regla de decisión *DR12*. Para el concepto *used*, $W_{used}^{F1} = 1$ y $W_{used}^{F2} = 1$; su suma constituye el peso otorgado en el concepto *used*.

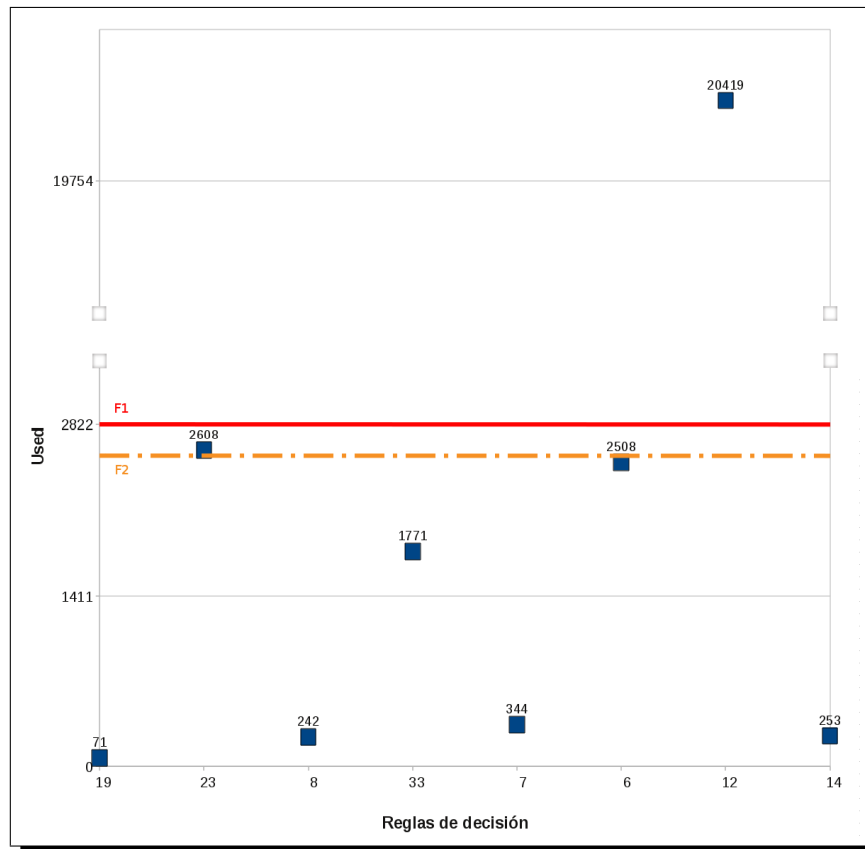


Figura 5.10: Fronteras de selección $F1$ y $F2$ en las observaciones del concepto *used*. Nota: la gráfica aparece segmentada en el eje de ordenadas a partir del número 3000 con el objetivo de mostrar las diferencias entre las dos fronteras, ya que la separación entre ambas es muy pequeña en comparación con el rango que tiene el eje de ordenadas que alcanza valores superiores a 20000.

El resultado de esta fase es un conjunto de reglas de decisión, cada una con su correspondiente precisión, y un *ranking* de reglas con los siguientes conceptos: *size*, *error*, *used*, *wrong*, *advantage* y *class*.

- (3) **Seleccionar las reglas relevantes para la característica de evaluación.** Seleccionar del *ranking* de reglas de decisión aquellas en las que la *variable de evaluación* cumpla la característica que se quiere evaluar.
- (4) **Establecer la frontera de selección.** Esta frontera o punto a partir del cual se realiza la selección de las reglas, se establece en función de los conceptos del *ranking* de reglas. Su función es hacer posible la selección de las reglas más relevantes respecto a los conceptos del *ranking*. La frontera utilizada corresponde al tercer cuartil, de forma que:

- Frontera: tercer cuartil, Q_3 , respecto de los conceptos: *used*, *size* y *error*. La forma de calcular el tercer cuartil es la siguiente:

1. Siendo N el número total de elementos en los datos, se calcula el rango, r , del cuartil mediante:

$$r = \frac{3}{4} \cdot (N - 1) \quad (5.19)$$

2. Comprobar si $|r|$ es entero. Siendo $f(x)$ la función de la distribución de los datos ordenados de menor a mayor respecto al *parámetro* seleccionado, y siendo x un valor entre 0 y $N-1$ (i.e. $f(5)$ devuelve el dato que ocupe la posición 5), se calcula el tercer cuartil de la siguiente manera:

$$Q_3 = f(r) \quad (5.20)$$

3. En caso de que $|r|$ no sea entero, el tercer cuartil se calcula mediante una ponderación entre el valor correspondiente al rango inmediatamente anterior, $f(r_{ant})$, y el correspondiente al rango del cuartil, $f(r)$. La siguiente fórmula realiza el cálculo del *tercer cuartil* o *percentil 75*:

$$Q_3 = f(r_{ant}) + (f(r) - f(r_{ant})) \cdot \frac{1}{4} \quad (5.21)$$

- (5) **Seleccionar las mejores reglas.** Esta fase está dividida en tres etapas: generación de la precisión, asignación de pesos a las reglas y selección de las reglas.
 - (5.1) Generación de la *precisión*. Se genera el parámetro *precision* a partir del parámetro *error*, pues $precision = 100 - error$, ya que el *error* está expresado en porcentajes.
 - (5.2) Asignación de pesos a las reglas. Se utiliza la función *sigmoidal* modificada de la ecuación 5.18 para la asignación de pesos de cada uno de los conceptos (*used*, *size* y *precision*):

$$w_c(x) = \frac{1}{1 + (1/4) \cdot e^{-\frac{x - Q_3^c}{Q_3^c - Q_1^c}}} \quad (5.22)$$

La ecuación 5.22 contiene los siguientes elementos: w_c hace referencia al peso que obtiene el concepto c para la regla, Q_3^c y Q_1^c son el tercer y primer cuartil de la tabla de *ranking* respecto al concepto c , y x es la observación. El *peso total* de cada regla se obtiene sumando los pesos de los conceptos, ponderados de acuerdo a la importancia de cada uno de los parámetros, siendo esta ponderación un número entre 0 y 1.

$$W_r = w_{used} \cdot s_{used} + w_{size} \cdot s_{size} + w_{precision} \cdot s_{precision} \quad (5.23)$$

Donde, W_r es el peso total de la regla r , w_c es el peso respecto al concepto c ($c \in \{used, size, precision\}$), y s_c es la importancia que tiene c ($s \in [0, 1]$).

- (5.3) Selección de las reglas. Se ordenan las reglas respecto al parámetro *peso total* y se seleccionan aquellas que mayores pesos han obtenido.

Aplicando el método *GeSES* se obtiene el conjunto de signos de situaciones problemáticas más representativas respecto al criterio de evaluación seleccionado. Así, este conjunto de signos sirve para diagnosticar los problemas del curso, y el profesor puede modificar o mejorar ciertos contenidos relacionados con los signos detectados. Para facilitar la comprensión del lector se expone en el siguiente apartado un gráfico del método expuesto.

5.3.2. Exposición gráfica del método

La figura 5.11 muestra el proceso llevado a cabo por el método *GeSES* (sección 5.3.1.5) de forma más intuitiva. De esta manera, se pretende que el lector pueda obtener una visión global del presente método. En dicha figura se han numerado las fases, tal y como se han expuesto, para facilitar la comprensión. El proceso comienza aplicando la fase de limpieza, señalada por (1) en la figura 5.11. Esta fase es equivalente a la fase de limpieza del método *Key-node*. Una vez se han obtenido los *datos de análisis* se aplica el algoritmo *C4.5* para la *generación de las reglas de decisión - DR* (señalado en la figura con (2)). La figura muestra las reglas de decisión obtenidas, que se pueden ver de forma más detallada en la figura 5.12. En dicha figura se puede observar que siete *DR* son de la clase *SI*, y siete de la clase *NO* (incluyendo a la regla *default*). Además, se puede observar que las reglas están ordenadas de mayor a menor precisión y por clase. El algoritmo *C4.5* para generación de *DR* incluye en la parte final de resultados un *ranking* de reglas de decisión ordenado en función del error esperado. Este *ranking* se puede ver de manera más detallada en la figura 5.13. Se puede observar en esta figura que las *DR* que mayor cobertura (columna *used* en la figura 5.13) presentan son las reglas *R7*, *R2*

y *R1*. Siguiendo en la figura 5.11, el siguiente paso es seleccionar las *DR* respecto al criterio de evaluación, que en la figura corresponde a seleccionar las reglas de la clase *NO* (la figura 5.14 muestra de forma más detallada las reglas seleccionadas). El paso cuarto corresponde a establecer la *frontera de selección*, Q_3 , (señalado en la figura con (4)) que sirve para seleccionar las mejores reglas a partir de la asignación de pesos aplicando la fórmula 5.22 (señalado en la figura 5.11 con (5)). En la parte inferior de la figura 5.11 se muestran las reglas seleccionadas, que son las que mayor peso total (aplicación de la fórmula 5.23) han obtenido, y desde el punto de vista del criterio de evaluación seleccionado las más representativas. A partir de estas reglas se obtienen los signos de situaciones problemáticas detectados, que es la información mostrada al profesor.

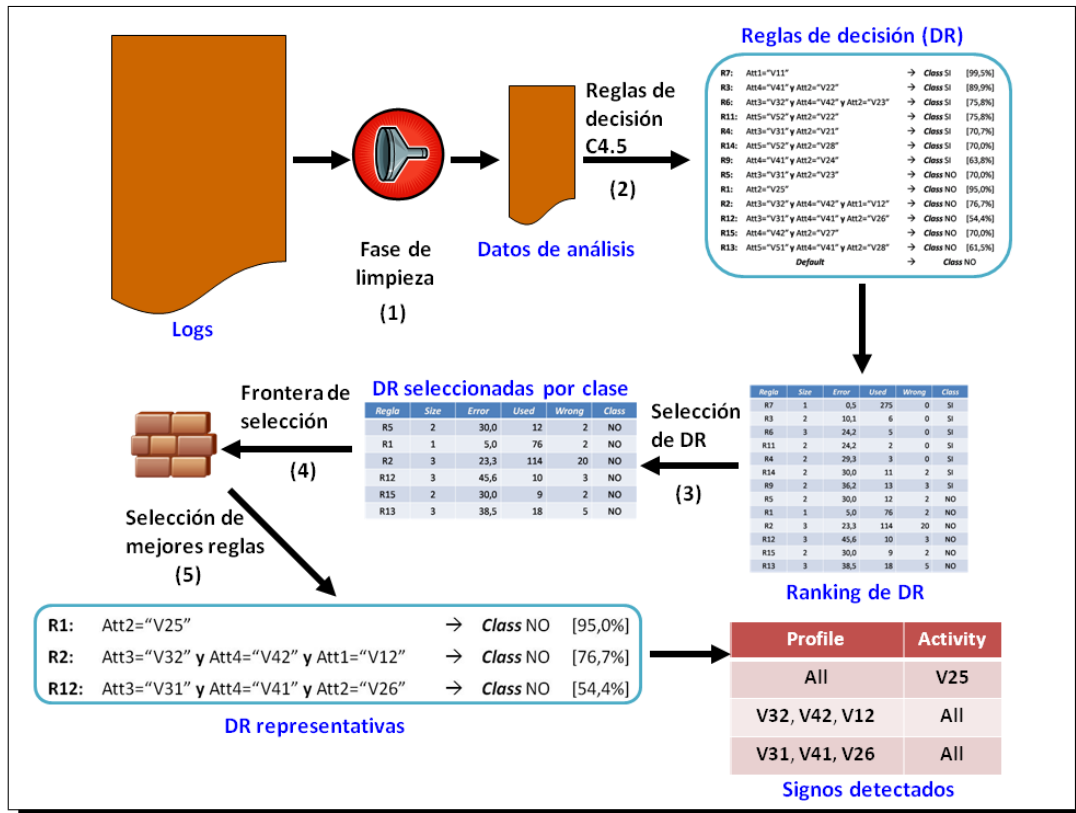


Figura 5.11: Detección de signos mediante el método *GeSES*. La figura de las *reglas de decisión* se puede ver con mayor detalle en la figura 5.12 (página 115), el *ranking de DR* se presenta con mayor detalle en la figura 5.13 (página 115) y las *DR seleccionadas por clase* pueden verse en la figura 5.14 (página 116).

R7:	Att1="V11"	→	Class SI	[99,5%]
R3:	Att4="V41" y Att2="V22"	→	Class SI	[89,9%]
R6:	Att3="V32" y Att4="V42" y Att2="V23"	→	Class SI	[75,8%]
R11:	Att5="V52" y Att2="V22"	→	Class SI	[75,8%]
R4:	Att3="V31" y Att2="V21"	→	Class SI	[70,7%]
R14:	Att5="V52" y Att2="V28"	→	Class SI	[70,0%]
R9:	Att4="V41" y Att2="V24"	→	Class SI	[63,8%]
R5:	Att3="V31" y Att2="V23"	→	Class NO	[70,0%]
R1:	Att2="V25"	→	Class NO	[95,0%]
R2:	Att3="V32" y Att4="V42" y Att1="V12"	→	Class NO	[76,7%]
R12:	Att3="V31" y Att4="V41" y Att2="V26"	→	Class NO	[54,4%]
R15:	Att4="V42" y Att2="V27"	→	Class NO	[70,0%]
R13:	Att5="V51" y Att4="V41" y Att2="V28"	→	Class NO	[61,5%]
	Default	→	Class NO	

Figura 5.12: Fase 2: Reglas de decisión generadas con el algoritmo *C4.5*

Regla	Size	Error	Used	Wrong	Class
R7	1	0,5	275	0	SI
R3	2	10,1	6	0	SI
R6	3	24,2	5	0	SI
R11	2	24,2	2	0	SI
R4	2	29,3	3	0	SI
R14	2	30,0	11	2	SI
R9	2	36,2	13	3	SI
R5	2	30,0	12	2	NO
R1	1	5,0	76	2	NO
R2	3	23,3	114	20	NO
R12	3	45,6	10	3	NO
R15	2	30,0	9	2	NO
R13	3	38,5	18	5	NO

Figura 5.13: *Ranking* de reglas de decisión. Las *DR* están ordenadas de menor a mayor error

Regla	Size	Error	Used	Wrong	Class
R5	2	30,0	12	2	NO
R1	1	5,0	76	2	NO
R2	3	23,3	114	20	NO
R12	3	45,6	10	3	NO
R15	2	30,0	9	2	NO
R13	3	38,5	18	5	NO

Figura 5.14: Fase 3: *DR* seleccionadas por clase

5.3.3. Experimentos realizados

Esta sección muestra los experimentos realizados para comprobar si la propuesta inicial del método *GeSES* es adecuada para conseguir el objetivo que se persigue. Además, dichos experimentos fueron útiles para mejorar los puntos débiles que presenta la propuesta inicial del método *GeSES*, y conseguir realizar una propuesta final de dicho método. Los experimentos consistieron en utilizar el método *GeSES* para analizar los logs de dos sistemas distintos: *QuizPACK/QuizGuide* y *CoMoLE*. A continuación se detallan ambos experimentos.

5.3.3.1. Análisis de los logs de *QuizPACK/QuizGuide*

En este experimento se analizaron los logs proporcionados por la *School of Information Sciences* de la *University of Pittsburgh (Pittsburgh, USA)*. Los datos se recogieron durante varios semestres, y responden a la interacción de los estudiantes con los sistemas *QuizPACK* y *QuizGuide* mientras realizaban un curso adaptativo de “Introducción a la Programación en el Lenguaje C”. Sólo se seleccionaron los datos del año 2007, pues los resultados obtenidos por los estudiantes en distintos años no son comparables, ya que tanto contenidos como estructura del curso cambian de forma anual. Por tanto, se analizaron 52734 interacciones generadas por 55 estudiantes.

QuizPACK y *QuizGuide*

QuizGuide [Brusilovsky 2004] es un sistema *Web* que proporciona ejercicios de auto-evaluación personalizados de programación en *C*. La función de este sistema no es la de generar y presentar los ejercicios, sino añadir ciertas características adaptativas relacionadas con la navegación entre las actividades. De esta forma, un estudiante es guiado mediante iconos adaptativos que indican su progreso individual en una determinada actividad y la relevancia que tiene ésta. *QuizPACK* [Brusilovsky 2005] es el sistema que genera y evalúa las actividades realizadas por los estudiantes (véase apéndice A.4.2, página 218, para mayor información sobre *QuizGuide* y *QuizPACK*).

Descripción de los logs

Las interacciones de los estudiantes con el sistema *QuizGuide* quedan almacenadas en una base de datos relacional. Así, cuando un estudiante responde a un determinado ejercicio o actividad se genera una nueva entrada de log que es añadida a dicha base de datos. Con el objetivo de integrar los datos a cerca de las interacciones, los estudiantes y los ejercicios, se construyó una tabla de logs (véase el apéndice A.5.1, página 222, para conocer una descripción más detallada sobre los datos de interacción). De esta forma, un log de esta tabla está formado por los siguientes campos:

- **userId**: identificador del estudiante en el sistema.
- **groupId**: identificador del grupo al que el estudiante pertenece. Siete grupos son posibles y corresponden a seis diferentes universidades, y un grupo adicional llamado “*world group*”. En este último grupo están los estudiantes que siguieron el curso de forma libre y no pertenecen a ninguna de las seis universidades anteriores.
- **result**: evaluación del resultado del estudiante en la actividad. Dos valores son posibles: 0 (respuesta incorrecta) y 1 (respuesta correcta).
- **activity**: nombre o identificador del ejercicio o actividad.
- **concept**: nombre del concepto que cubre la actividad.
- **conceptParent**: nombre del concepto superior (padre).
- **session**: identificador de la sesión Web.
- **success**: variable discreta generada a partir de la variable *result*. Si *result* es 1 entonces esta variable toma el valor de *yes*, en caso contrario su valor es *no*.
- **timeStamp**: marca de tiempo en la que el estudiante inicia la actividad.

La tabla 5.5 muestra un extracto de la tabla de logs, correspondiente a tres logs de los estudiantes con identificador *uid_199* y *uid_359*. Es importante señalar que, un estudiante puede realizar una actividad más de una vez, pero sin embargo, aunque la actividad repetida conceptualmente sea la misma, no es idéntica a la actividad mostrada en el intento anterior. En esta tabla se puede observar que el estudiante *uid_199* realizó la actividad *2dimensional_array1* más de una vez. En el primer intento no obtuvo éxito (*success = no*), mientras que en el segundo consiguió superar la actividad con éxito (*success = yes*). Además, las filas 1 y 2 de la tabla muestran información sobre el resultado que obtuvo en el ejercicio (0,00 en la primera fila y 1,00 en la segunda), el grupo al que pertenece el estudiante (*gid_190*), el concepto inherente en esta actividad (*printf*), el concepto superior (*output*) y la sesión que inició el estudiante (*21BC5*). La última fila de la tabla muestra a otro estudiante que obtuvo éxito en la misma actividad.

userId	groupId	result	activity	concept	concept-Parent	session	success
uid_199	gid_190	0,00	2dimensional_array1	printf	output	21BC5	no
uid_199	gid_190	1,00	2dimensional_array1	printf	output	21BC5	yes
uid_359	gid_72	1,00	2dimensional_array1	printf	output	A0B33	yes

Tabla 5.5: Ejemplos de logs en *QuizGuide*

Análisis mediante el método *GeSES*

Se aplicó el método *GeSES* de la sección 5.3.1.1 (página 98) a los logs que se generaron en los sistemas *QuizGuide/QuizPACK*. Las fases que se realizaron fueron:

(1) **Fase de limpieza.** Se seleccionaron las interacciones correspondientes al año 2007, y se diseñó la tabla de logs, cuyo extracto puede verse en la tabla 5.5. La tabla completa contiene 52734 filas, y constituye los *datos de análisis*. Además, se asignó como *variable de evaluación* a la variable *success*. La tabla 5.6 muestra un análisis estadístico preliminar de los *datos de análisis*. Las dos primeras filas de la tabla indican los valores que posee la variable *success*, así como la distribución de estos valores. Se puede observar, que la proporción entre las clases *yes* y *no* es similar (46 % frente a 54 %). Las siguientes siete filas contienen información sobre los distintos grupos de estudiantes y el número de instancias para cada uno. Es claro que, el grupo *gid_67* tiene el menor número de instancias o logs (70), y el grupo *gid_190* agrupa a la mayoría de las instancias (38382). Finalmente, la última fila de la tabla ofrece los nombres de las actividades (46 actividades) que contiene el curso adaptativo “Introducción a la Programación en el Lenguaje C”. Por ejemplo, las tres primeras actividades (*2dimensional_array1*, *2dimensional_array2*, *2dimensional_array3*) hacen referencia a ejercicios sobre el elemento *array* en el lenguaje C.

(2) **Generación de las reglas de decisión.** Se aplica el algoritmo *C4.5* a los *datos de análisis* con los siguientes parámetros:

- Atributos: *groupId*, *activity*.
- Variable de clasificación: *success*.

El conjunto de reglas de decisión obtenido puede verse en el apéndice A.6.1, página 243. La tabla 5.7 muestra el *ranking* que ofrece el programa *C4.5rules*. Este *ranking* es el que se utilizará para seleccionar las reglas de decisión más representativas respecto a la característica de evaluación.

(3) **Seleccionar las reglas más relevantes para la característica de evaluación.** La *variable de evaluación* es *success*, luego se seleccionan las *DR* en las que dicha variable contenga el valor *no*. La tabla 5.8 muestra el conjunto de reglas

Campo	Valor	Número de instancias
success	yes	24010
	no	28724
groupId	gid_67	70
	gid_72	1629
	gid_190	38382
	gid_373	3879
	gid_394	280
	gid_441	6536
	gid_442	1958
activity	2dimensional_array1, 2dimensional_array2, 2dimensional_array3, arithmetic_expresion1, arithmetic_expresion2, character_array1, character_array2, character_array3, character_processing1, character_processing2, character_processing3, conditional_operator1, complex_conditional1, complex_conditional2, function1, function2, function3, printing1, printing2, variable1, variable2, variable3, constant1, constant2, globvar_simplefunc1, globvar_simplefunc2, increment_decrement1, compound_assignment1, logical_expresion1, logical_operator1, if_else1, if_else2, do1, do2, for1, for2, while1, while2, nested_loop1, switch1, pointer1, pointer2, pointer3, array1, array2, array3	

Tabla 5.6: Distribución de los datos de análisis de *QuizGuide*

seleccionadas del *ranking* de la tabla 5.7. Se puede apreciar que se han seleccionado 8 reglas de decisión: *DR19*, *DR23*, *DR8*, *DR33*, *DR7*, *DR6*, *DR12* y *DR14*. La regla que mayor cobertura (columna *used*) tiene es *DR12*, la que menor error presenta es *DR19* y la más pequeña (columna *size*) la *DR33*.

(4) **Establecer la frontera de selección.** Se establecen para los conceptos: *size*, *error* y *used* de la tabla 5.8 las siguientes fronteras de selección: *F1* (diez por ciento) y *F2* (tercer cuartil). La tabla 5.9 muestra los valores de estas fronteras para cada uno de los conceptos. Por ejemplo, el valor que toma *F1* para el concepto *used* es 2822, pues la suma de todos los valores para este concepto es 28216, y el 10 % de esta cantidad redondeado es 2822. Hay que notar, que en el caso del concepto *error* se utiliza el primer cuartil en vez del tercero, ya que es importante que la regla tenga el menor error posible.

(5) **Seleccionar las mejores reglas.** En esta fase se asignan pesos a cada regla, aplicando la fórmula 5.5 (página 100). Es importante destacar que, se ha fijado el

Rule	Size	Error	Used	Wrong	Advantage
DR21	2	0,0 %	74	0 (0,0 %)	74 (74 0) yes
DR20	2	19,2 %	159	30 (18,9 %)	99 (129 30) yes
DR13	2	22,5 %	60	13 (21,7 %)	34 (47 13) yes
DR9	2	25,6 %	236	60 (25,4 %)	116 (176 60) yes
DR25	1	32,5 %	3657	1189 (32,5 %)	391 (1297 906) yes
DR16	2	35,5 %	280	99 (35,4 %)	82 (181 99) yes
DR40	1	38,9 %	16061	6526 (40,6 %)	3009 (9535 6526) yes
DR17	2	46,4 %	738	342 (46,3 %)	54 (396 342) yes
DR3	2	47,3 %	278	131 (47,1 %)	16 (147 131) yes
DR19	2	17,6 %	71	12 (16,9 %)	0 (0 0) no
DR23	2	18,2 %	2608	475 (18,2 %)	0 (0 0) no
DR8	2	20,9 %	242	50 (20,7 %)	0 (0 0) no
DR33	1	29,8 %	1771	494 (27,9 %)	0 (0 0) no
DR7	2	32,1 %	344	110 (32,0 %)	0 (0 0) no
DR6	2	32,4 %	2508	813 (32,4 %)	0 (0 0) no
DR12	2	34,2 %	20419	7522 (36,8 %)	0 (0 0) no
DR14	2	39,7 %	253	100 (39,5 %)	0 (0 0) no

Tabla 5.7: *Ranking* de *DR* de los datos de análisis de *QuizGuide*

Rule	Size	Error	Used	Wrong	Advantage
DR19	2	17,6 %	71	12 (16.9 %)	0 (0 0) no
DR23	2	18,2 %	2608	475 (18.2 %)	0 (0 0) no
DR8	2	20,9 %	242	50 (20.7 %)	0 (0 0) no
DR33	1	29,8 %	1771	494 (27.9 %)	0 (0 0) no
DR7	2	32,1 %	344	110 (32.0 %)	0 (0 0) no
DR6	2	32,4 %	2508	813 (32.4 %)	0 (0 0) no
DR12	2	34,2 %	20419	7522 (36.8 %)	0 (0 0) no
DR14	2	39,7 %	253	100 (39.5 %)	0 (0 0) no

Tabla 5.8: Reglas de decisión seleccionadas del *ranking* de *DR* de los datos de análisis de *QuizGuide*

peso máximo en el valor 1, y que se ha considerado que cada uno de los conceptos tenga la misma importancia, i.e., $s_{used} = s_{size} = s_{error} = 1$. Teniendo lo anterior en cuenta, la fórmula 5.5 se modifica de la siguiente forma:

$$W_r = \sum_{j \in \{1,2\}} w_{used}^{Fj} + w_{size}^{Fj} + w_{error}^{Fj} , \quad \text{siendo } w_c^{Fj} \in [0,1]$$

La tabla 5.10 muestra los pesos que obtiene cada regla respecto a cada una de las fronteras. Como se puede observar, en la tabla están señaladas en negrita las tres reglas de decisión que mayor peso total (W_r en la ecuación anterior) obtienen: *DR23*, *DR19* y *DR12*. Parece claro que la regla más representativa es *DR23*, pero

Rule	Size	Error	Used
DR19	2	17,6 %	71
DR23	2	18,2 %	2608
DR8	2	20,9 %	242
DR33	1	29,8 %	1771
DR7	2	32,1 %	344
DR6	2	32,4 %	2508
DR12	2	34,2 %	20419
DR14	2	39,7 %	253
Σ	15	224,9	28216
F1	2	22	2822
F2	2	20,23	2533
σ	0,33	7,66	6460,38
σ_r	–	–	1088,63

Tabla 5.9: Fronteras de selección para los conceptos del *ranking* de *DR* de los datos de análisis de *QuizGuide*

se produce un empate en el peso total en las otras dos. Este empate es debido a que el sistema de selección propuesto en el método *GeSES* otorga el valor de 1 si la observación supera la frontera y 0 si no la supera, por tanto, como el peso total es la suma de los pesos es probable la existencia de empates. Además, observando la tabla 5.9 veremos que el peso de la regla *DR6* ha sido infravalorado, ya que en el concepto *used* recibe un peso de 0 (columnas 2 y 3 en la tabla), y sin embargo, es una observación muy cercana, su valor es 2508, a la frontera *F2*, con valor 2533. La misma situación se puede observar para la regla *DR8*, que toma un valor de 20,9 en el concepto *error*. En este caso, el valor de *F2* para el concepto *error* es 20,23, que está por debajo¹¹ del error de la regla *DR8*, y por esta razón el peso respecto de la frontera *F2* que recibe en este concepto es 0 (columna 7 en la tabla).

Con el objetivo de evitar los efectos de infravaloración producidos en las observaciones cercanas a alguna de las fronteras, se decidió realizar el experimento modificando la asignación de pesos mediante la aplicación de una asignación gradual por debajo de la frontera. Parece lógico pensar que, el peso debe ser proporcional a la distancia que se encuentre la observación de la frontera. De esta forma, se considera que esta distancia debe estar en proporción con la dispersión de los datos. Por esta razón, se decidió utilizar la desviación típica (σ) como medida de distancia. Por tanto, la función de asignación de pesos para cada una de las fronteras (fórmulas 5.6 y 5.7 - página 102) se puede resumir en:

¹¹Es importante que las reglas de decisión seleccionadas tengan el menor error posible, por eso en el concepto *error* se asigna el peso máximo si la observación está por debajo de la frontera, y 0 si está por encima. En este caso, el valor 20,9 es claramente mayor que 20,23, luego su peso respecto de esta frontera es 0.

Rule	w_{used}^{F1}	w_{used}^{F2}	w_{size}^{F1}	w_{size}^{F2}	w_{error}^{F1}	w_{error}^{F2}	W_r
DR19	0	0	1	1	1	1	4
DR23	0	1	1	1	1	1	5
DR8	0	0	1	1	1	0	3
DR33	0	0	0	0	0	0	0
DR7	0	0	1	1	0	0	2
DR6	0	0	1	1	0	0	2
DR12	1	1	1	1	0	0	4
DR14	0	0	1	1	0	0	2

Tabla 5.10: Pesos otorgados por las fronteras para los conceptos del *ranking* de *DR* de los datos de análisis de *QuizGuide*

$$\begin{aligned}
 w_c^F &= 1, & si & \quad x \geq F \\
 w_c^F &= \frac{1}{2^n}, & si & \quad F - n \cdot \sigma \leq x < F - (n - 1) \cdot \sigma
 \end{aligned}
 \tag{5.24}$$

Aplicando la fórmula 5.24 a los datos anteriores se obtienen nuevos pesos para cada concepto, que pueden verse en la tabla 5.11. Sin embargo, para el concepto cobertura (columna *used* en la tabla 5.10) se ha utilizado σ_r en vez de σ . La razón de elegir este nuevo valor es porque el valor de σ es superior al valor de cualquiera de las fronteras, por lo que su aplicación no produciría el efecto de asignación de pesos graduales. Por eso, es necesario utilizar un valor reducido de σ hasta que sea inferior a las fronteras. Observando la tabla 5.9 (página 121), existe una observación, 20419, que es la que produce que *sigma* sea tan grande. Por tanto, la solución propuesta es eliminar esta observación del cálculo de la desviación típica. Realizando de nuevo el cálculo de la desviación típica (σ_r) se obtiene el valor de 1088,63, que es inferior a las fronteras.

Nuevamente, se observa en la tabla 5.11 que las mejores reglas son: *DR23*, *DR12* y *DR19*. Claramente, la mejor sigue siendo la regla *DR23*, que es la que obtiene mayor peso total, y además, las dos reglas siguientes obtienen distintos pesos totales, por lo que no hay empates. Además, se puede observar que las reglas *DR8* y *DR6* obtenían en el caso anterior una puntuación infravalorada. En el caso de la regla *DR8*, la distancia entre ésta y la tercera mejor regla (*DR19*) es de 1 para el sistema de asignación de pesos dicotómicos, mientras que en el actual sistema de pesos gradual esta distancia es de 0,5. Por tanto, la distancia de una regla respecto a poder ser elegida en el conjunto de mejores reglas, se reduce con este nuevo sistema de asignación de pesos. La misma situación se produce con la regla *DR6*, incluso en este caso la distancia se ha reducido más que el caso anterior, es decir, $distancia_{sistema-anterior}(DR19, DR6) = 2$ y $distancia_{sistema-gradual}(DR19, DR6) = 0,75$.

La figura 5.5 (página 104) muestra la gráfica de la función de asignación de pesos (fórmula 5.24 - página 122) para el concepto cobertura respecto a la frontera *F2*.

Rule	w_{used}^{F1}	w_{used}^{F2}	w_{size}^{F1}	w_{size}^{F2}	w_{error}^{F1}	w_{error}^{F2}	W_r
DR19	0,125	0,125	1	1	1	1	4,25
DR23	0,5	1	1	1	1	1	5,5
DR8	0,125	0,125	1	1	1	0,5	3,75
DR33	0,5	0,5	0	0	0,25	0,25	1,5
DR7	0,125	0,125	1	1	0,25	0,25	2,75
DR6	0,5	0,5	1	1	0,25	0,25	3,5
DR12	1	1	1	1	0,25	0,25	4,5
DR14	0,125	0,125	1	1	0,125	0,125	2,5

Tabla 5.11: Pesos graduados otorgados por las fronteras para los conceptos del *ranking* de *DR* de los datos de análisis de *QuizGuide*

Teniendo en cuenta que se utiliza como desviación típica σ_r , los pesos asignados responden a:

$$\begin{aligned}
 1444,4 \leq x < 2533 &\Rightarrow w_{used}^{F2} = \frac{1}{2} \\
 355,74 \leq x < 1444,4 &\Rightarrow w_{used}^{F2} = \frac{1}{4} \\
 0 \leq x < 355,74 &\Rightarrow w_{used}^{F2} = \frac{1}{8}
 \end{aligned} \tag{5.25}$$

Observando la tabla 5.9 (página 121) veremos que no hay observaciones para el segundo intervalo, por eso, en la tabla 5.11 no aparece ninguna observación con peso 0,25 en la columna 3. Tal situación indica que la asignación gradual de pesos obtiene mejores resultados que la asignación dicotómica, pero es aún mejorable. Además, se nota que la observación 2608, correspondiente a *DR23*, obtiene el mismo peso en el concepto cobertura que la observación 20419, correspondiente a la regla *DR12*, a pesar de que la distancia entre ambas es muy grande. De igual forma que se utilizó una asignación gradual de pesos para las observaciones por debajo de la frontera, el siguiente paso consiste en utilizar esta misma asignación gradual por encima de la frontera.

Después de una serie de pruebas con diversas funciones de asignación de pesos, como las funciones parabólicas, las basadas en deciles, y la función *sigmoidal*, se decidió utilizar esta última. Ahora bien, esta función como se conoce por la fórmula 5.8 (página 103) no responde a lo que se está buscando, ya que en su dominio hay valores negativos, y está centrada en el eje de ordenadas, tal y como se aprecia en la figura 5.6 (página 105). Por tanto, es necesario realizar una serie de transformaciones para adecuar la función *sigmoidal*, con el fin de conseguir la asignación de pesos gradual por encima y por debajo de la frontera.

En esta fase fue necesario substituir el concepto *error* por el concepto *precision*, de forma que $precision = 100 - error$. La razón de esta substitución responde a que para el concepto *error* se necesita una función diferente a la *sigmoidal*, su “inversa”.

Tal es la función *logit*, pero reproducir en ésta las transformaciones realizadas en la función *sigmoidal* es una tarea con cierta complejidad, por eso se prefirió trabajar con el concepto opuesto al error, i.e. precisión.

Tal y como se explica en 5.3.1.4 (página 103), la fórmula que se utiliza finalmente para la asignación de pesos es la ecuación 5.18 (página 110). La figura 5.15 muestra la forma que tiene la función *sigmoidal* respecto al concepto cobertura para distintos valores del parámetro a , tomando la frontera $F2$ como referencia. En esta figura se aprecia que las funciones para los parámetros a_{32} , a_{46} y a_{64} asignan valores muy bajos a las observaciones más alejadas por debajo de la frontera, lo cual es una característica deseable. El objetivo es conseguir una función de asignación de pesos gradual de forma que otorgue pesos muy bajos a las observaciones más alejadas por debajo de la frontera, y pesos suficientemente altos a las observaciones que se encuentren por encima de ésta. Respecto a este objetivo, el parámetro “ a = recorrido intercuartílico” (a_{IQ}) produce la peor función. La función obtenida con el parámetro a_{IQ2} ($a = (Q_3 - Q_1)/2$) es aceptable por encima de la frontera, pero asigna pesos por encima de 0,25 a las observaciones más alejadas por debajo de la frontera.

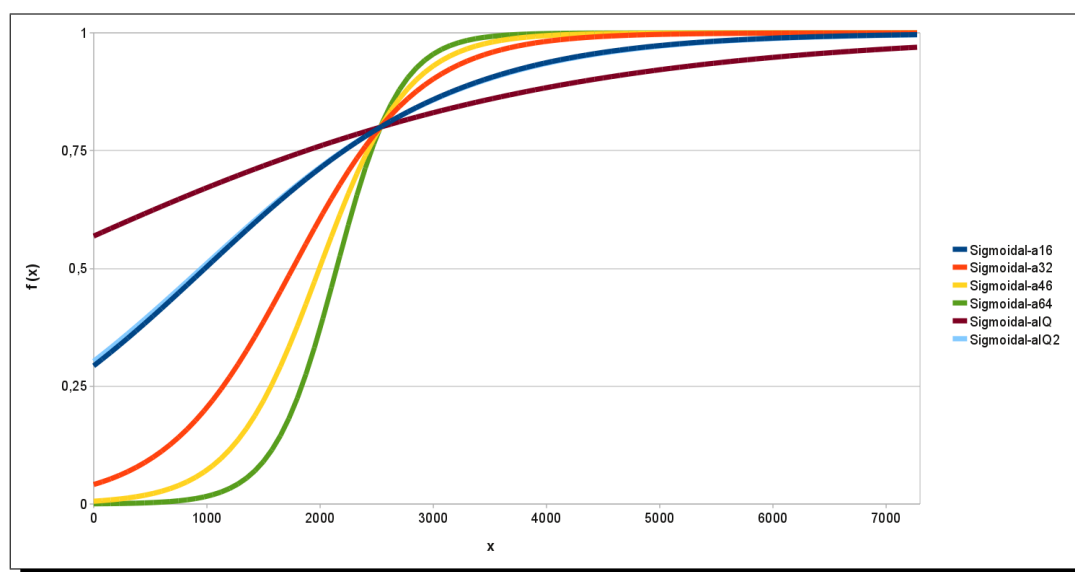


Figura 5.15: Funciones sigmoideas para distintos valores de la k para los datos de *QuizGuide*

Realizando estas pruebas para el concepto *precision* se obtuvo que el valor de $a_{46} = 0,06$ es muy bajo, mientras que el valor de $a_{IQ2} = 6,31$ es aceptable. El valor de a_{46} produce que la función tenga una pendiente muy pronunciada, ya que el primer valor por encima de 0 se alcanza en $x = 79,38$, y la función alcanza el valor 1 en $x = 82,4$. Esta situación no es deseable, ya que no se produce una graduación en los pesos asignados, pues observaciones por debajo de 79,38 obtendrán peso 0, y observando los datos veremos que el 75 % de las observaciones son inferiores a este

valor. Por tanto, el valor de $a = a_{IQ2}$ cumple el objetivo perseguido. En la tabla 5.12 se muestran los pesos obtenidos mediante la fórmula 5.18.

Otro problema que se encontró en esta fase de experimentación es que el tercer cuartil y el primero sean es mismo, es decir, que IQ sea 0. Tal situación ocurre en el concepto tamaño de la regla (columna *size*). En este caso, se decidió utilizar el enfoque primario, es decir, asignación de pesos dicotómica. Esto es debido a que no parece razonable realizar una asignación de pesos gradual, ya que, si IQ es cero, indica que no hay casi variación en los datos. Por tanto, no tiene sentido aplicar una asignación de pesos gradual a observaciones prácticamente iguales.

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
DR19	0,32	1	0,86	2,17
DR23	0,81	1	0,85	2,66
DR8	0,35	1	0,78	2,13
DR33	0,67	0	0,47	1,14
DR7	0,37	1	0,38	1,75
DR6	0,80	1	0,37	2,16
DR12	1	1	0,30	2,30
DR14	0,35	1	0,15	1,51

Tabla 5.12: Pesos otorgados por la función sigmoïdal modificada para los datos de análisis de *QuizGuide*

En la tabla 5.12 se puede observar que las reglas de decisión con mayor peso son las mismas (*DR23*, *DR12* y *DR19*) que en el sistema de asignación de pesos anterior, pero se observa que hay dos reglas, *DR6* y *DR8*, que se encuentran muy cercanas en puntuación a la tercera mejor. Hay que tener en cuenta que el sistema de pesos puntúa sobre 3, esto quiere decir que el mayor peso total posible es 3. En concreto, el mayor peso total lo obtiene la regla *DR23* con una puntuación de 2,66. Los pesos de la reglas *DR6* y *DR8*, 2,16 y 2,13 respectivamente, son ligeramente inferiores al peso de la regla *DR19*, 2,17. En concreto, esta diferencia es de sólo 0,01 para la primera regla y 0,03 para la segunda. Observando la tabla, se aprecia que la regla *DR6* es mejor en el concepto cobertura, pero es peor en el concepto precisión que la regla *DR19*. Sin embargo, la cobertura de la regla *DR8* es ligeramente mejor que la de la regla *DR19*, pero es ligeramente peor la precisión. Por tanto, en este caso se observa como cobra especial relevancia el grado de importancia de los conceptos (s_c). Es importante recordar que al principio del experimento se decidió que todos los conceptos tuvieran el mismo grado de importancia, pero si la importancia del concepto cobertura fuera mayor que la de la precisión, la tercera mejor regla sería *DR6*, en vez de *DR19* y la regla *DR8* seguiría siendo peor que las otras dos.

La tabla 5.13 muestra las reglas más relevantes, siendo la más relevante *DR23*. Ésta indica que la mayoría de los alumnos pertenecientes a alguno de los grupos: *gid_190*, *gid_441*, *gid_373* y *gid_394* tuvieron dificultades al realizar alguna de las actividades: *2dimensional_array2*, *do2*, *array1* y *array3*. La siguiente regla, *DR12*

Rule 23:
<pre> groupid in {gid_190, gid_441, gid_373, gid_394} activity in {2dimensional_array2, do2, array1, array3} ->class no [81,8 %] </pre>
Rule 12:
<pre> groupid = gid_190 activity in {2dimensional_array1, 2dimensional_array2, 2dimensional_array3, arithmetic_expression1, arithmetic_expression2, character_array1, character_array2, character_array3, character_processing1, character_processing2, character_processing3, conditional_operator1, do2, function1, logical_expression1, nested_loop1, pointer2, pointer3, printing2, variable2, array1, array2, array3, do1, for1, for2, function3, if_else1, if_else2, logical_operator1, pointer1, while2, globvar_simplefunc2} ->class no [65,8 %] </pre>
Rule 19:
<pre> groupid = gid_441 activity = pointer2 ->class no [82,4 %] </pre>

Tabla 5.13: Reglas más relevantes respecto a la característica de evaluación para los datos de análisis de *QuizGuide*

indica que gran parte de los alumnos del grupo *gid_190* mostraron problemas en la mayoría de las actividades del curso. Finalmente la regla *DR19* exhibe que la mayoría de estudiantes del grupo *gid_441* tuvieron dificultades con la actividad *pointer2*. Adicionalmente se incluye la tabla 5.14 con las dos reglas que se encuentran más cercanas a la tercera mejor regla. Por tanto, como se puede observar la información expuesta en este párrafo puede ser especialmente útil para el profesor, pues de forma precisa puede conocer las actividades en las que mostraron más dificultades los estudiantes, así como el perfil o perfiles de estudiantes afectados por estos problemas.

La tabla 5.15 presenta un resumen de los signos detectados con el método *GeSES* (sección 5.3.1.5), por eso se la denomina “tabla de signos”. La información contenida en esta tabla es relevante para el profesor, ya que con la ayuda de esta información, puede conocer fácilmente los problemas que presentan los estudiantes, así como las actividades de mayores dificultades. Por esta razón, los resultados de la aplicación del método, que se resumen en esta tabla, serían los mostrados al profesor.

Rule 6:
groupid in {gid_190, gid_373}
activity in {character_array2, printing2, while2}
->class no [67.6 %]
Rule 8:
groupid = gid_373
activity = character_processing3
->class no [79.1 %]

Tabla 5.14: Reglas cercanas a las más relevantes para los datos de análisis de *QuizGuide*

Perfil	Actividad
groupid in {gid_190, gid_441, gid_373, gid_394}	{2dimensional_array2, do2, array1, array3}
groupid = gid_190	{2dimensional_array1, 2dimensional_array2, 2dimensional_array3, arithmetic_expression1, arithmetic_expression2, character_array1, character_array2, character_array3, character_processing1, character_processing2, character_processing3, conditional_operator1, do2, function1, logical_expression1, nested_loop1, pointer2, pointer3, printing2, variable2, array1, array2, array3, do1, for1, for2, function3, if_else1, if_else2, logical_operator1, pointer1, while2, globvar_simplefunc2}
groupid = gid_441	pointer2

Tabla 5.15: Signos detectados para los datos de análisis de *QuizGuide*

5.3.3.2. Análisis de los logs de *CoMoLE*

CoMoLE

CoMoLE (*Context-based Adaptive Mobile Learning Environments*) [Martín 2009] es un sistema *Web* que ofrece recomendaciones y diferentes tipos de actividades, de las que se pueden destacar: ejemplos, tests, ejercicios de respuesta rápida (*short_text*) y colaborativos. Este sistema recomienda las actividades más adecuadas a cada estudiante no sólo teniendo en cuenta sus características personales, sino también considerando su contexto actual en el momento de la recomendación. Las recomendaciones se realizan en base a reglas estructurales, filtros generales de contexto y restricciones para ciertas actividades (véase el apéndice A.4.3, página 220, para mayor información).

Descripción de los logs

Los datos fueron proporcionados por la *EPS* de la Universidad Autónoma de Madrid (Madrid, España). Fueron recogidos durante el segundo cuatrimestre del curso 2007/08. Corresponden a las interacciones generadas por los estudiantes de la asignatura de “Sistemas Operativos I” (*SO1*), mientras realizaban actividades de aprendizaje, ofrecidas por el sistema *CoMoLE*, para reforzar sus conocimientos teóricos. El sistema realizó recomendaciones de acuerdo a: los estilos de aprendizaje, actividades que fueron completadas, resultados en anteriores actividades y el contexto (dispositivo utilizado, tiempo disponible, localización física y tiempo de recomendación). El número de estudiantes que realizó las actividades fue de 131, generando un total de 3610 interacciones.

Este sistema genera dos ficheros *XML* para cada estudiante: uno para almacenar el perfil del estudiante, y otro para guardar sus interacciones. Consecuentemente, con el objetivo de generar un “gran” fichero de logs de todos los estudiantes fue necesario procesar cada uno de los ficheros de cada estudiante, seleccionando las partes útiles para este experimento. Así, este gran fichero contiene información relativa a los perfiles de los estudiantes, sus contextos, las actividades realizadas en cada contexto y los resultados obtenidos (véase el apéndice A.5.2, página 229, para una descripción más detallada sobre los logs). Cada uno de los registros (entradas del fichero) del fichero de logs contienen las siguientes variables:

- Variables que describen el perfil: se almacenan las variables relativas al identificador del estudiantes y las componentes del estilo de aprendizaje (modelo de Felder-Silverman [Felder 1988]).
 - **IdUser**: identificador del estudiante en el sistema.
 - **Visual**: dimensión *visual/verbal* del modelo de estilos de aprendizaje de Felder-Silverman. Esta dimensión está relacionada con la forma en la que los contenidos son mostrados. Puede tener dos valores: “y” (indica mayor dimensión *visual* que *verbal*) y “n” (indica mayor dimensión *verbal* que *visual*).
 - **Sequential**: dimensión *sequential/global* del modelo de Felder-Silverman. Esta dimensión se refiere a la manera en la que los estudiantes entienden la información. Dos valores son posibles: “y” (indica mayor dimensión *sequential* que *global*) y “n” (indica mayor dimensión *global* que *sequential*).
 - **active**: dimensión *active/reflexive* del modelo anterior. Esta dimensión indica la manera preferida de los estudiantes para procesar la información. Dos valores están disponibles: “y” (más *active* que *reflexive*) y “n” (más *reflexive* que *active*).
 - **Sensitive**: dimensión *sensitive/intuitive* del modelo de Felder-Silverman. Esta dimensión indica la forma en la que los estudiantes perciben la información. Puede tener dos valores: “y” (más *sensitive* que *intuitive*) y “n” (más *intuitive* que *sensitive*).

- Variables respecto al contexto:
 - **Device:** tipo de dispositivo en el que las actividades son presentadas a los estudiantes. Dos tipos son posibles: *PC* y *PDA*.
 - **Location:** lugar en el que se encuentra el estudiante al realizar las actividades. Existen cuatro posibles lugares: *home*, *laboratory*, *class* y *others*.
 - **TimeUser:** tiempo del que dispone el estudiante para finalizar las actividades en la sesión. El estudiante fija este tiempo al principio de la sesión. Valores posibles para esta variable son los siguientes: 10 minutos, 20 minutos, 30 minutos, 1 hora, 1 – 2 horas y más de 2 horas.

- Variables que describen la actividad:
 - **NameAct:** identificador de la actividad de aprendizaje.
 - **Type:** tipo de actividad. Hay cinco tipos de actividades disponibles en este curso: teóricas (*theory*), ejemplos (*examples*), test de auto-evaluación (*self-assessment test*), ejercicios de respuesta rápida (*short_test answer exercise*) y actividades colaborativas (*collaborative activities*).
 - **finalization:** indica si la actividad se terminó o no. Dos valores son posibles: “F” (actividad finalizada) y “N” (actividad no finalizada).
 - **Score:** almacena el resultado obtenido en la actividad cuando ésta ha sido finalizada. Es una variable continua que toma valores entre 0 (peor puntuación que se puede obtener) y 10 (mejor puntuación).
 - **Grade:** variable generada de forma artificial a partir de la variable *score*. Esto quiere decir, que esta variable no es generada por *CoMoLE*. Puede recibir dos valores: *LOW* (si $0 \leq score < 5$) y *HIGH* (si $5 \leq score \leq 10$).

La tablas 5.16 y 5.17 muestran un extracto de la tabla de logs, correspondiente a tres entradas de tres estudiantes (*user23*, *user73*, *user90*) del fichero de logs. En la primera entrada (tabla 5.16) se muestra la información del estudiante *user23*, cuyo perfil es *visual* (valor “y” en la variable *visual*), *sequential* (valor “y” en la variable *sequential*), *active* (valor “y” en la variable *active*) y *sensitive* (valor “y” en la variable *sensitive*). También se puede observar en esta la primera fila de la tabla que este estudiante realizó la actividad en su casa (*location = home*) y utilizando su ordenador (*device = PC*). Además, este estudiante fijó un tiempo de 30 minutos para realizar las actividades. Las variables de la tabla 5.17 describen atributos de la actividad que realizó el estudiante. Se puede observar que la actividad que realizada fue *Mem_Test1*, es de tipo test, y fue finalizada sin éxito (valor “F” en *finalization* y valor “LOW” en *grade*). La siguiente entrada muestra que el estudiante *user73*, cuyo estilo de aprendizaje es visual, global, activo y reflexivo, realizó la actividad *MV_Test19* en su casa, utilizando su ordenador personal. Se puede ver que esta actividad es de tipo *short_text*, y fue finalizada sin éxito por el estudiante. La última entrada de la tabla muestra al estudiante *user90* con los mismos estilos de aprendizaje y contexto, pero finalizó la actividad *PagSimpleTest2* con éxito.

IdUser	Visual	Sequential	Active	Sensitive	Device	Location	TimeUser
user23	y	y	y	y	pc	home	30
user73	y	n	y	y	pc	home	10
user90	y	n	y	y	pc	home	10

Tabla 5.16: Ejemplos de logs en *CoMoLE* (1/2)

IdUser	NameAct	Type	Finalization	Score	Grade
user23	Mem_Test1	test	F	0,1	LOW
user73	MV_Test19	short_text	F	0,1	LOW
user90	PagSimpleTest2	test	F	10,0	HIGH

Tabla 5.17: Ejemplos de logs en *CoMoLE* (2/2)

Análisis mediante el método *GeSES*

Con el objetivo de analizar los resultados obtenidos de la aplicación del método *GeSES*, se siguieron los mismos pasos que en el experimento anterior. En primer lugar, se aplicó la propuesta inicial del método *GeSES* (sección 5.3.1.1, página 98). En segundo lugar, se analizaron los resultados obtenidos. En tercer lugar, se aplicaron las mejoras propuestas y se analizaron de nuevo los resultados.

La propuesta inicial del método consta de cinco fases:

(1) **Fase de limpieza.** Se seleccionaron las interacciones correspondientes al curso de 2007/08 en la asignatura *SO1*, y se diseñó la tabla de logs cuyo extracto se expone en las tablas 5.16 y 5.17. La tabla completa contiene 3610 filas (logs) y 13 columnas (variables). Esta tabla constituye los *datos de análisis*, y la *variable de evaluación* asignada fue la variable *grade*. La tabla 5.18 muestra un análisis estadístico de la distribución de los datos, mostrando cada variable y su frecuencia absoluta. Así, se muestra que la variable *visual* tiene dos valores posibles: “y” y “n”, siendo la frecuencia del primero de 3140, frente a 470. Dicho de otro modo, hay una mayor proporción de estudiantes visuales que verbales. La misma interpretación puede darse de las otras tres dimensiones del estilo de aprendizaje. Existe mayor número de estudiantes secuenciales que globales, mayor proporción de reflexivos que de activos y mayor proporción de sensitivos que intuitivos. Se observa que el dispositivo más utilizado fue el *PC*, que hubo tres tipos de localizaciones, aunque la más frecuente fue realizar las actividades en “casa”. Respecto a los tipos de actividades realizadas existen tres tipos: *test*, *short_text* y *collaborative*. Fundamentalmente, los estudiantes realizaron actividades de los dos primeros tipos. Para la variable *grade*, se observa que hubo mayor proporción de valores *LOW* que de *HIGH*, lo cual puede ser signo de situaciones problemáticas. El número de actividades disponibles fue de 46. Sus nombres aparecen en la última fila de la tabla 5.18. Se sigue la siguiente notación para indicar los nombres de las actividades: {nombre-actividad}({número-actividad-inicial}-{número-actividad-final}) o {nombre-actividad}{número-actividad}. Por ejemplo, el primer grupo de activida-

des es *PagSimpleTest(1-3)*, este nombre indica que las actividades disponibles son *PagSimpleTest1*, *PagSimpleTest2* y *PagSimpleTest3*. Otro ejemplo es la actividad *TablasRptaLibre1*, que es una única actividad.

Campo	Valor	Número de instancias
Visual	y	3140
	n	470
Sequential	y	2670
	n	940
Active	y	1364
	n	2246
Sensitive	y	3181
	n	429
Device	PC	3496
	PDA	114
Location	home	3042
	lab	68
	others	500
Type	test	2306
	short_text	1235
	collaborative	69
Grade	LOW	2799
	HIGH	811
nameAct	PagSimpleTest(1-3), PagSimpleRptaLibre(1-2), SegSimpleTest(1-2), Mem_Test(1-5), PSRptaLibre(1-2), MV_Test(1-19), MultiRptaLibre(1-2), PagMV-Ejer(1-7), SegM-VEjer, ColSO1, TablasRptaLibre1, TablasTest2	

Tabla 5.18: Distribución de los datos de análisis de los logs de *CoMoLE*

(2) **Generación de las reglas de decisión.** Se aplica el algoritmo *C4.5* a los *datos de análisis* con los siguientes parámetros:

- Atributos: *visual*, *sequential*, *active*, *sensitive*, *device*, *location*, *type*, *nameAct*.
- Variable de clasificación: *grade*.

Después de generar las *DR* con este algoritmo se obtuvieron 43 (13 para la clase *LOW* y 30 para la clase *HIGH*). El conjunto de reglas de decisión obtenido se puede examinar en el apéndice A.6.2, página 246. La tabla 5.19 muestra el *ranking* inicial obtenido con el programa *c4.5rules* al analizar los datos anteriores¹². Las reglas se seleccionan a partir de este *ranking* en las siguientes fases.

¹²En el apéndice se puede observar como el algoritmo genera un *ranking* inicial con las 43 reglas; sucesivamente se van descartando reglas (se descartan 5 reglas) hasta obtener el *ranking* final que consta de 38 reglas.

Rule	Size	Error	Used	Wrong	Advantage
DR112	1	2,0 %	69	0 (0,0 %)	69 (69 0) HIGH
DR22	3	33,8 %	7	1 (14,3 %)	5 (6 1) HIGH
DR32	3	37,0 %	3	0 (0,0 %)	3 (3 0) HIGH
DR56	2	37,0 %	3	0 (0,0 %)	3 (3 0) HIGH
DR88	5	37,0 %	3	0 (0,0 %)	3 (3 0) HIGH
DR31	3	42,2 %	52	19 (36,5 %)	14 (33 19) HIGH
DR46	3	45,4 %	5	1 (20,0 %)	3 (4 1) HIGH
DR47	3	50,0 %	2	0 (0,0 %)	2 (2 0) HIGH
DR101	3	50,0 %	2	0 (0,0 %)	2 (2 0) HIGH
DR55	3	51,3 %	11	4 (36,4 %)	3 (7 4) HIGH
DR72	3	52,5 %	17	7 (41,2 %)	3 (8 5) HIGH
DR74	1	53,9 %	79	39 (49,4 %)	1 (40 39) HIGH
DR16	5	54,7 %	4	1 (25,0 %)	2 (3 1) HIGH
DR5	4	54,9 %	49	24 (49,0 %)	1 (25 24) HIGH
DR10	2	55,5 %	56	28 (50,0 %)	0 (28 28) HIGH
DR12	3	55,7 %	6	2 (33,3 %)	2 (4 2) HIGH
DR51	2	55,9 %	10	4 (40,0 %)	2 (5 3) HIGH
DR79	2	58,5 %	76	41 (53,9 %)	-6 (35 41) HIGH
DR36	2	59,5 %	52	28 (53,8 %)	-4 (24 28) HIGH
DR52	6	61,2 %	9	4 (44,4 %)	1 (5 4) HIGH
DR70	4	61,2 %	5	2 (40,0 %)	1 (3 2) HIGH
DR68	2	61,3 %	69	39 (56,5 %)	-9 (30 39) HIGH
DR48	4	62,7 %	7	3 (42,9 %)	1 (4 3) HIGH
DR81	2	62,7 %	7	3 (42,9 %)	1 (4 3) HIGH
DR110	3	64,8 %	5	2 (40,0 %)	1 (3 2) HIGH
DR60	1	68,4 %	81	52 (64,2 %)	-23 (29 52) HIGH
DR50	2	68,4 %	23	14 (60,9 %)	-5 (9 14) HIGH
DR8	2	68,6 %	3	1 (33,3 %)	1 (2 1) HIGH
DR14	2	68,6 %	3	1 (33,3 %)	1 (2 1) HIGH
DR21	2	68,6 %	3	1 (33,3 %)	1 (2 1) HIGH
DR104	1	8,1 %	1225	85 (6,9 %)	0 (0 0) LOW
DR19	1	8,5 %	99	6 (6,1 %)	0 (0 0) LOW
DR30	1	8,5 %	99	6 (6,1 %)	0 (0 0) LOW
DR26	1	9,0 %	93	6 (6,5 %)	0 (0 0) LOW
DR15	2	11,8 %	11	0 (0,0 %)	0 (0 0) LOW
DR57	1	15,8 %	81	10 (12,3 %)	0 (0 0) LOW
DR62	1	17,1 %	81	11 (13,6 %)	0 (0 0) LOW
DR1	2	17,2 %	74	10 (13,5 %)	0 (0 0) LOW
DR54	2	17,8 %	418	93 (22,2 %)	0 (0 0) LOW
DR37	2	18,4 %	56	9 (16,1 %)	0 (0 0) LOW
DR7	1	19,7 %	95	17 (17,9 %)	0 (0 0) LOW
DR20	2	20,4 %	54	12 (22,2 %)	0 (0 0) LOW
DR45	3	22,3 %	27	4 (14,8 %)	0 (0 0) LOW

Tabla 5.19: *Ranking* de *DR* de los datos de análisis de *CoMoLE*

(3) **Seleccionar las reglas más relevantes para la característica de evaluación.** La variable de evaluación es *grade*. Como el objetivo el método es detectar signos de baja eficacia y el valor *LOW* en la variable *grade* indica que la actividad no se superó adecuadamente, se seleccionaron las reglas de decisión de la clase *LOW* (13 reglas). La tabla 5.20 muestra las reglas seleccionadas en esta fase. Se observa que la regla que más cobertura presenta es *DR104*, que es además la que menor error presenta. La regla *DR45* es la de mayor tamaño.

Rule	Size	Error	Used	Wrong	Advantage
DR104	1	8,1 %	1225	85 (6,9 %)	0 (0 0) LOW
DR19	1	8,5 %	99	6 (6,1 %)	0 (0 0) LOW
DR30	1	8,5 %	99	6 (6,1 %)	0 (0 0) LOW
DR26	1	9,0 %	93	6 (6,5 %)	0 (0 0) LOW
DR15	2	11,8 %	11	0 (0,0 %)	0 (0 0) LOW
DR57	1	15,8 %	81	10 (12,3 %)	0 (0 0) LOW
DR62	1	17,1 %	81	11 (13,6 %)	0 (0 0) LOW
DR1	2	17,2 %	74	10 (13,5 %)	0 (0 0) LOW
DR54	2	17,8 %	418	93 (22,2 %)	0 (0 0) LOW
DR37	2	18,4 %	56	9 (16,1 %)	0 (0 0) LOW
DR7	1	19,7 %	95	17 (17,9 %)	0 (0 0) LOW
DR20	2	20,4 %	54	12 (22,2 %)	0 (0 0) LOW
DR45	3	22,3 %	27	4 (14,8 %)	0 (0 0) LOW

Tabla 5.20: Reglas de decisión seleccionadas del *ranking* de *DR* de los datos de análisis de *CoMoLE*

(4) **Establecer la frontera de selección.** Se seleccionan las columnas (conceptos) *size*, *error* y *used* de la tabla 5.20. El resto de las columnas son redundantes, o no contienen información relevante para la fase siguiente. Para los conceptos seleccionados se calcularon las fronteras de selección: *F1* (diez por ciento) y *F2* (tercer cuartil). La tabla 5.21 muestra los valores de estas fronteras para cada concepto. Por ejemplo, el valor de *F1* para el concepto cobertura es de 241, pues la suma de todos los valores de esta columna es de 2413; por tanto, su diez por ciento redondeado es 241. Hay que indicar que, en el caso del concepto *error* la frontera *F2* es el primer cuartil, ya que es importante que la regla tenga el menor error posible.

(5) **Seleccionar las mejores reglas.** Esta fase consiste en asignar pesos a cada regla, aplicando la fórmula 5.5 (página 100), y seleccionar las que mayor peso total presenten. Es importante destacar que, el peso otorgado por cada frontera puede recibir un valor entre 0 y el peso máximo (valor entre 0 y 1). En este experimento, al igual que el anterior, se ha fijado el peso máximo en el valor 1, y se ha considerado que cada uno de los conceptos tenga la misma importancia, i.e., $s_{used} = s_{size} = s_{error} = 1$. Aplicando estos últimos valores a la fórmula 5.5, ésta se modifica de la siguiente manera:

Rule	Size	Error	Used
DR104	1	8,1 %	1225
DR54	2	17,8 %	418
DR19	1	8,5 %	99
DR30	1	8,5 %	99
DR7	1	19,7 %	95
DR26	1	9 %	93
DR57	1	15,8 %	81
DR62	1	17,1 %	81
DR1	2	17,2 %	74
DR37	2	18,4 %	56
DR20	2	20,4 %	54
DR45	3	22,3 %	27
DR15	2	11,8 %	11
\sum	20	194,6	2413
F1	2	19	241
F2	2	9	99
σ	0,63	4,9	315,04
σ_{r1}	–	–	99,54
σ_{r2}	–	–	27,53

Tabla 5.21: Fronteras de selección para los conceptos del *ranking* de *DR* de los datos de análisis de *CoMoLE*

$$W_r = \sum_{j \in \{1,2\}} w_{used}^{Fj} + w_{size}^{Fj} + w_{error}^{Fj}, \quad \text{siendo } w_c^{Fj} \in [0,1]$$

La tabla 5.22 muestra los pesos que obtiene cada regla respecto a cada una de las fronteras. Como se puede observar, en la tabla están señaladas en negrita las reglas de decisión que mayor peso total (W_r en la ecuación anterior) obtienen: *DR54* y *DR104*. Las siguientes mejores son: *DR19*, *DR30*, *DR1*, *DR37* y *DR15*. Las reglas más representativas son *DR54* y *DR104*, pero se produce un empate en el peso total en las cinco siguientes. Este empate es debido a que el sistema de selección propuesto en el método *GeSES* otorga el valor de 1 si la observación supera la frontera y 0 si no la supera. Por tanto, como el peso total es la suma de los pesos individuales, es probable la existencia de empates en el peso total. Además, observando la tabla 5.21 veremos que los pesos de las reglas *DR7* y *DR26* han sido infravalorados, ya que en el concepto cobertura reciben un peso de 0 (columnas 2 y 3 en la tabla 5.22), y sin embargo, se trata de observaciones muy cercanas, con valores de 95 y 93, a la frontera *F2*, con valor 99. La misma situación se puede observar para la regla *DR15*, que toma un valor de 11,8 en el concepto *error*. En este caso, el valor de *F2* para el concepto *error* es 9, que está por debajo¹³ del error de la regla *DR15*, y por esta

¹³Es importante que las reglas de decisión seleccionadas tengan el menor error posible, por eso

razón el peso respecto de la frontera $F2$ que recibe en este concepto es 0 (columna 7 en la tabla 5.22).

Rule	w_{used}^{F1}	w_{used}^{F2}	w_{size}^{F1}	w_{size}^{F2}	w_{error}^{F1}	w_{error}^{F2}	W_r
DR104	1	1	0	0	1	1	4
DR54	1	1	1	1	1	0	5
DR19	0	1	0	0	1	1	3
DR30	0	1	0	0	1	1	3
DR7	0	0	0	0	0	0	0
DR26	0	0	0	0	1	1	2
DR57	0	0	0	0	1	0	1
DR62	0	0	0	0	1	0	1
DR1	0	0	1	1	1	0	3
DR37	0	0	1	1	1	0	3
DR20	0	0	1	1	0	0	2
DR45	0	0	1	1	0	0	2
DR15	0	0	1	1	1	0	3

Tabla 5.22: Pesos otorgados por las fronteras para los conceptos del *ranking* de *DR* de los datos de análisis de *CoMoLE*

Con el objetivo de evitar los efectos de infravaloración producidos en las observaciones cercanas a alguna de las fronteras, y los empates en los pesos totales, se decidió realizar el experimento modificando la asignación de pesos mediante la aplicación de una asignación gradual por debajo de la frontera. Ya que el objetivo fue establecer una valoración para las observaciones en relación con la frontera, el peso debe ser proporcional a su distancia con la frontera. De esta forma, se considera que esta distancia debe estar en proporción con la dispersión de los datos, y por esta razón, se utilizó la desviación típica (σ) como medida de unidad de distancia. Por tanto, la función de asignación de pesos para cada una de las fronteras (fórmulas 5.6 y 5.7 - página 102) se puede resumir en:

$$\begin{aligned}
 w_c^F &= 1, & \text{si } x &\geq F \\
 w_c^F &= \frac{1}{2^n}, & \text{si } F - n \cdot \sigma &\leq x < F - (n - 1) \cdot \sigma
 \end{aligned}
 \tag{5.26}$$

Aplicando la fórmula 5.26 a los datos anteriores se obtienen nuevos pesos para cada concepto, que pueden verse en la tabla 5.23. Sin embargo, para el concepto cobertura (columna *used* en la tabla 5.21) se han utilizado σ_{r1} y σ_{r2} en vez de σ . La razón de elegir estos nuevos valores es porque el valor de σ , 315,04, es superior al valor de cualquiera de las fronteras (241 para *F1* y 99 para *F2*), por lo que

en el concepto *error* se asigna el peso máximo si la observación está por debajo de la frontera, y 0 si está por encima. En este caso, el valor 11,8 es claramente mayor que 9, por eso, su peso respecto de esta frontera es 0.

su aplicación generaría un solo intervalo en la formula anterior, y no se produciría el efecto de asignación de pesos graduales. Por eso, es necesario utilizar un valor reducido de σ hasta que sea inferior a las fronteras. Observando la tabla 5.21 (página 134), existe una observación, 1225, que es la que produce que σ sea mayor que las fronteras. Por tanto, la solución propuesta es eliminar esta observación para el cálculo de la desviación típica. Realizando de nuevo el cálculo de la desviación típica (σ_{r1}) se obtiene el valor de 99,54, que es inferior sólo a la frontera $F1$, y por tanto, se utilizará este valor para calcular los pesos respecto a $F1$. Como el valor de σ aún es superior a $F2$, se elimina de su cálculo la siguiente observación con mayor valor, que es 418 (cobertura de la regla $DR54$). El nuevo valor de σ para la frontera $F2$ es 27,53.

Nuevamente, se observa en la tabla 5.23 que las mejores reglas son: $DR54$, $DR104$, pero la siguiente mejor regla es $DR1$, seguida de las reglas $DR19$ y $DR30$. Claramente, la mejor sigue siendo la regla $DR54$, que es la que obtiene mayor peso total. Además, las dos reglas siguientes obtienen distintos pesos totales, por lo que no hay empates entre ellas. Además, se puede observar que la regla $DR26$ obtenía en el caso anterior una puntuación infravalorada, pues la distancia entre ésta y la tercera mejor ($DR1$) es de 1 para el sistema de asignación de pesos anterior, mientras que, en el actual sistema de pesos esta distancia es de 0,75.

Rule	w_{used}^{F1}	w_{used}^{F2}	w_{size}^{F1}	w_{size}^{F2}	w_{error}^{F1}	w_{error}^{F2}	W_r
DR104	1	1	0,25	0,25	1	1	4,5
DR54	1	1	1	1	1	0,25	5,25
DR19	0,25	1	0,25	0,25	1	1	3,75
DR30	0,25	1	0,25	0,25	1	1	3,75
DR7	0,25	0,5	0,25	0,25	0,5	0,13	1,88
DR26	0,25	0,5	0,25	0,25	1	1	3,25
DR57	0,25	0,5	0,25	0,25	1	0,25	2,5
DR62	0,25	0,5	0,25	0,25	1	0,25	2,5
DR1	0,25	0,5	1	1	1	0,25	4
DR37	0,25	0,25	1	1	1	0,25	3,75
DR20	0,25	0,25	1	1	0,5	0,13	3,13
DR45	0,13	0,13	1	1	0,5	0,13	2,88
DR15	0,13	0,06	1	1	1	0,5	3,69

Tabla 5.23: Pesos graduados otorgados por las fronteras para los conceptos del *ranking* de DR de los datos de análisis de *CoMoLE*

La figura 5.16 (página 137) muestra la gráfica de la función de asignación de pesos (fórmula 5.26) para el concepto cobertura respecto a la frontera $F2$. Teniendo en cuenta que se utiliza como desviación típica σ_{r2} , los pesos asignados respecto a la frontera $F2$ responden a:

$$\begin{aligned}
 71,47 \leq x < 99 &\Rightarrow w_{used}^{F2} = \frac{1}{2} \\
 55,06 \leq x < 71,47 &\Rightarrow w_{used}^{F2} = \frac{1}{4} \\
 16,41 \leq x < 55,06 &\Rightarrow w_{used}^{F2} = \frac{1}{8} \\
 0 \leq x < 16,41 &\Rightarrow w_{used}^{F2} = \frac{1}{16}
 \end{aligned}
 \tag{5.27}$$

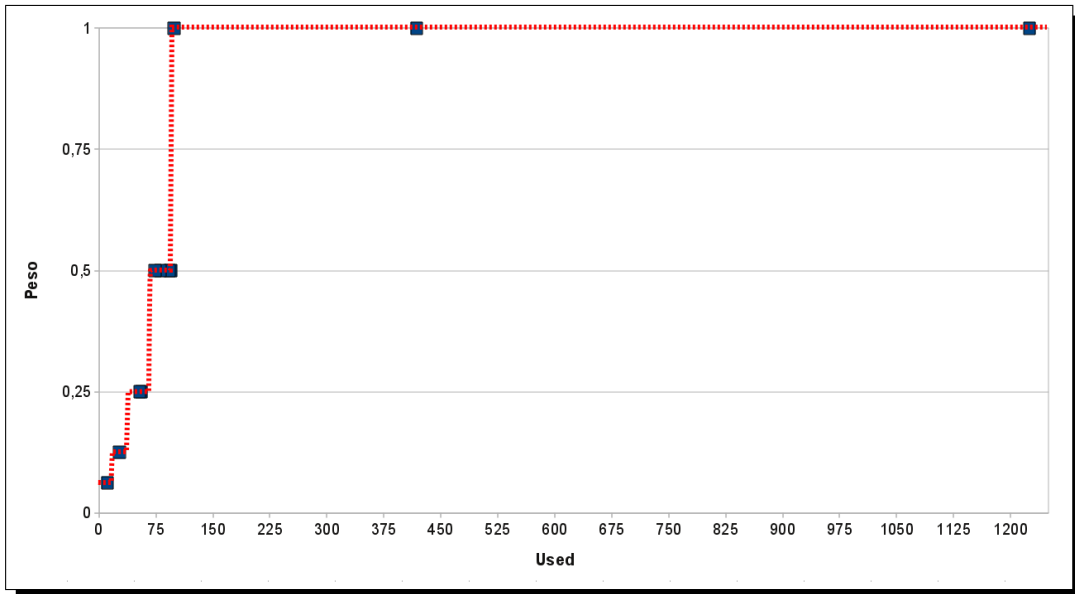


Figura 5.16: Pesos graduados otorgados por la frontera $F2$ a las reglas para el parámetro $used$

Observando la tabla 5.21 (página 134) veremos que las observaciones del concepto cobertura recibirán distintos pesos aplicando la ecuación 5.27. Esto indica que la asignación gradual de los pesos obtiene mejores resultados que la asignación dicotómica, pero aún es mejorable, ya que tres observaciones obtienen peso 1, y una de ellas (observación 1225 correspondiente a *DR104*) se encuentra muy alejada de las otras dos (ver parte superior de la figura 5.16). De igual forma que se utilizó una asignación gradual de pesos para las observaciones por debajo de la frontera, el siguiente paso es utilizar esta misma por encima de la frontera.

Al igual que en el experimento anterior, se probó con diversas funciones de asignación de pesos, como las funciones parabólicas, las basadas en deciles, y la función *sigmoidal*. Se decidió utilizar esta última realizando una serie de modificaciones sobre su fórmula (la fórmula general de la función sigmoidal se puede ver en la ecuación 5.8 - página 103) con el propósito de conseguir una asignación gradual por encima y por debajo de la frontera. La idea es conseguir que esta función otorgue pesos muy bajos

a las observaciones más alejadas por debajo de la frontera, y pesos suficientemente altos a las observaciones que se encuentren por encima.

Además, en esta fase fue necesario substituir el concepto *error* por el concepto *precision*, de forma que $precision = 100 - error$. La razón de esta substitución responde a que para el concepto *error* se necesita otro tipo de función, con características distintas de la *sigmoidal*. Ya que, reproducir las transformaciones realizadas en la función *sigmoidal* en esta nueva función es una tarea con cierta complejidad, se prefirió trabajar con el concepto opuesto al error, i.e., precisión.

Tal y como se explica en 5.3.1.4 (página 103), la fórmula que se utiliza finalmente para la asignación de pesos es la ecuación 5.18 (página 110). La figura 5.17 muestra la forma que tiene la función *sigmoidal* respecto al concepto cobertura para distintos valores del parámetro a , tomando la frontera $F2$ como referencia. En esta figura se aprecia que la función *sigmoidal-a16* (color azul claro) es la que peores valores asigna, pues el mínimo valor que toma es 0,49 y alcanza valores cercanos a 1 para valores de x por encima de 400. Claramente, ésta es la función que más lentamente crece, aunque su valor base es muy alto (0,49) para el propósito que se busca. Por tanto, esta función no parece adecuada, ya que las observaciones más alejadas por debajo de la frontera obtienen al menos un peso cercano a 0,5.

La siguiente función, *sigmoidal-aIQ* (color marrón), obtiene un valor mínimo más bajo que la anterior, pero aún no adecuado para el propósito buscado, pues, las observaciones más alejadas por debajo de la frontera obtendrían un peso cercano a 0,3. En consecuencia, esta función tampoco parece adecuada. De igual forma, la función *sigmoidal-a32* (color amarillo) tampoco es adecuada, ya que observaciones alejadas por debajo de la frontera obtendrían un peso mínimo por encima de 0,20.

Las funciones más adecuadas son las sigmoideas para los valores del parámetro a : a_{64} , a_{IQ2} y a_{46} , ya que asignan valores muy bajos, por debajo de 0,10, a las observaciones más alejadas por debajo de la frontera. Sin embargo, existen diferencias para las observaciones por encima de la frontera, pues la función *sigmoidal-a64* (color azul oscuro) crece más rápido que las otras dos. Pero, esta característica tampoco es deseable, ya que observaciones por encima de la frontera, y cercanas a ésta, obtendrán pesos demasiado altos, muy cercanos a los pesos obtenidos por las observaciones más alejadas por encima de la frontera. El nivel de crecimiento de las dos funciones restantes es razonable por encima de la frontera, por lo que para elegir una de las dos hay que observar qué ocurre debajo de ésta. La *sigmoidal-aIQ2* (coloreada en verde en la figura 5.17) asigna pesos ligeramente más bajos que la *sigmoidal-a46* (color rojo en la figura 5.17) por debajo de la frontera. Es por esta razón, por la que la función *sigmoidal-aIQ2* es la más adecuada para este concepto. Además, respecto al concepto precisión el valor para a_{46} es 0,02, valor muy bajo como ocurría en el experimento anterior. Este valor origina que la pendiente de la función sea muy pronunciada, tanto que lo que en realidad produce es un efecto dicotómico en la asignación. Esto así, porque esta función con este parámetro asigna valores superiores a 0 para valores de $x = 90,87$, es decir, un 70 % de las observaciones obtendrían peso 0. El valor del parámetro a_{IQ2} , que es 4,7, es más razonable que el anterior, y produce el efecto de graduación buscado. Por esta razón, el valor adecuado para el

parámetro a es a_{IQ2} .

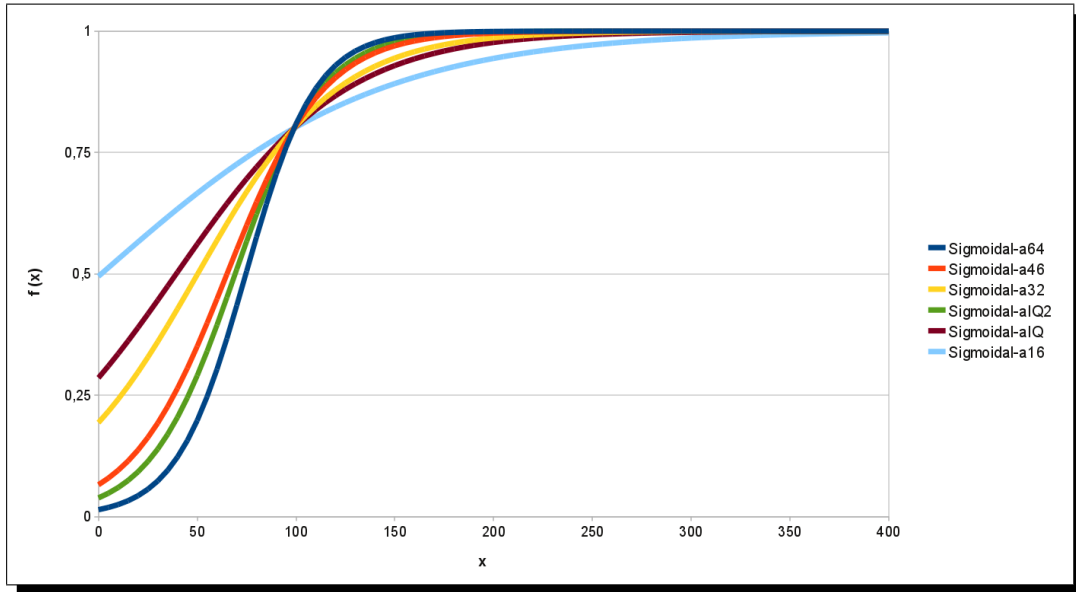


Figura 5.17: Funciones sigmoideas para distintos valores de la k para los datos de *CoMoLE*

En la tabla 5.24 se pueden ver los pesos que han sido obtenidos aplicando la fórmula 5.18 que utiliza $a = a_{IQ2}$.

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
DR104	1	0,35	0,83	2,18
DR54	1	0,8	0,38	2,18
DR19	0,8	0,35	0,82	1,97
DR30	0,8	0,35	0,82	1,97
DR7	0,77	0,35	0,29	1,41
DR26	0,75	0,35	0,8	1,9
DR57	0,63	0,35	0,48	1,47
DR62	0,63	0,35	0,42	1,4
DR1	0,56	0,8	0,41	1,77
DR37	0,35	0,8	0,35	1,5
DR20	0,33	0,8	0,26	1,39
DR45	0,12	0,97	0,19	1,28
DR15	0,06	0,8	0,69	1,55

Tabla 5.24: Pesos otorgados por la función sigmoideal modificada para los datos de análisis de *CoMoLE*

En el experimento anterior se encontró el problema de que el *IQ* de uno de los

conceptos sea cero. Esto es fácil que ocurra en el caso del tamaño de las reglas. Sin embargo, en este experimento no se detectó este problema, pues hay mayor variabilidad entre los tamaños. Por lo tanto, se pudo aplicar la función sigmoïdal en todos los conceptos.

En la tabla 5.24 se puede observar que las reglas de decisión con mayor peso son *DR104* y *DR54*, seguidas de las reglas *DR30* y *DR19*. Se observa que la regla *DR1*, que anteriormente había sido seleccionada, no se encuentra entre las tres mejores. Sin embargo, la regla *DR26* que anteriormente se encontraba a distancia de 0,75 de ser seleccionada, ahora se encuentra tan sólo a 0,07 de la tercera mejor regla¹⁴. Hay que tener en cuenta que el sistema de pesos puntúa sobre 3, esto quiere decir que el mayor peso total posible es 3. En concreto, el mayor peso total se registra en las reglas *DR104* y *DR54* con una puntuación de 2,18. En este experimento se observa que también cobra especial relevancia el grado de importancia de los conceptos (s_c), como ocurría en el anterior experimento. Es importante recordar que al principio del experimento se decidió que todos los conceptos tuvieran el mismo grado de importancia, pero si la importancia del concepto tamaño fuera mayor que la del concepto precisión, la regla *DR1* sería seleccionada en lugar de las reglas *DR19* y *DR30*.

Rule 104:
<pre> tipo = short_text ->class LOW [91,9 %] </pre>
Rule 54:
<pre> active = y location = home ->class LOW [82,2 %] </pre>
Rule 19:
<pre> nameAct = Mem_Test2 ->class LOW [91,5 %] </pre>
Rule 30:
<pre> nameAct = MV_Test1 ->class LOW [91,5 %] </pre>

Tabla 5.25: Reglas más relevantes respecto a la característica de evaluación para los datos de análisis de *CoMoLE*

La tabla 5.25 muestra las reglas más relevantes: *DR104* y *DR54*. La primera indica que la mayoría de los estudiantes que realizaron actividades de tipo *short_text* obtuvieron baja puntuación en estos ejercicios. Y la segunda indica que la mayoría de

¹⁴Estas distancias no son comparables de forma absoluta, ya que en el *ranking* gradual el peso total máximo que puede obtener una regla es 6, mientras que utilizando la función sigmoïdal este valor máximo se sitúa en 3. No obstante, la distancia relativa para el primer caso es 0,125 (0,75/6), mientras que para el segundo 0,023 (0,07/3). Es claro, que la regla *DR26* ha mejorado su distancia respecto a ser seleccionada.

estudiantes activos que realizaron sus actividades en casa, presentaron dificultades en la mayoría de actividades. Estas dos reglas contienen información interesante desde el punto de vista de los profesores, ya que les alerta de que es posible que exista un problema con las actividades de tipo *short_text*. Por tanto, es posible que estas actividades no fueran adecuadas para los estudiantes activos que trabajaron en casa. Las dos siguientes reglas indican que pueden existir problemas en actividades concretas. En el caso de la regla *DR19*, se indica que la actividad *Mem_Test2* presentó problemas para la mayoría de estudiantes, pues no existe información sobre el perfil de éstos. Las conclusiones son las mismas para la última regla, pero referidas a la actividad *MV_Test1*.

Adicionalmente, se incluye la tabla 5.26 para mostrar las dos reglas que podían haber sido seleccionadas. Por tanto, como se puede observar la información expuesta en el párrafo anterior puede ser especialmente útil para el profesor, pues de forma precisa puede conocer las actividades en las que mostraron más dificultades los estudiantes, el tipo de actividades con más problemas, así como el perfil o perfiles de estudiantes que los experimentaron.

Rule 26:
nameAct = Mem_Test5
->class LOW [91.0 %]
Rule 1:
location = home
nameAct = PagSimpleTest1
->class LOW [82.8 %]

Tabla 5.26: Reglas cercanas a las más relevantes para los datos de análisis de *Co-MoLE*

La tabla 5.27 (tabla de signos) presenta un resumen de los signos detectados con el método *GeSES* (sección 5.3.1.5). Esta información sería la que se mostraría al profesor, pues de esta manera éste puede entender qué problemas han sido detectados sin tener grandes conocimientos de *DM*. De esta forma, el profesor consultando esta tabla puede conocer fácilmente los problemas que presentan los estudiantes y el contexto en el que se desarrollan estas situaciones problemáticas. Hay que señalar, que el término “cualquier” se utiliza para indicar que no existe información relativa a los perfiles en la regla de decisión asociada a la fila de la tabla. Este tipo de situaciones se producen más frecuentemente en las reglas de menor tamaño.

Estos dos experimentos han mostrado la aplicación de la propuesta inicial del método *GeSES* y las pruebas realizadas para perfeccionar las dos últimas fases del método. Se han mostrado diversas opciones para conseguir el objetivo buscado, a fin de mejorar el método en sí. En cada uno de los dos experimentos se han expuesto las mejoras llevadas a cabo en el método, así como sus puntos débiles.

Perfil	Actividad
cualquier	tipo = short_text
active = y location = home	cualquier
cualquier	nameAct = Mem_Test2
cualquier	nameAct = MV_Test1

Tabla 5.27: Signos detectados para los datos de análisis de *CoMoLE*

5.4. Conclusiones

Aunque en el capítulo 8 se expondrán las conclusiones principales de los resultados de esta investigación, en esta sección se exponen las conclusiones relativas a las cuestiones analizadas anteriormente.

El objetivo que se ha perseguido en la presentación de este capítulo fue mostrar al lector las etapas seguidas por el autor para conseguir formular la propuesta final del método *GeSES*, así como las dificultades encontradas en la realización de la presente investigación. Por este motivo, se ha expuesto el proceso llevado a cabo para obtener un método de detección eficaz de los signos expuestos en el capítulo 4. Se partió de un método inicial, como es el método *Key-node*, hasta llegar a la propuesta final del método *GeSES*. Asimismo, se han explicado y discutido los puntos débiles en el método inicial, cuya mejora constituyó la base de la propuesta inicial del método *GeSES*. Esta propuesta inicial fue analizada de forma exhaustiva mediante la realización de dos experimentos con datos de distintas fuentes, en los que se realizaron diversas mejoras a sus puntos delicados, y fueron aplicadas de forma progresiva. Ambos experimentos demostraron su utilidad, ya que los problemas encontrados en las fases 4 y 5 del método fueron diferentes, y permitieron mejorar sus resultados.

Como se ha visto en los experimentos realizados, las reglas de decisión seleccionadas por el método de selección, basado en la función de pesos, pueden tener un tamaño pequeño, lo que se traduce en pérdida de información en la tabla de signos, y la consecuente aparición del término “cualquier” en dicha tabla. Sin embargo, este problema es subsanable asignando mayor importancia al concepto tamaño frente a los conceptos precisión y cobertura. No obstante, la información contenida en una regla con tamaño pequeño puede ser relevante para el profesor, por lo que sería conveniente que esta figura fije el grado de importancia de cada concepto. Por ejemplo, asignar mayor importancia al tamaño de la regla significa que las reglas seleccionadas contendrán mayor información (las reglas son más expresivas), es decir, que la información será más completa. Por otro lado, si la prioridad es la precisión, las reglas seleccionadas serán más precisas, es decir, la información tendrá menor error. Y por último, si el concepto cobertura es prioritario, se obtendrán reglas más representativas, es decir, la información contenida será más extrapolable.

El método *GeSES* se apoya fuertemente en las *DR*, técnica cuya eficacia ha sido probada ampliamente en la literatura. Sin embargo, aunque se han realizado

diversos experimentos en este capítulo y se han mostrado los resultados obtenidos, parece conveniente evaluar los límites que presenta este método, es decir, conocer qué signos incluidos en los logs deberían haber sido detectados, y por qué no fueron detectados. El siguiente capítulo comprende la aplicación de la propuesta final del método *GeSES*, expuesta en la sección 5.3.1.5, con un nuevo conjunto de datos y el procedimiento seguido para la evaluación del método.

Aplicación y Evaluación de la propuesta

Índice de contenidos

6.1. Introducción	145
6.2. Aplicación de la propuesta	146
6.2.1. Análisis inicial de los datos	146
6.2.2. Análisis mediante el método <i>GeSES</i>	147
6.3. Procedimiento de evaluación	154
6.4. Introducción a <i>SimuLog</i>	155
6.5. Evaluación	155
6.5.1. Preparación de los datos (logs)	156
6.5.2. Análisis de los datos mediante el método <i>GeSES</i>	157
6.6. Conclusiones	168

Este capítulo presenta la forma de aplicar y evaluar el método GeSES. En primer lugar se presenta la aplicación del método con un nuevo conjunto de datos, y en segundo lugar el procedimiento de evaluación efectuado. Dicho procedimiento, grosso modo consiste en utilizar logs sintéticos con unas determinadas características para analizar la eficacia del método.

6.1. Introducción

Este capítulo se divide en dos partes claramente diferenciadas: aplicación del método *GeSES* y evaluación de la eficacia de este método. En la primera parte, se expone la aplicación de la propuesta con un nuevo conjunto de datos, diferente de los utilizados hasta este momento en la investigación. El objetivo que se persigue es mostrar la forma de aplicar el método y la utilidad de los resultados obtenidos. La segunda parte de este capítulo contiene el procedimiento seguido para evaluar la eficacia del método. En síntesis, el procedimiento consiste en generar logs sintéticos con determinados signos en determinadas condiciones, aplicar el método *GeSES* a estos logs, y analizar si se han detectado los signos generados de forma artificial.

6.2. Aplicación de la propuesta

En esta sección se presenta la aplicación del método *GeSES* en un nuevo conjunto de datos. Este conjunto corresponde a las interacciones recogidas en el sistema *CoMoLE* de los estudiantes de la *EPS* en la Universidad Autónoma de Madrid durante el curso académico 2007/08.

Estos estudiantes realizaron una serie de actividades de aprendizaje, suministradas por *CoMoLE* en el contexto de la asignatura “Estructura de Datos y de la Información I” (EDI1). La estructura de los logs de las interacciones, así como de los datos de los perfiles de los usuarios es muy similar a la de los datos recogidos en la sección 5.3.3.2 en el capítulo 5. En este caso, también existe información en el perfil del estudiante sobre el estilo de aprendizaje (se utiliza el modelo de Felder-Silverman), y en los datos de interacción existe información sobre el tipo de dispositivo utilizado, localización del estudiante, tipo de actividad, estado de la actividad, nombre de la actividad, tiempo del que se dispone y puntuación en la actividad. Puesto que el método *GeSES* requiere que se defina una *variable de evaluación*, se creó una variable dicotómica a partir de la variable “puntuación en la actividad” (*score*), y se la denominó *grade*. Así, su valor es el siguiente:

$$\begin{aligned} \textit{grade} &= \textit{LOW}, \text{ si } 0 \leq \textit{score} < 5 \\ \textit{grade} &= \textit{HIGH}, \text{ si } 5 \leq \textit{score} \leq 10 \end{aligned}$$

Hay que indicar que, previo a la aplicación del método, es importante tener una visión global de los datos. Por esta razón, en el siguiente apartado se presenta un análisis inicial de los datos, seguido de la aplicación del método *GeSES*.

6.2.1. Análisis inicial de los datos

Los pasos seguidos en la creación de los *datos de análisis* son los mismos que se siguieron en el experimento en el que se analizaron los logs del curso de *SO1* en el sistema *CoMoLE*. Un extracto de la tabla final de logs se exhibe en el apéndice A.5.3.

El análisis inicial de esta tabla refleja que el sistema registró un total de 2309 interacciones, correspondientes a 91 estudiantes. La tabla consta de 11 variables o columnas: *visual*, *sequential*, *active*, *sensitive*, *device*, *location*, *type*, *finalization*, *timeUser*, *nameAct* y *grade*. Con el objetivo de mejorar el conocimiento de los datos se realizó un análisis estadístico descriptivo de cada una de estas variables. La tabla 6.1 muestra las variables, los valores que pueden tomar y las frecuencias absolutas de estos valores. Este análisis muestra que la proporción de estudiantes visuales es mayor que los verbales, la proporción entre secuenciales y globales es muy similar, lo mismo ocurre con los sensitivos respecto de los intuitivos, y los estudiantes reflexivos están en mayor proporción que los activos. Además, se puede observar que el dispositivo más utilizado fue el ordenador personal, principalmente fue utilizado en casa del estudiante, y la mayor parte de las actividades fueron finalizadas. Respecto al

tipo de actividad, las actividades de tipo *test* y *short_text* tienen mayor frecuencia respecto al otro tipo. La variable *grade* muestra proporciones muy similares entre las puntuaciones bajas (valor *LOW*) y las altas (valor *HIGH*). Finalmente, respecto a la variable tiempo disponible, la mayor frecuencia se registra cuando esta variable es 60 minutos, lo que quiere decir que un amplio número de estudiantes indicó una disponibilidad de tiempo de 60 minutos.

6.2.2. Análisis mediante el método *GeSES*

Con el objetivo de demostrar la aplicación de la propuesta final de este método, se presenta a continuación la manera en la que se realizaron las fases del método:

(1) **Fase de limpieza.** En esta fase fue necesaria la construcción de una tabla final de logs, agrupando los datos de los perfiles de los estudiantes y sus datos de interacción. Se asignó como *variable de evaluación* la variable *grade*. De esta forma, los logs junto con la *variable de evaluación* forman los *datos de análisis*. Un extracto de los logs analizados se puede ver en el apéndice A.5.3 (página 238), donde se exponen los ficheros utilizados para la segunda fase de este método.

(2) **Generación de las reglas de decisión.** Se aplica el algoritmo *C4.5* a los *datos de análisis* para obtener las reglas de decisión:

- Atributos: *visual, sequential, active, sensitive, device, location, type, finalization, timeUser, nameAct*.
- Variable de clasificación: *grade*.

Se generaron un total de 84 reglas de decisión (véase el apéndice A.6.3, página 258, para examinar el conjunto de reglas obtenido). De estas reglas, las pertenecientes a la clase *LOW* fueron 32, y las pertenecientes a la clase *HIGH* fueron 52.

(3) **Seleccionar las reglas más relevantes para la característica de evaluación.** Como la variable *grade* indica si la puntuación fue alta o baja en una actividad, y las puntuaciones bajas pueden ser signo de dificultades en la realización de las actividades, se seleccionan las reglas de la clase *LOW*. Por tanto, en esta fase se seleccionaron 32 reglas de decisión, expuestas en la tabla 6.2 (p. 149). Se observa en esta tabla que la regla que presenta mayor cobertura es *DR65*, la que contiene menor error es *DR166*, y las que mayor tamaño tienen son las reglas *DR1*, *DR205* y *DR25*.

(4) **Establecer la frontera de selección.** Se seleccionan las columnas *size*, *error* y *used* de la tabla anterior. Seguidamente se calcula el tercer cuartil, *frontera de selección*, de cada una de ellas. La tabla 6.3 (p. 150) presenta a los conceptos seleccionados, así como el valor de la *frontera de selección* (denotado por **F** en la tabla) para cada uno. Se puede observar en esta tabla, que como se explicó en el capítulo 5, para el cálculo de la frontera del concepto *error* se ha utilizado el primer cuartil en lugar del tercero.

(5) **Seleccionar las mejores reglas.** Esta fase está dividida en tres etapas:

Campo	Valor	Número de instancias
Visual	y	1942
	n	367
Sequential	y	1335
	n	974
Active	y	813
	n	1496
Sensitive	y	2063
	n	246
Device	PC	2286
	PDA	23
Location	home	1932
	lab	189
	others	188
Type	test	1200
	short_text	834
	review	275
Finalization	F	2297
	N	12
TimeUser	10	353
	20	272
	30	280
	60	557
	120	391
	240	456
Grade	LOW	1119
	HIGH	1190
nameAct	EnumTest(1-2), OperadoresTest(1-3), OperadoresRptaLibre(1-5), IfTest, SwitchRptaLibre(1-2), CondReview, IfEjemRC, IfTestRC, SwitchRptaLibre1RC, SwitchRptaLibre2RC, WhileTest, DoWhileTest, ForTest, DoWhileRptaLibre, ESTest(1-3), EstructurasRptaLibre, NuevosDatosTest, FuncTest(1-3), PasoArgumentosRptaLibre(1-2), ArraysTest(1-7), ArraysRptaLibre(1-3), MatricesRptaLibre, PunterosTest(1-5), PunterosRptaLibre(1-4)	

Tabla 6.1: Distribución de los datos de análisis de los logs de CoMoLE en el curso de EDII

Rule	Size	Error	Used	Wrong	Class
DR65	4	23,30 %	114	20 (17,5 %)	LOW
DR157	1	16,40 %	78	10 (12,8 %)	LOW
DR166	1	5,00 %	76	2 (2,6 %)	LOW
DR1	5	39,30 %	67	20 (29,9 %)	LOW
DR142	1	20,20 %	58	7 (12,1 %)	LOW
DR205	5	22,20 %	50	8 (16,0 %)	LOW
DR179	3	28,80 %	46	11 (23,9 %)	LOW
DR47	1	5,20 %	46	1 (2,2 %)	LOW
DR170	3	32,40 %	38	6 (15,8 %)	LOW
DR196	3	21,00 %	34	5 (14,7 %)	LOW
DR69	2	37,70 %	27	8 (29,6 %)	LOW
DR224	1	29,20 %	26	6 (23,1 %)	LOW
DR149	1	31,90 %	20	4 (20,0 %)	LOW
DR229	1	18,50 %	20	2 (10,0 %)	LOW
DR230	1	13,30 %	19	1 (5,3 %)	LOW
DR22	4	38,50 %	18	5 (27,8 %)	LOW
DR59	4	46,00 %	16	6 (37,5 %)	LOW
DR228	1	33,30 %	15	5 (33,3 %)	LOW
DR182	3	23,90 %	15	4 (26,7 %)	LOW
DR152	1	40,80 %	12	3 (25,0 %)	LOW
DR60	4	30,00 %	12	2 (16,7 %)	LOW
DR33	3	45,60 %	10	3 (30,0 %)	LOW
DR85	2	22,30 %	10	1 (10,0 %)	LOW
DR114	3	30,00 %	9	2 (22,2 %)	LOW
DR218	2	22,30 %	8	1 (12,5 %)	LOW
DR132	3	15,90 %	8	0 (0,0 %)	LOW
DR78	3	18,00 %	7	0 (0,0 %)	LOW
DR130	3	18,00 %	6	0 (0,0 %)	LOW
DR219	2	37,00 %	3	0 (0,0 %)	LOW
DR139	3	37,00 %	3	0 (0,0 %)	LOW
DR25	5	24,20 %	3	0 (0,0 %)	LOW
DR103	2	24,20 %	3	0 (0,0 %)	LOW

Tabla 6.2: Reglas de decisión seleccionadas de los datos del curso de *EDI1* en *Co-MoLE*

Rule	Size	Error	Precision	Used
DR65	4	23,30 %	76,7	114
DR157	1	16,40 %	83,6	78
DR166	1	5,00 %	95	76
DR1	5	39,30 %	60,7	67
DR142	1	20,20 %	79,8	58
DR205	5	22,20 %	77,8	50
DR179	3	28,80 %	71,2	46
DR47	1	5,20 %	94,8	46
DR170	3	32,40 %	67,6	38
DR196	3	21,00 %	79	34
DR69	2	37,70 %	62,3	27
DR224	1	29,20 %	70,8	26
DR149	1	31,90 %	68,1	20
DR229	1	18,50 %	81,5	20
DR230	1	13,30 %	86,7	19
DR22	4	38,50 %	61,5	18
DR59	4	46,00 %	54	16
DR228	1	33,30 %	66,7	15
DR182	3	23,90 %	76,1	15
DR152	1	40,80 %	59,2	12
DR60	4	30,00 %	70	12
DR33	3	45,60 %	54,4	10
DR85	2	22,30 %	77,7	10
DR114	3	30,00 %	70	9
DR218	2	22,30 %	77,7	8
DR132	3	15,90 %	84,1	8
DR78	3	18,00 %	82	7
DR130	3	18,00 %	82	6
DR219	2	37,00 %	63	3
DR139	3	37,00 %	63	3
DR25	5	24,20 %	75,8	3
DR103	2	24,20 %	75,8	3
Σ	81	851,4	2348,6	877
F	3	19,78	80,23	40
Q₁	1	19,78	65,78	8,75
Q₃	3	34,23	80,23	40
σ	1,3	10,33	10,33	26,47

Tabla 6.3: Reglas y valores de la frontera de selección para los datos del curso de *EDI1* en *CoMoLE*

- (5.1) Generación de la precisión. Se genera el parámetro *precision* a partir del parámetro *error* mediante la fórmula $precision = 100 - error$. En la tabla anterior se pueden ver los valores de este nuevo concepto, el valor de su frontera (80,23), el tercer (80,23) y primer cuartil (65,78) y la desviación típica (10,33). Además, se observa que las desviaciones típicas de *error* y *precision* son iguales, lo cual quiere decir que la distribución de los datos no cambia, lo que es deseable para la siguiente etapa.
- (5.2) Asignación de pesos a las reglas. Se utiliza la ecuación 5.22 (página 113) para calcular el peso de cada uno de los conceptos (fijando el peso máximo al valor 1), y la ecuación 5.23 (página 113) para calcular el peso total de cada regla de decisión. Para este último cálculo se tomó la decisión de asignar a cada concepto la misma importancia. Consecuentemente, la fórmula utilizada para el cálculo de los pesos totales es:

$$W_r = w_{used} + w_{size} + w_{precision}$$

Por tanto, el peso total máximo que puede tener una regla de decisión es 3, y el peso total mínimo es 0. La tabla 6.4 muestra el valor de los pesos para cada concepto y los pesos totales de las reglas de decisión¹.

- (5.3) Selección de las reglas. De la tabla anterior se deduce que las reglas de mayor peso total son: *DR65* (con peso total 2,62), *DR205* (con 2,59), *DR166* (con 2,3) y *DR196* (con 2,3).

La tabla 6.5 muestra las reglas de decisión más representativas respecto de la *característica de evaluación* elegida. En esta tabla se observa que la regla *DR65* hace referencia a que los estudiantes globales y reflexivos que dispusieron más de 60 minutos para realizar las actividades, tuvieron dificultades al realizar las actividades de tipo *test*. La regla *DR205* indica que los estudiantes con estilo de aprendizaje visual y global, que trabajaron desde su casa, y que dispusieron de más de 60 minutos para realizar las actividades, presentaron dificultades en las actividades de tipo *short_text*. Por otro lado, las dos reglas siguientes muestran información sobre grupos de estudiantes que mostraron problemas en dos actividades concretas. Dichas actividades son *OperadoresRptaLibre3* y *DoWhileRptaLibre*. Sin embargo, en la regla *DR166* no existe información sobre el grupo de estudiantes que tuvo problemas en esta actividad, pero sí existe esta información para la otra actividad en la otra regla. La regla *DR196* indica que los estudiantes que trabajaron en sus

¹Los valores mostrados en esta tabla fueron calculados con el programa *Calc*, y para mejorar la claridad se decidió redondear los números a 2 cifras decimales. Sin embargo, el lector puede apreciar que los valores de la última columna no coinciden con la suma exacta de los valores redondeados. Esto es así, porque los cálculos se realizan internamente con los valores sin redondear, y después se realiza el redondeo. Así, por ejemplo, puede parecer que el cálculo de W_r en la primera fila es erróneo, y no es así, ya que los valores (redondeados a 4 decimales) de w_{used} , w_{size} y $w_{precision}$ son 0,9978, 0,9158 y 0,7106 respectivamente. Por tanto, sumando estos tres valores se obtiene 2,6242, que redondeado a 2 decimales es 2,62, valor que aparece en la tabla.

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
DR65	1	0,92	0,71	2,62
DR157	0,98	0,35	0,86	2,19
DR166	0,98	0,35	0,97	2,3
DR1	0,96	0,97	0,21	2,14
DR142	0,93	0,35	0,79	2,07
DR205	0,88	0,97	0,74	2,59
DR179	0,85	0,8	0,53	2,19
DR47	0,85	0,35	0,97	2,17
DR170	0,78	0,8	0,41	1,99
DR196	0,73	0,8	0,77	2,3
DR69	0,64	0,6	0,25	1,48
DR224	0,62	0,35	0,52	1,49
DR149	0,53	0,35	0,43	1,31
DR229	0,53	0,35	0,83	1,7
DR230	0,51	0,35	0,91	1,77
DR22	0,49	0,92	0,23	1,64
DR59	0,46	0,92	0,1	1,47
DR228	0,45	0,35	0,38	1,18
DR182	0,45	0,8	0,69	1,94
DR152	0,4	0,35	0,18	0,93
DR60	0,4	0,92	0,49	1,81
DR33	0,37	0,8	0,1	1,27
DR85	0,37	0,6	0,74	1,7
DR114	0,35	0,8	0,49	1,65
DR218	0,34	0,6	0,74	1,67
DR132	0,34	0,8	0,87	2,01
DR78	0,33	0,8	0,84	1,96
DR130	0,31	0,8	0,84	1,95
DR219	0,27	0,6	0,27	1,14
DR139	0,27	0,8	0,27	1,34
DR25	0,27	0,97	0,68	1,92
DR103	0,27	0,6	0,68	1,55

Tabla 6.4: Pesos totales para las reglas de los datos del curso de *EDI1* en *CoMoLE*

Rule 65:
sequential = n
active = n
timeUser >60
type = test
->class LOW [76,7 %]
Rule 205:
visual = y
sequential = n
location = home
timeUser >60
type = short_text
->class LOW [77,8 %]
Rule 166:
nameAct = OperadoresRptaLibre3
->class LOW [95,0 %]
Rule 196:
location = home
timeUser <= 120
nameAct = DoWhileRptaLibre
->class LOW [79,0 %]

Tabla 6.5: Reglas más relevantes respecto a la característica de evaluación para los datos de análisis de *EDI1* en *CoMoLE*

casas y dispusieron como mucho de 120 minutos tuvieron muchas dificultades en la actividad *DoWhileRptaLibre*. No obstante, no existe información sobre el tipo de estilo de aprendizaje que presentó problemas en este ejercicio, por lo cual se asume que el problema puede ser independiente del mismo.

La información explicada en el párrafo anterior está resumida en la tabla 6.6. Se puede observar en esta tabla que dos actividades concretas requieren revisión por parte del profesor: *OperadoresRptaLibre3* y *DoWhileRptaLibre*. El término “cualquier” se utiliza en esta tabla para indicar que no existe información en la regla de decisión, asociada a la fila, sobre los perfiles. En particular, la regla *DR166* no contiene información sobre los perfiles, pues su tamaño es 1, pero sí contiene información sobre el nombre de la actividad. Por esta razón, en el perfil de la penúltima fila se indica que “cualquier” perfil puede presentar problemas con la actividad *OperadoresRptaLibre3*. Nótese que cuanto mayor sea el tamaño de la regla, menor es la posibilidad de pérdida de caracterización del signo, i.e., que aparezca el término “cualquier” en la tabla de signos.

Además, la tabla proporciona información sobre los tipos de actividades que presentaron mayores dificultades y los perfiles relacionados con éstas. En este caso, las actividades de tipo *test* y *short_text* fueron más problemáticas. Respecto a las

Perfil	Actividad
sequential = n active = n timeUser >60	type = test
visual = y sequential = n location = home timeUser >60	type = short_text
cualquier	nameAct = OperadoresRptaLibre3
location = home timeUser <= 120	nameAct = DoWhileRptaLibre

Tabla 6.6: Signos detectados para los datos de análisis del curso *EDI1* en *CoMoLE*

actividades de tipo *test*, existe información sobre los perfiles que mostraron mayores dificultades al realizarlas, que son los estudiantes globales y reflexivos. En relación con las actividades de tipo *short_text* la tabla de signos contiene información sobre el perfil, localización y tiempo disponible de los estudiantes que mostraron dificultades al realizar este tipo de actividades.

Como se ha podido observar, la información presentada en la tabla de signos proporciona una manera rápida y eficaz para los profesores de detectar aquellas actividades y/o perfiles que mostraron mayores dificultades, o que requieren mayor atención por parte de ellos.

6.3. Procedimiento de evaluación

Esta sección expone una forma de valorar si los resultados obtenidos en la aplicación del método *GeSES* son fiables. Por tanto, el procedimiento que se ha seguido trata de responder a dos cuestiones:

- ¿Los signos detectados están realmente presentes en los logs analizados?
- ¿Existe algún signo presente en los logs que no fue capaz de detectar el método?

Con el objetivo de dar respuesta a estas cuestiones se propone utilizar un enfoque sintético, consistente en la generación de forma automática de logs mediante el uso de técnicas de simulación. Con este fin, se utilizó la herramienta *SimuLog*, desarrollada por el autor en el transcurso de esta investigación. La figura 6.1 muestra el procedimiento de evaluación seguido para valorar la eficacia del método. En esta figura, el evaluador (persona que evalúa la eficacia del método) especifica los signos a generar (representados en la figura por **ST**) y las características o atributos de los estudiantes simulados (representados por **PE**). *SimuLog* utiliza estos parámetros de entrada junto con la descripción del curso para generar los logs sintéticos. Los logs generados contendrán en su interior los signos solicitados previamente por el

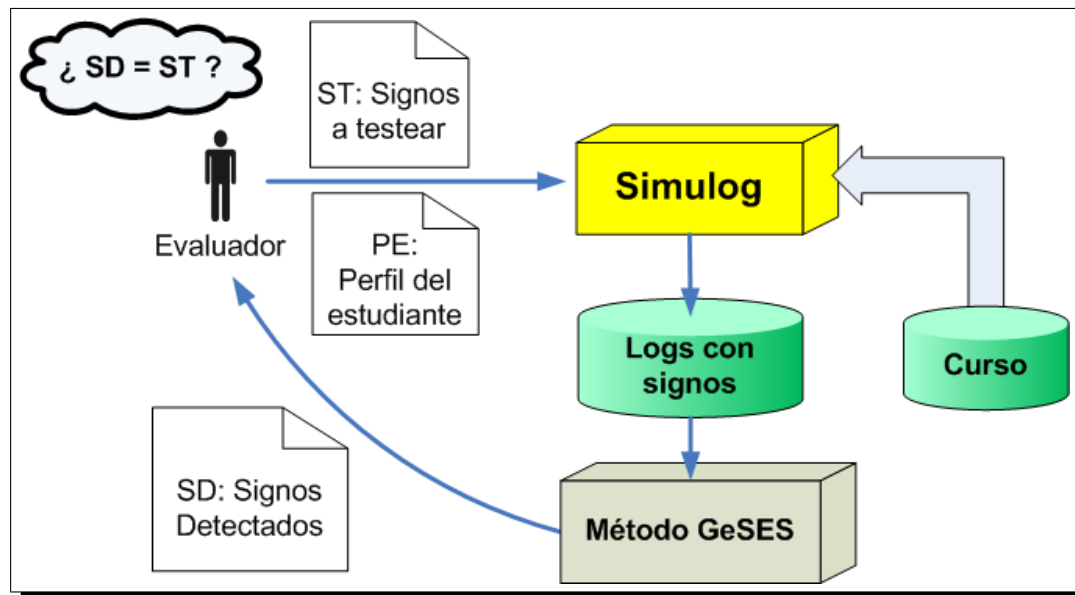


Figura 6.1: Procedimiento de evaluación de la propuesta

evaluador (representados por **ST** en la figura). Siguiendo con el diagrama de la figura citada, se utiliza el método *GeSES* para procesar los logs generados en la fase anterior con el objetivo de encontrar signos. Por último, el evaluador puede comparar los signos detectados (representados en la figura con **SD**) por el método con los que se solicitaron como parámetros en la generación de logs sintéticos. De esta manera, es posible evaluar la capacidad de detección del método *GeSES*.

6.4. Introducción a *SimuLog*

SimuLog (*Simulación de Usuarios a través de Logs*) es una herramienta capaz de generar interacciones sintéticas de los estudiantes de acuerdo a una serie de parámetros indicados por el usuario. La característica que le hace especialmente útil es su capacidad para incluir supuestos fallos en las actividades realizadas por los estudiantes simulados. De esta forma, es posible indicar que *SimuLog* genere un determinado tipo de fallo en una actividad o grupo de actividades cometidos por un determinado perfil de estudiantes, en una determinada proporción (véase la sección 7.3 del capítulo 7 (página 173) para mayor información sobre la herramienta *SimuLog*).

6.5. Evaluación

Esta sección expone el procedimiento que se ha seguido para evaluar el método *GeSES*. Con este propósito se diseñó un curso sintético para el sistema *Wotan*, cons-

tituido por 8 actividades prácticas que deben ser resueltas por los estudiantes de forma secuencial. Esto quiere decir que todos los estudiantes seguirán el mismo recorrido de actividades. De esta forma se generaron, mediante *SimuLog*, logs sintéticos de las interacciones de distintos perfiles de estudiantes, que incluyen diversos signos de bajo rendimiento para determinados perfiles. Las siguientes subsecciones presentan los parámetros de simulación, la aplicación del método para los logs generados y la valoración de los resultados obtenidos.

6.5.1. Preparación de los datos (logs)

En esta fase se diseñó un curso sintético formado por 8 actividades prácticas, que deben ser resueltas de forma secuencial por todos los estudiantes. Además, se diseñó el abanico de los distintos perfiles que pueden realizar dicho curso. Un perfil está formado por una secuencia de valores correspondientes a las distintas dimensiones. Así, para este experimento un perfil está constituido por tres dimensiones:

- Dimensión 1: valores posibles: v_{11} , v_{12} y v_{13} .
- Dimensión 2: valores posibles: v_{21} y v_{22} .
- Dimensión 3: valores posibles. v_{31} , v_{32} y v_{33} .

Por ejemplo, el perfil (v_{11}, v_{21}, v_{31}) está formado por el valor v_{11} para la dimensión 1, el valor v_{21} para la dimensión 2, y el valor v_{31} para la dimensión 3. De esta forma, ejemplos de perfiles concretos son (v_{11}, v_{21}, v_{31}) y (v_{13}, v_{21}, v_{32}) . Consecuentemente, el número total de perfiles distintos es 18.

SimuLog permite al evaluador indicar el porcentaje de generación de los valores para cada dimensión. En el presente experimento se utilizaron los siguientes porcentajes:

- Dimensión 1: v_{11} (32,5 %), v_{12} (35 %) y v_{13} (32,5 %).
- Dimensión 2: v_{21} (50 %) y v_{22} (50 %).
- Dimensión 3: v_{31} (32,5 %), v_{32} (35 %) y v_{33} (32,5 %).

De esta forma, el 35 % de los perfiles contendrán el valor v_{12} en la dimensión 1, el 32,5 % contendrán el valor v_{11} y el 32,5 % el valor v_{13} , y así sucesivamente para las dimensiones 2 y 3. Por lo tanto, la probabilidad de generar un determinado tipo de perfil es, $P(\text{perfil}) = 0,33 \cdot 0,50 \cdot 0,33 = 0,055$, tomando como aproximación $P(\text{valores de dimensión 1}) = 0,33$ y $P(\text{valores de dimensión 3}) = 0,33$. Esto quiere decir que, si se generaran 20 estudiantes, la esperanza sería $E[x] = 0,055 \cdot 20 = 1,100$. Dicho de otro modo, en media se generará 1 estudiante para cada perfil. *SimuLog* realiza sus simulaciones teniendo en cuenta la *Ley de los Grandes Números*, lo que implica que hay que generar un número suficientemente grande de estudiantes para que los porcentajes indicados en el simulador sean efectivos.

Asimismo, se diseñó un conjunto de 12 signos para distintos tipos de perfiles localizados en tres actividades. El conjunto de signos introducidos en el simulador fueron los siguientes:

- **Signo 1:** 100 % del perfil (v11,v21,v31) fallaron la actividad act1.
- **Signo 2:** 80 % del perfil (v11,v22,v31) fallaron la actividad act1.
- **Signo 3:** 60 % del perfil (v11,v21,v32) fallaron la actividad act1.
- **Signo 4:** 50 % del perfil (v11,v21,v33) fallaron la actividad act1.
- **Signo 5:** 40 % del perfil (v11,v22,v32) fallaron la actividad act1.
- **Signo 6:** 20 % del perfil (v11,v22,v33) fallaron la actividad act1.
- **Signo 7:** 90 % del perfil (v12,v21,v31) fallaron la actividad act3.
- **Signo 8:** 70 % del perfil (v12,v22,v31) fallaron la actividad act3.
- **Signo 9:** 30 % del perfil (v12,v21,v32) fallaron la actividad act3.
- **Signo 10:** 10 % del perfil (v12,v21,v33) fallaron la actividad act3.
- **Signo 11:** 40 % del perfil (v11,v21,v31) fallaron la actividad act5.
- **Signo 12:** 60 % del perfil (v11,v22,v32) fallaron la actividad act5.

Además, se fijó la probabilidad de éxito para cualquier actividad en 0,8. Esto quiere decir que, si un estudiante de un determinado perfil no está afectado por alguno de los signos anteriores resolverá correctamente la actividad un 80 % de las veces. La figura 6.2 muestra el estado de *SimuLog* mientras se generaban los logs sintéticos bajo los parámetros anteriormente indicados.

6.5.2. Análisis de los datos mediante el método *GeSES*

El objetivo de la evaluación es valorar la eficacia del método, comprobando cuántos y cuáles de los anteriores signos simulados se detectaron. Hay que tener en cuenta dos consideraciones importantes: a) el método *GeSES* se apoya sobre las reglas de decisión, técnica que necesita ser aplicada a un volumen suficientemente grande de datos, y b) *SimuLog* requiere un número suficientemente grande de estudiantes a ser simulados para conseguir generar un amplio abanico de perfiles. En este caso, es importante que los datos generados contengan estudiantes de distintos perfiles, y cada perfil esté reflejado en los datos. Aunque es cierto que en las clases tradicionales no es común que existan estudiantes de todos los perfiles, en este experimento es importante disponer de esta variedad de perfiles, pues si existe algún signo asociado a algún perfil que no haya sido generado en los datos, no será posible su detección, no por falta de eficacia en el método, sino porque no existen estos estudiantes en los datos. Por este motivo, se realizaron distintas pruebas variando el número de estudiantes simulados desde 20 hasta 6000.

Se aplicó el método *GeSES* a cada uno de los conjuntos de datos y se comprobó que el mínimo de estudiantes para obtener alguna conclusión era de 100, generando un total de 800 entradas en el log. No obstante, la precisión de las reglas obtenidas fue

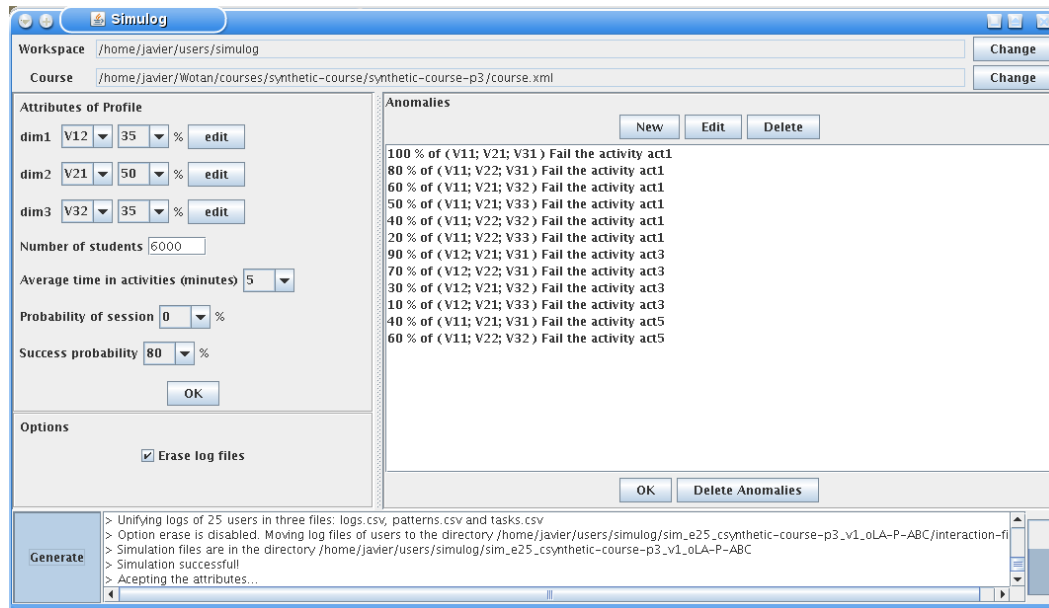


Figura 6.2: *SimuLog* generando logs sintéticos

inferior a las obtenidas en el caso de generación de 300 estudiantes (2400 entradas), continuándose esta tendencia con cada incremento del número de estudiantes.

Otro aspecto importante a destacar es cómo valorar si los signos detectados corresponden a los simulados (caso de detección correcta). En este sentido, se considera una detección correcta si al menos dos de los valores de las dimensiones coinciden con los de algún signo simulado y la actividad coincide con la del signo simulado. Por ejemplo, para el conjunto de datos de 100 estudiantes se obtuvo la siguiente regla: “dimension 2 = *v21*, dimension 3 = *v31* y activity = *act1* ->class *no* [82,2%]”. Esta regla indica que se detectó de forma correcta el **signo 1**, ya que los valores **v21**, **v31** y **act1** coinciden con los de este signo. En el caso de que la detección sea incorrecta, implica que el signo detectado no está incluido en el conjunto de signos simulados, y por tanto, la denominaremos *falso positivo*. Hay que indicar que un *falso positivo* puede ser debido a efectos aleatorios (ruido) en los datos generados, normalmente presente en los conjuntos de datos reales.

La tabla 6.7 muestra un resumen de las pruebas realizadas para distintos tamaños de datos. La primera columna (N) corresponde al número de estudiantes simulados, la segunda (S) a las interacciones generadas por estos estudiantes, la siguiente (Pf) indica el número de perfiles generados, la cuarta columna (Fase (3)) contiene las reglas que se obtuvieron en la fase (3) del método *GeSES* (*Seleccionar las reglas más relevantes para la característica de evaluación*), la siguiente columna (Fase (5)) muestra el número de reglas seleccionadas en la última fase del método (*Seleccionar las mejores reglas*), y las dos últimas columnas contienen los signos detectados (SD) y el número de *falsos positivos* (FP). Esta última columna indica tanto si hubo *falsos*

N	S	Pf	Fase (3)	Fase (5)	SD	FP
20	160	6	1	1	0	+1
50	400	6	2	2	0	0
80	640	6	1	1	0	+1
100	800	12	3	3	signo 1, 7 y 11	0
150	1200	10	2	2	signo 7	0
300	2400	12	3	3	signo 1 y [7]	+1
600	4800	12	3	3	signo 1, 7 y 9	0
900	7200	16	5	3	signo 7,9 y 11	-1
1200	9600	16	4	3	signo 1,4, 7,8 y 11	0
1500	12000	14	3	3	signo 1,3,4, 7,8 y 11	0
1800	14400	16	3	3	signo 7,8 y 11	0
2100	16800	15	4	3	signo [7] y 11	-1
2400	19200	17	4	3	signo 1,3,4, 7,8 y [12]	0
2700	21600	16	5	3	signo 1,3,4, 7 y 8	+1
3100	24800	16	5	4	signo [7], 11 y [12]	-1
3500	28000	17	4	3	signo 1,3,4, [7] y 11	-1
4000	32000	15	5	3	signo 7 y 11	-1
5000	40000	16	4	3	signo 7,8 y 11	0
6000	48000	18	5	3	signo 1,2, 3,4 y [11]	0

Tabla 6.7: Resumen de la aplicación del método *GeSES* para distintos tamaños de datos

positivos (número positivo) como si se evitó su existencia (número negativo). Además, se indica si una regla coincide totalmente con algún signo simulado encerrando entre corchetes el número del signo en la columna SD. Por ejemplo, en la fila sexta, correspondiente a la generación de 300 estudiantes sintéticos, y la columna SD, se puede observar que los signos detectados son el 1 y [7], lo que indica que la detección del signo 7 no sólo es correcta, sino que es completa.

Cabe comentar que en el caso del análisis de los logs sintéticos de 50 estudiantes, que generaron 400 interacciones, no se encontraron falsos positivos, pero tampoco se puede hablar de detección correcta. En este caso, se obtuvieron las siguientes reglas:

- *DR1*: $\text{dim1}=v11, \text{activity}=\text{act1} \rightarrow \text{class no}$ [93 %]
- *DR3*: $\text{dim1}=v11, \text{activity}=\text{act5} \rightarrow \text{class no}$ [42,2 %]

Las dimensiones de la regla *DR1* coinciden con los signos 1 a 6, pero no se la puede considerar una detección correcta, pues sólo coincide una de las dimensiones, la dimensión 1. De igual forma ocurre en el caso de la regla *DR3*, que coincide con los signos 11 y 12, pero tampoco se la considera una detección correcta.

En general, se observa que el número de detecciones de signos que están incluidos en el conjunto de los signos sintéticos aumenta a medida que aumenta el número de estudiantes analizados. Hay que destacar que, esta tendencia también se observa en la calidad de las reglas de decisión obtenidas, la cual aumenta con el tamaño de estudiantes analizados. Igualmente, el análisis de la tabla revela que el número de *falsos positivos* encontrados disminuye conforme aumenta el tamaño de estudiantes analizados.

Es importante señalar que, los resultados de las fases 3 y 5 de la aplicación del método *GeSES* a los conjuntos de datos, reflejados en la tabla 6.7, se han incluido en el apéndice A.7 (página 272), que el lector puede consultar para mayor información. Con el propósito de mostrar un análisis más detallado de los datos generados, a continuación se presentan los resultados que se consideran más relevantes, correspondientes a los tamaños $N = 900$, $N = 2100$ y $N = 6000$.

6.5.2.1. Análisis para $N = 900$

- (1) **Fase de limpieza.** El tamaño de los datos es de 7200 filas. Cada fila corresponde a una interacción de un determinado estudiante con el sistema *Wotan*. Se añadió la *variable de evaluación success*, que contiene dos valores posibles: *yes* ($\text{grade} \geq 0,5$) y *no* ($\text{grade} < 0,5$). La distribución de esta variable es la siguiente:
 - *yes* : 5351 instancias (74,3 %).
 - *no* : 1849 instancias (25,7 %).
- (2) **Generación de las reglas de decisión.** Se aplicó el algoritmo *C4.5* a los *datos de análisis* obteniéndose 13 reglas de decisión, de las cuales 8 corresponden a la clase *yes*, y 5 a la clase *no*.
- (3) **Seleccionar las reglas relevantes para la característica de evaluación.** Se seleccionaron las cinco reglas de la clase *no*, que se pueden ver en la tabla 6.8.
- (4) **Establecer la frontera de selección.** Se estableció como frontera de selección el tercer cuartil para cada uno de los conceptos *used*, *size* y *error*, siendo sus valores los siguientes: $Q_3^{\text{used}} = 290$, $Q_3^{\text{size}} = 3$ y $Q_3^{\text{error}} = 55,9$.

Rule 2:
dim1 = v11
activity = act1
->class no [98,0 %]
Rule 19:
dim1 = v11
dim3 = v31
activity = act5
->class no [49,2 %]
Rule 10:
dim1 = v12
dim2 = v21
activity = act3
->class no [45,0 %]
Rule 21:
dim1 = v11
dim2 = v22
dim3 = v32
->class no [44,1 %]
Rule 8:
dim1 = v13
dim2 = v21
activity = act3
->class no [35,2 %]

Tabla 6.8: Reglas obtenidas en la fase 3 para $N = 900$

- (5) **Seleccionar las mejores reglas.** En esta fase se generó el parámetro precisión ($Q_3^{precision} = 49,2$) y se asignó peso a las reglas. La tabla 6.9 refleja los pesos obtenidos de cada regla en cada concepto y el peso total. Se puede observar que la regla que mayor peso obtiene es la *DR19*, bastante distanciada de las demás reglas. La regla que peor peso total obtiene es la *DR8*.

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
19	0,80	1,00	0,80	2,60
10	0,63	1,00	0,44	2,06
2	0,81	0,00	1,00	1,81
21	0,35	1,00	0,35	1,70
8	0,35	1,00	0,02	1,36

Tabla 6.9: *Ranking* de reglas en la fase 5 para $N = 900$

Se seleccionaron las tres primeras reglas, como reglas más representativas. La regla *DR19* coincide con el signo 11, y por tanto se puede decir que se detectó correctamente este signo. La regla *DR10* coincide con los signos 7 y 9, pues el análisis de los datos (véase tabla 6.10) nos indica que se generaron 151 estudiantes para el perfil (v12,v21,v32) (signo 9), 17 estudiantes de perfil (v12,v21,v31), afectado por el signo 7, y sólo se generó un estudiante de perfil (v12,v21,v33), (signo 10). La regla *DR2* no responde a una detección correcta (posibles signos 1 a 6), pues sólo contiene información sobre la dimensión 1 y la actividad. La regla *DR21* tampoco responde a una detección correcta, pues no existe información sobre la actividad. La última regla, la *DR8*, es un *falso positivo* ya que el perfil (v13,v21,-) no está dentro de los perfiles que experimentan los signos 7 a 10. Como se puede observar en el *ranking*, esta última regla obtuvo una valoración muy baja, lo que evitó su selección.

Este análisis demuestra que el método *GeSES* consigue detectar problemas en perfiles mayoritarios y en perfiles con menor número de estudiantes. En este caso, se detectó un problema, en los estudiantes del perfil (v11,v21,v31) que representa un

Perfil	Instancias	Perfil	Instancias
(v11,v21,v31)	2320	(v13,v21,v31)	32
(v13,v22,v33)	2088	(v11,v22,v33)	24
(v12,v21,v32)	1208	(v13,v22,v31)	16
(v12,v22,v32)	1096	(v11,v22,v32)	8
(v12,v21,v31)	136	(v11,v21,v33)	8
(v12,v22,v33)	104	(v12,v21,v33)	8
(v13,v22,v32)	72	(v12,v22,v31)	8
(v11,v21,v32)	64	(v13,v21,v32)	8

Tabla 6.10: Estadísticas de los perfiles generados para $N = 900$

32 % del total de estudiantes, experimentado por el 40 % de este perfil. Dicho de otro modo, se detectó de un total de 900 estudiantes, que 116 estudiantes, pertenecientes a un conjunto de estudiantes con el mismo perfil de tamaño 290, mostraron dificultades en la actividad act1.

Además, se detectó otro problema relativo a otro tipo de perfil, pero con menor número de estudiantes, representando un 19 %² del total de estudiantes. Según los parámetros de simulación el problema afecta a un 36 %³ de los estudiantes que tienen dimensión 1 con valor v12 y dimensión 2 con valor v21.

Hay que indicar que, el método selecciona sólo las mejores reglas, es decir, aquellas que tengan una cobertura alta, precisión alta y tamaño alto en relación con las demás. Por tanto, en el caso de que una regla obtenga una precisión por debajo del 30 %, deberá poseer un tamaño y cobertura muy altos para contrarrestar la penalización sufrida por la baja precisión. Por esta razón, es muy poco probable la detección de signos que afecten a una proporción por debajo de 30 %, y en este caso particular no fueron detectados los signos 6 y 10. Por otro lado, los parámetros de simulación indicaban la generación de estudiantes con un porcentaje de éxito del 80 %, lo que quiere decir que el porcentaje de fracaso será del 20 %. Consecuentemente, reglas con precisión cercana a este valor tienen una alta probabilidad de ser confundidas con efectos aleatorios, y por el razonamiento anterior el método evita su selección.

6.5.2.2. Análisis para N = 2100

- (1) **Fase de limpieza.** Los datos analizados contienen 16800 filas. Se añadió la *variable de evaluación success*, que contiene dos valores posibles: *yes* ($grade \geq 0,5$) y *no* ($grade < 0,5$). La distribución de esta variable es la siguiente:
 - *yes* : 12501 instancias (74,4 %).
 - *no* : 4229 instancias (25,6 %).
- (2) **Generación de las reglas de decisión.** Se aplicó el algoritmo *C4.5* a los *datos de análisis* obteniéndose 15 reglas de decisión, de las cuales 11 corresponden a la clase *yes*, y 4 a la clase *no*.
- (3) **Seleccionar las reglas relevantes para la característica de evaluación.** Se seleccionaron las cuatro reglas de la clase *no*, que se pueden ver en la tabla 6.11.

²Los tres perfiles afectados son aquellos que tienen factor común a v12 y v21, i.e., (v12,v21,v31), (v12,v21,v32) y (v12,v21,v33). Sumando el número de estudiantes perteneciente a cada uno se obtiene: $17 + 151 + 1 = 169$, que representa sobre 900 estudiantes aproximadamente un 19 %.

³El signo que puede afectar a estudiantes del perfil (v12,v21,v31) es el signo 7, cuya proporción es 90 %, el correspondiente al perfil (v12,v21,v32) es el signo 9, con proporción 30 %, y el correspondiente al perfil (v12,v21,v33) es el signo 10, que afecta sólo al 10 %. Por tanto, el número medio de estudiantes afectados es: $17 \cdot 0,9 + 151 \cdot 0,3 + 1 \cdot 0,1 = 60,7$, que es aproximadamente 61, y cuya proporción sobre 169 es 36 %.

Rule 3:
dim1 = v11
activity = act1
->class no [96,7 %]
Rule 12:
dim1 = v12
dim2 = v21
dim3 = v31
activity = act3
->class no [86,7 %]
Rule 19:
dim1 = v11
dim3 = v31
activity = act5
->class no [53,3 %]
Rule 24:
dim1 = v13
dim2 = v21
dim3 = v31
activity = act7
->class no [44,1 %]

Tabla 6.11: Reglas obtenidas en la fase 3 para N = 2100

- (4) **Establecer la frontera de selección.** Se estableció como frontera de selección el tercer cuartil para cada uno de los conceptos *used*, *size* y *error*, siendo sus valores los siguientes: $Q_3^{used} = 664,5$, $Q_3^{size} = 4$ y $Q_3^{error} = 49$.
- (5) **Seleccionar las mejores reglas.** En esta fase se generó el parámetro precisión ($Q_3^{precision} = 89,2$) y se asignó peso a las reglas. La tabla 6.12 refleja los pesos obtenidos por cada regla y el peso total. Se puede observar que las reglas que mayor peso obtienen son *DR12* y *DR3*. La regla que peor peso total obtiene es *DR24*.

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
12	0,36	0,80	0,78	1,93
3	0,82	0,14	0,86	1,81
19	0,79	0,45	0,38	1,62
24	0,34	0,80	0,27	1,41

Tabla 6.12: *Ranking* de reglas en la fase 5 para $N = 2100$

Se seleccionaron las tres primeras reglas, como reglas más representativas. La regla *DR12* coincide totalmente con el signo 7, y por tanto se puede decir que se detectó de forma correcta y completa este signo. La regla *DR3* no responde a una detección correcta (posibles signos 1 a 6), pues sólo contiene información sobre la dimensión 1 y la actividad. La regla *DR19* coincide con el signo 11, y su detección fue correcta. La última regla es un *falso positivo*, ya que la actividad *act7* no está afectada por ninguno de los signos simulados. Al igual que con 900 estudiantes, el *falso positivo* es el peor valorado en el *ranking*.

El análisis de estos datos ($N = 2100$), como sucedió en el caso anterior, refleja que el método *GeSES* no sólo detecta problemas en los perfiles mayoritarios (véase tabla 6.13), sino también detecta problemas en perfiles minoritarios, como es el caso del perfil (v12,v21,v31) con 37 estudiantes (representando a 1,8 % del total de estudiantes). En este sentido, se detectó que el 90 % de los estudiantes de este perfil,

Perfil	Instancias	Perfil	Instancias
(v11,v21,v31)	5232	(v13,v21,v31)	64
(v13,v22,v33)	5032	(v11,v22,v33)	64
(v12,v21,v32)	2976	(v11,v21,v33)	64
(v12,v22,v32)	2472	(v11,v22,v32)	40
(v12,v21,v31)	296	(v12,v22,v31)	24
(v13,v22,v32)	168	(v13,v22,v31)	24
(v11,v21,v32)	168	(v13,v21,v32)	16
(v12,v22,v33)	160		

Tabla 6.13: Estadísticas de los perfiles generados para $N = 2100$

i.e., 33 estudiantes, mostraron problemas en la actividad act3. También, se detectó un problema relacionado con los estudiantes que tienen los valores v11 y v31 en sus dimensiones. Concretamente, este problema se presentó al realizar la actividad act5. Como de los estudiantes generados, sólo el perfil (v11,v21,v31) contiene los valores anteriores, y 654 estudiantes (31 % del total) pertenecen a este perfil, revisando los parámetros de simulación (signo 11), veremos que el 40 % (262 estudiantes) de éstos presentaron signos de bajo rendimiento en la actividad act5, representando un 13 % del total de estudiantes. Por último, se evitan detecciones incorrectas, ya que la regla que representa un *falso positivo* es la peor valorada en el *ranking*.

Por las mismas razones que en el anterior caso, no se detectaron signos cuyas proporciones estuvieran por debajo del 30 %, pero como se ha descrito en el párrafo anterior, se detectó de forma correcta y completa un signo cuya proporción fue del 40 %, lo que muestra la eficacia del método.

6.5.2.3. Análisis para $N = 6000$

- (1) **Fase de limpieza.** El tamaño de los datos es de 48000 filas. Se añadió la *variable de evaluación success*, que contiene dos valores posibles: *yes* ($grade \geq 0,5$) y *no* ($grade < 0,5$). La distribución de esta variable es la siguiente:
 - *yes* : 36017 instancias (75,0 %).
 - *no* : 11983 instancias (25,0 %).
- (2) **Generación de las reglas de decisión.** Se aplicó el algoritmo *C4.5* a los *datos de análisis* obteniéndose 16 reglas de decisión, de las cuales 11 corresponden a la clase *yes*, y 5 a la clase *no*.
- (3) **Seleccionar las reglas relevantes para la característica de evaluación.** Se seleccionaron las cinco reglas de la clase *no*, que se pueden ver en la tabla 6.14.
- (4) **Establecer la frontera de selección.** Se estableció como frontera de selección el tercer cuartil para cada uno de los conceptos *used*, *size* y *error*, siendo sus valores los siguientes: $Q_3^{used} = 1608$, $Q_3^{size} = 4$ y $Q_3^{error} = 41,2$.
- (5) **Seleccionar las mejores reglas.** En esta fase se generó el parámetro precisión ($Q_3^{precision} = 95,5$) y se asignó peso a las reglas. La tabla 6.15 refleja los pesos obtenidos por cada regla y el peso total. Se puede observar que las reglas que mayor peso obtienen son *DR5* y *DR22*. Las reglas con peores pesos totales son *DR23* y *DR10*.

Se seleccionaron las tres primeras reglas, como reglas más representativas. La regla *DR5* coincide con los signos 1 y 2, y por tanto se puede decir que ambos signos fueron detectados de forma correcta y parcial. La regla *DR22* coincide completamente con el signo 11, por tanto, su detección es correcta y completa. La información que contiene la regla *DR4* está relacionada con los signos 1, 3 y 4, pues existen instancias suficientes (véase tabla 6.16) de

Rule 5:
dim1 = v11
dim3 = v31
activity = act1
->class no [99,9 %]
Rule 4:
dim1 = v11
dim2 = v21
activity = act1
->class no [95,5 %]
Rule 10:
dim1 = v12
dim3 = v31
activity = act3
->class no [85,5 %]
Rule 23:
dim1 = v12
dim2 = v22
dim3 = v32
activity = act5
->class no [58,8 %]
Rule 22:
dim1 = v11
dim2 = v21
dim3 = v31
activity = act5
->class no [51,4 %]

Tabla 6.14: Reglas obtenidas en la fase 3 para N = 6000

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
5	0,80	0,35	0,84	1,99
22	0,80	0,80	0,27	1,87
4	0,35	0,35	0,80	1,50
23	0,28	0,80	0,35	1,43
10	0,35	0,35	0,70	1,40

Tabla 6.15: *Ranking* de reglas en la fase 5 para N = 6000

los perfiles (v11,v21,v31) (1608 estudiantes), (v11,v21,v32) (213 estudiantes) y (v11,v21,v33) con 32 estudiantes. La regla *DR23* es un caso de detección correcta y completa del signo 12. Finalmente, la última regla contiene infor-

Perfil	Instancias	Perfil	Instancias
(v13,v22,v33)	12904	(v11,v22,v33)	416
(v11,v21,v31)	12864	(v13,v22,v31)	376
(v12,v21,v32)	6744	(v11,v21,v33)	256
(v12,v22,v32)	6248	(v12,v21,v33)	248
(v12,v21,v31)	1776	(v11,v22,v32)	240
(v11,v21,v32)	1704	(v13,v21,v32)	200
(v13,v22,v32)	1680	(v12,v22,v31)	176
(v12,v22,v33)	1640	(v11,v22,v31)	40
(v13,v21,v31)	464	(v13,v21,v33)	24

Tabla 6.16: Estadísticas de los perfiles generados para $N = 6000$

mación relacionada con los signos 7 y 8.

El análisis de estos datos ($N = 6000$) refleja que el método *GeSES* ha detectado un problema que afecta a un 40 % de uno de los perfiles mayoritarios (signo 11), representando un 27 % de los estudiantes. En otras palabras, revisando los parámetros de simulación veremos que 643 estudiantes (11 % del total y 40 % de su perfil) de los 1608 estudiantes (27 % del total) con este mismo perfil, mostraron problemas en la actividad act5.

En este caso, también se detectó un problema en un perfil minoritario, (v11,v22,32), representando un 0,5 % del total de las instancias, que afectó al 60 % de estudiantes con este perfil (18 estudiantes que representan un 0,3 % del total de estudiantes). No obstante, este problema aparece en cuarta posición en el *ranking* final, pues pocos estudiantes mostraron este problema, y como se ha explicado anteriormente el método *GeSES* selecciona las mejores reglas. Por tanto, es muy poco probable que una regla que sea fuertemente penalizada en un concepto de los tres que se evalúan, en este caso la cobertura (concepto *used*), aparezca en las posiciones superiores del *ranking*. Finalmente, no se obtuvieron *falsos positivos*, pues debido fundamentalmente a la riqueza de los *datos de análisis* ninguna de las reglas obtenidas es un *falso positivo*.

6.6. Conclusiones

Este capítulo muestra la forma de aplicar el método *GeSES*, presentando la aplicación de cada una de las cinco fases con un nuevo conjunto de datos. Asimismo, se incluyó al final de esta aplicación un análisis de los resultados obtenidos por dicho método, así como de la utilidad que pueden tener para los profesores. En este análisis se evidencia la utilidad de la tabla de signos para los profesores, ya que éstos sin necesidad de conocimientos en *DM* pueden conocer de forma rápida y eficaz determinados problemas que se han detectado en el curso. Por ejemplo, la tabla de signos contiene información sobre qué actividades o qué tipo de actividades presentaron

mayores dificultades para los estudiantes, o qué perfiles o grupos de estudiantes realizaron las actividades con dificultades, o incluso qué perfiles tuvieron más dificultades en determinadas actividades. Esta información puede resultar muy valiosa para los profesores o diseñadores de cursos, pues conociendo ésta ellos pueden tomar las acciones pertinentes en cada caso, bien reforzando determinadas actividades con nuevos contenidos, bien mejorando la capacidad adaptativa de éstas, o bien guiando de mejor forma a los estudiantes en el recorrido seguido.

También se ha expuesto en este capítulo la evaluación de la eficacia del método *GeSES*. En concreto, se presentó un procedimiento de evaluación basado en simulación de logs que incluyen signos de bajo rendimiento. De esta forma, se generaron con *SimuLog* 19 conjuntos de datos de distinto tamaño, pero con los mismos 12 signos de bajo rendimiento. El objetivo fue comprobar si el tamaño y proporción del problema influyen en la eficacia del método, y en qué medida. Los resultados obtenidos demostraron que tanto el tamaño de los datos como la proporción del problema influyen en la calidad de los signos detectados. Esto es fundamentalmente debido a que el método *GeSES* se apoya fuertemente en la técnica *reglas de decisión* y un modelo estadístico, que requieren un tamaño de *datos de análisis* suficientemente grande para obtener resultados fiables. En este sentido, con las condiciones impuestas en la simulación se ha demostrado que aplicar el método a un número de estudiantes inferior a 300 puede producir resultados poco fiables, aumentando el grado de fiabilidad a medida que se aumenta el número de estudiantes. No obstante, hay que tener en cuenta que la proporción total de valores “no” (fallos) y valores “yes” (aciertos) en la *variable de evaluación* también es un factor que se debe tener en cuenta en la calidad de las *reglas de decisión* obtenidas. Por tanto, no sólo es importante el número de estudiantes analizados, sino también la proporción de fallos que se produzca en ellos o en un grupo de ellos. Hay que tener presente que, si se aumenta el tamaño de los datos manteniendo constante el porcentaje de fallos, es decir, aumento de estudiantes que aciertan, no se produce una mejora significativa de la calidad de las reglas correspondientes a la clase “no”, que son finalmente las que analiza el método presentado en esta tesis.

Finalmente, el proceso seguido en la evaluación del método ha demostrado la eficacia del método para detectar signos que afectan tanto a perfiles mayoritarios como minoritarios. Esta característica es especialmente importante, pues para los profesores en el ámbito de los sistemas *e-Learning* resulta muy complicado descubrir estos problemas en perfiles con menor presencia. No obstante, si los perfiles afectados por los signos están relacionados, es decir, existe al menos una dimensión de factor común, es complicado que se detecten estos signos, pues las mejores reglas (las que tengan mayor precisión, mayor cobertura y mayor tamaño de regla) serán las seleccionadas, ocultando de esta forma a las otras que son peores. Tal situación se puede observar en la evaluación presentada en este capítulo, pues determinadas reglas, cuya información nos indica la presencia de un signo, fueron ocultadas por otras, que también indicaban la presencia de otro signo, pero más importante o relevante que el fue ocultado.

Herramientas para la Evaluación

Índice de contenidos

7.1. Introducción	171
7.2. Ciclo de creación y mantenimiento de cursos <i>e-Learning</i> . .	172
7.3. <i>SimuLog</i>	173
7.3.1. Arquitectura	174
7.3.2. Entradas de simulación	174
7.3.3. Motor de simulación	178
7.3.4. Procesador de logs	186
7.4. <i>ASquare</i>	189
7.4.1. Elementos de <i>ASquare</i>	189
7.4.2. Interfaz de <i>ASquare</i>	190
7.5. Conclusiones	192

*Este capítulo expone la propuesta de arquitectura para evaluación de cursos e-Learning utilizando el análisis de las interacciones de los estudiantes. También se muestran las dos herramientas implementadas por el autor: *SimuLog* y *ASquare*. *SimuLog* grosso modo es una herramienta que permite generar interacciones de estudiantes de forma sintética. *ASquare* es una herramienta construida para dar soporte a los profesores en la aplicación del método propuesto en la presente tesis.*

7.1. Introducción

El análisis de las interacciones es necesario para conocer si los contenidos fueron adecuados, ya que estos datos están relacionados con sus comportamientos. Debido a que el tamaño de estos datos es suficientemente grande para ser inviable su análisis sin la ayuda de herramientas automáticas o semi-automáticas, es útil disponer de métodos y herramientas que les ayuden en la ejecución de esta tarea. En el caso concreto de esta tesis se desarrolló *ASquare* (*Author Assistant*), herramienta de evaluación que implementa los métodos de detección de signos *Key-node* y *GeSES*.

En la siguiente sección se expone el contexto en el que se aplica *ASquare*, así como la utilidad de disponer de una herramienta como *SimuLog*. En este sentido, se expone el ciclo del desarrollo de un curso *e-Learning* añadiendo las dos herramientas anteriores con el objetivo de mejorar el proceso. Las últimas secciones presentan de forma detallada las herramientas *SimuLog* y *ASquare*.

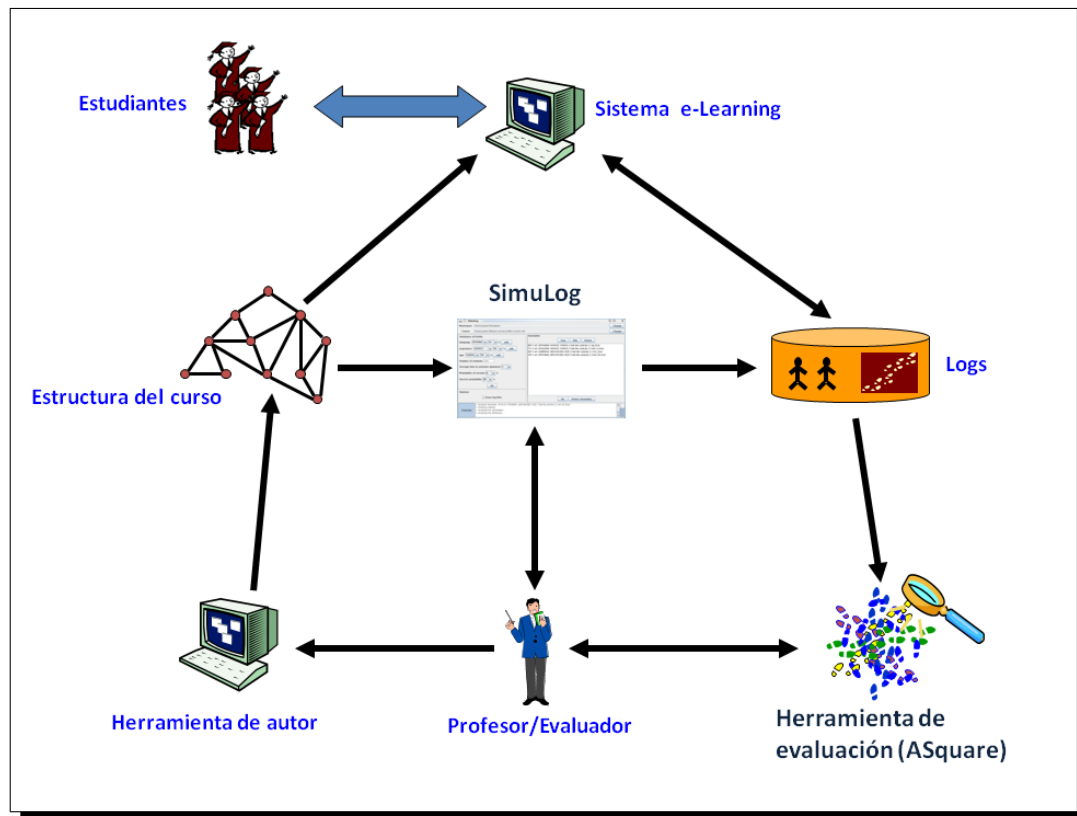


Figura 7.1: Ciclo de creación y mantenimiento de cursos *e-Learning*

7.2. Ciclo de creación y mantenimiento de cursos *e-Learning*

La figura 7.1 representa la propuesta que se realiza en este estudio sobre el ciclo de creación y mantenimiento de cursos *e-Learning*. Esta propuesta es similar al estudio realizado por [Romero 2002] (véase capítulo 2, página 46) que divide la construcción de un curso adaptativo en cuatro etapas: construcción, ejecución, aplicación de la Minería de Datos y mejorar el curso.

La figura citada presenta al profesor en varias facetas, la de creador y evaluador. El profesor crea el curso mediante la herramienta de autor, ésta genera la estructura del curso, que es introducida en el sistema *e-Learning*. El curso es ofrecido por este último a los estudiantes, y éstos lo realizan, generando acciones y resultados de sus interacciones, que el sistema almacena en forma de logs. Los logs contienen tanto los perfiles de los estudiantes como los recorridos de actividades que han seguido. Esta información es especialmente importante para analizar si los contenidos del curso fueron adecuados para los estudiantes, pues contiene el comportamiento de éstos con el sistema.

En la faceta de evaluador, el profesor puede utilizar la herramienta de evaluación

para detectar si hubo contenidos en los que los estudiantes mostraron problemas. Este es el papel que desempeña *ASquare*, y será explicada con más detalle más adelante en este capítulo. El objetivo de *ASquare* es aplicar los métodos de detección de signos, expuestos en el capítulo 5, para ofrecer al evaluador un resumen sobre aquellos elementos del curso que necesitan mayor atención.

Por otro lado, el profesor puede también comprobar si el diseño del curso es el esperado utilizando técnicas de simulación para generar las interacciones de los estudiantes (*SimuLog*). De este modo, antes de ofrecerlo a los estudiantes se puede testar si el recorrido de actividades para cada perfil responde a lo diseñado. Además, el evaluador puede comprobar la calidad de los resultados ofrecidos por la herramienta de evaluación utilizando *SimuLog*, tal y como se expuso en el capítulo anterior (véase capítulo 6).

7.3. *SimuLog*

SimuLog fue diseñada para validar métodos y herramientas de evaluación de sistemas *AEH*, basados en análisis de logs. La idea es generar ficheros de logs de una manera controlada para que puedan ser utilizados como datos de entrada en el método de evaluación. Estos datos de entrada contendrán signos generados de forma artificial por el simulador. De esta forma, se puede comprobar si el método de evaluación es capaz de detectar los signos anteriores en los logs sintéticos. La primera versión de *SimuLog* fue presentada en el *workshop* sobre evaluación de sistemas adaptativos, celebrado en el congreso *Adaptive Hypermedia and Adaptive Web-Based Systems - AH2006* [Bravo 2006c]. Además, esta herramienta se realizó en el contexto del Trabajo de Iniciación a la Investigación presentado en septiembre del año 2006 [Bravo 2006a]. Hay que indicar que, en los últimos años se han realizado mejoras en la interfaz y en la forma de generación de los logs de *SimuLog*.

Originalmente, *SimuLog* fue diseñado buscando la mayor independencia posible tanto de la herramienta de evaluación a validar como del sistema adaptativo. Debido a este motivo, se buscó la máxima generalidad en la definición de sus entradas. Por eso, el diseño original preveía utilizar el estándar *SCORM* [Dodds 2004a] para la descripción de cursos adaptativos orientados a la enseñanza, y una ontología para describir los signos. El primero sirve para describir un curso adaptativo en formato *XML* de forma generalizada, y de esta manera, conseguir que la descripción del curso sea independiente del método de evaluación y del sistema adaptativo. En particular, *SCORM 1.3.1* [Dodds 2004b] describe un curso mediante unidades de aprendizaje y reglas. Las reglas pueden ser modos de secuencias de control y cláusulas de tipo *if-then*. La decisión de cuál es el siguiente nodo a mostrar es tomada por el Sistema Gestor de Aprendizaje (*Learning Management System - LMS*). No obstante, el prototipo actual sólo soporta el formato de descripción de cursos adaptativos existente en la versión mejorada de *TANGOW*, cuyo nombre es *Wotan*. Además, dicho prototipo implementa una parte de la ontología para describir determinados tipos de signos, los relacionados con el bajo rendimiento académico de los estudiantes.

7.3.1. Arquitectura

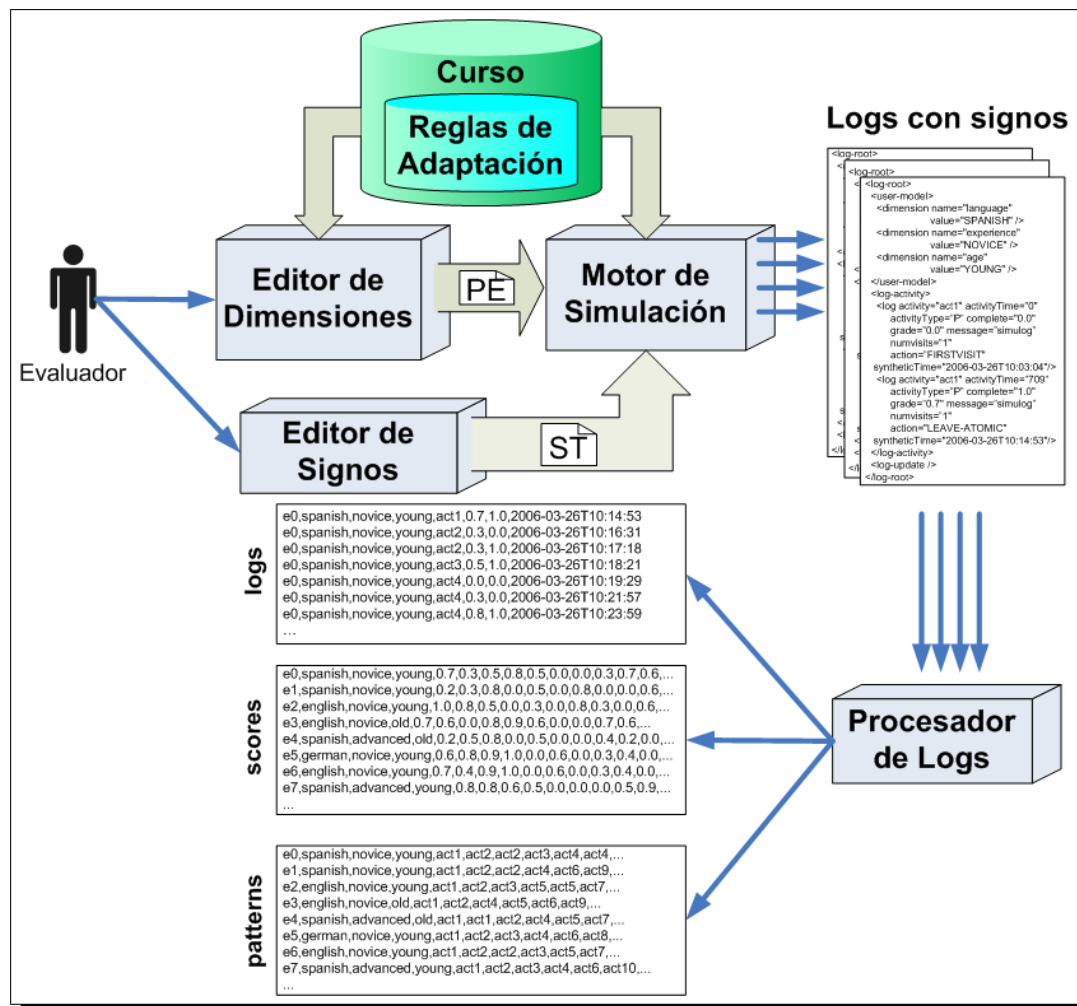
SimuLog está dividido en tres partes claramente diferenciadas: el procesamiento de las entradas de simulación, el motor de simulación y el procesador de logs (véase figura 7.2). El funcionamiento de *SimuLog* es el siguiente: el evaluador introduce los parámetros de simulación deseados (utilizando el editor de perfiles y de signos), la herramienta recibe estos parámetros, el núcleo de simulación los procesa y produce la simulación solicitada, consultando la descripción del curso adaptativo, que es traducida en ficheros de logs. Seguidamente, el procesador de logs procesa los ficheros de logs generando tres ficheros: fichero de puntuaciones en las actividades (llamado *scores*), fichero de interacciones (*logs*) y fichero de recorridos (*patterns*). El primero de ellos contiene las características de los estudiantes y sus resultados en cada una de las actividades. Así, cada línea de este fichero contiene información sobre el perfil del estudiante, y las puntuaciones que obtuvo en cada actividad. El fichero de interacciones está formado por los perfiles de estudiantes, las actividades que realizaron, características de estas actividades (tipo de actividad, estado de la actividad, número de visitas, etc.), puntuación en las actividades y contexto en el que fueron realizadas. El último fichero guarda los datos relativos a los perfiles de estudiantes y los recorridos de actividades realizados por los estudiantes. De esta forma, una fila contiene información sobre el perfil del estudiante y los nombres de las actividades que realizó (respetando el orden en el que fueron realizadas).

7.3.2. Entradas de simulación

Como se ha visto anteriormente, *SimuLog* recibe tres tipos de parámetros de entrada: descripción del curso, atributos o dimensiones de los estudiantes y los signos a generar. La descripción del curso hace referencia a la estructura del curso, a partir de la cual se van a generar los logs sintéticos. Hay que señalar que el prototipo actual sólo admite cursos adaptativos con el formato *Wotan* (versión mejorada de *TANGOW*).

7.3.2.1. Editor de perfiles

Las características o atributos de los estudiantes reciben el nombre en esta herramienta de “dimensiones”. Por tanto, un perfil de estudiante está formado por una o más dimensiones. *SimuLog* permite que el evaluador pueda definir los perfiles de estudiantes que se desean simular. En el caso de los cursos del sistema *Wotan*, las dimensiones generalmente están asociadas al tipo de adaptación implementada en un curso, por eso, la estructura del curso incluye la definición de las dimensiones. Por ejemplo, la figura 7.3 muestra un ejemplo de simulación de logs sintéticos para el “curso sobre normas viales”. En esta figura se puede observar que la herramienta ha extraído de forma automática las dimensiones de los perfiles de los estudiantes: idioma del curso (*language* en la figura), conocimiento previo (*experience* en la figura) y edad (*age* en la figura). Además, se le puede indicar al simulador el número de estudiantes a simular, tiempo medio empleado en una actividad, probabilidad de

Figura 7.2: Arquitectura de *SimuLog*

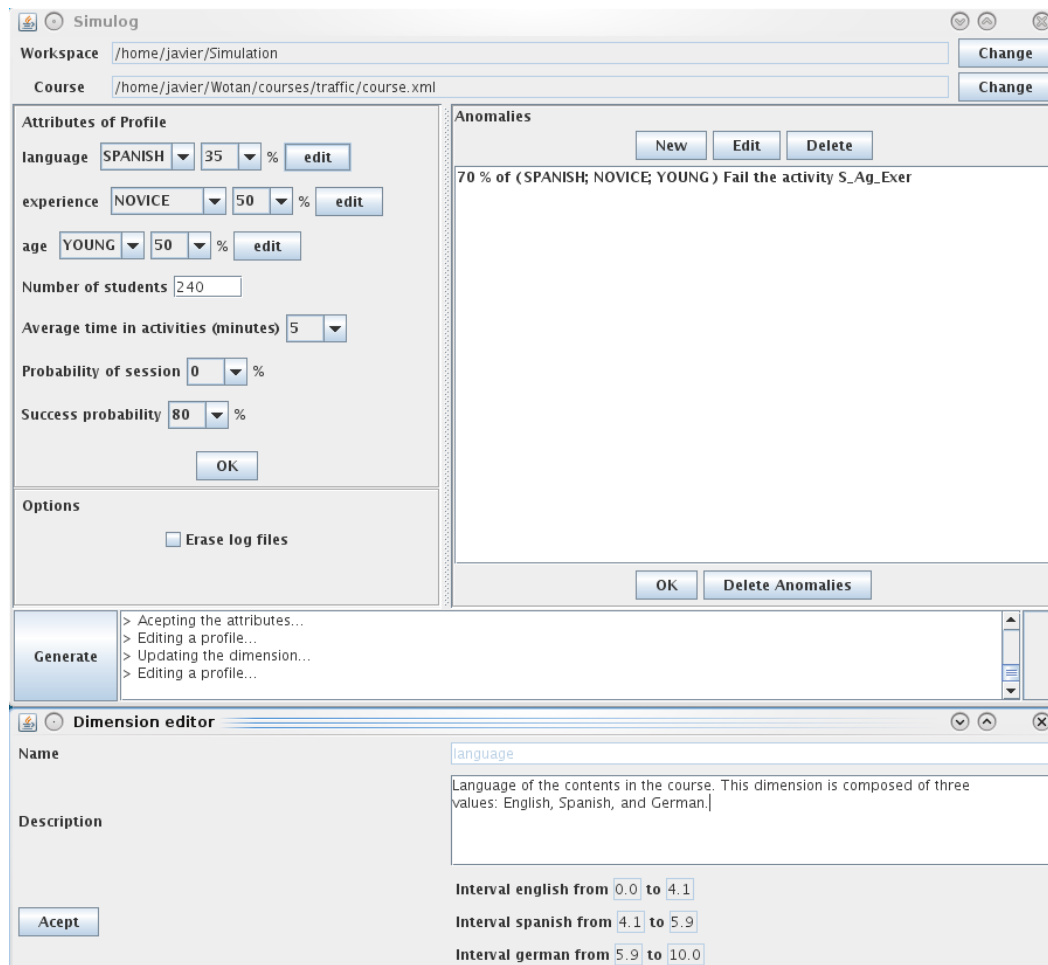


Figura 7.3: Interfaz de *SimuLog*: el editor de dimensiones

cambiar de sesión y probabilidad de éxito. Respecto al tiempo medio empleado, se mide en minutos, y se genera mediante una distribución *Normal* de media el tiempo medio introducido en esta casilla (μ) y desviación típica $\mu/1,96$ ¹. La probabilidad de sesión hace referencia al cambio de sesión una vez el estudiante haya abandonado o terminado la actividad. De esta manera, cada vez que el estudiante abandona o termina una actividad, se realiza un sorteo aleatorio, basado en esta probabilidad, para determinar si se genera una nueva sesión o no. Consecuentemente, cuanto más cercano se fije este valor a 1, más frecuente es que el estudiante inicie nuevas sesiones. El último parámetro indica la probabilidad de éxito que tiene todo estudiante al resolver una actividad, siempre que su perfil no esté afectado por un signo.

La idea que subyace en la simulación de logs es que los distintos perfiles de-

¹El valor 1,96 es el valor que toma z para un nivel de significación 0,05. De esta forma, como el objetivo es obtener tiempos superiores a 0, resolviendo $P(z \leq (0 - \mu)/\sigma) = 0,025$ se obtiene el valor de σ . Por tanto, el valor de σ es $\mu/1,96$.

terminarán la forma en la que los estudiantes responderán a los ejercicios o, la probabilidad de abandonar una actividad antes de completarla o, la probabilidad de fallo en una determinada actividad por un grupo de estudiantes, etc. Dicho de otro modo, un modelo de probabilidad es el que determina que se produzca alguna de las situaciones anteriores. De hecho, este modelo depende de las proporciones de los perfiles, es decir, los valores de las dimensiones y sus proporciones. En el ejemplo de la figura 7.3 puede verse como las proporciones para las dimensiones son 35 %, 50 % y 50 %. Esto quiere decir que, la herramienta generará más perfiles con el idioma del curso en español, ya que, el resto (65 %) se reparte por igual entre los otros dos idiomas (32,5 % para el inglés y 32,5 % para el alemán). Los perfiles se reparten por igual entre los estudiantes novatos y avanzados, y lo mismo ocurre con la dimensión edad entre los jóvenes y los mayores.

Otra característica destacable de esta herramienta es la posibilidad de que el evaluador conozca los parámetros que describen una dimensión. Tal información es mostrada cada vez que se pulsa el botón “edit” por el “editor de dimensiones” (parte inferior de la figura 7.3). Este editor muestra los parámetros que describen una dimensión: *name* (identificador de la dimensión), *description* (descripción de la dimensión) y los intervalos de cada uno de los valores respecto a la *Normal* generada. Por ejemplo, en la figura citada se puede observar que el identificador de la dimensión editada es “*language*”, su descripción, es decir, la semántica de esta dimensión es “*Language of the contents in the course. This dimension is composed of three values: English, Spanish, and German*”, y los intervalos para los valores son: de 0,0 a 4,1 para el valor inglés, 4,1 a 5,9 para el idioma español y 5,9 a 10,0 para el idioma alemán².

7.3.2.2. Editor de signos

SimuLog genera ficheros que contienen signos de potenciales problemas en el rendimiento académico de los estudiantes. Estos signos responden a comportamientos anómalos o no esperados de los estudiantes. Un ejemplo de un signo de bajo rendimiento puede ser: los estudiantes de un determinado perfil “P”, obtienen puntuaciones bajas en una determinada actividad “A”, con una probabilidad “X”. El módulo de signos (parte derecha, *anomalies*, de la figura 7.3) permite generar un nuevo signo (botón “New”), modificar los parámetros de un signo existente (botón “Edit”), borrar un signo (botón “Delete”) y borrar todos los signos (botón “Delete Anomalies”). *SimuLog* proporciona al evaluador un editor de signos (figura 7.4) al que se accede a través de los botones “New” o “Edit”. Este editor permite al evaluador definir los signos que se desean añadir en la simulación. La figura 7.4 muestra el estado del editor de signos después de haber pulsado el botón “New”. Se puede observar que éste se divide en: parámetros relacionados con el tipo de signo (*Property*), actividad a la afecta (*Activity*), porcentaje de presencia (*Percent*), parámetros relacionados con el perfil de estudiantes que experimenta el signo (*language*,

²En la sección **Motor de simulación** se explica detalladamente el significado de estos intervalos.

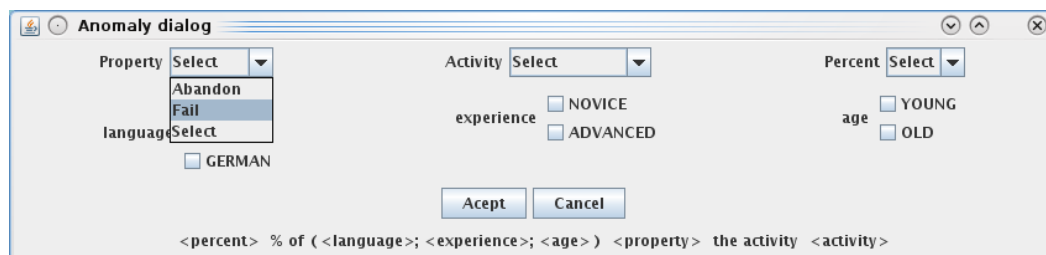


Figura 7.4: Interfaz de *SimuLog*: el editor de signos en el estado inicial

experience y *age* en la figura), y una descripción del signo, que se va actualizando según cambian los valores de los anteriores parámetros.

La figura 7.5 muestra la definición de un signo que está relacionado con fallar la actividad *S_Ag_Exer* y cuyo porcentaje es del 70%. El perfil afectado es el correspondiente a los estudiantes que siguen el curso en español, son novatos y jóvenes. La parte final de esta figura presenta la descripción del signo de la forma siguiente:

“70 % of (SPANISH;NOVICE;YOUNG) Fail the activity S_Ag_Exer”

Esta descripción es una ayuda que se proporciona al evaluador para entender el signo que ha definido de una manera simplificada. Nótese que si se deseara incluir en el perfil del signo a los que siguen el curso en el idioma alemán, bastaría con editar el signo anterior y seleccionar además el idioma “GERMAN” (véase figura 7.6).

La figura 7.7 muestra el estado de *SimuLog* después de haber añadido el signo presentado en la figura 7.5. Obsérvese que, en la parte inferior de dicha figura se muestra el log del simulador, que sirve para indicar al evaluador el éxito o el fracaso en las operaciones llevadas a cabo por el simulador, así como para guiar al evaluador con sugerencias. Por ejemplo, en la parte inferior del simulador se indican una serie de mensajes como si la definición de un signo es correcta, los pasos que se deben seguir para realizar la simulación, etc. En este caso se indica que se añadió con éxito el signo mediante el mensaje: “*Accepted anomaly: 70 % of (SPANISH; NOVICE; YOUNG) Fail the activity S_Ag_Exer*”. Además, se observa que tanto los botones “OK” del módulo de generación de perfiles, como “Generate” permanecen desactivados. Lo cual indica que, mientras no se haya definido un conjunto de signos (botón “OK” del módulo de generación de signos), no es posible iniciar la generación de logs.

Hay que indicar que, en la parte inferior izquierda de la figura 7.7 se presentan al evaluador las opciones disponibles para el procesador de logs. La versión actual de *SimuLog* presenta al evaluador la posibilidad de conservar o eliminar los ficheros individuales de los estudiantes (desactivando o activando la opción “Erase log files”).

7.3.3. Motor de simulación

El motor de simulación realiza una doble función: generación de perfiles de estudiantes con la proporción de dimensiones solicitada y la generación de logs. Las

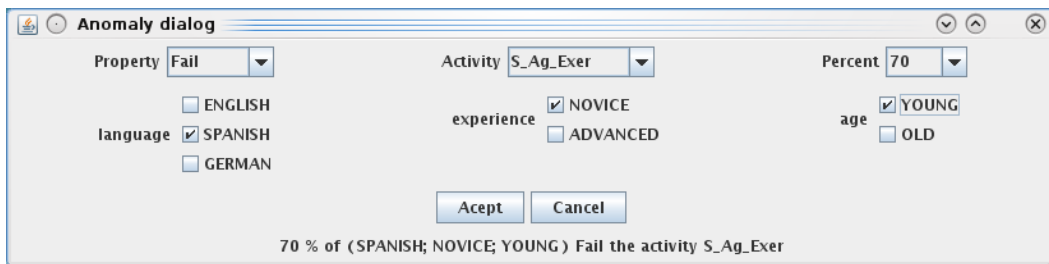


Figura 7.5: Interfaz de *SimuLog*: el editor de signos con la descripción de un signo

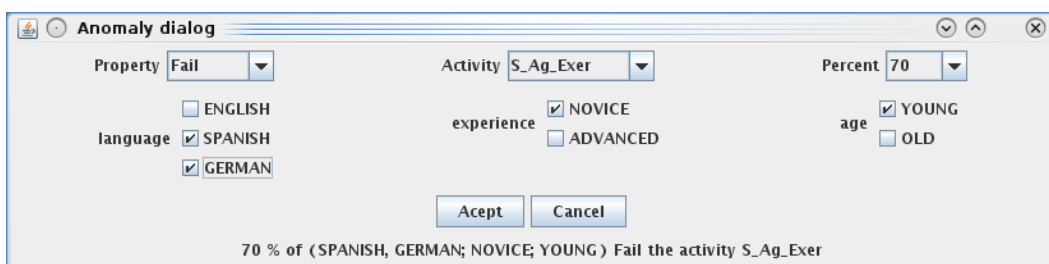


Figura 7.6: Interfaz de *SimuLog*: el editor de signos con la modificación de un signo

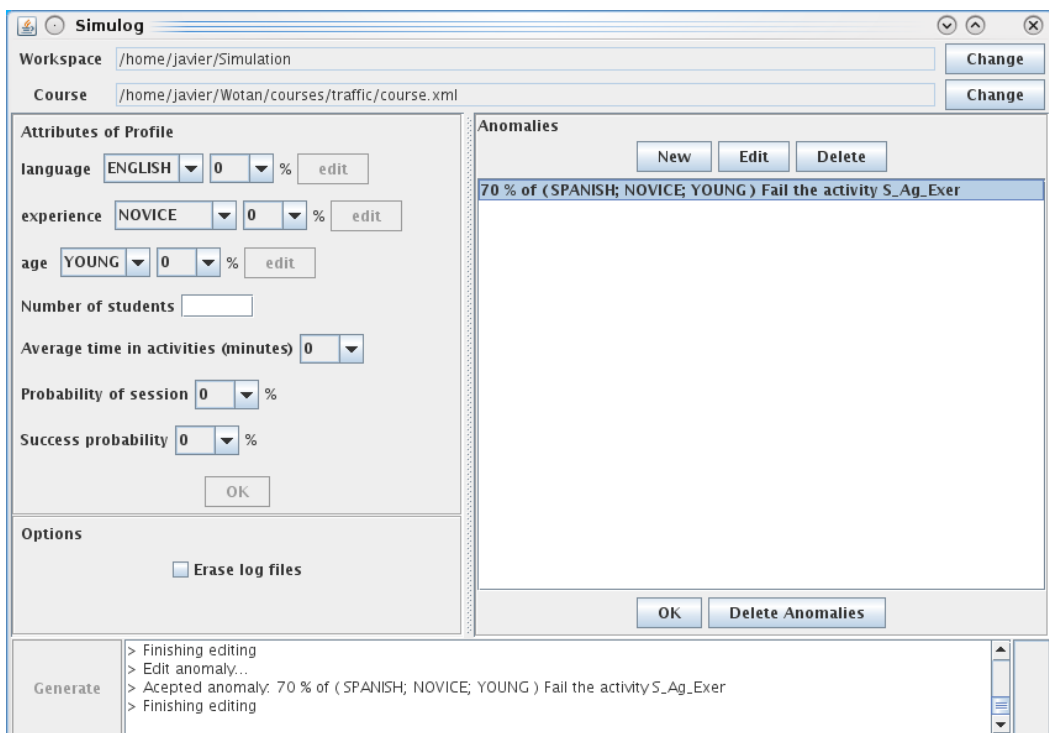


Figura 7.7: Interfaz de *SimuLog* después de haber añadido un signo

siguientes secciones exponen el funcionamiento interno de esta doble función.

7.3.3.1. Generación de perfiles

En la simulación de perfiles es importante que se cumplan dos propiedades ampliamente conocidas en el ámbito de la Estadística: representatividad y aleatoriedad. Si estas dos propiedades no se cumplieran, no sería posible extrapolar los resultados de la simulación de estudiantes. La primera propiedad mide el grado por el que una muestra representa a la población, es decir, mide como es de buena la extrapolación de resultados a partir de esta muestra. La otra propiedad, aleatoriedad, se refiere a la forma de selección de las unidades muestrales, es decir, garantiza que no haya efectos externos o condicionantes que afecten a la selección de la muestra. Por tanto, el reto en esta simulación es conseguir una generación de perfiles aleatoria sin pérdida de representatividad.

La idea en la simulación de perfiles consiste en generar números de forma aleatoria entre el rango de valores de cada dimensión. Se puede considerar que las dimensiones toman valores entre el valor mínimo “a” y el máximo “b”. Por tanto, empleando esta generación aleatoria se obtendrá un valor entre “a” y “b” para cada dimensión. A primera vista, la solución a este problema parece sencilla, pues la mayoría de estudios realizados en la Literatura aplican la distribución *Uniforme*, $U(0,1)$, como medio de generación aleatorio. La fórmula para obtener valores entre “a” y “b” en una distribución *Uniforme* es la siguiente:

$$U(a,b) = a + U(0,1) \cdot (b - a) \quad (7.1)$$

No obstante, este método presenta un grave problema, ya que, aunque dicho método garantiza la aleatoriedad en la muestra, ésta puede no ser representativa de la población de individuos. Por tanto, es necesario encontrar otra distribución que cumpla las dos propiedades anteriores. Tal distribución es la distribución *gaussiana* o, también conocida como distribución *Normal*. Cabe destacar que, es ampliamente utilizada en otros campos como la *Sociología*, debido a que produce resultados muy próximos a los reales cuando se trabaja con variables relacionadas con las personas. De hecho, muchos parámetros relacionados con la Naturaleza también se distribuyen bajo una distribución *Normal* ([García Ferrando 1995] y [DeGroot 1986]), e.g., la longitud de un brazo, la altura de un individuo, etc. Veremos ahora un procedimiento para generar números aleatorios que sigan una distribución *Normal* con media μ y desviación típica σ , i.e., $N(\mu, \sigma)$.

En primer lugar, supongamos que una dimensión está formada por valores continuos. El objetivo es generar valores, de forma que no superen un cierto porcentaje p . Por ejemplo, la dimensión “nota media de un estudiante” es una variable continua entre 0 y 10^3 . Supongamos que se quieren generar notas medias de estudiantes tal que el 75% estén por debajo de 6,5. La nota media puede considerarse una variable sociológica que generalmente se distribuye mediante la distribución *Normal*, por

³Esto es cierto en el sistema educativo español, donde las notas de las asignaturas están entre 0 y 10. En otros países las notas pueden variar entre valores distintos, e.g., 0 y 20.

tanto, una forma de simularlas sería generar números mediante esta distribución. Es bien conocido que, para una variable aleatoria X con distribución $N(\mu, \sigma)$ ⁴ se cumplen las siguientes relaciones:

$$\begin{aligned} P(X \leq x) &= p \\ P(X > x) &= 1 - p \end{aligned} \tag{7.2}$$

El proceso para hallar x consiste en transformar la variable X a una *Normal tipificada*, o también conocida como $N(0, 1)$, y utilizar las tablas de la *Normal* (véase las tablas de la distribución $N(0, 1)$ en [Owen 1962]). De esta forma, se obtiene una nueva variable $Z \sim N(0, 1)$. Se sabe que para tipificar una variable, basta con restarle la media y dividirla por la desviación típica, es decir, $Z = (X - \mu)/\sigma$ ⁵. Realizando estas modificaciones en la fórmula anterior se obtiene:

$$\begin{aligned} P\left(Z \leq \frac{x - \mu}{\sigma}\right) &= p \equiv \\ &\equiv P(Z \leq z) = p \end{aligned} \tag{7.3}$$

Una vez se ha hallado z , hay que revertir la tipificación para conocer el valor de x , que se realiza mediante la siguiente fórmula, ampliamente conocida en el campo de la Estadística:

$$x = z \cdot \sigma + \mu \tag{7.4}$$

Teniendo esta base estadística, el problema consiste en hallar μ y σ , tal que se cumplan los condicionantes anteriores. Realizando una serie de pruebas se obtuvo que los valores que garantizan los supuestos anteriores son 5 y 2 respectivamente. De esta forma, X es una variable aleatoria que se distribuye por una $N(5, 2)$. Esta distribución garantiza la generación de valores entre 0 y 10, ya que, la probabilidad de generar valores menores que 0 es de 0,0062⁶, y la probabilidad de generar valores mayores que 10 es de 0,0062⁷. Además, calculando el valor de x para una probabilidad de 0,75, se obtiene el valor 6,34, que está próximo al valor 6,5. Para conseguir generar elementos de manera que el porcentaje solicitado en media sea p , una solución consiste en trasladar la *Normal* hacia la derecha o hacia la izquierda una serie de unidades, que llamaremos d . Esta distancia, d , en nuestro caso es $|\mu - x|$, es decir, el número de unidades que hay que desplazar la media de la *Normal*. Por tanto, en el caso de que $\mu \leq x$, o en otras palabras, si $p \geq 0,5$, se desplaza la media hacia la izquierda d unidades, y si $p < 0,5$, se desplaza la media hacia la derecha d unidades. Volviendo al ejemplo, y realizando los cálculos siguientes, se obtiene que la nueva media, μ' , cuyo valor es 3,66:

$$\begin{aligned} d &= |\mu - x| = |5 - 6,34| = 1,34 \\ p &= 0,75 > 0,5 \Rightarrow \mu' = \mu - d = 5 - 1,34 = 3,66 \end{aligned}$$

⁴De forma matemática se expresa de la siguiente forma: $X \sim N(\mu, \sigma)$.

⁵La variable Z sigue siendo *Normal*, pues si una variable aleatoria X se distribuye mediante una distribución *Normal*, entonces cualquier función lineal de X también se distribuirá como una *Normal* (véase teorema en [DeGroot 1986, pp. 253-255]).

⁶ $P(X \leq 0) = P(Z \leq (0 - 5)/2) = P(Z \leq -2,5) = P(Z > 2,5) = 1 - P(Z \leq 2,5) = 0,0062$

⁷ $P(X > 10) = P(Z > (10 - 5)/2) = P(Z > 2,5) = 1 - P(Z \leq 2,5) = 0,0062$

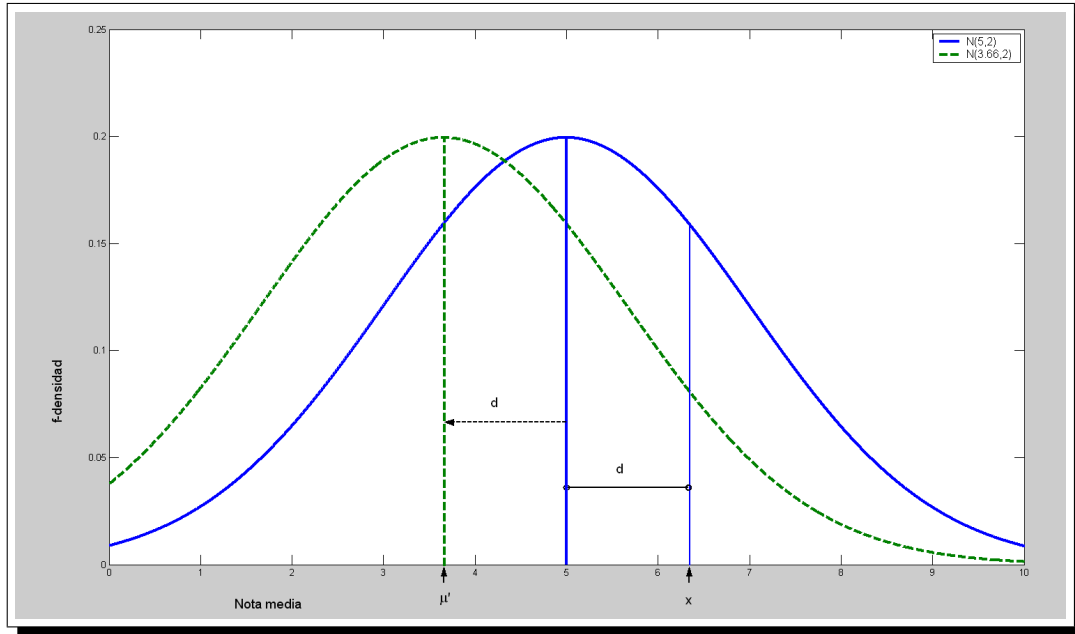


Figura 7.8: Traslación de una distribución Normal d unidades hacia la izquierda

La figura 7.8 ilustra de forma gráfica la traslación de una distribución $N(5, 2)$ a una $N(3,66, 2)$.

Esta forma de simulación es sólo válida para variables continuas; sin embargo, es práctica habitual en el ámbito de los sistemas *e-Learning* utilizar variables de tipo discreto, como por ejemplo, conocimiento previo (novato, avanzado), edad (joven, mayor), idioma (inglés, español, alemán), etc. Con el objetivo de generar valores discretos para estas variables, se modificó la forma anterior de simulación en el siguiente sentido. El objetivo perseguido en este caso es generar valores que siguen una *Normal*, y asignar intervalos de estos valores para cada categoría de la dimensión. Sea una variable (dimensión) con tres categorías, $v1$, $v2$, $v3$, y $p1$, $p2$, $p3$ ⁸, las proporciones a generar para cada categoría, el objetivo es encontrar los intervalos adyacentes de la *Normal* que satisfacen estas proporciones. Utilizando las ecuaciones 7.3 y 7.4 se pueden obtener los valores para $x1$, $x2$ y $x3$ que garantizan las probabilidades siguientes:

$$P(X \leq x1) = p1$$

$$P(X \leq x2) = p1 + p2$$

$$P(X \leq x3) = p1 + p2 + p3$$

⁸La suma de las proporciones es 1.

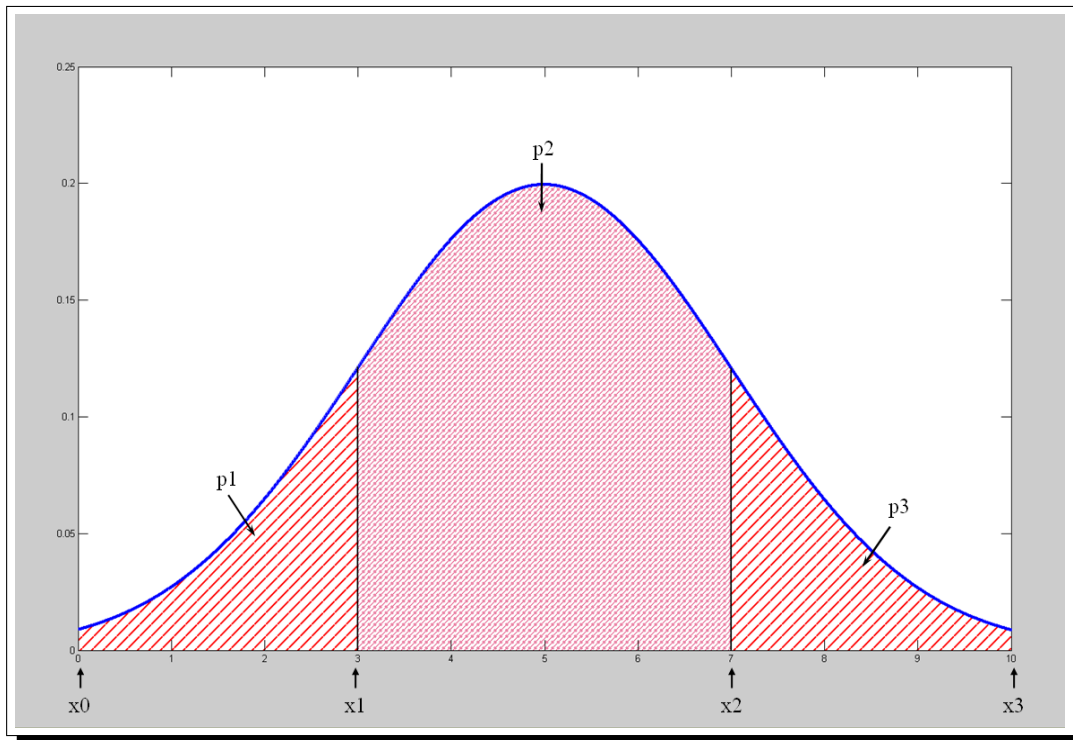


Figura 7.9: Intervalos de la distribución Normal para tres categorías

Por tanto, los intervalos para cada categoría serán:

$$v1 \in [x0, x1]$$

$$v2 \in (x1, x2]$$

$$v3 \in (x2, x3]$$

La figura 7.9 muestra la forma de repartir los valores de la distribución *Normal* en las proporciones $p1$, $p2$, y $p3$. Se puede observar que en esta figura se han sombreado tres áreas por debajo de la *Normal*, correspondientes a las proporciones $p1$, $p2$ y $p3$. Asimismo, se muestran los tres intervalos correspondientes a las proporciones anteriores. Nótese, que el proceso anterior se ha presentado para una dimensión con tres categorías, pero se puede generalizar para n categorías de la siguiente forma:

$$\begin{aligned}
 P(X \leq x1) &= p1 \Rightarrow x1 ; v1 \in [x0, x1] \\
 P(X \leq x2) &= p1 + p2 \Rightarrow x2 ; v2 \in (x1, x2] \\
 &\dots \\
 P(X \leq xn) &= p1 + p2 + \dots + pn \Rightarrow xn ; vn \in (x(n-1), xn]
 \end{aligned}
 \tag{7.5}$$

Hay que indicar, que en la generación de perfiles la escala que se utiliza para las dimensiones es de 0 a 10. Como se vio anteriormente, la *Normal* que más se ajusta en la generación de valores entre 0 y 10 es la $N(5,2)$. Veamos un ejemplo para ilustrar

el funcionamiento de la generación expuesta en la ecuación 7.5. Supongamos que se desean simular 100 perfiles de la dimensión “idioma de curso”, cuyas categorías son “inglés” ($v1$), “español” ($v2$) y “alemán” ($v3$), de forma que el 65 % de los perfiles sean con idioma “español”, y el resto de los idiomas estén distribuidos en igual proporción en los perfiles. Resumiendo los datos del problema: $p2 = 0,65$ y $p1 = p3 = 0,175$. Por lo tanto, realizando los siguientes cálculos se obtienen los valores de $x1$, $x2$ y $x3$:

$$P(X \leq x1) = 0,175 \Rightarrow x1 = 3,09 ; v1 \in [0, 3,09]$$

$$P(X \leq x2) = 0,825 \Rightarrow x2 = 6,84 ; v2 \in (3,09, 6,84]$$

$$P(X \leq x3) = 1 \Rightarrow x3 = 10 ; v3 \in (6,84, 10]$$

Una vez finalizado el proceso de cálculo de los intervalos se generan 100 valores aleatorios de la $N(5,2)$, comprobando para cada valor en qué intervalo está incluido, y asignándolo a la categoría correspondiente. Por ejemplo, los primeros cinco valores fueron: 3,16, 4,86, 2,12, 5,35 y 7,28, consecuentemente, los idiomas generados son: *español, español, inglés, español y alemán*.

7.3.3.2. Generación de logs

La generación de logs se realiza aplicando un modelo de probabilidad dependiente del tipo de perfil del estudiante y los signos generados. En otras palabras, los valores generados en las dimensiones de un perfil condicionan que el estudiante responda de forma correcta a una determinada cuestión, o que abandone de forma prematura una actividad antes de ser acabada. De esta forma, la probabilidad de que un estudiante falle una actividad viene dada por:

$$P(\text{fallar}_{\text{actividad}}) = P(\text{signo}) + (1 - P(\text{signo})) \cdot (1 - P(\text{exito})) \quad (7.6)$$

Así, la probabilidad de fallar una determinada actividad está relacionada con la probabilidad de que el perfil del estudiante esté afectado por el signo, y la probabilidad de éxito en las actividades (valor fijado por el evaluador, “Success probability”). Por tanto, el simulador comprueba en primer lugar si el perfil del estudiante y la actividad están afectados por algún signo, en caso de que lo estén, se verifica el tipo de signo. Se genera un número aleatorio para comprobar si el signo afectará al estudiante (probabilidad del signo). En el caso afirmativo, si el signo fue *abandonar la actividad* se fijan los valores de *grade* y *complete* a 0,0; pero, si fue *fallar*, se genera un nuevo número aleatorio entre 0 y la puntuación mínima para superar la actividad para la variable *grade*, y se asigna a *complete* el valor de 0,0. En caso negativo, se asigna el valor a la variable *grade* dependiendo de la probabilidad de éxito, y la variable *complete* toma el valor de 1,0 si hubo éxito, y 0,0 en caso contrario.

En esta fase se generan tantos ficheros de logs como estudiantes se hayan generado. Un fichero de logs está compuesto por tres partes: perfil del estudiante (*user-model*), interacciones del estudiante con las actividades (*log-activity*) y actualizaciones del perfil del estudiante (*log-update*). El siguiente esquema ilustra la estructura de un fichero de logs:


```
<log-root>
  <user-model>
    <dimension name=" " value=" " \>
      ...
  </user-model>
  <log-activity>
    <log
      activity=" "
      activityTime=" "
      activityType=" "
      complete=" "
      grade=" "
      message=" "
      numvisits=" "
      syntheticTime=" "
      timestamp=" "
      action=" " \>
      ...
  </log-activity>
  <log-update>
    <update
      name=" "
      value=" " \>
      ...
  </log-update>
</log-root>
```

La primera parte, perfil del estudiante, está compuesta por tantos elementos *dimension* como atributos o dimensiones tenga el perfil del estudiante. Este perfil, a su vez, está compuesto por las dimensiones relativas al curso y los atributos estáticos del estudiante (lenguaje, edad, etc.). Cada uno de estos elementos está compuesto por el par *name-value*, que definen el identificador de la dimensión y el valor que tiene asignado dicha dimensión.

La parte central, está compuesta por tantos elementos *log* como interacciones haya realizado el estudiante con el sistema adaptativo. Así, los parámetros que forman este elemento son:

- *activity*: representa la actividad que está realizando el estudiante.
- *activityTime*: tiempo desde que se inició la actividad hasta que se finalizó o abandonó (tomando como referencia el tiempo indicado en *syntheticTime*).
- *activityType*: hace referencia al tipo de actividad.
- *complete*: representa el grado de completación de la actividad.

- *grade*: indica el grado de éxito del estudiante en la actividad.
- *message*: reservado para *SimuLog*.
- *numvisits*: número de veces que se ha realizado la actividad. Si es la primera visita, *numvisits* tendrá de valor 1.
- *syntheticTime*: tiempo generado por *SimuLog* para simular un tiempo lo más aproximado al tiempo real.
- *timestamp*: muestra el tiempo absoluto de la máquina.
- *action*: variable que indica el tipo de comportamiento del estudiante. Puede tomar los siguientes valores:
 - *START-SESSION*: inicio de sesión.
 - *FIRSTVISIT*: inicio de actividad.
 - *REVISIT*: inicio de una actividad ya visitada, pero todavía no completada.
 - *LEAVE-ATOMIC*: abandono o finalización de una actividad elemental.
 - *LEAVE-COMPOSITE*: abandono o finalización de una actividad compuesta.

La última parte de los ficheros de logs es la relacionada con las actualizaciones del perfil del estudiante (*log-update*). Los parámetros que la componen no están fijados *a priori*, ya que dependen de si se han producido modificaciones en algún atributo del perfil del estudiante. Por tanto, los elementos que componen esta parte son las dimensiones del perfil que hayan actualizado su valor, existiendo tantos elementos *update* como actualizaciones se hayan producido.

7.3.4. Procesador de logs

El procesador de logs es el módulo encargado de extraer los datos relevantes de cada fichero de logs, y ordenar estos datos en un fichero unificado (con formato *CSV*), de forma que, éste pueda ser procesado por una herramienta o método de evaluación. Este módulo genera tres tipos de ficheros unificados: fichero con las interacciones de los estudiantes (*logs*), con el recorrido de actividades de los estudiantes (*patterns*) y con el resultado de las actividades realizadas por los estudiantes (*scores*).

En esta investigación, únicamente se utiliza el fichero *logs* para valorar la eficacia del método *GeSES*. Sin embargo, *SimuLog* se desarrolló no sólo para evaluar el método que se ha presentado en este estudio, sino también para evaluar otros tipos de herramientas o métodos de evaluación, en los que pueden ser útiles los otros dos tipos de ficheros unificados.

7.3.4.1. Fichero *logs.csv*

El fichero está formado por una primera línea, donde se indican los nombres de las variables (separadas por comas), y el resto de las líneas, formadas por los datos correspondientes a estas variables. Cada fila, excepto la primera, contiene una interacción del estudiante. Se almacenan los datos relativos al identificador del estudiante, perfil del estudiante (edad, lenguaje, conocimiento previo, tipo de estilo de aprendizaje, etc.), y la actividad realizada (nombre de la actividad, tipo de actividad, puntuación en la actividad, grado de compleción, tiempo empleado, etc.). Por tanto, el tamaño de este fichero dependerá del número de estudiantes y número de actividades que forman el curso.

7.3.4.2. Fichero *patterns.csv*

Contiene los datos relativos al identificador del estudiante, perfil del estudiante y la secuencia de nombres de las actividades realizadas, en el mismo orden en el que fueron realizadas. El fichero está compuesto por tantas filas como estudiantes se hayan solicitado simular. Cada fila contiene el identificador del estudiante, características del estudiante y nombre de las actividades realizadas por el estudiante. A continuación, se presenta la generación de 20 estudiantes para un curso sobre números enteros (compuesto por 25 actividades) con edades comprendidas entre 13 y 15 años, el idioma de los contenidos puede ser presentado en: inglés, español, alemán e italiano, y el conocimiento previo de los estudiantes puede tener tres valores: novato, normal y avanzado. Hay que indicar que, en este fichero, generalmente las filas no contienen el mismo número de columnas. Esto es debido a que, en un curso adaptativo dos estudiantes pueden realizar distintos recorridos de actividades, dependiendo del tipo de adaptación realizada en el sistema.

```
e0,13,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,valorAbs,v1,ev1_b1,o2,...
e1,12,german,normal,curso_enteros,ordenacion,o1,eo1_n1,o2,valorAbs,v1,ev1_n1,...
e2,13,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,valorAbs,v1,ev1_b1,o2,...
e3,14,german,advanced,curso_enteros,ordenacion,o1,eo1_a1,o2,valorAbs,v1,ev1_n1,...
e4,14,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,o2,valorAbs,v1,ev1_b1,...
e5,15,german,novice,curso_enteros,ordenacion,o1,eo1_b1,o2,valorAbs,v1,ev1_b1,...
e6,15,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,valorAbs,v1,ev1_b1,o2,...
e7,12,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,o2,valorAbs,v1,ev1_b1,...
e8,15,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,o2,valorAbs,v1,ev1_b1,...
e9,12,english,novice,curso_enteros,ordenacion,o1,eo1_b1,o2,valorAbs,v1,ev1_b1,...
e10,13,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,o2,valorAbs,v1,ev1_b1,...
e11,12,english,novice,curso_enteros,ordenacion,o1,eo1_b1,valorAbs,v1,ev1_b1,o2,...
e12,14,spanish,advanced,curso_enteros,ordenacion,o1,eo1_a1,o2,valorAbs,v1,ev1_n1,...
e13,12,spanish,normal,curso_enteros,ordenacion,o1,eo1_n1,o2,valorAbs,v1,ev1_n1,...
e14,12,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,o2,valorAbs,v1,ev1_b1,...
e15,15,spanish,novice,curso_enteros,ordenacion,o1,eo1_b1,valorAbs,v1,ev1_b1,o2,...
e16,14,italian,advanced,curso_enteros,ordenacion,o1,eo1_a1,o2,valorAbs,v1,ev1_n1,...
```

```
e17,15,spanish,normal,curso_enteros,ordenacion,o1,eo1_n1,valorAbs,v1,ev1_n1,o2,...
e18,14,german,advanced,curso_enteros,ordenacion,o1,eo1_a1,valorAbs,v1,ev1_n1,o2,...
e19,14,spanish,normal,curso_enteros,ordenacion,o1,eo1_n1,o2,valorAbs,v1,ev1_n1,
suma,s1,es1_n1,s2,es2_n1,resta,r1,er1_b1,er1_a1,r2,er2_a1,er2_b1,parentesis,p1,
ep1_b1,ep1_a1,multiplicacion,m1,m2,em2_a1,d1,operCombinadas,c1,c2,c3,ec3_n1,
ec3_a1
```

7.3.4.3. Fichero *scores.csv*

Este fichero contiene los datos relativos al identificador del estudiante, perfil del estudiante y puntuación obtenida en las actividades realizadas. El fichero está compuesto por una primera fila, donde se indican los nombres de las variables (separadas por comas), y el resto de las filas contiene los valores asociados a estas variables para cada estudiante. Por tanto, el fichero contiene tantas filas + 1, como el número de estudiantes a simular. A continuación, se presenta el fichero generado a partir de los datos del anterior ejemplo. Hay que señalar que, en este fichero es probable que aparezcan valores “0.0”, indicando que la actividad no ha sido realizada, y por tanto, no hay puntuación disponible para ella. El razonamiento es el mismo que en el anterior tipo de fichero, debido a que en un curso adaptativo varios estudiantes pueden seguir distinto recorrido de actividades, pueden existir actividades que nunca sean realizadas por los estudiantes. Por ejemplo, en un curso adaptativo pueden existir actividades de refuerzo, que no necesiten ser realizadas por estudiantes con conocimiento avanzado. Como consecuencia, la puntuación de estas actividades para estos estudiantes en este fichero es “0.0”.

```
id,age,language,experience,grade_curso_enteros,grade_ordenacion,grade_o1,...
e0,13,spanish,novice,0.7464285714285713,0.645,0.29000000000000004,0.48,...
e1,12,german,normal,0.7404761904761904,1.0,1.0,0.65,0.0,1.0,1.0,0.0,0.0,...
e2,13,spanish,novice,0.8521428571428574,1.0,1.0,1.0,1.0,1.0,0.0,0.0,0.0,...
e3,14,german,advanced,0.758095238095238,1.0,1.0,0.8099999999999999,0.0,...
e4,14,spanish,novice,0.770952380952381,0.73,0.46,0.975,0.46,1.0,0.0,...
e5,15,german,novice,0.8583333333333333,1.0,1.0,0.845,1.0,1.0,0.0,0.0,...
e6,15,spanish,novice,0.7866666666666665,0.825,0.65,0.98,0.65,1.0,0.0,...
e7,12,spanish,novice,0.7335714285714285,0.945,0.8899999999999999,0.965,...
e8,15,spanish,novice,0.8066666666666665,0.63,0.26,0.675,0.26,1.0,0.0,...
e9,12,english,novice,0.593095238095238,0.655,0.31,0.405,0.31,1.0,0.0,...
e10,13,spanish,novice,0.7573809523809523,0.755,0.51,0.45999999999999996,...
e11,12,english,novice,0.8621428571428572,1.0,1.0,0.995,1.0,1.0,0.0,0.0,...
e12,14,spanish,advanced,0.8254761904761904,0.93,0.8600000000000001,0.89,...
e13,12,spanish,normal,0.6935714285714285,0.65,0.30000000000000004,0.595,...
e14,12,spanish,novice,0.6352380952380952,0.85,0.7,0.31500000000000006,0.7,...
e15,15,spanish,novice,0.7864285714285716,0.88,0.76,0.735,0.76,1.0,0.0,0.0,...
e16,14,italian,advanced,0.845952380952381,0.735,0.47000000000000003,0.940,...
e17,15,spanish,normal,0.7995238095238095,0.975,0.95,0.6699999999999999,...
e18,14,german,advanced,0.8719047619047621,0.8,0.6000000000000001,...
```

```
e19,14,spanish,normal,0.8371428571428571,0.965,0.9299999999999999,0.84,
0.0,1.0,0.9299999999999999,0.0,0.0,0.0,0.73,0.95,0.0,0.73,0.0,0.0,0.95,
0.0,0.410000000000000003,0.410000000000000003,0.0,0.410000000000000003,
0.845,0.72,0.97,0.26,0.0,0.72,1.0,1.0,0.9366666666666666,1.0,0.81,1.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.8633333333333333,1.0,1.0,0.5900000000000001,
0.0,0.0,0.71,0.0,1.0,0.0,0.0,0.81,0.37,0.0,0.97,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.4,0.5900000000000001
```

7.4. *ASquare*

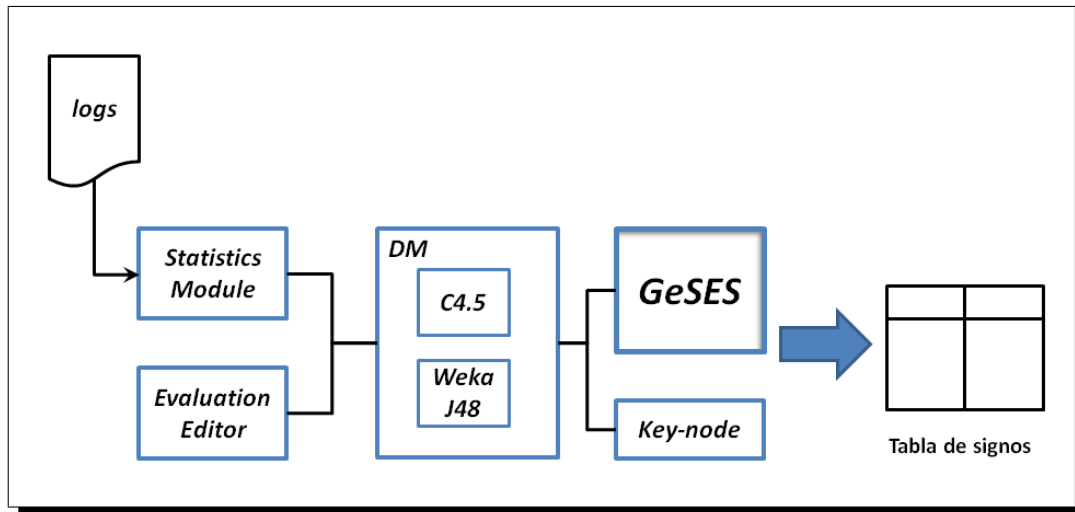
ASquare es la herramienta que da soporte a la metodología, pues contiene implementaciones tanto del método *Key-node* como del método *GeSES*. Esta herramienta proporciona una manera fácil y eficaz de aplicar ambos métodos por parte de los profesores. Actualmente, ha sido desarrollado e implementado en JAVA un prototipo que permite la detección de signos de baja eficiencia de forma *off-line*. Esto quiere decir que *ASquare* requiere que el conjunto de logs a analizar sea estable y completo, es decir, una vez los estudiantes terminen o abandonen las actividades de aprendizaje de un curso *e-Learning* es cuando se puede utilizar esta herramienta. En otro caso, se estarían analizando de forma sesgada las interacciones de una parte del curso.

7.4.1. Elementos de *ASquare*

ASquare está compuesto por cinco módulos o componentes, tal y como se muestra en la figura 7.10. El primer módulo es el *Statistics Module* que es el encargado de realizar estadísticas generales, previas a la aplicación de algún método de detección. Estas estadísticas consisten en un recuento de número de valores para la variable de evaluación y para los atributos. De esta forma, se muestran estos datos al profesor, con el fin de que tenga un conocimiento mínimo sobre los datos.

El siguiente módulo es el *Evaluation Editor*. Dicho módulo es el responsable de adquirir los datos relacionados con la evaluación. En este sentido, se le presentan al profesor una serie de opciones que éste debe escoger, como la variable de evaluación, atributos que son interesantes para la evaluación, y grado de importancia en los conceptos relacionados con la evaluación. Este grado de importancia indicará a la herramienta si el profesor considera que la información obtenida debe tener un alto grado de representatividad, o si se quiere obtener una información más completa, o si se requiere que la información sea muy precisa. No obstante, como estos conceptos no son fáciles de entender la herramienta considera por defecto que todos los conceptos tienen la misma importancia.

El módulo *DM* contiene los procedimientos necesarios para aplicar los árboles de decisión y las reglas de decisión. En este sentido, dicho módulo es el encargado de aplicar los algoritmos *C4.5* y *J48* para obtener los árboles de decisión (*C4.5* y *J48*) y las reglas de decisión (*C4.5*). Por tanto, el módulo *DM* obtiene el árbol de

Figura 7.10: Arquitectura de *ASquare*

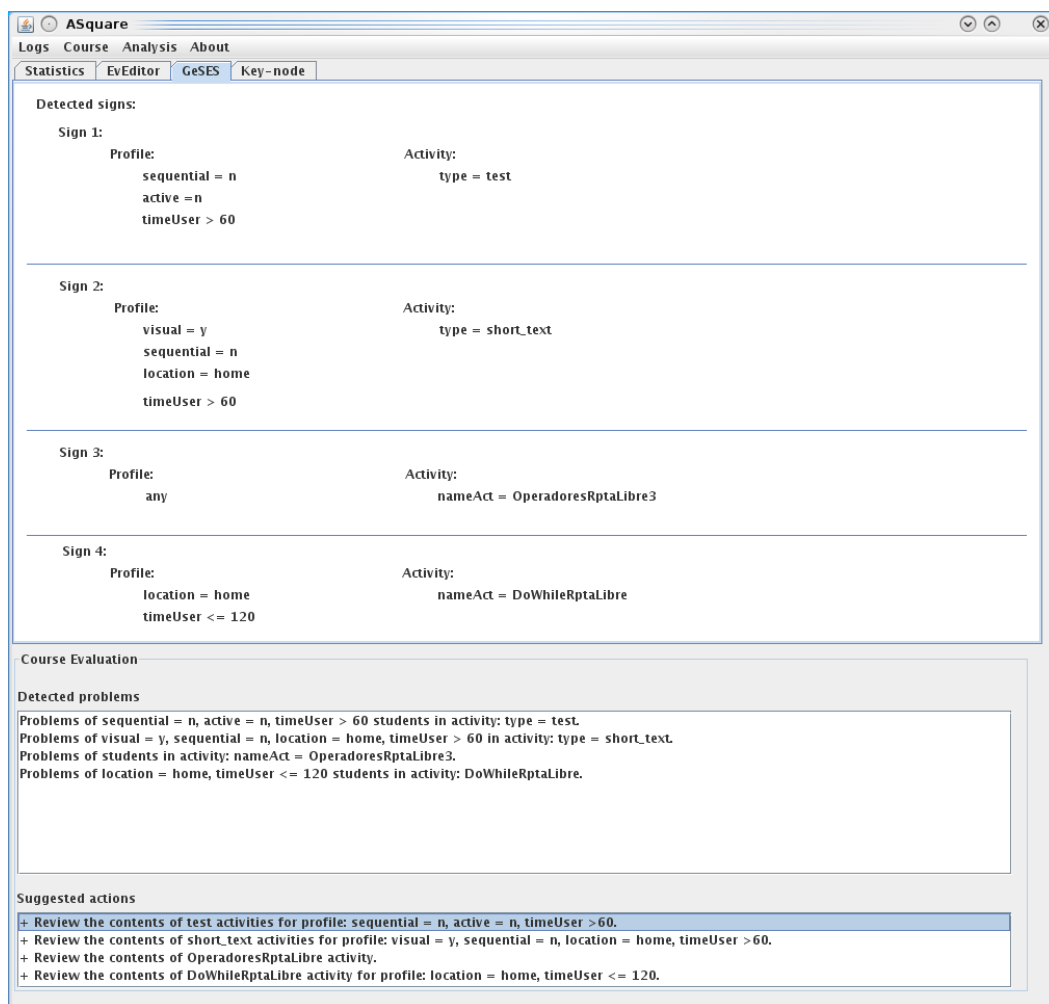
decisión que será procesado por el método *Key-node*, o bien las reglas de decisión y el *ranking* obtenido que será procesado por el método *GeSES*.

El módulo *Key-node* contiene el método *Key-node* y es el encargado de su aplicación. Recibe el árbol de decisión del submódulo *J48*, y lo procesa de acuerdo a los pasos establecidos por el método *Key-node*. Proporciona la información de todos los signos encontrados con la información extraída del árbol. Hay que indicar, que este método se ha ido progresivamente substituyendo por el método *GeSES*, más actual y con mayor eficiencia.

El módulo *GeSES* es el módulo que da soporte al método *GeSES*. En este sentido, recibe las reglas de decisión así como el *ranking* de reglas proporcionado por el submódulo *C4.5*. En él se produce el procesamiento de las reglas según los pasos establecidos en el método *GeSES*. Una vez finalizado este procesamiento proporciona la tabla de signos, que contiene la información relevante a los signos más representativos respecto al grado de importancia de los conceptos fijados en el *Evaluation Editor*.

7.4.2. Interfaz de *ASquare*

La interfaz de *ASquare*, como se muestra en la figura 7.11, está claramente dividida en tres partes: menús, parte central y parte de recomendaciones. Los menús disponibles son: *File*, *Analysis* y *Help*. Con el primero se puede cargar los logs de un curso en el formato *CSV* (opción *load*), o bien se puede abandonar la herramienta (*exit*). Es importante que el fichero de logs esté en el formato *CSV* de forma que la primera fila debe contener los nombres de las variables separadas por comas, la última variable de esta columna debe ser la variable de evaluación, y las siguientes filas deben contener los datos de las variables separados por comas. De no seguir

Figura 7.11: *ASquare*

este formato la aplicación mostrará un mensaje de error indicando que el formato de los logs es incorrecto. El menú *Analysis* contiene los submenús *Settings* y *Execute*. *Settings* muestra al profesor el *Evaluation Editor* para que éste pueda seleccionar las columnas de los datos que quiere evaluar, el método elegido para la evaluación (método *GeSES* o *Key-node*). En caso de elegir el método *GeSES*, se solicita el grado de importancia de los conceptos utilizados para obtener los signos de baja eficacia más representativos. Finalmente, el menú *Help* permite que el profesor sea consciente, si lo desea, de los pasos seguidos por la aplicación, así como los comentarios ofrecidos por *ASquare* (submenú *log*), y muestra la información sobre el autor de la herramienta (submenú *about*).

La parte central está dividida en pestañas que se activan desde el menú principal. Las pestañas posibles son: *Statistics*, *EvEditor*, *GeSES* y *Key-node*.

- *Statistics*: se activa después de cargar los datos (submenú *load*), y presenta la

información de los estadísticos descriptivos correspondientes a la variable de clase y los atributos o variables.

- *EvEditor*: se activa desde el submenú *settings* del menú *Analysis*. Es necesario elegir las opciones de evaluación antes de realizar la evaluación, en caso contrario la herramienta elegirá las opciones por defecto (selección de todos los atributos, utilizar el método *GeSES*, todos los conceptos tienen el mismo grado de importancia). Esta pestaña presenta al profesor la variable de evaluación que se utilizará, los posibles atributos relacionados con la evaluación, el método de evaluación elegido (*GeSES* o *Key-node*), y el grado de importancia de los conceptos.
- *Key-node*: se activa desde el submenú *execute* del menú *Analysis*, sólo si el método de análisis seleccionado en el *EvEditor* fue el método *Key-node*. Se proporciona la información sobre los signos encontrados mediante este método. Hay que indicar, que la parte de recomendaciones o sugerencias sólo se activa cuando esta pestaña está activada.
- *GeSES*: se activa desde el submenú *execute*, sólo si el método de análisis seleccionado fue el método *GeSES*. Proporciona la información resumida de los signos de baja eficacia en el curso *e-Learning* detectados por el método por medio de la tabla de signos. Hay que indicar, al igual que en la anterior pestaña que la parte de recomendaciones o sugerencias sólo se activa cuando esta pestaña está activada.

La parte de recomendaciones ofrece sugerencias de acciones que puede realizar el profesor para solucionar los problemas detectados. Hay que indicar que el motor de sugerencias está en su primera fase de desarrollo y sólo ofrece sugerencias del tipo: revisa los contenidos de una determinada actividad o actividades, revisa los contenidos ofrecidos a determinado perfil de estudiantes, etc.

7.5. Conclusiones

Este capítulo ha mostrado una propuesta para el ciclo de creación y mantenimiento de un curso *e-Learning*, que consiste en añadir en éste las herramientas de evaluación y herramientas de simulación. Asimismo, se ofrecen las descripciones detalladas de: la herramienta de simulación desarrollada, *SimuLog*, y la herramienta de evaluación que da soporte a la metodología, *ASquare*.

En el capítulo anterior se mostró de la necesidad de disponer de una herramienta de simulación para valorar la eficiencia del método *GeSES*. Con este propósito, se desarrolló *SimuLog* y por esta razón en este capítulo se incluyen la descripción y funcionamiento de forma detallada, para que el lector pueda entender de manera completa la forma y los procesos necesarios para generar los logs sintéticos. Es importante señalar que la comprensión de esta herramienta es necesaria para entender los resultados obtenidos en la evaluación del capítulo anterior. Del mismo modo,

se incluyó la descripción de *ASquare*, herramienta que da soporte tanto al método *GeSES* como al método *Key-node*.

Conclusiones y trabajo futuro

Índice de contenidos

8.1. Conclusiones	195
8.2. Limitaciones	200
8.2.1. Consideraciones respecto al método GeSES	200
8.2.2. Consideraciones respecto a SimuLog	201
8.2.3. Consideraciones respecto al proceso de evaluación del método	202
8.2.4. Consideraciones respecto a <i>ASquare</i>	202
8.3. Trabajo futuro	202
8.4. Publicaciones a las que ha dado lugar este trabajo	204
8.4.1. Publicaciones en revistas con índice de impacto	205
8.4.2. Publicaciones en congresos internacionales	205
8.4.3. Publicaciones en congresos nacionales	206
8.4.4. Capítulos de libro	207
8.4.5. Trabajo de Iniciación a la Investigación	207

Este capítulo presenta los principales resultados de investigación, limitaciones de la propuesta, futuras líneas de trabajo y un listado de las publicaciones a las que ha dado lugar este trabajo.

8.1. Conclusiones

Aunque ya a lo largo de este estudio se han ido exponiendo las conclusiones parciales relativas a las cuestiones concretas analizadas, se sintetizarán en esta sección los principales resultados de investigación.

Este trabajo propone un nuevo método para evaluar un curso *e-Learning*. El problema de evaluar este tipo de cursos es muy amplio, y esta investigación se ha concentrado fundamentalmente en el estudio de los sistemas *AEH*. El objetivo principal de esta propuesta fue **facilitar la evaluación de un curso *e-Learning***, en el sentido de diseñar métodos y herramientas para permitir que profesores sin conocimiento técnico específico puedan evaluar la eficacia de sus cursos *e-Learning*, y en especial los adaptativos. Cuando un profesor se enfrenta a la tarea de evaluar si el sistema funciona como él espera, el principal problema es determinar si el sistema es beneficioso para los estudiantes. Este reto plantea el problema de que percibir el

comportamiento de los estudiantes, tal como se hacía en la enseñanza tradicional no es posible. Esto es debido a que las ventajas ofrecidas por estos sistemas (utilización en cualquier localización, dispositivo, espacio de tiempo, entre otras) no hacen posible que el profesor pueda observar físicamente los comportamientos de los estudiantes. Sin embargo, obtener esta información es factible por medio del análisis de las interacciones de los estudiantes. Este análisis presenta dos principales obstáculos para los docentes: cómo pueden analizar los logs, pues contienen gran cantidad de datos, lo que hace difícil su análisis, y determinar qué información es relevante para la evaluación. Con el propósito de dar solución a estos dos problemas se plantearon los siguientes subobjetivos:

1. *Analizar y seleccionar las técnicas que permitan el análisis de grandes volúmenes de datos.* Se analizaron las técnicas y métodos de los principales trabajos que se encontraron en la literatura (capítulo 2). Asimismo, se eligieron las técnicas de *DM* como las más adecuadas para conseguir este objetivo. De entre ellas, se consideró que las más adecuadas inicialmente eran las reglas de asociación y los árboles de decisión (capítulo 3), pues sus resultados son, en un principio, fáciles de interpretar y procesar. Esta última fue más adecuada, pues su interpretación es más fácil por la forma en la que presenta sus resultados. No obstante, los logs analizados en este trabajo generan árboles muy frondosos y difíciles de procesar. Por esta razón, fue necesario encontrar una técnica que ayudara a su procesamiento. Las reglas de decisión, derivadas de los *DTs*, permitieron procesar la información contenida en un árbol de forma eficiente por un ordenador. Por este motivo fueron elegidas para realizar este análisis.
2. *Definir la información que se desea conocer.* Este objetivo fue imprescindible para la evaluación de los cursos *e-Learning* planteada en esta investigación. Como consecuencia, se definieron el tipo de situaciones que pueden indicar una reducción de la eficacia de estos cursos. A este tipo de situaciones se las denominó signos de baja eficacia y se realizó una clasificación de los principales. Las pruebas realizadas aquí sólo incluyen la detección de signos de baja eficacia, referidos al bajo rendimiento académico de los estudiantes (capítulo 4). No obstante, una línea de trabajo futuro consiste en probar la detección de otro tipo de signos.
3. *Desarrollar un método de evaluación a partir de la consecución de los dos objetivos anteriores.* El método de evaluación desarrollado inicialmente utilizó los árboles de decisión. Debido a que hubo experimentos en los que resultaba muy difícil su aplicación, se mejoró el método basándolo en las reglas de decisión y un modelo de selección de aquellas que contienen la información más relevante (capítulo 5). Este modelo se realizó para evitar la sobrecarga de información, ya que de un *DT* se pueden obtener muchas reglas, mostrando sólo las más relevantes. El modelo de selección consistió en generar un nuevo *ranking* a partir del *ranking* de reglas que proporciona el programa *c4.5rules*

y decidir de qué reglas se obtendrá la información que se mostrará al profesor. Un problema consistió en decidir en base a qué criterios se realiza este nuevo *ranking*. En este sentido, se utilizaron los conceptos: cobertura, tamaño y precisión. Se comprobó que estos conceptos son los más interesantes para este objetivo, pues los otros, contenidos en el *ranking*, contienen datos redundantes. Otro problema consistió en establecer un punto de corte a partir del cual se pondera lo buena que es una regla respecto a estos conceptos. Siguiendo esta línea, se desarrolló un proceso de asignación de pesos y se fijaron dos fronteras de selección: diez por ciento y tercer cuartil. Estas fronteras sirvieron para establecer distintos pesos entre las reglas, cuya cobertura, tamaño, o precisión, estuvieran por encima o debajo de las fronteras. La propuesta inicial sólo tuvo en cuenta si una observación (valor del concepto relacionado con una regla) estaba por encima o por debajo de la frontera. Dicho de otro modo, las observaciones por debajo de la frontera obtuvieron un peso mínimo, y a las que estaban por encima se les asignó peso máximo. Se utilizó el peso total, suma de los pesos de la regla en cada concepto ponderados por su grado de importancia, para seleccionar las reglas más relevantes. El método da la libertad de que el profesor fije estos grados de importancia, según lo que considere más importante en la detección. El proceso de asignación mostró dos problemas, uno relacionado con los empates en los pesos totales, y otro con que determinadas reglas no fueron seleccionadas por haber sido infravaloradas. Con el propósito de mejorarlo, el método se fue modificando paulatinamente a partir de los resultados de las pruebas realizadas con distintos tipos de datos. La primera mejora consistió en aplicar un proceso de selección distinto para las reglas cuyos conceptos quedaron cercanos a la frontera, pero por debajo. De esta forma, dicho proceso se hizo de forma gradual por debajo de la frontera utilizando una función escalonada dependiente de σ . Los resultados obtenidos con esta nueva forma de asignación fueron mejores, pero aún era necesario mejorar el proceso. Se advirtió que observaciones cercanas y por encima de la frontera obtenían el mismo peso que las más alejadas. Además, la frontera del diez por ciento parecía prescindible, pues penalizaba o valoraba en demasía determinadas observaciones alejadas. Por este motivo, se seleccionó la frontera tercer cuartil y se siguió el enfoque de asignación de pesos mediante una función continua. Se buscó dicha función de forma que asignara pesos bajos a observaciones alejadas y por debajo de la frontera, pesos intermedios a las cercanas y altos a las más alejadas por encima. Se comprobó que una función que cumple estos requisitos es la sigmoïdal, pero fue necesario adecuarla de forma que su dominio fueran sólo números positivos, puesto que los conceptos de una regla son siempre positivos, y que cambiara de curvatura en la frontera de selección. Este proceso constituyó una parte importante del método *GeSES*, y fue aplicado con éxito a un conjunto de datos de estudiantes reales (capítulo 6).

4. *Presentar la información extraída con el método de forma que sea entendible*

por un profesor. Este objetivo estuvo muy relacionado con el anterior. Las mejoras llevadas a cabo se realizaron con el fin de que el método *GeSES* fuera capaz de seleccionar la información más relevante. Este método sirvió para seleccionar las reglas de decisión que contienen la información más relevante para la evaluación. La información extraída de estas reglas se resumió en una tabla dividida en dos columnas, referidas a la información sobre el perfil de estudiantes que experimentaron un problema y el contexto en el que se produjo: la actividad, el tipo de actividad, el dispositivo utilizado, la localización, etc. Ésta se denominó tabla de signos, cuyo objetivo fue proporcionar la información extraída por el método de forma que pueda ser entendible por un profesor, dado que no siempre se dispone del conocimiento necesario para entender los resultados de las técnicas de análisis de logs. En líneas generales, esta tabla informa sobre los problemas más relevantes detectados en el sistema, de forma que el profesor pueda tomar las acciones que considere oportunas para resolverlos: puede modificar la actividad problemática, añadir nuevas actividades de refuerzo, entre otras cosas. Es relevante destacar que los signos detectados informan al profesor sobre posibles problemas encontrados en el curso, de los que generalmente no era consciente. Por tanto, esta información puede ser muy útil para las futuras mejoras del curso.

5. *Evaluar la eficacia del método.* Este objetivo fue de vital importancia para conocer los límites propios del método y medir su fiabilidad bajo determinadas condiciones. Se propuso utilizar métodos de simulación para comprobar la eficacia de herramientas o métodos de evaluación. Éste fue el motivo por el que se desarrolló *SimuLog*, cuyo fin no es otro sino ayudar a valorar la eficacia del método *GeSES*. Este simulador es capaz de producir logs que incluyen signos de baja eficacia generados de forma artificial (capítulo 7). Por tanto, es posible indicar el tipo de perfiles a generar, número de estudiantes, determinados parámetros generales (tiempo medio empleado, porcentaje de éxito medio, probabilidad de sesión) y definir los signos de baja eficacia. El proceso de evaluación del método consistió en: i) diseñar un curso sintético con unas características muy sencillas para evitar ruido en los datos, ii) diseñar un conjunto de perfiles de estudiantes, iii) diseñar una lista de signos de baja eficacia, iv) generar los logs del conjunto de perfiles siguiendo el curso sintético y conteniendo los signos de la lista, v) aplicar el método *GeSES* al conjunto de logs generado, y vi) observar los signos que se encontraron con el método. Se generaron logs para distinto número de estudiantes, empezando en 20 estudiantes y terminando en 6000. En los experimentos realizados se comprobó que el método empezó a producir resultados fiables a partir de 100 estudiantes, y esta fiabilidad aumentó, como se esperaba, a medida que el número de estudiantes era mayor. Esto es lógico, pues bajo estas condiciones (los porcentajes de cada perfil permanecieron constantes y los porcentajes de signos también) un mayor número de datos produce reglas de mejor calidad. Además, es posible que este aumento de datos produzca que aparezcan

determinadas reglas que no lo hacían antes por falta de datos. Tal situación se detectó a partir de un tamaño de 1200 estudiantes, en la que el método empezó a detectar un mayor número de signos. Además, la calidad de los signos detectados fue mayor a partir de 2100 estudiantes, pues se detectaron no sólo un mayor número de signos correctos sino también completos, si bien, el tamaño de 300 estudiantes fue suficiente para encontrar una detección completa de algún signo. También es importante considerar en qué proporción se encuentran los perfiles afectados por un signo. En este sentido, se comprobó que el método consiguió detectar signos cuya presencia era al menos del 40 % en el perfil afectado. Incluso, el método consiguió detectar este tipo de signos en perfiles minoritarios. Cabe destacar que si bien estos números surgen de analizar una situación (logs sintéticos), se cree que la configuración analizada representa adecuadamente el contexto de situaciones reales. Por lo tanto, sus resultados pueden ser usados como guía para la utilización del método.

6. *Implementar una herramienta de evaluación con la que se pueda aplicar el método.* Una vez que se diseñó el método *GeSES*, se comprobó su aplicación con datos reales, se valoró su eficacia, y el siguiente paso consistió en desarrollar una herramienta que sirviera de soporte. *ASquare*, nombre que se le dio a esta herramienta, es capaz de analizar los logs y de extraer mediante la aplicación del método *GeSES* la información relevante para los profesores. Por tanto, esta herramienta satisface el objetivo principal de esta tesis, ya que facilita la tarea de evaluar un sistema o curso *e-Learning*. Además, esta herramienta proporciona sugerencias en base a los signos detectados, que informan al profesor sobre las posibles acciones que son necesarias, como por ejemplo, “revisar los contenidos de la actividad A1”, “revisar las actividades de tipo T1 para los perfiles P1”, “revisar la actividad A2 para el perfil P1”, etc.

El principal aporte que realiza esta tesis es proporcionar métodos y herramientas a los profesores para que puedan evaluar la eficacia de sus cursos *e-Learning*. No obstante, el método desarrollado no resuelve el problema de la evaluación de cursos *e-Learning*, pero sí supone un pilar importante que puede permitir solventar este problema en el futuro. En este sentido el método consta de una serie de pasos, especificados de forma concreta y precisa, evitando ambigüedades, para que sea más fácil detectar determinado tipo de situaciones que para el profesor pueden pasar desapercibidas. Esta contribución se puede dividir en tres:

1. *Evaluación de un sistema o curso e-Learning mediante el análisis de logs.* En este trabajo se ha demostrado que el análisis de logs es útil para detectar aquellas situaciones que se han denominado signos de baja eficacia.
2. *Aplicación de las reglas de decisión para la detección de signos de baja eficacia.* El trabajo presentado ha demostrado que las *DR* es una técnica no sólo útil para realizar predicciones, sino que se pueden utilizar para detectar signos de baja eficacia.

3. *Selección de los signos de baja eficacia más relevantes para los profesores.* El método *GeSES* ha demostrado que es capaz de seleccionar la información que pueda resultar más útil para los profesores. En este sentido, se les informa de un conjunto reducido de los puntos débiles del sistema/curso.

Otra importante contribución que se realiza es la *propuesta de valoración de la eficacia de una herramienta o método de evaluación*. Se presentó y probó una nueva forma de evaluar la eficacia de un método o herramienta de evaluación mediante técnicas de simulación. De esta forma, es posible valorar si los resultados proporcionados por el método o herramienta son fiables. En este sentido, se presentó una manera de conocer los límites del método *GeSES*.

8.2. Limitaciones

Este trabajo ha presentado, entre otras cosas, un nuevo método para la evaluación de sistemas *e-Learning*, una herramienta de simulación de logs, un proceso de evaluación del método y la herramienta que lo da soporte. No obstante, es importante analizar las distintas limitaciones que se pueden encontrar en cada uno de estos elementos. Respecto al método *GeSES* las principales limitaciones se deben a las propias de las reglas de decisión y el modelo de selección. Con respecto a *SimuLog* sus limitaciones están relacionadas con la Ley de los Grandes Números y el modo de generar los perfiles. El proceso de evaluación del método *GeSES* está limitado fundamentalmente por las características del simulador, y *ASquare* hereda las limitaciones del método *GeSES*. En los siguientes párrafos se discuten las limitaciones de cada uno de estos elementos.

8.2.1. Consideraciones respecto al método GeSES

El método propuesto tiene las limitaciones propias de las reglas de decisión y el modelo matemático de selección. En relación a las reglas de decisión es importante que sean capaces de clasificar los datos de entrenamiento con el menor error posible. De este modo, cada conjunto de datos es caracterizado de forma diferente, y esto hace posible que un signo pueda ser definido de forma correcta. Debido a que las reglas derivan de los árboles de decisión, y en éstos es muy importante la cantidad y distribución de los datos, es necesario satisfacer estos dos factores para obtener buenas reglas. El adjetivo “buenas” debe ser entendido respecto a que las reglas tengan amplia cobertura, su precisión sea adecuada y su tamaño sea cercano al número de atributos evaluados. Es cierto que la variabilidad en los datos es difícil de controlar, pues depende en gran medida de la población de estudiantes que genera estos datos. Además, conocer la distribución de los datos no es una tarea fácil. Según expone [Quinlan 1993, cap. 10] se puede utilizar un modelo geométrico en el que los casos serían representados como vectores de números reales. Siguiendo esta idea un clasificador se puede entender como una división de las distintas regiones espaciales que agrupan a elementos de una clase. Esta noción puede ser generalizada para

atributos discretos (caso de los árboles de decisión *C4.5*), de forma que un atributo se representa en un eje, y es claro que cada caso se representa como un punto en el espacio de ejes. Esto nos da una idea de lo difícil que es saber si el árbol es la técnica apropiada, pues el conocimiento de la distribución de los datos es complejo (4 atributos serían representados en espacios de cuatro dimensiones). El tamaño sí es una variable controlable, y debe ser tenida en cuenta en el momento de interpretar los resultados obtenidos por el método *GeSES*. Se comprobó que tamaños muy pequeños, por debajo de 100 estudiantes, teniendo en cuenta que existen distintos perfiles de estudiantes que realizan el curso *e-Learning*, unos más numerosos que otros, no garantizan que los resultados obtenidos con el método *GeSES* tengan una cierta fiabilidad. No se ha comprobado la fiabilidad si sólo existiera un único perfil de estudiantes en los logs, probablemente 100 estudiantes no serían suficientes para obtener resultados fiables.

Respecto al modelo matemático de selección, que efectúa la asignación de pesos mediante una modificación de la función sigmoideal, se pueden encontrar limitaciones cuando las reglas obtenidas tienen conceptos con poca variabilidad (valores de los conceptos de cada regla = observaciones). Por ejemplo, es muy probable que exista poca variabilidad respecto al tamaño de la regla si el número de atributos en el modelo es de tan sólo 2, pues los tamaños variarán de 1 a 2. En esta situación, recordando la fórmula 5.22 veremos que en este caso la diferencia entre los dos cuartiles será 0, con lo que esta fórmula no será aplicable. Se comprobó que una solución a este problema era asignar los pesos utilizando la fórmula inicial, ya que al no existir prácticamente variabilidad no tiene sentido una aplicar una asignación gradual. Sin embargo, es muy poco probable que este problema se manifieste en los conceptos cobertura y precisión, pues por la forma de construir las reglas normalmente se obtendrán reglas con mayor cobertura que otras, y del mismo modo ocurrirá con la precisión. Por otro lado, el sobreajuste no es un problema para el método. Es más, cuanto más ajustado esté el modelo a los datos de entrenamiento mejores resultados se obtendrán con el método. Esto es así porque en este trabajo las *DR* no se utilizan para hacer predicciones, sino para examinar los datos mediante su clasificación. Por tanto, el sobreajuste no es un problema que preocupe en exceso.

8.2.2. Consideraciones respecto a SimuLog

SimuLog se apoya en la *Ley de los Grandes Números*, por lo que es necesario simular gran cantidad de datos para que las proporciones solicitadas en la simulación se aproximen a las de los datos simulados. Los perfiles se simulan bajo la premisa de que muchas características físicas humanas (peso, altura, longitud del brazo, etc.) se distribuyen mediante una distribución Normal. En consecuencia, se infirió que otras variables cognitivas inherentes al ser humano (e.g. conocimiento previo) también se distribuyen bajo una Normal. Así, el simulador genera los perfiles por medio de distribuciones Normales, de forma que los datos generados sean lo más aproximados posibles a los reales. Por tanto, es importante asegurarse de que los datos a simular se distribuyen bajo una Normal, pues en caso contrario las simulaciones se alejarían

de la realidad.

8.2.3. Consideraciones respecto al proceso de evaluación del método

El proceso de evaluación del método está limitado por el propio simulador, en el sentido de que se necesita un número amplio de estudiantes simulados para conseguir un mayor abanico de perfiles. A consecuencia de esto, las conclusiones obtenidas respecto al número de estudiantes deben ser entendidas en el contexto de la simulación. Un factor que se puede extrapolar es que el método *GeSES* no detecta signos cuya presencia sea menor del 40 % dentro del grupo en el que se manifiestan. Además, los resultados obtenidos demostraron que el método detectó signos en grupos de estudiantes minoritarios. Esta característica hace que *GeSES* sea especialmente útil para los profesores, ya que este tipo de situaciones son muy difíciles de detectar.

8.2.4. Consideraciones respecto a *ASquare*

La herramienta que da soporte al método tiene las limitaciones propias de éste y las relacionadas con la interfaz. En este sentido, *ASquare* actualmente sólo es capaz de procesar ficheros de logs con un formato determinado. En efecto, es cierto que dicho formato es ampliamente utilizado por otras herramientas de *DM* como *Weka*, pero no es un formato estándar. Por ejemplo, otro tipo de formato es el utilizado por el paquete de programas *C4.5*, que utiliza un fichero para la definición de los datos (atributos y variables de clase) y otro para los datos. Otro aspecto a tener en cuenta es que las tareas de preparación y limpieza de datos deben ser realizadas por el profesor, ya que *ASquare* no proporciona soporte para aplicar filtros, generar nuevas variables y limpiar datos. No obstante, estos procesos no deberían representar grandes dificultades, pues los sistemas *e-Learning* suelen requerir procesos de registro y autenticación de los estudiantes, con lo que se garantiza que los logs no incluyan datos sobre usuarios ajenos al sistema. El proceso más tedioso puede consistir en generar el fichero de logs a partir de una base de datos relacional, o a partir de otros ficheros, pues es necesario seleccionar los datos que serán incluidos en el fichero de logs, así como generar y añadir la variable de evaluación.

8.3. Trabajo futuro

Algunos investigadores piensan que el desarrollo de una investigación nunca es finalizado, sino que es interrumpido. Puede haber varios factores que favorezcan esta afirmación. Por un lado, es importante transmitir los logros obtenidos a la comunidad científica, y durante este proceso la investigación permanece temporalmente detenida. Por otro, es posible que la investigación haya llegado a un punto en el que no es posible continuar por la falta de medios (económicos, tecnológicos, sociales, entre otros). Por ejemplo, en noviembre de 1985 se presentó la primera propuesta de *SMS*, pero su desarrollo fue parcialmente interrumpido a partir de 1987 por falta de

medios tecnológicos, y fue en 1993 cuando se retomó de nuevo con la aparición de los primeros teléfonos móviles [Trosby 2004]. Igualmente, se dice que cualquier tesis, y ésta no quiere ser una excepción, no tiene un punto final, sino que es un puente hacia futuras investigaciones. Dentro del trabajo de investigación desarrollado se han identificado diversos puntos en los que sería interesante profundizar su estudio.

El método *GeSES* realiza el proceso de identificación de signos que obtiene a partir de la información suministrada por las *DR*. Se detectaron situaciones en las que las reglas no contienen la información completa sobre el perfil del estudiante. En estos casos sería interesante complementar esta información de alguna forma. Una posibilidad es examinar la estructura del curso *e-Learning*, y otra realizar un examen descriptivo de los datos. Por ejemplo, si la información contenida en la *DR* indicara que los estudiantes de perfil (v11,-,v31) mostraron problemas con la actividad act5, es claro, que según esta información no sería posible conocer el valor de la segunda dimensión de este perfil. Sin embargo, un examen de los datos pondría de manifiesto que los estudiantes del perfil (v11,v21,v31) realizaron esta actividad, y que ésta no fue realizada por ningún otro estudiante de perfil con los valores v11 en la dimensión 1 y v31 en la dimensión 3. Por tanto, se podría complementar la información de la regla con esta nueva información, de manera que se identificaría el perfil concreto que tuvo problemas. También, sería interesante enriquecer la información que ofrece el método utilizando otros tipos de técnicas, como por ejemplo las reglas de asociación. Éstas últimas pueden averiguar si los signos identificados están relacionados entre sí, es decir, relacionar que un estudiante de un determinado perfil tuvo problemas en una actividad por los problemas mostrados en otra. Esta información también es importante desde el punto de vista del profesor, ya que es posible que el problema esté localizado en la primera actividad y no en la segunda. En otras palabras, los problemas de la segunda pueden deberse a los problemas de la primera. Otra línea de futuro trabajo consiste en aplicar el método a otros tipos de datos, de forma que se identifiquen otros signos de baja eficacia. Hay que indicar que este estudio se concentró en detectar signos de baja eficacia referidos al rendimiento académico. Sin embargo, este método puede ser aplicado sin problema para la detección de otro tipo de signos, por ejemplo los relacionados con el tiempo empleado. Un objetivo más ambicioso consiste en modificar el método de forma que no sólo detecte signos de baja eficacia en un sistema *e-Learning*, sino en otro tipo de sistemas no necesariamente enfocados a la enseñanza.

SimuLog ha sido desarrollado y mejorado durante esta investigación, sin embargo, diversos aspectos relacionados con la usabilidad pueden ser todavía mejorados. Por ejemplo, el simulador no permite que se especifique cada una de las proporciones de los valores que componen una dimensión, sólo se permite especificar la proporción de uno de ellos, de forma que los otros quedan asignados automáticamente por el simulador. Por ejemplo, si la dimensión 1 está compuesta por los valores v11, v12 y v13, si se asigna que el 35 % de los perfiles contengan el valor v11 en la dimensión 1 el porcentaje respectivo para los otros dos sería de 32,5 %. Por tanto, no sería posible generar, por ejemplo, que el 50 % tengan el valor v11, el 10 % el valor v12, y el 40 % el valor v13. Ésta es una mejora inmediata que será realizada en el simu-

lador. Otro aspecto a tener en cuenta es que se utiliza la distribución Normal para simular perfiles, sería interesante implementar la simulación mediante otro tipo de distribuciones. Mirando más adelante, un objetivo mucho más ambicioso consistiría en que mediante un análisis inicial de los datos que se quieren simular, el simulador fuera capaz de deducir diversos parámetros de la distribución de los datos, de modo que generara datos con una distribución muy similar a los datos originarios.

En esta tesis se ha propuesto una clasificación de signos de baja eficacia expresada mediante una ontología. En esta línea, se pretende mejorar la ontología añadiendo nuevas clases y mejores relaciones que permitan una caracterización más amplia de un signo de baja eficacia.

ASquare es la herramienta desarrollada para dar soporte al método *GeSES*. El prototipo de *ASquare* que se presenta en este trabajo necesita que determinadas cuestiones relacionadas con la interfaz sean mejoradas. Por ejemplo:

- Sería útil disponer de un conversor de ficheros, de forma que no sea un obstáculo para el profesor adaptar sus datos al formato utilizado en *ASquare*.
- Mejorar la presentación de las estadísticas, de forma que se puedan mostrar de forma gráfica mediante diagramas de barras, diagramas de dispersión, etc.

Asimismo, una línea futura que se abre es la mejora del motor de sugerencias. El motor actual está únicamente basado en la información contenida en la tabla de signos. Se propone ampliar esta información mediante el examen de la estructura del curso *e-Learning* y las reglas de adaptación. Dicho de otro modo, relacionar la información de la tabla con actividades (no especificadas en la tabla) y las adaptaciones realizadas por el sistema. Por ejemplo, si los estudiantes de un determinado perfil mostraron problemas en una determinada actividad práctica, puede ser que el problema esté relacionado con que los estudiantes no hayan comprendido los contenidos teóricos, y esta información no aparece en la tabla de signos. Por tanto, se puede sugerir al profesor que revise estos contenidos teóricos. Finalmente, un objetivo a largo plazo consiste en integrar en una base de conocimiento las sugerencias ofrecidas, de modo que las siguientes pudieran ser inferidas.

8.4. Publicaciones a las que ha dado lugar este trabajo

Esta tesis ha sido realizada dentro del marco de los proyectos *U-CAT* (*Ubiquitous Collaborative Adaptive Training*) y su continuación *HADA* (*Hipermedia Adaptativa para la atención a la Diversidad en entornos de inteligencia Ambiental*), ambos financiados por el Ministerio de Educación y Ciencia con TIN2004-03140 y TIN2007-64718 respectivamente. El objetivo principal de *U-CAT* fue el desarrollo de un entorno integrado que facilite la realización de actividades educativas desde cualquier localización utilizando diferentes dispositivos. El proyecto *HADA*, en cambio, persigue desarrollar metodologías y herramientas que integren la *hypermedia* adaptativa y entornos de inteligencia ambiental para facilitar el uso de las nuevas tecnologías

a usuarios con necesidades especiales y específicas, concretamente a personas con el síndrome de Down y personas mayores.

Asimismo, gran parte del trabajo llevado a cabo en esta investigación ha sido posible gracias a la beca **Predoctoral de FPI** (con referencia BES-2005-8178) asociada al proyecto de investigación TIN2004-03140 y adjudicada por el Ministerio de Educación y Ciencia, cofinanciada por el Fondo Social Europeo (BOE del 29 de julio de 2005).

Aunque, ya las aportaciones y los resultados de esta tesis se han expuesto de forma detallada en el capítulo 1, se ofrecen a continuación de forma resumida por categorías.

8.4.1. Publicaciones en revistas con índice de impacto

- César Vialardi, Javier Bravo y Alvaro Ortigosa. Improving AEH Courses through Log Analysis. *Journal of Universal Computer Science - J.UCS* 14(17), páginas 2777-2798, febrero 2009. [Vialardi 2009a]
- David Camacho, Estrella Pulido, Maria D. Rodríguez-Moreno, Rosa M. Carro, Alvaro Ortigosa y Javier Bravo. Automatic Course Redesign: global vs. individual adaptation. *International Journal of Engineering Education* 25(6), páginas 1270-1282, diciembre 2009. (ISSN: 0949-149X/91) [Camacho 2009]

8.4.2. Publicaciones en congresos internacionales

- Javier Bravo y Alvaro Ortigosa. *Validating the Evaluation of Adaptive Systems by User Profile Simulation*. En Proceedings of Fifth Workshop on User-Centred Design and Evaluation of Adaptive Systems held at the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), páginas 479-483, National College of Irland, Dublín, Irlanda, junio 2006. (ISSN: 1649-8623) [Bravo 2006c]
- César Vialardi, Javier Bravo y Alvaro Ortigosa. *Empowering AEH Authors Using Data Mining Techniques*. En Proceedings of Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H2007) held at the 11th International Conference on User Modeling (UM2007), páginas 33-43, Corfu, Grecia, junio 2007. [Vialardi 2007]
- Javier Bravo, César Vialardi y Alvaro Ortigosa. *A Problem-Oriented Method for Supporting AEH Authors through Data Mining*. En Proceedings of International Workshop on Applying Data Mining in e-Learning (ADML07) held at the Second European Conference on Technology Enhanced Learning (EC-TEL2007),

páginas 53-62, Creta, Grecia, septiembre 2007. (ISSN: 1613-0073) [Bravo 2007]

- Javier Bravo, César Vialardi y Alvaro Ortigosa. *ASquare: A Powerful Evaluation Tool for Adaptive Hypermedia Course System*. En Proceedings of Hypertext 2008 Conference, páginas 219-220, University of Pittsburgh, Pittsburgh, USA, junio 2008. (ISBN: 978-1-59593-998-2) [Bravo 2008a]
- Javier Bravo, César Vialardi y Alvaro Ortigosa. *Using Decision Trees for Discovering Problems on Adaptive Courses*. En Proceedings of E-Learn 2008: World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, páginas 268-277, Las Vegas, USA, noviembre 2008. (ISBN: 1-880094-66-5) [Bravo 2008b]
- Javier Bravo y Alvaro Ortigosa. *Detecting Symptoms of Low Performance Using Production Rules*. En Proceedings of Second Educational Data Mining conference, editado por: T. Barnes; M. Desmarais; C. Romero; S. Ventura, páginas 31-40, Universidad de Córdoba, Córdoba, España, julio 2009. (ISBN: 978-84-613-2308-1) [Bravo 2009b]
- César Vialardi, Javier Bravo, Leila Shafti y Alvaro Ortigosa. *Recommendation in Higher Education Using Data Mining Techniques*. En Proceedings of Second Educational Data Mining conference, editado por: T. Barnes; M. Desmarais; C. Romero; S. Ventura, páginas 190-199, Universidad de Córdoba, Córdoba, España, julio 2009. (ISBN: 978-84-613-2308-1) [Vialardi 2009b]
- Javier Bravo, Estefanía Martín, Alvaro Ortigosa y Rosa M Carro. *Checking the Reliability of GeSES: Method for Detecting Symptoms of Low Performance*. En Proceedings of workshop on Educational Data Mining held at the 9th International Conference on Intelligent System Design and Applications (ISDA09), páginas 1108-1113, Pisa, Italia. IEEE press. (ISBN: 978-1-4244-4735-0) [Bravo 2009a]

8.4.3. Publicaciones en congresos nacionales

Javier Bravo y Alvaro Ortigosa. *Integración y Prueba de Herramientas de Evaluación en un Entorno Hipermedia Adaptativo*. En Proceedings of 8th International Symposium on Computers in Education (SIIE2006), páginas 253-261, Universidad de León, León, España, octubre 2006. (ISBN: 84-9773-302-9) [Bravo 2006b]

8.4.4. Capítulos de libro

Javier Bravo, César Vialardi y Alvaro Ortigosa. Using decision trees for improving AEH courses, capítulo 26. *Handbook of Educational Data Mining*. Editorial Taylor & Francis, 2010 (octubre). [Bravo 2010]

8.4.5. Trabajo de Iniciación a la Investigación

Javier Bravo. Mecanismos de Integración y Prueba de Herramientas Automáticas de Evaluación de Calidad de Cursos Adaptativos. *Trabajo de Iniciación a la Investigación*, Escuela Politécnica Superior, UAM, Madrid, España, septiembre 2006. [Bravo 2006a]

APÉNDICE A

Apéndice

Índice de contenidos

A.1. Ficheros de datos	210
A.1.1. Fichero <i>weather.csv</i>	210
A.1.2. Fichero <i>weather.names</i>	211
A.1.3. Fichero <i>weather.data</i>	211
A.2. Reglas de asociación	211
A.2.1. Reglas de asociación de los datos de la tabla 3.1 (página 58)	211
A.3. Árboles de decisión	214
A.3.1. Árbol de decisión de los datos de la tabla 3.1, página 58 (algoritmo <i>J48</i>)	214
A.3.2. Árbol de decisión de los datos de la tabla 3.1, página 58 (algoritmo <i>C4.5</i>)	215
A.3.3. Reglas de decisión de los datos de la tabla 3.1, página 58	216
A.4. Sistemas <i>e-Learning</i> evaluados	217
A.4.1. <i>Wotan</i>	217
A.4.2. <i>QuizGuide</i> y <i>QuizPACK</i>	218
A.4.3. <i>CoMoLE</i>	220
A.5. Ficheros de logs	222
A.5.1. Curso de “Introducción a la programación en el lenguaje C” en los sistemas <i>QuizGuide</i> y <i>QuizPACK</i>	222
A.5.2. Curso de “Sistemas Operativos I” en el sistema <i>CoMoLE</i>	229
A.5.3. Curso de “Estructura de Datos y de la Información I” en el sistema <i>CoMoLE</i>	238
A.6. Reglas de decisión	242
A.6.1. Reglas de decisión para el curso “Introducción a la programación en el lenguaje C” en los sistemas <i>QuizGuide</i> y <i>QuizPACK</i>	243
A.6.2. Reglas de decisión para el curso “Sistemas Operativos I” en el sistema <i>CoMoLE</i>	246
A.6.3. Reglas de decisión para el curso “Estructura de Datos y de la Información I” en el sistema <i>CoMoLE</i>	258
A.7. Método <i>GeSES</i> para distintos tamaños de datos	272
A.7.1. Resultados para $N = 20$	273
A.7.2. Resultados para $N = 50$	273
A.7.3. Resultados para $N = 80$	273

A.7.4. Resultados para N = 100	274
A.7.5. Resultados para N = 150	274
A.7.6. Resultados para N = 300	275
A.7.7. Resultados para N = 600	275
A.7.8. Resultados para N = 900	276
A.7.9. Resultados para N = 1200	277
A.7.10. Resultados para N = 1500	278
A.7.11. Resultados para N = 1800	278
A.7.12. Resultados para N = 2100	279
A.7.13. Resultados para N = 2400	280
A.7.14. Resultados para N = 2700	281
A.7.15. Resultados para N = 3100	282
A.7.16. Resultados para N = 3500	283
A.7.17. Resultados para N = 4000	284
A.7.18. Resultados para N = 5000	285
A.7.19. Resultados para N = 6000	286

A.1. Ficheros de datos

Esta sección contiene los ficheros utilizados en este trabajo. Se presentan los ficheros con los formatos .csv y los formatos propios del programa C4.5 (*.names y *.data).

A.1.1. Fichero *weather.csv*

```

outlook,temperature,humidity,windy,play
sunny,hot,high,false,no
sunny,hot,high,true,no
overcast,hot,high,false,yes
rainy,mild,high,false,yes
rainy,cool,normal,false,yes
rainy,cool,normal,true,no
overcast,cool,normal,true,yes
sunny,mild,high,false,no
sunny,cool,normal,false,yes
rainy,mild,normal,false,yes
sunny,mild,normal,true,yes
overcast,mild,high,true,yes
overcast,hot,normal,false,yes
rainy,mild,high,true,no

```

A.1.2. Fichero *weather.names*

no,yes.
 outlook: sunny,overcast,rainy.
 temperature: hot,mild,cool.
 humidity: high,normal.
 windy: false,true.

A.1.3. Fichero *weather.data*

sunny,hot,high,false,no
 sunny,hot,high,true,no
 overcast,hot,high,false,yes
 rainy,mild,high,false,yes
 rainy,cool,normal,false,yes
 rainy,cool,normal,true,no
 overcast,cool,normal,true,yes
 sunny,mild,high,false,no
 sunny,cool,normal,false,yes
 rainy,mild,normal,false,yes
 sunny,mild,normal,true,yes
 overcast,mild,high,true,yes
 overcast,hot,normal,false,yes
 rainy,mild,high,true,no

A.2. Reglas de asociación

Esta sección contiene las salidas de la aplicación del algoritmo *A-priori* de las reglas de asociación, utilizando *Weka* (versión 3.6.0) en los diversos ejemplos expuestos en este trabajo.

A.2.1. Reglas de asociación de los datos de la tabla 3.1 (página 58)

1. outlook=overcast 4 ==> play=yes 4 conf:(1)
2. temperature=cool 4 ==> humidity=normal 4 conf:(1)
3. humidity=normal windy=false 4 ==> play=yes 4 conf:(1)
4. outlook=sunny play=no 3 ==> humidity=high 3 conf:(1)
5. outlook=sunny humidity=high 3 ==> play=no 3 conf:(1)
6. outlook=rainy play=yes 3 ==> windy=false 3 conf:(1)
7. outlook=rainy windy=false 3 ==> play=yes 3 conf:(1)
8. temperature=cool play=yes 3 ==> humidity=normal 3 conf:(1)
9. outlook=sunny temperature=hot 2 ==> humidity=high 2 conf:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2 conf:(1)
11. outlook=sunny temperature=hot 2 ==> play=no 2 conf:(1)
12. outlook=sunny play=yes 2 ==> humidity=normal 2 conf:(1)

13. outlook=sunny humidity=normal 2 ==> play=yes 2 conf:(1)
14. windy=false play=no 2 ==> outlook=sunny 2 conf:(1)
15. outlook=overcast windy=false 2 ==> temperature=hot 2 conf:(1)
16. outlook=overcast temperature=hot 2 ==> windy=false 2 conf:(1)
17. temperature=hot play=yes 2 ==> outlook=overcast 2 conf:(1)
18. outlook=overcast temperature=hot 2 ==> play=yes 2 conf:(1)
19. outlook=overcast humidity=high 2 ==> play=yes 2 conf:(1)
20. outlook=overcast humidity=normal 2 ==> play=yes 2 conf:(1)
21. outlook=overcast windy=false 2 ==> play=yes 2 conf:(1)
22. outlook=overcast windy=true 2 ==> play=yes 2 conf:(1)
23. outlook=rainy humidity=high 2 ==> temperature=mild 2 conf:(1)
24. outlook=rainy temperature=cool 2 ==> humidity=normal 2 conf:(1)
25. outlook=rainy play=no 2 ==> windy=true 2 conf:(1)
26. outlook=rainy windy=true 2 ==> play=no 2 conf:(1)
27. temperature=hot play=no 2 ==> humidity=high 2 conf:(1)
28. temperature=hot play=yes 2 ==> windy=false 2 conf:(1)
29. temperature=mild play=no 2 ==> humidity=high 2 conf:(1)
30. temperature=mild humidity=normal 2 ==> play=yes 2 conf:(1)
31. temperature=cool windy=false 2 ==> humidity=normal 2 conf:(1)
32. temperature=cool windy=true 2 ==> humidity=normal 2 conf:(1)
33. temperature=cool windy=false 2 ==> play=yes 2 conf:(1)
34. windy=false play=no 2 ==> humidity=high 2 conf:(1)
35. temperature=hot humidity=high play=no 2 ==> outlook=sunny 2 conf:(1)
36. outlook=sunny temperature=hot play=no 2 ==> humidity=high 2 conf:(1)
37. outlook=sunny temperature=hot humidity=high 2 ==> play=no 2 conf:(1)
38. temperature=hot play=no 2 ==> outlook=sunny humidity=high 2 conf:(1)
39. outlook=sunny temperature=hot 2 ==> humidity=high play=no 2 conf:(1)
40. humidity=high windy=false play=no 2 ==> outlook=sunny 2 conf:(1)
41. outlook=sunny windy=false play=no 2 ==> humidity=high 2 conf:(1)
42. outlook=sunny humidity=high windy=false 2 ==> play=no 2 conf:(1)
43. windy=false play=no 2 ==> outlook=sunny humidity=high 2 conf:(1)
44. temperature=hot windy=false play=yes 2 ==> outlook=overcast 2 conf:(1)
45. outlook=overcast windy=false play=yes 2 ==> temperature=hot 2 conf:(1)
46. outlook=overcast temperature=hot play=yes 2 ==> windy=false 2 conf:(1)
47. outlook=overcast temperature=hot windy=false 2 ==> play=yes 2 conf:(1)
48. temperature=hot play=yes 2 ==> outlook=overcast windy=false 2 conf:(1)
49. outlook=overcast windy=false 2 ==> temperature=hot play=yes 2 conf:(1)
50. outlook=overcast temperature=hot 2 ==> windy=false play=yes 2 conf:(1)
51. temperature=mild windy=false play=yes 2 ==> outlook=rainy 2 conf:(1)
52. outlook=rainy temperature=mild play=yes 2 ==> windy=false 2 conf:(1)
53. outlook=rainy temperature=mild windy=false 2 ==> play=yes 2 conf:(1)
54. outlook=rainy humidity=normal play=yes 2 ==> windy=false 2 conf:(1)
55. outlook=rainy humidity=normal windy=false 2 ==> play=yes 2 conf:(1)
56. temperature=cool windy=false play=yes 2 ==> humidity=normal 2 conf:(1)

57. temperature=cool humidity=normal windy=false 2 ==> play=yes 2 conf:(1)
58. temperature=cool windy=false 2 ==> humidity=normal play=yes 2 conf:(1)
59. temperature=hot windy=true 1 ==> outlook=sunny 1 conf:(1)
60. outlook=sunny temperature=cool 1 ==> humidity=normal 1 conf:(1)
61. outlook=sunny temperature=cool 1 ==> windy=false 1 conf:(1)
62. outlook=sunny temperature=cool 1 ==> play=yes 1 conf:(1)
63. temperature=hot humidity=normal 1 ==> outlook=overcast 1 conf:(1)
64. outlook=overcast temperature=mild 1 ==> humidity=high 1 conf:(1)
65. outlook=overcast temperature=mild 1 ==> windy=true 1 conf:(1)
66. outlook=overcast temperature=mild 1 ==> play=yes 1 conf:(1)
67. outlook=overcast temperature=cool 1 ==> humidity=normal 1 conf:(1)
68. outlook=overcast temperature=cool 1 ==> windy=true 1 conf:(1)
69. outlook=overcast temperature=cool 1 ==> play=yes 1 conf:(1)
70. temperature=cool play=no 1 ==> outlook=rainy 1 conf:(1)
71. humidity=normal play=no 1 ==> outlook=rainy 1 conf:(1)
72. temperature=hot windy=true 1 ==> humidity=high 1 conf:(1)
73. temperature=hot humidity=normal 1 ==> windy=false 1 conf:(1)
74. temperature=hot humidity=normal 1 ==> play=yes 1 conf:(1)
75. temperature=hot windy=true 1 ==> play=no 1 conf:(1)
76. humidity=normal play=no 1 ==> temperature=cool 1 conf:(1)
77. temperature=cool play=no 1 ==> humidity=normal 1 conf:(1)
78. temperature=cool play=no 1 ==> windy=true 1 conf:(1)
79. humidity=normal play=no 1 ==> windy=true 1 conf:(1)
80. outlook=sunny temperature=hot windy=false 1 ==> humidity=high 1 conf:(1)
81. temperature=hot humidity=high windy=true 1 ==> outlook=sunny 1 conf:(1)
82. outlook=sunny humidity=high windy=true 1 ==> temperature=hot 1 conf:(1)
83. outlook=sunny temperature=hot windy=true 1 ==> humidity=high 1 conf:(1)
84. temperature=hot windy=true 1 ==> outlook=sunny humidity=high 1 conf:(1)
85. temperature=hot windy=false play=no 1 ==> outlook=sunny 1 conf:(1)
86. outlook=sunny temperature=hot windy=false 1 ==> play=no 1 conf:(1)
87. temperature=hot windy=true play=no 1 ==> outlook=sunny 1 conf:(1)
88. outlook=sunny windy=true play=no 1 ==> temperature=hot 1 conf:(1)
89. outlook=sunny temperature=hot windy=true 1 ==> play=no 1 conf:(1)
90. temperature=hot windy=true 1 ==> outlook=sunny play=no 1 conf:(1)
91. outlook=sunny temperature=mild windy=false 1 ==> humidity=high 1 conf:(1)
92. outlook=sunny temperature=mild humidity=high 1 ==> windy=false 1 conf:(1)
93. outlook=sunny temperature=mild play=no 1 ==> humidity=high 1 conf:(1)
94. outlook=sunny temperature=mild humidity=high 1 ==> play=no 1 conf:(1)
95. temperature=mild humidity=normal windy=true 1 ==> outlook=sunny 1 conf:(1)
96. outlook=sunny humidity=normal windy=true 1 ==> temperature=mild 1 conf:(1)
97. outlook=sunny temperature=mild windy=true 1 ==> humidity=normal 1 conf:(1)
98. outlook=sunny temperature=mild humidity=normal 1 ==> windy=true 1 conf:(1)
99. outlook=sunny temperature=mild play=yes 1 ==> humidity=normal 1 conf:(1)
100. outlook=sunny temperature=mild humidity=normal 1 ==> play=yes 1 conf:(1)

A.3. Árboles de decisión

Esta sección muestra el resultado de aplicar el algoritmo de los árboles de decisión a los diversos ejemplos expuestos en este trabajo. Asimismo, se muestra la salida de los algoritmos *J48* (*Weka* versión 3.6.0), y *C4.5* (versión 8).

A.3.1. Árbol de decisión de los datos de la tabla 3.1, página 58 (algoritmo *J48*)

```
=== Run information ===
```

```
Scheme:      weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:    weather
Instances:   14
Attributes:  5
              outlook
              temperature
              humidity
              windy
              play
Test mode:   10-fold cross-validation
```

```
=== Classifier model (full training set) ===
```

```
J48 pruned tree
```

```
-----
```

```
outlook = sunny
|  humidity = high: no (3.0)
|  humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|  windy = false: yes (3.0)
|  windy = true: no (2.0)
```

```
Number of Leaves : 5
```

```
Size of the tree : 8
```

```
Time taken to build model: 0.04 seconds
```

```
=== Stratified cross-validation ===
```

```
=== Summary ===
```

Correctly Classified Instances	7	50	%
Incorrectly Classified Instances	7	50	%
Kappa statistic	-0.0426		
Mean absolute error	0.4167		
Root mean squared error	0.5984		
Relative absolute error	87.5	%	
Root relative squared error	121.2987	%	
Total Number of Instances	14		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.4	0.444	0.333	0.4	0.364	0.633	no
	0.556	0.6	0.625	0.556	0.588	0.633	yes
Weighted Avg.	0.5	0.544	0.521	0.5	0.508	0.633	

=== Confusion Matrix ===

```

a b  <-- classified as
2 3 | a = no
4 5 | b = yes

```

A.3.2. Árbol de decisión de los datos de la tabla 3.1, página 58 (algoritmo C4.5)

C4.5 [release 8] decision tree generator Wed Sep 2 21:15:20 2009

```

Options:
File stem <weather>

```

Read 14 cases (4 attributes) from weather.data

Decision Tree:

```

outlook = overcast: yes (4.0)
outlook = sunny:
|  humidity = high: no (3.0)
|  humidity = normal: yes (2.0)
outlook = rainy:
|  windy = false: yes (3.0)
|  windy = true: no (2.0)

```

Tree saved

Evaluation on training data (14 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
8	0(0.0%)	8	0(0.0%)	(38.5%) <<

A.3.3. Reglas de decisión de los datos de la tabla 3.1, página 58

C4.5 [release 8] rule generator Wed Sep 2 21:16:49 2009

Options:
File stem <weather>

Read 14 cases (4 attributes) from weather

Processing tree 0

Final rules from tree 0:

Rule 1:
outlook = sunny
humidity = high
-> class no [63.0%]

Rule 5:
outlook = rainy
windy = true
-> class no [50.0%]

Rule 3:
outlook = overcast
-> class yes [70.7%]

Rule 2:
humidity = normal
-> class yes [66.2%]


```

Rule 4:
    outlook = rainy
    windy = false
-> class yes [63.0%]

```

Default class: yes

Evaluation on training data (14 items):

Rule	Size	Error	Used	Wrong	Advantage	
----	----	-----	----	-----	-----	
1	2	37.0%	3	0 (0.0%)	3 (3 0)	no
5	2	50.0%	2	0 (0.0%)	2 (2 0)	no
3	1	29.3%	4	0 (0.0%)	0 (0 0)	yes
2	1	33.8%	4	0 (0.0%)	0 (0 0)	yes
4	2	37.0%	1	0 (0.0%)	0 (0 0)	yes

Tested 14, errors 0 (0.0%) <<

```

(a) (b) <-classified as
-----
5      (a): class no
      9 (b): class yes

```

A.4. Sistemas *e-Learning* evaluados

A.4.1. *Wotan*

Wotan (*Web-based Online Task-based Adaptive Learning*) [Freire 2007, pp. 110-113] es el nombre que recibe la versión mejorada de *TANGOW* (*Task-based Adaptive Learner Guidance On the Web*) [Carro 2001, cap. 6]. Este sistema diseñado por Rosa M. Carro permite ofrecer cursos *AEH* a través de *Internet* que se ajustan a las necesidades y preferencias de cada estudiante durante el proceso de aprendizaje. Con este propósito, este sistema genera dinámicamente los contenidos presentados a los estudiantes durante la realización de un curso en función de: la petición realizada por el estudiante, la descripción del curso realizada por el diseñador (generalmente es el profesor el que realiza esta tarea) y la información del estudiante que incluye datos de su perfil y las acciones realizadas durante su interacción con el curso (modelo del estudiante).

Wotan fue implementado como un *JAVA servlet container* y está compuesto por

cuatro módulos: administración, ejercicios, creación de cursos y presentación. El módulo de administración permite realizar las tareas de control de acceso y gestión de usuarios y cursos. El segundo módulo es una aplicación *Web* que permite a los diseñadores crear y probar distintos tipos de ejercicios, que posteriormente pueden ser añadidos a un curso. El módulo de creación de cursos, llamado *Woted*¹ (*Wotan Editor*), es una herramienta de autor desarrollada por Manuel Freire que permite crear y editar cursos adaptativos para este sistema (consúltese [Freire 2007, cap. 7] para mayor información sobre su funcionamiento).

Un curso de este sistema está formado por un conjunto de actividades y reglas de adaptación. Una actividad a su vez está compuesta por fragmentos de código *HTML*, que puede contener cualquier tipo de contenido *hypermedia*, o por subactividades.

- Los contenidos incluidos en las actividades pueden ser de varios tipos: teóricos, prácticos y ejemplos.
- Las reglas de adaptación indican al sistema cómo y cuando realizar la adaptación, es decir, qué actividades presentar, en qué orden deben ser presentadas y en qué momento.

Por tanto, la estructura de un curso *Wotan* no sólo incluye los contenidos educativos sino también las reglas de adaptación. Así, la adaptación puede cambiar en cada curso y no depende únicamente del sistema, ya que está implícita en la propia definición del curso. No obstante, esta flexibilidad para añadir adaptación a los cursos *e-Learning* produce que la tarea de diseñarlos sea muy compleja para los profesores, pues éstos tienen que especificar el tipo de adaptación elegida en cada caso además de diseñar distintos contenidos para cada tipo de adaptación (consúltese la *web* de *Wotan*: <http://tangow.ii.uam.es/wotan/> para más información).

A.4.2. *QuizGuide* y *QuizPACK*

QuizGuide [Brusilovsky 2004] es un sistema adaptativo que ayuda a los estudiantes a seleccionar los ejercicios más relevantes respecto de sus objetivos de aprendizaje. Los ejercicios o actividades, llamados *quizzes*, están agrupados en temas, llamados *topics*. La función de este sistema no es la de generar y presentar los ejercicios, sino añadir ciertas características adaptativas relacionadas con la navegación entre las actividades. El objetivo es mostrar a cada estudiante los *topics* que son importantes para su estado actual, y cuáles requieren mayor esfuerzo. La interfaz de la aplicación (ver figura A.1) se divide en dos partes: *quiz-navigation-area* y *quiz-presentation-area*. La primera proporciona enlaces (*links*) a 40 *quizzes* agrupados en 20 *topics*. Cuando un estudiante selecciona un *topic*, por ejemplo el *topic* “*loops (do while)*” en la figura A.1, éste se expande mostrando los enlaces a los *quizzes* disponibles. En el área de presentación se muestra el ejercicio seleccionado. Un ejercicio está formado por un código en el lenguaje *C* y una cuestión relacionada con el código,

¹ *Woted* es una herramienta basada la herramienta de visualización de grafos *CLOVER* [Freire 2007, cap. 6].

que puede ser de dos tipos: *¿Cuál es el valor final de una variable?* y *¿Cuál es el resultado?* Una vez que el estudiante ha respondido la cuestión, se le muestra la respuesta correcta y si lo ha hecho bien o mal, ofreciéndole dos opciones: continuar con el siguiente ejercicio o repetir el ejercicio (véase parte “b”) de la figura A.1). En caso de repetición, el ejercicio mostrado conceptualmente es el mismo, pero la respuesta a la cuestión es diferente. De esta forma, un estudiante puede repetir los ejercicios para reforzar sus conocimientos.

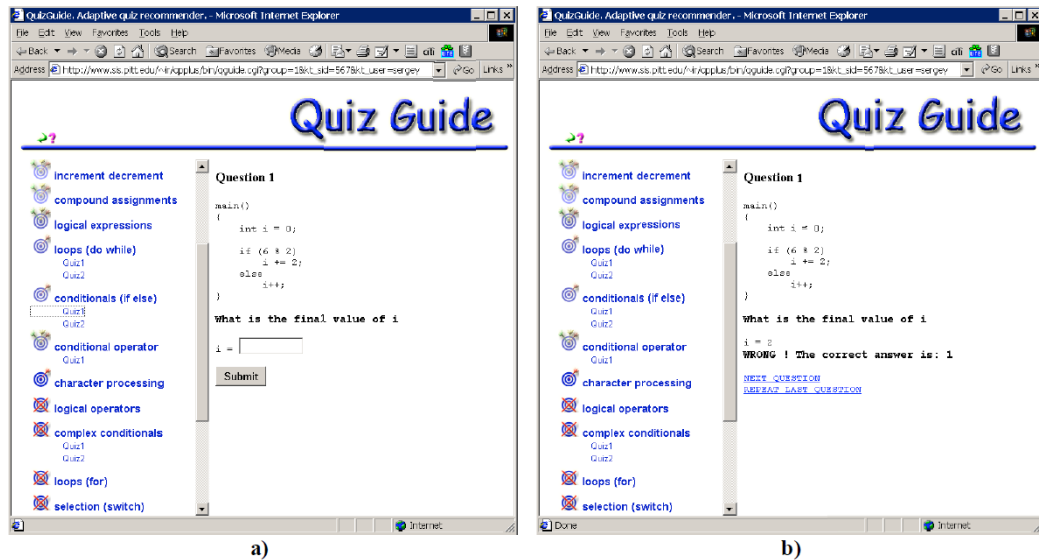


Figura A.1: Interfaz de *QuizGuide*. Fuente: figura 2: *Student interface of QuizGuide* en [Brusilovsky 2004].

La adaptación se realiza en el área de navegación mediante el uso de iconos adaptativos a la izquierda de cada *topic* (véase figura A.1). Estos iconos indican el nivel de conocimiento y la relevancia que tiene el *topic* respecto a alcanzar el objetivo de aprendizaje. Se utiliza una diana con flechas para indicar el nivel de conocimiento del estudiante en el *topic*. Así, el número de flechas en la diana (la diana puede tener de 0 a 3 flechas) indica el nivel de conocimiento del estudiante en el *topic*. Cuantas más flechas haya en la diana mayor es el conocimiento (en la figura A.1, el *topic* “*logical expressions*” tiene una diana con tres flechas). La intensidad del color de la diana indica la relevancia del *topic* respecto al objetivo de aprendizaje. De esta manera, se indica que un *topic* es muy relevante con un color muy intenso. Finalmente, una diana tachada con una cruz roja indica que el *topic* no es alcanzable con el nivel de conocimientos actual del estudiante (en la figura A.1, se muestra que el *topic* “*logical operators*” no es alcanzable).

QuizPACK (*Quizzes for Parameterized Assessment of C Knowledge*) [Brusilovsky 2005] es un sistema que crea ejercicios, y evalúa las respuestas de los estudiantes. Como en la aplicación anterior, un ejercicio está formado por un fragmento de código (proporcionado por el profesor) y una cuestión sobre el valor de un

parámetro del código (el profesor señala en el sistema el parámetro sobre el que se realiza la cuestión). Cuando un estudiante responde a la cuestión que se le plantea en un ejercicio, *QuizPACK* interpreta el código para generar la respuesta; seguidamente, la compara con la proporcionada por el estudiante, le devuelve su puntuación, almacena el resultado, y le presenta dos opciones: ir al siguiente ejercicio, o repetir el ejercicio. Como *QuizPACK* genera de forma aleatoria el valor del parámetro del código cada vez que se repite el ejercicio, el estudiante puede repetir el ejercicio, que es el mismo conceptualmente, pero con distinta respuesta, para reforzar sus conocimientos.

A.4.3. *CoMoLE*

CoMoLE (*Context-based Adaptive Mobile Learning Environments*) [Martín 2009, Martín 2006] es un sistema *Web* que ofrece recomendaciones y diferentes tipos de actividades (i.e. ejemplos, tests, ejercicios de respuesta rápida (*short_text*) y colaborativos, etc). Este sistema recomienda las actividades más adecuadas a cada estudiante no sólo teniendo en cuenta sus características personales, sino también considerando el contexto actual en el que se encuentra en el momento de la recomendación. El contexto tiene información relativa al dispositivo usado, tiempo disponible, localización física y tiempo de recomendación. Las recomendaciones se definen en el sistema mediante las reglas de recomendación (reglas estructurales, filtros generales de contexto y restricciones).

Cuando un estudiante accede al sistema e inicia una sesión, el sistema le pregunta el tiempo que va a interactuar con él. Mediante esta información *CoMoLE* es capaz de estimar las actividades que se le pueden recomendar teniendo en cuenta el tiempo del que dispone el estudiante. Las actividades ofrecidas a los estudiantes versan sobre actividades de refuerzo para los contenidos teóricos, revisión de ejemplos, tests, cuestiones, resolver actividades colaborativas y actividades de repaso. El sistema realiza las recomendaciones en base a reglas estructurales, filtros generales de contexto (i.e. mínimo tiempo necesario para realizar los ejercicios *short_text* y actividades colaborativas), y restricciones para ciertas actividades (tiempo de comienzo y final, dispositivos que deben ser utilizados en actividades colaborativas, etc).

La figura A.2 muestra la interfaz de una página *web* generada por *CoMoLE*. En ésta se distinguen tres partes: área de recomendación, índice de actividades y área de contenidos. Las recomendaciones se realizan en el área de recomendación (señalada en la figura con (1)). En dicha figura se puede ver que se recomiendan las actividades “Switch. Ejemplo” e “If. Teoría”. Además, en la parte derecha de este área se muestra la actividad que el estudiante está realizando actualmente, actividad “If. Ejemplos” en la figura. De esta forma, los estudiantes pueden ver tanto la actividad actual como las sugerencias ofrecidas por el sistema. El índice de actividades se muestra en la parte inferior izquierda de la figura (área señalada con (2)), donde cada actividad está coloreada según su estado. Los estados de una actividad son: actividad disponible y recomendada por el sistema (color verde), actividad disponible, pero no

The screenshot shows a web browser window titled "CoMoLE - Windows Internet Explorer". The address bar contains the URL "http://tangow.i.i.uam.es/console/NewActivity?T=OperadorasRptalibre5". The page layout is as follows:

- Top Bar:** Contains a "Salir" button, a "ENCUESTA!!!!!!" link, and an "Oculta indice" button.
- Left Panel (Navigation):** A tree view under "ED11" with categories like "Lenguaje C", "Tipos atómicos", "Enumeraciones", "Operadores", "Condicionales", "If", "Switch", "Bucles", "Entrada-Salida", "Estructuras", "Nuevos Tipos de Datos", "Funciones", "Paso de Argumentos", and "Arrays".
- Top Right Panel (Actividad Actual):** Displays "If. Ejemplos".
- Main Content Area:**
 - Ejemplos Condicional IF:** Introduction text: "Veamos unos ejemplos del condicional if, para entender mejor su funcionamiento:".
 - Ejemplo 1:** "Imaginemos que tenemos el siguiente fragmento de código:" followed by C code:

```
int numero;

scanf("%d", &numero);
if (numero % 2 == 0)
    printf("El numero es par\n");
else
    printf("El numero es impar\n");
```
 - Ejemplo 2:** "Un uso muy común del if es para controlar errores. Veamos un sencillo ejemplo:" followed by C code:

```
int n;

if (scanf("%d", &n) != 1){
    printf("Error al introducir los datos");
    return (-1);
}
```

Figura A.2: Ejemplo de página *web* generada por *CoMoLE*. Fuente: figura 4.28: Ejemplo de una página *web* generada por *CoMoLE* en [Martín 2008]

recomendada por el sistema (color amarillo), actividad no disponible (color rojo), y actividad realizada (color negro). La tercera parte (señalada en la figura con (3)) muestra los contenidos de la actividad seleccionada. Véase [Martín 2008] para mayor información.

A.5. Ficheros de logs

Esta sección contiene los logs de los distintos sistemas que fueron evaluados con el método *GeSES*.

A.5.1. Curso de “Introducción a la programación en el lenguaje C” en los sistemas *QuizGuide* y *QuizPACK*

Las interacciones de los estudiantes con el sistema *QuizGuide* al realizar dicho curso, se almacenan en una base de datos relacional. Esta base de datos relacional consta principalmente de una serie de tablas con información sobre actividades, conceptos, usuarios, relaciones entre usuarios y actividades, y relación entre conceptos y actividades. Por ejemplo, la tabla que relaciona usuarios y actividades contiene los siguientes campos:

- *groupId*: grupo al que pertenece el estudiante.
- *result*: resultado obtenido en la actividad por el estudiante. Almacena valores entre -1 y 1 .
- *activityId*: identificador de la actividad.
- *session*: identificador de sesión.
- *dateNTime*: marca de tiempo al iniciar la actividad.
- *appId*: identificador de la aplicación.
- *svc*: reservado para aplicaciones.
- *allParameters*: contiene los anteriores parámetros en una cadena de caracteres.
- *userId*: identificador del estudiante.

Hay que indicar que las aplicaciones desarrolladas por el grupo *PAWS* (*Personalized Adaptive Web Systems*) en la *University of Pittsburgh*, cuyo director de grupo es el Dr. Peter Brusilovsky, son centralizadas a través del sistema *ADAPT2* (*Advanced Distributed Architecture for Personalized Teaching and Training*). Este sistema almacena, por medio de *CUMULATE* (*Centralized User Modeling Architecture for Teaching*), todas las interacciones de los usuarios con las aplicaciones en una única base de datos, por esta razón aparece el campo *appId* en la tabla anterior. En este sentido, tanto *QuizGuide* como *QuizPACK* son aplicaciones con identificadores distintos.

Con el objetivo de integrar información a cerca de las interacciones, sobre los estudiantes, y sobre los ejercicios, se construyó una tabla de logs, mediante una serie de consultas *SQL* a la base de datos. De esta forma, un log de esta tabla está formado por los siguientes campos:

- ***userId***: identificador del estudiante en el sistema.
- ***groupId***: identificador del grupo al que el estudiante pertenece. Siete grupos son posibles y corresponden a seis diferentes universidades, y un grupo adicional llamado “world group”. En este último grupo están los estudiantes que siguieron el curso de forma libre y no pertenecen a ninguna de las seis universidades anteriores.
- ***result***: evaluación del resultado del estudiante en la actividad. Dos valores son posibles: 0 (respuesta incorrecta) y 1 (respuesta correcta).
- ***activity***: nombre o identificador del ejercicio o actividad.
- ***question***: identificador de la cuestión. Una actividad está formada por una o varias cuestiones.
- ***concept***: nombre del concepto que cubre la actividad.
- ***conceptParent***: nombre del concepto superior (padre).
- ***session***: identificador de la sesión *Web*.
- ***success***: variable discreta generada a partir de la variable *result*. Si *result* es 1 entonces esta variable toma el valor de *yes*, en caso contrario su valor es *no*.
- ***timeStamp***: momento en el que el estudiante inicia la actividad (equivalente a *dateNTime*).

A continuación, se presenta el fichero de definiciones (*.names) para la tabla de logs anterior² y un extracto del fichero de datos (.data).

Fichero de descripción de los datos (*.names):

```
no,yes.
userid: uid_199,uid_194,uid_359,uid_456,uid_200,uid_191,uid_277,uid_193,
uid_383,uid_377,uid_374,uid_454,uid_446,uid_448,uid_466,uid_443,uid_371,
uid_457,uid_468,uid_459,uid_578,uid_445,uid_582,uid_585,uid_398,uid_583,
uid_198,uid_366,uid_279,uid_186,uid_364,uid_196,uid_367,uid_399,uid_384,
uid_189,uid_499,uid_510,uid_531,uid_565,uid_581,uid_584,uid_588,uid_400,
uid_580,uid_284,uid_586,uid_402,uid_185,uid_281,uid_503,uid_360,uid_577,
uid_587,uid_534.
groupid: gid_190,gid_72,gid_441,gid_373,gid_442,gid_394,gid_67.
```

²Los campos *session* y *dateNTime* han sido reducidos, y tan sólo se muestran los cinco primeros valores y el último.

```

result: continuous.
activity: 2dimensional_array1,2dimensional_array2,2dimensional_array3,
arithmetic_expression1,arithmetic_expression2,character_array1,
character_array2,character_array3,character_processing1,
character_processing2,character_processing3,complex_conditional1,
conditional_operator1,do2,function1,function2,logical_expression1,
nested_loop1,pointer2,pointer3,printing1,printing2,switch1,
variable2,variable3,array1,array2,array3,complex_conditional2,
compound_assignment1,constant1,constant2,do1,for1,for2,function3,
if_else1,if_else2,increment_decrement1,logical_operator1,
pointer1,variable1,while1,while2,globvar_simplefunc1,
globvar_simplefunc2.
question: q_1,q_2,q_3,q_0,q_4.
concept: printf,main_function,function_declaration,include,define,char,int,float,
void,pointer_declaration,reference,dereference,string,array_declaration,
explicit_type_cast,variable_declaration,variable_initialization,if,if_else,
switch,while,do,for,break,add,sub,mul,div,mod,pre_increment,post_increment,
post_decrement,simple_assignment,add_assignment,sub_assignment,mul_assignment,
div_assignment,mod_assignment,equal,not_equal,less,greater,less_equal,
greater_equal,and,or,not,conditional_operator,multi_dimensional_array,
return,printing (printf),arithmetic expressions,variables,constants (define),
increment decrement,compound assignments,loops (while),logical expressions,
loops (do while),conditionals (if else),conditional operator,
character processing,logical operators,complex conditionals,loops (for),
selection (switch),arrays,functions,character arrays,nested loops,pointers,
two-dimensional arrays,Printing,Arithmetic Expressions,Local Variables,
Global Variables & Simple Functions,Constants,Increment and Decrement,
Compound Assignments,while-Loop,Logical Expressions,do-Loop,Simple Decisions,
Conditional Operator,Character Processing,Boolean Operators,Complex Decisions,
for-Loop,Selection,Arrays,Functions with Parameters,Character Arrays,
Nested Loops,Pointers,Multi-Dimensional Arrays.
conceptParent: output,function,preprocessor_directive,simple_type,pointer,
structure_type,array,type_cast,variable_manipulation,selection,iteration,jump,
arithmetical_expression,increment_decrement,assignment,logical_expression,
Expression,printing (printf),arithmetic expressions,variables,increment decrement,
logical expressions,conditionals (if else),complex conditionals,
character processing,loops (while),arrays,hidden,Printing,Arithmetic Expressions,
Local Variables,Increment and Decrement,Logical Expressions,Simple Decisions,
Complex Decisions,Global Variables & Simple Functions,Character Processing,
while-Loop,Arrays.
session: 21BC5,DB414,A0B33,56604,7ACB9,...,78AA0.
datentime: 2007-04-21T17:51:41,2007-04-21T17:52:09,2007-04-21T17:52:21,
2007-04-21T17:52:52,2007-04-23T15:13:24,...,2007-04-17T22:21:34.
svc: 30.0,0.0,0.1,23.31,20.2,23.0,30.2,30.1,32.0,quizguide,32.31,23.3,31.2,23.1,

```


23.21,31.0,31.1,33.0,23.41,11.1,13.4,13.31,23.4,10.0,33.01,13.0,13.01,0.01,
 23.01,33.2,33.21,32.1,33.31,30.01,12.0,13.3,12.11,33.1,33.3,32.2,0.11,20.0,
 0.2,31.11,13.41,13.2,33.4,31.21,32.21,23.2,22.0,31.3,32.3,33.41,20.1,
 31.31,32.11,0.3,0.4,0.31,0.41,0.21,33.11,11.01.

Extracto del fichero de datos (*.data)

```
uid_199,gid_190,0,2dimensional_array1,q_1,printf,output,21BC5,2007-04-21T17:51:41,,no
uid_199,gid_190,1,2dimensional_array1,q_1,printf,output,21BC5,2007-04-21T17:52:09,,yes
uid_199,gid_190,0,2dimensional_array1,q_1,printf,output,21BC5,2007-04-21T17:52:21,,no
uid_199,gid_190,1,2dimensional_array1,q_1,printf,output,21BC5,2007-04-21T17:52:52,,yes
uid_194,gid_190,0,2dimensional_array1,q_1,printf,output,DB414,2007-04-23T15:13:24,30,no
uid_359,gid_72,1,2dimensional_array1,q_1,printf,output,A0B33,2007-05-02T18:35:05,30,yes
uid_456,gid_441,1,2dimensional_array1,q_1,printf,output,56604,2007-09-25T22:41:40,30,yes
uid_200,gid_190,0,2dimensional_array2,q_2,printf,output,7ACB9,2007-04-09T19:28:09,0.00,
uid_199,gid_190,0,2dimensional_array2,q_2,printf,output,21BC5,2007-04-21T18:01:59,,no
uid_199,gid_190,1,2dimensional_array2,q_2,printf,output,21BC5,2007-04-21T18:02:40,,yes
uid_194,gid_190,0,2dimensional_array2,q_2,printf,output,DB414,2007-04-23T15:14:22,30,no
uid_456,gid_441,0,2dimensional_array2,q_2,printf,output,01E22,2007-10-02T22:16:22,.10,n
uid_200,gid_190,0,2dimensional_array3,q_1,printf,output,7ACB9,2007-04-09T19:29:14,0.00,
uid_199,gid_190,1,2dimensional_array3,q_1,printf,output,21BC5,2007-04-21T18:08:29,,yes
uid_194,gid_190,0,2dimensional_array3,q_1,printf,output,DB414,2007-04-23T15:15:28,30,no
uid_200,gid_190,0,2dimensional_array3,q_3,printf,output,7ACB9,2007-04-09T19:29:50,0.00,
uid_199,gid_190,0,2dimensional_array3,q_3,printf,output,21BC5,2007-04-21T18:12:47,,no
uid_199,gid_190,0,2dimensional_array3,q_3,printf,output,21BC5,2007-04-21T18:13:07,,no
uid_199,gid_190,1,2dimensional_array3,q_3,printf,output,21BC5,2007-04-21T18:13:23,,yes
uid_194,gid_190,0,2dimensional_array3,q_3,printf,output,DB414,2007-04-23T15:15:58,30,no
```

Para el experimento realizado en la sección 5.3.3.1 (página 116) los atributos son las variables *groupid* y *activity*. Por esta razón, en los siguientes ficheros se eliminó la información relativa a los parámetros: *userid*, *concept*, *conceptParent*, *result*, *session*, *dateNTime* y *svc*.

Fichero de descripción de los datos (*.names):

```
no,yes.
groupid: gid_190,gid_72,gid_441,gid_373,gid_442,gid_394,gid_67.
activity: 2dimensional_array1,2dimensional_array2,2dimensional_array3,
arithmetic_expression1,arithmetic_expression2,character_array1,character_array2,
character_array3,character_processing1,character_processing2,character_processing3,
complex_conditional1,conditional_operator1,do2,function1,function2,
logical_expression1,nested_loop1,pointer2,pointer3,printing1,printing2,switch1,
variable2,variable3,array1,array2,array3,complex_conditional2,compound_assignment1,
constant1,constant2,do1,for1,for2,function3,if_else1,if_else2,
increment_decrement1,logical_operator1,pointer1,variable1,while1,while2,
globvar_simplefunc1,globvar_simplefunc2.
```

Fichero de datos (*.data): el fichero completo contiene 52734 líneas, por esta razón sólo se incluye una parte (desde la línea 1 hasta la 101, y desde la 52699 hasta la 52734) del fichero de datos. De esta forma, el lector puede tener una imagen real de los datos analizados.

```
gid_190,2dimensional_array1,no
gid_190,2dimensional_array1,yes
gid_190,2dimensional_array1,no
gid_190,2dimensional_array1,yes
gid_190,2dimensional_array1,no
gid_72,2dimensional_array1,yes
gid_441,2dimensional_array1,yes
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,yes
gid_190,2dimensional_array2,no
gid_441,2dimensional_array2,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,no
gid_190,arithmetric_expression1,no
gid_190,arithmetric_expression1,yes
gid_190,arithmetric_expression1,no
gid_190,arithmetric_expression1,yes
gid_190,arithmetric_expression1,yes
gid_190,arithmetric_expression1,no
gid_190,arithmetric_expression1,no
gid_190,arithmetric_expression1,yes
gid_190,arithmetric_expression1,no
gid_190,arithmetric_expression1,no
gid_190,arithmetric_expression1,no
gid_72,arithmetric_expression1,yes
gid_190,arithmetric_expression1,yes
gid_190,arithmetric_expression1,yes
gid_190,arithmetric_expression1,no
gid_190,arithmetric_expression1,yes
gid_190,arithmetric_expression1,no
gid_72,arithmetric_expression1,no
gid_190,arithmetric_expression1,no
```

gid_373,arithmetic_expression1,yes
gid_373,arithmetic_expression1,no
gid_373,arithmetic_expression1,no
gid_373,arithmetic_expression1,no
gid_373,arithmetic_expression1,yes
gid_441,arithmetic_expression1,yes
gid_441,arithmetic_expression1,no
gid_441,arithmetic_expression1,no
gid_442,arithmetic_expression1,yes
gid_442,arithmetic_expression1,yes
gid_441,arithmetic_expression1,yes
gid_441,arithmetic_expression1,yes
gid_442,arithmetic_expression1,yes
gid_442,arithmetic_expression1,yes
gid_373,arithmetic_expression1,no
gid_373,arithmetic_expression1,no
gid_441,arithmetic_expression1,yes
gid_441,arithmetic_expression1,yes
gid_441,arithmetic_expression1,no
gid_441,arithmetic_expression1,yes
gid_441,arithmetic_expression1,no
gid_441,arithmetic_expression1,no
gid_441,arithmetic_expression1,no
gid_72,arithmetic_expression1,no
gid_442,arithmetic_expression1,no
gid_442,arithmetic_expression1,yes
gid_441,arithmetic_expression1,no
gid_441,arithmetic_expression1,yes
gid_72,arithmetic_expression1,no
gid_72,arithmetic_expression1,yes
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,yes
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,yes
gid_190,arithmetic_expression1,yes
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,yes
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,yes

gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,yes
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,yes
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_190,arithmetic_expression1,no
gid_373,arithmetic_expression1,yes
gid_373,arithmetic_expression1,yes
gid_441,arithmetic_expression1,no
gid_441,arithmetic_expression1,yes
gid_442,arithmetic_expression1,yes
gid_442,arithmetic_expression1,no
gid_442,arithmetic_expression1,no
gid_441,arithmetic_expression1,yes
gid_442,arithmetic_expression1,yes
gid_442,arithmetic_expression1,no
...
...
...
gid_441,2dimensional_array1,yes
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,yes
gid_190,2dimensional_array2,no
gid_441,2dimensional_array2,no
gid_441,2dimensional_array2,no
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,yes
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,no
gid_190,2dimensional_array2,yes
gid_190,2dimensional_array2,no
gid_441,2dimensional_array2,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,no

```
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,no
gid_190,2dimensional_array3,yes
gid_190,2dimensional_array3,no
```

A.5.2. Curso de “Sistemas Operativos I” en el sistema *CoMoLE*

Los logs que genera el sistema *CoMoLE* para cada estudiante se guardan en dos ficheros *XML*: perfil del usuario e interacciones con el sistema. El perfil del usuario contiene campos relativos al nombre del estudiante (campo *fullName*), su dirección de correo electrónico (campo *eaddress*), su edad (campo *age*), el idioma en el que se mostrarán las actividades (campo *language*), los cursos a los que está inscrito (campo *courses*), y el estilo de aprendizaje del estudiante (campo *LS*). En las siguientes líneas se puede ver el perfil del estudiante José Suárez³. Se observa que este estudiante está inscrito en el curso de *SO1*, y que su estilo de aprendizaje se corresponde con las dimensiones: verbal, global, reflexivo y sensitivo.

```
<?xml version="1.0" encoding="UTF-8"?>
<user password="123456">
  <fullName>
    <name>jose</name>
    <surname>suarez</surname>
  </fullName>
  <eaddress>
    <email>jose.suarez@estudiante.uam.es</email>
  </eaddress>
  <age>20</age>
  <language>spanish</language>
  <courses>
    <course id="SO1" location="" />
  </courses>
  <LS visual="n" secuencial="n" active="n" sensitive="y" />
</user>
```

³Se ha usado un nombre ficticio para este estudiante, con el fin de preservar su privacidad. El resto de los campos, excepto el *e-mail*, son los que aparecen en el perfil real de este estudiante.

Las interacciones del usuario con el sistema se guardan en un fichero *XML*, llamado “datos dinámicos del usuario X”. Este fichero está dividido en varias partes. Al principio del fichero se hace una descripción sobre el contexto en el que se encuentra el estudiante, es decir, dispositivo en el que se le muestran las actividades (campo *device*), tiempo del que dispone para realizar las actividades (campo *time*), localización física del usuario (campo *location*), árbol de actividades que se presentan al usuario (campo *tree*), valoración por parte de los usuarios de la adaptación realizada (campo *feedback*), y características de la actividad realizada (campo *activity*). A continuación se muestra un extracto del fichero de interacciones para el estudiante José Suárez. Se puede observar que este estudiante utilizó un ordenador personal en un sitio que no era ni su casa, ni el laboratorio, ni en clase (localización *others*), y dispuso de 10 minutos para realizar las actividades. Realizó una serie de actividades, entre ellas la actividad “PagEjem1”, que es de tipo ejemplo, no es compuesta y fue finalizada con éxito⁴. Para mayor información consúltese [Martín 2008].

```
<?xml version="1.0" encoding="UTF-8"?>
<DynamicData>
  <device>pc</device>
  <Time>10</Time>
  <location>others</location>
  <Tree>
    <Nodo id="GestionMem" estado="2" nota="0.0" cont="">
      <Nodo id="PagSimple" padre="GestionMem" estado="2" nota="0.0" cont="">
        <Nodo id="PagEjem1" padre="PagSimple" estado="2" nota="0.0" cont="" />
        <Nodo id="PagEjem2" padre="PagSimple" estado="0" nota="0.0" cont="" />
        <Nodo id="PagIntroTeo" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="TradPagEjem" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="TradPagTeo" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleTest1" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleTest2" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleTest3" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleRptaLibre1" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleRptaLibre2" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
      </Nodo>
    </Nodo>
  </Tree>
</DynamicData>
```

⁴Esta descripción corresponde a la línea: <Activity id = “PagEjem1” ma = “SO1” fFin = “Mon Jun 23 09:27:54 CEST 2008” compuesta = “n” finalizada = “F” tipo = “example” nota = “10.0” />.

```

    </Nodo>
    ...
</Tree>
<Tree>
  <Nodo id="GestionMem" estado="2" nota="0.0" cont="">
    <Nodo id="PagSimple" padre="GestionMem" estado="2" nota="0.0" cont="">
      <Nodo id="PagEjem1" padre="PagSimple" estado="4" nota="0.0"
        cont="UbiPag_ejem1_pc.htm" />
      <Nodo id="PagEjem2" padre="PagSimple" estado="2" nota="0.0" cont="" />
      <Nodo id="PagIntroTeo" padre="PagSimple" estado="0" nota="0.0" cont="" />
      <Nodo id="TradPagEjem" padre="PagSimple" estado="0" nota="0.0" cont="" />
      <Nodo id="TradPagTeo" padre="PagSimple" estado="0" nota="0.0" cont="" />
      <Nodo id="PagSimpleTest1" padre="PagSimple" estado="0" nota="0.0"
        cont="" />
      <Nodo id="PagSimpleTest2" padre="PagSimple" estado="0" nota="0.0"
        cont="" />
      <Nodo id="PagSimpleTest3" padre="PagSimple" estado="0" nota="0.0"
        cont="" />
      <Nodo id="PagSimpleRptaLibre1" padre="PagSimple" estado="0" nota="0.0"
        cont="" />
      <Nodo id="PagSimpleRptaLibre2" padre="PagSimple" estado="0" nota="0.0"
        cont="" />
    </Nodo>
    ...
  </Tree>
  <Feedback id="PagEjem1" adecuacion="0" />
  <Activity id="PagEjem1" ma="S01" fFin="Mon Jun 23 09:27:54 CEST 2008"
    compuesta="n" finalizada="F" tipo="example" nota="10.0" />
  <Tree>
    <Nodo id="GestionMem" estado="2" nota="0.1" cont="">
      <Nodo id="PagSimple" padre="GestionMem" estado="2" nota="1.0" cont="">
        <Nodo id="PagEjem1" padre="PagSimple" estado="4" nota="10.0"
          cont="UbiPag_ejem1_pc.htm" />
        <Nodo id="PagEjem2" padre="PagSimple" estado="2" nota="0.0" cont="" />
        <Nodo id="PagIntroTeo" padre="PagSimple" estado="0" nota="0.0" cont="" />
        <Nodo id="TradPagEjem" padre="PagSimple" estado="0" nota="0.0" cont="" />
        <Nodo id="TradPagTeo" padre="PagSimple" estado="0" nota="0.0" cont="" />
        <Nodo id="PagSimpleTest1" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleTest2" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleTest3" padre="PagSimple" estado="0" nota="0.0"
          cont="" />
        <Nodo id="PagSimpleRptaLibre1" padre="PagSimple" estado="0" nota="0.0"

```

```

        cont="" />
        <Nodo id="PagSimpleRptaLibre2" padre="PagSimple" estado="0" nota="0.0"
        cont="" />
    </Nodo>
    ...
</Tree>
<Feedback id="CompRes" adecuacion="0" />
<Activity id="CompRes" ma="S01" fFin="Mon Jun 23 09:30:17 CEST 2008"
    compuesta="n" finalizada="F" tipo="example" nota="10.0" />
<Tree>
    ...
</DynamicData>

```

Como se puede observar, la información relevante para los experimentos realizados está contenida en el campo *activity*. Este campo indica el estado de una actividad, el tipo de actividad y la nota o puntuación obtenida en una actividad. Hay que indicar que, también son relevantes los campos relativos al contexto en el que se encuentra el estudiante, tales son: *device*, *time* y *location*. Con el objetivo de integrar esta información con los perfiles de los estudiantes, se construyó la tabla de logs, que contiene todos los perfiles de los estudiantes junto con sus interacciones. Esta tabla contiene las siguientes variables:

- Variables que describen el perfil: guardan la información relativa a las variables identificador del estudiante en el sistema y las componentes del estilo de aprendizaje (dimensiones del modelo de Felder-Silverman [Felder 1988]).
 - ***IdUser***: identificador del estudiante en el sistema.
 - ***Visual***: dimensión *visual/verbal* del modelo de estilos de aprendizaje de Felder-Silverman. Esta dimensión está relacionada con la forma en la que los contenidos son mostrados. Puede tener dos valores: “y” (indica mayor dimensión *visual* que *verbal*) y “n” (indica mayor dimensión *verbal* que *visual*).
 - ***Sequential***: dimensión *sequential/global* del modelo de Felder-Silverman. Esta dimensión se refiere a la manera en la que los estudiantes entienden la información. Dos valores son posibles: “y” (indica mayor dimensión *sequential* que *global*) y “n” (indica mayor dimensión *global* que *sequential*).
 - ***Active***: dimensión *active/reflexive* del modelo de estilos de aprendizaje anterior. Esta dimensión indica la manera preferida de los estudiantes para procesar la información. Dos valores están disponibles: “y” (indica mayor dimensión *active* que *reflexive*) y “n” (indica mayor dimensión *reflexive* que *active*).
 - ***Sensitive***: dimensión *sensitive/intuitive* del modelo de Felder-Silverman. Esta dimensión indica la forma en la que los estudiantes perciben la información. Puede tener dos valores: “y” (indica mayor dimensión *sensitive* que *intuitive*) y “n” (indica mayor dimensión *intuitive* que *sensitive*).

- Variables respecto al contexto: guardan la información relativa a la localización del usuario, dispositivo utilizado para mostrar las actividades y el tiempo disponible por el usuario.
 - **Device**: tipo de dispositivo en el que las actividades son presentadas a los estudiantes. Dos tipos son posibles:
 1. *PC (Personal Computer)*
 2. *PDA (Personal Digital Assistant)*
 - **Location**: lugar en el que se encuentra el estudiante al realizar las actividades. Existen tres posibles lugares:
 1. *home*: el usuario se encuentra en su casa realizando las actividades.
 2. *laboratory*: el usuario se encuentra en la sala de ordenadores de la *EPS* realizando las actividades.
 3. *class*: el usuario se encuentra en clase de teoría.
 4. *others*: el usuario se encuentra en cualquier otro sitio.
 - **TimeUser**: tiempo del que dispone el estudiante para finalizar las actividades en la sesión. El estudiante fija este tiempo al principio de la sesión. Valores posibles para esta variable son los siguientes: 10 minutos, 20 minutos, 30 minutos, 1 hora, 1–2 horas y más de 2 horas. Este tiempo es utilizado por el sistema para realizar una estimación sobre qué actividades se le pueden ofertar al estudiante en función del tiempo disponible.
- Variables que describen la actividad: guardan la información relacionada con la actividad: su nombre, tipo, su estado (si se ha finalizado o no), y la nota obtenida por el estudiante en la actividad.
 - **NameAct**: identificador de la actividad de aprendizaje.
 - **Type**: indicador del tipo de actividad. Hay cinco tipos de actividades disponibles en este curso:
 1. teóricas (*theory*).
 2. ejemplos (*examples*).
 3. test de auto-evaluación (*test*).
 4. ejercicios de respuesta rápida (*short_text*).
 5. actividades colaborativas (*collaborative activities*).
 - **Finalization**: indica si la actividad se terminó o no. Dos valores son posibles: “F” (actividad finalizada) y “N” (actividad no finalizada).
 - **Score**: almacena el resultado obtenido en la actividad cuando ésta ha sido finalizada. Es una variable continua que toma valores entre 0, la peor puntuación que se puede obtener, y 10, la mejor puntuación o nota.
 - **Grade**: variable generada de forma artificial a partir de la variable *score*. Esto quiere decir, que esta variable no es generada por *CoMoLE*. Puede recibir dos valores:

1. *LOW* : toma este valor cuando $0 \leq score < 5$.
2. *HIGH* : toma este valor cuando $5 \leq score \leq 10$.

Para generar el fichero de logs final, se decidió eliminar las variables *finalization* y *score*, así como la información relativa a éstas en los datos de interacción. La variable *finalization* indica si una actividad se finalizó o no, pero en el conjunto de datos actual todas las actividades fueron finalizadas. Por tanto, no tiene sentido añadir esta variable en el estudio. La variable *score* almacena la puntuación en la actividad, pero incluir esta información es redundante en los datos de análisis, ya que la variable *grade* indica si ha sido o no superada la actividad. El programa *c4.5rules* exige que el fichero de datos (*.data) sea descrito mediante un fichero de descripción (*.names). A continuación se presentan los ficheros: fichero de descripción y fichero de *datos de análisis*.

Fichero de descripción de los datos (*.names):

```
LOW,HIGH.
visual: y,n.
sequential: y,n.
active: n,y.
sensitive: n,y.
device: pc,pda.
location: home,lab,others, class.
nameAct: PagSimpleTest1,PagSimpleTest2,PagSimpleTest3,PagSimpleRptaLibre1,PagSimpleRptaLibre2,SegSimpleTest1,SegSimpleTest2,Mem_Test1,Mem_Test2,Mem_Test3,Mem_Test4,Mem_Test5,PSRptaLibre1,PSRptaLibre2,MV_Test1,MV_Test2,MV_Test3,MV_Test4,MV_Test5,MV_Test6,MV_Test7,MV_Test8,MV_Test9,MV_Test10,MV_Test11,MV_Test12,MV_Test13,MV_Test14,MV_Test15,MV_Test16,MV_Test17,MV_Test18,MV_Test19,MultiRptaLibre1,MultiRptaLibre2,PagMVEjer1,PagMVEjer2,PagMVEjer3,PagMVEjer4,PagMVEjer5,PagMVEjer6,PagMVEjer7,SegMVEjer,ColS01,TablasRptaLibre1,TablasTest2.
tipo: test,short_text,collaborative.
```

Fichero de *datos de análisis* (*.data): las siguientes líneas son un extracto (las 100 primeras líneas y las 50 últimas) del fichero de datos original.

```
y,y,n,n,pc,home,PagSimpleTest1,test,LOW
y,y,n,n,pc,home,PagSimpleTest2,test,HIGH
y,y,n,n,pc,home,PagSimpleTest3,test,LOW
y,y,n,n,pc,home,PagSimpleRptaLibre1,short_text,LOW
y,y,n,n,pc,home,PagSimpleRptaLibre2,short_text,LOW
y,y,n,n,pc,home,SegSimpleTest1,test,HIGH
y,y,n,n,pc,home,SegSimpleTest2,test,LOW
y,y,n,n,pc,home,Mem_Test1,test,LOW
y,y,n,n,pc,home,Mem_Test2,test,LOW
y,y,n,n,pc,home,Mem_Test3,test,LOW
y,y,n,n,pc,home,Mem_Test4,test,HIGH
y,y,n,n,pc,home,Mem_Test5,test,LOW
```

```
y,y,n,n,pc,home,PSRptaLibre1,short_text,LOW
y,y,n,n,pc,home,PSRptaLibre2,short_text,LOW
y,y,n,n,pc,home,MV_Test1,test,LOW
y,y,n,n,pc,home,MV_Test2,test,LOW
y,y,n,n,pc,home,MV_Test3,short_text,LOW
y,y,n,n,pc,home,MV_Test4,test,LOW
y,y,n,n,pc,home,MV_Test5,test,LOW
y,y,n,n,pc,home,MV_Test6,test,LOW
y,y,n,n,pc,home,MV_Test7,test,LOW
y,y,n,n,pc,home,MV_Test8,test,LOW
y,y,n,n,pc,home,MV_Test9,test,LOW
y,y,n,n,pc,home,MV_Test10,test,LOW
y,y,n,n,pc,home,MV_Test11,test,LOW
y,y,n,n,pc,home,MV_Test12,short_text,LOW
y,y,n,n,pc,home,MV_Test13,short_text,LOW
y,y,n,n,pc,home,MV_Test14,test,LOW
y,y,n,n,pc,home,MV_Test15,test,LOW
y,y,n,n,pc,home,MV_Test16,test,LOW
y,y,n,n,pc,home,MV_Test17,test,LOW
y,y,n,n,pc,home,MV_Test18,test,LOW
y,y,n,n,pc,home,MV_Test19,short_text,LOW
y,y,n,n,pc,home,MultiRptaLibre1,short_text,LOW
y,y,n,n,pc,home,MultiRptaLibre2,short_text,LOW
y,y,n,n,pc,home,PagMVEjer1,test,LOW
y,y,n,n,pc,home,PagMVEjer2,short_text,LOW
y,y,n,n,pc,home,PagMVEjer3,short_text,LOW
y,y,n,n,pc,home,PagMVEjer4,short_text,LOW
y,y,n,n,pc,home,PagMVEjer5,short_text,LOW
y,y,n,n,pc,home,PagMVEjer6,short_text,LOW
y,y,n,n,pc,home,PagMVEjer7,short_text,LOW
y,y,n,n,pc,home,SegMVEjer,short_text,LOW
y,y,n,n,pc,home,ColS01,collaborative,HIGH
y,y,n,n,pc,home,TablasRptaLibre1,short_text,LOW
y,y,n,n,pc,home,TablasTest2,test,LOW
y,n,n,y,pc,home,Mem_Test1,test,HIGH
y,n,n,y,pc,home,Mem_Test2,test,LOW
y,n,n,y,pc,home,Mem_Test3,test,HIGH
y,n,n,y,pc,home,Mem_Test4,test,HIGH
y,n,n,y,pc,home,Mem_Test5,test,LOW
y,n,n,y,pc,home,MV_Test1,test,LOW
y,n,n,y,pc,home,MV_Test7,test,LOW
y,n,n,y,pc,home,MV_Test2,test,HIGH
y,n,n,y,pc,home,MV_Test3,short_text,LOW
y,n,n,y,pc,home,MV_Test4,test,HIGH
```

y,n,n,y,pc,home,MV_Test5,test,HIGH
y,n,n,y,pc,home,MV_Test6,test,HIGH
y,n,n,y,pc,home,MV_Test8,test,HIGH
y,n,n,y,pc,home,MV_Test9,test,HIGH
y,n,n,y,pc,home,MV_Test10,test,LOW
y,n,n,y,pc,home,MV_Test11,test,LOW
y,n,n,y,pc,home,MV_Test12,short_text,HIGH
y,n,n,y,pc,home,MV_Test13,short_text,HIGH
y,n,n,y,pc,home,MV_Test14,test,HIGH
y,n,n,y,pc,home,MV_Test15,test,HIGH
y,n,n,y,pc,home,MV_Test16,test,HIGH
y,n,n,y,pc,home,MV_Test17,test,HIGH
y,n,n,y,pc,home,MV_Test18,test,HIGH
y,n,n,y,pc,home,MV_Test19,short_text,LOW
y,n,n,y,pc,home,ColSO1,collaborative,HIGH
y,n,n,y,pc,home,PagSimpleTest1,test,LOW
y,n,n,y,pc,home,PagSimpleTest2,test,LOW
y,n,n,y,pc,home,PagSimpleTest3,test,LOW
y,n,n,y,pc,home,PagSimpleRptaLibre1,short_text,LOW
y,n,n,y,pc,home,SegSimpleTest1,test,LOW
y,n,n,y,pc,home,SegSimpleTest2,test,LOW
y,n,n,y,pc,home,PagSimpleRptaLibre2,short_text,LOW
y,n,n,y,pc,home,PSRptaLibre1,short_text,LOW
y,n,n,y,pc,home,PSRptaLibre2,short_text,LOW
y,n,n,y,pc,home,TablasRptaLibre1,short_text,LOW
y,n,n,y,pc,home,TablasTest2,test,LOW
y,n,n,y,pc,home,SegMVEjer,short_text,LOW
y,n,n,y,pc,home,MultiRptaLibre1,short_text,LOW
y,n,n,y,pc,home,MultiRptaLibre2,short_text,LOW
y,n,n,y,pc,home,PagMVEjer1,test,LOW
y,n,n,y,pc,home,PagMVEjer2,short_text,LOW
y,n,n,y,pc,home,PagMVEjer3,short_text,LOW
y,n,n,y,pc,home,PagMVEjer4,short_text,LOW
y,n,n,y,pc,home,PagMVEjer5,short_text,LOW
y,n,n,y,pc,home,PagMVEjer6,short_text,LOW
y,n,n,y,pc,home,PagMVEjer7,short_text,LOW
n,y,n,y,pc,home,PagSimpleTest1,test,LOW
n,y,n,y,pc,home,PagSimpleTest2,test,LOW
n,y,n,y,pc,home,PagSimpleTest3,test,LOW
n,y,n,y,pc,home,SegSimpleTest1,test,LOW
n,y,n,y,pc,home,PagSimpleRptaLibre1,short_text,LOW
n,y,n,y,pc,home,SegSimpleTest2,test,LOW
n,y,n,y,pc,home,Mem_Test1,test,LOW
n,y,n,y,pc,home,Mem_Test2,test,LOW

```
...
...
...
y,y,y,y,pc,home,MV_Test7,test,LOW
y,y,y,y,pc,home,MV_Test8,test,LOW
y,y,y,y,pc,home,MV_Test9,test,LOW
y,y,y,y,pc,home,MV_Test10,test,LOW
y,y,y,y,pc,home,MV_Test11,test,LOW
y,y,y,y,pc,home,MV_Test12,short_text,LOW
y,y,y,y,pc,home,MV_Test13,short_text,LOW
y,y,y,y,pc,home,MV_Test14,test,LOW
y,y,y,y,pc,home,MV_Test15,test,LOW
y,y,y,y,pc,home,MV_Test16,test,LOW
y,y,y,y,pc,home,MV_Test17,test,LOW
y,y,y,y,pc,home,MV_Test18,test,LOW
y,y,y,y,pc,home,PagMVEjer1,test,LOW
y,y,y,y,pc,home,PagMVEjer2,short_text,LOW
y,y,y,y,pc,home,PagMVEjer3,short_text,LOW
y,y,y,y,pc,home,PagMVEjer4,short_text,LOW
y,y,y,y,pc,home,PagMVEjer5,short_text,LOW
y,y,y,y,pc,home,PagMVEjer6,short_text,LOW
y,y,y,y,pc,home,PagMVEjer7,short_text,LOW
y,n,n,y,pc,others,MV_Test1,test,LOW
y,y,n,y,pda,others,SegSimpleTest1,test,LOW
y,y,n,y,pc,home,SegSimpleTest2,test,LOW
y,y,n,y,pc,home,Mem_Test1,test,HIGH
y,y,n,y,pc,home,Mem_Test2,test,LOW
y,y,n,y,pc,home,Mem_Test3,test,LOW
y,y,n,y,pc,home,Mem_Test4,test,LOW
y,y,n,y,pc,home,Mem_Test5,test,LOW
y,y,n,y,pc,home,MV_Test1,test,LOW
y,y,n,y,pc,home,MV_Test2,test,HIGH
y,y,n,y,pc,home,MV_Test4,test,HIGH
y,y,n,y,pc,home,MV_Test5,test,LOW
y,y,n,y,pc,home,MV_Test6,test,LOW
y,y,n,y,pc,home,MV_Test7,test,LOW
y,y,n,y,pc,home,MV_Test8,test,HIGH
y,y,n,y,pc,home,MV_Test9,test,LOW
y,y,n,y,pc,home,MV_Test10,test,LOW
y,y,n,y,pc,home,MV_Test11,test,LOW
y,y,n,y,pc,home,MV_Test14,test,LOW
y,y,n,y,pc,home,MV_Test15,test,LOW
y,y,n,y,pc,home,MV_Test16,test,LOW
y,y,n,y,pc,home,MV_Test17,test,HIGH
```

```

y,y,n,y,pc,home,MV_Test18,test,LOW
y,y,n,y,pc,home,MV_Test19,short_text,LOW
y,y,n,y,pc,home,TablasTest2,test,HIGH
y,y,n,y,pc,home,PagMVEjer1,test,LOW
y,y,n,y,pc,home,PagSimpleRptaLibre1,short_text,LOW
y,y,n,y,pc,home,PagMVEjer5,short_text,LOW
y,y,n,y,pc,home,PagMVEjer6,short_text,LOW
y,y,n,y,pc,home,PagMVEjer7,short_text,LOW
y,y,n,y,pc,home,SegMVEjer,short_text,LOW

```

A.5.3. Curso de “Estructura de Datos y de la Información I” en el sistema *CoMoLE*

La estructura de los ficheros para este curso es exactamente la misma que en el curso de *SO1*. Por tanto, la creación de un fichero de logs que agrupe la información de los perfiles de los usuarios y sus interacciones es también idéntica. No obstante, existe una ligera diferencia respecto a las actividades ofrecidas por *CoMoLE*, ya que en este caso se substituyen las actividades de tipo colaborativo por actividades de repaso. Hay que indicar, que este tipo de actividades se mostrarán automáticamente a los estudiantes que obtengan puntuaciones bajas. De este modo, los tipos de actividades disponibles son:

- Teóricas (*theory*).
- Ejemplos (*examples*).
- Test de auto-evaluación (*test*).
- Ejercicios de respuesta rápida (*short_text*).
- Actividades de repaso (*review*).

A continuación se muestran los ficheros finales de logs: fichero de descripción y fichero de datos.

Fichero de descripción de los datos (*.names):

```

HIGH,LOW.
visual: y,n.
sequential: y,n.
active: n,y.
sensitive: y,n.
device: pc,pda.
location: home,others,lab.
timeUser: continuous.
nameAct: EnumTest1,EnumTest2,OperadoresTest1,OperadoresTest2,OperadoresTest3,
OperadoresRptaLibre1,OperadoresRptaLibre2,OperadoresRptaLibre3,

```

OperadoresRptaLibre4, OperadoresRptaLibre5, IfTest, SwitchRptaLibre1, SwitchRptaLibre2, CondReview, IfEjemRC, IfTestRC, SwitchEjemRC, SwitchRptaLibre1RC, SwitchRptaLibre2RC, WhileTest, DoWhileTest, ForTest, DoWhileRptaLibre, ESTest1, ESTest2, ESTest3, EstructurasRptaLibre, NuevosDatosTest, FuncTest1, FuncTest2, FuncTest3, PasoArgumentosRptaLibre1, PasoArgumentosRptaLibre2, ArraysTest1, ArraysTest2, ArraysTest3, ArraysTest4, ArraysTest5, ArraysTest6, ArraysTest7, ArraysRptaLibre1, ArraysRptaLibre2, ArraysRptaLibre3, MatricesRptaLibre, PunterosTest1, PunterosTest2, PunterosTest3, PunterosTest4, PunterosTest5, PunterosRptaLibre1, PunterosRptaLibre2, PunterosRptaLibre3, PunterosRptaLibre4.
 tipo: test, short_text, review.
 finalizada: F, N.

Fichero de *datos de análisis* (*.data): las siguientes líneas son un extracto (las 100 primeras líneas y las 50 últimas) del fichero de datos original.

```

y,y,n,y,pc,home,120.0,EnumTest1,test,F,HIGH
y,y,n,y,pc,home,120.0,EnumTest2,test,F,LOW
y,y,n,y,pc,home,120.0,OperadoresTest1,test,F,HIGH
y,y,n,y,pc,home,120.0,OperadoresTest2,test,F,HIGH
y,y,n,y,pc,home,120.0,OperadoresTest3,test,F,HIGH
y,y,n,y,pc,home,120.0,OperadoresRptaLibre1,short_text,F,LOW
y,y,n,y,pc,home,120.0,OperadoresRptaLibre2,short_text,F,HIGH
y,y,n,y,pc,home,120.0,OperadoresRptaLibre3,short_text,F,LOW
y,y,n,y,pc,home,120.0,OperadoresRptaLibre4,short_text,F,HIGH
y,y,n,y,pc,home,120.0,OperadoresRptaLibre5,short_text,F,HIGH
y,y,n,y,pc,home,120.0,IfTest,test,F,LOW
y,y,n,y,pc,home,120.0,SwitchRptaLibre1,short_text,F,LOW
y,y,n,y,pc,home,120.0,SwitchRptaLibre2,short_text,F,LOW
y,y,n,y,pc,home,120.0,CondReview,review,F,HIGH
y,y,n,y,pc,home,120.0,IfEjemRC,review,F,HIGH
y,y,n,y,pc,home,120.0,IfTestRC,review,F,HIGH
y,y,n,y,pc,home,120.0,SwitchEjemRC,review,F,HIGH
y,y,n,y,pc,home,120.0,SwitchRptaLibre1RC,review,F,HIGH
y,y,n,y,pc,home,120.0,SwitchRptaLibre2RC,review,F,HIGH
y,y,n,y,pc,home,120.0,WhileTest,test,F,LOW
y,y,n,y,pc,home,120.0,DoWhileTest,test,F,LOW
y,y,n,y,pc,home,120.0,ForTest,test,F,HIGH
y,y,n,y,pc,home,120.0,DoWhileRptaLibre,short_text,F,HIGH
y,y,n,y,pc,home,120.0,ESTest1,test,F,LOW
y,y,n,y,pc,home,120.0,ESTest2,test,F,HIGH
y,y,n,y,pc,home,120.0,ESTest3,test,F,LOW
n,n,n,y,pc,home,20.0,EnumTest1,test,F,HIGH
n,n,n,y,pc,home,20.0,EnumTest2,test,F,LOW
n,n,n,y,pc,home,20.0,OperadoresTest1,test,F,HIGH
n,n,n,y,pc,home,20.0,OperadoresTest2,test,F,LOW

```

n,n,n,y,pc,home,20.0,OperadoresTest3,test,F,HIGH
n,n,n,y,pc,home,20.0,OperadoresRptaLibre1,short_text,F,LOW
n,n,n,y,pc,home,20.0,OperadoresRptaLibre3,short_text,F,LOW
n,n,n,y,pc,home,20.0,OperadoresRptaLibre4,short_text,F,HIGH
n,n,n,y,pc,home,20.0,OperadoresRptaLibre5,short_text,F,LOW
n,n,n,y,pc,home,30.0,IfTest,test,F,HIGH
n,n,n,y,pc,home,30.0,SwitchRptaLibre1,short_text,F,HIGH
n,n,n,y,pc,home,30.0,SwitchRptaLibre2,short_text,F,HIGH
n,n,n,y,pc,home,30.0,CondReview,review,F,HIGH
n,n,n,y,pc,home,30.0,IfEjemRC,review,F,HIGH
n,n,n,y,pc,home,30.0,IfTestRC,review,F,HIGH
n,n,n,y,pc,home,30.0,SwitchRptaLibre1RC,review,F,HIGH
n,n,n,y,pc,home,30.0,SwitchRptaLibre2RC,review,F,HIGH
n,n,n,y,pc,home,30.0,WhileTest,test,F,HIGH
n,n,n,y,pc,home,30.0,DoWhileTest,test,F,LOW
n,n,n,y,pc,home,30.0,ForTest,test,F,LOW
n,n,n,y,pc,home,30.0,DoWhileRptaLibre,short_text,F,LOW
y,n,n,y,pc,others,20.0,EnumTest1,test,F,LOW
y,n,n,y,pc,others,20.0,EnumTest2,test,F,HIGH
y,y,n,y,pc,home,30.0,EnumTest1,test,F,HIGH
y,y,n,y,pc,home,30.0,EnumTest2,test,F,LOW
y,y,n,y,pc,home,30.0,OperadoresTest1,test,F,HIGH
y,y,n,y,pc,home,30.0,OperadoresTest2,test,F,HIGH
y,y,n,y,pc,home,30.0,OperadoresTest3,test,F,HIGH
y,y,n,y,pc,home,30.0,OperadoresRptaLibre1,short_text,F,LOW
y,y,n,y,pc,home,30.0,OperadoresRptaLibre2,short_text,F,HIGH
y,y,n,y,pc,home,30.0,OperadoresRptaLibre3,short_text,F,LOW
y,y,n,y,pc,home,30.0,OperadoresRptaLibre4,short_text,F,HIGH
y,y,n,y,pc,home,30.0,OperadoresRptaLibre5,short_text,F,LOW
y,y,n,y,pc,home,30.0,IfTest,test,F,LOW
y,y,n,y,pc,home,30.0,SwitchRptaLibre1,short_text,F,HIGH
y,y,n,y,pc,home,30.0,SwitchRptaLibre2,short_text,F,HIGH
y,y,n,y,pc,home,30.0,CondReview,review,F,HIGH
y,y,n,y,pc,home,30.0,IfEjemRC,review,F,HIGH
y,y,n,y,pc,home,30.0,IfTestRC,review,F,HIGH
y,y,n,y,pc,home,30.0,SwitchEjemRC,review,F,HIGH
y,y,n,y,pc,home,30.0,SwitchRptaLibre1RC,review,F,HIGH
y,y,n,y,pc,home,30.0,SwitchRptaLibre2RC,review,F,HIGH
y,y,n,y,pc,home,120.0,DoWhileTest,test,F,LOW
y,y,n,y,pc,home,120.0,DoWhileRptaLibre,short_text,F,LOW
y,y,n,y,pc,home,120.0,ForTest,test,F,HIGH
n,n,n,y,pc,home,10.0,EnumTest1,test,F,HIGH
n,n,n,y,pc,home,10.0,EnumTest2,test,F,HIGH
n,n,n,y,pc,home,10.0,OperadoresTest1,test,F,HIGH


```
n,n,n,y,pc,home,10.0,OperadoresTest2,test,F,HIGH
n,n,n,y,pc,home,10.0,OperadoresTest3,test,F,LOW
n,n,n,y,pc,home,10.0,OperadoresRptaLibre1,short_text,F,LOW
n,n,n,y,pc,home,10.0,OperadoresRptaLibre2,short_text,F,HIGH
n,n,n,y,pc,home,10.0,OperadoresRptaLibre3,short_text,F,LOW
n,n,n,y,pc,home,10.0,OperadoresRptaLibre4,short_text,F,HIGH
n,n,n,y,pc,home,10.0,OperadoresRptaLibre5,short_text,F,HIGH
n,n,n,y,pc,home,10.0,IfTest,test,F,HIGH
n,n,n,y,pc,home,10.0,SwitchRptaLibre1,short_text,F,HIGH
n,n,n,y,pc,home,10.0,SwitchRptaLibre2,short_text,F,HIGH
n,n,n,y,pc,home,10.0,WhileTest,test,F,HIGH
n,n,n,y,pc,home,10.0,DoWhileTest,test,F,LOW
n,n,n,y,pc,home,10.0,ForTest,test,F,LOW
n,n,n,y,pc,home,10.0,DoWhileRptaLibre,short_text,F,LOW
n,n,n,y,pc,home,10.0,ESTest1,test,F,HIGH
n,n,n,y,pc,home,10.0,ESTest2,test,F,HIGH
n,n,n,y,pc,home,10.0,ESTest3,test,F,LOW
n,n,n,y,pc,home,10.0,EstructurasRptaLibre,short_text,F,HIGH
n,n,n,y,pc,home,10.0,NuevosDatosTest,test,F,HIGH
n,n,n,y,pc,home,10.0,FuncTest1,test,F,LOW
n,n,n,y,pc,home,10.0,FuncTest2,test,F,HIGH
n,n,n,y,pc,home,10.0,FuncTest3,test,F,HIGH
n,n,n,y,pc,home,10.0,PasoArgumentosRptaLibre1,short_text,F,LOW
y,n,y,y,pc,home,60.0,EnumTest1,test,F,HIGH
y,n,y,y,pc,home,60.0,EnumTest2,test,F,HIGH
y,n,y,y,pc,home,60.0,OperadoresTest1,test,F,LOW
...
...
...
n,n,y,y,pc,home,240.0,IfTestRC,review,F,HIGH
n,n,y,y,pc,home,240.0,SwitchEjemRC,review,F,HIGH
n,n,y,y,pc,home,240.0,SwitchRptaLibre1RC,review,F,HIGH
n,n,y,y,pc,home,240.0,SwitchRptaLibre2RC,review,F,HIGH
n,n,y,y,pc,home,240.0,WhileTest,test,F,HIGH
n,n,y,y,pc,home,240.0,DoWhileTest,test,F,LOW
n,n,y,y,pc,home,240.0,ForTest,test,F,LOW
n,n,y,y,pc,home,240.0,DoWhileRptaLibre,short_text,F,LOW
n,n,y,y,pc,home,240.0,ESTest1,test,F,LOW
n,n,y,y,pc,home,240.0,ESTest2,test,F,LOW
n,n,y,y,pc,home,240.0,ESTest3,test,F,LOW
n,n,y,y,pc,home,240.0,EstructurasRptaLibre,short_text,F,HIGH
n,n,y,y,pc,home,240.0,NuevosDatosTest,test,F,HIGH
n,n,y,y,pc,home,240.0,FuncTest1,test,F,HIGH
n,n,y,y,pc,home,240.0,FuncTest2,test,F,HIGH
```

```

n,n,y,y,pc,home,240.0,FuncTest3,test,F,LOW
n,n,y,y,pc,home,240.0,PasoArgumentosRptaLibre1,short_text,F,HIGH
n,n,y,y,pc,home,240.0,PasoArgumentosRptaLibre2,short_text,F,LOW
n,n,y,y,pc,home,240.0,ArraysTest1,test,F,HIGH
n,n,y,y,pc,home,240.0,ArraysTest2,test,F,HIGH
n,n,y,y,pc,home,240.0,ArraysTest3,test,F,HIGH
n,n,y,y,pc,home,240.0,ArraysTest4,test,F,LOW
n,n,y,y,pc,home,240.0,ArraysTest5,test,F,LOW
n,n,y,y,pc,home,240.0,ArraysTest6,test,F,HIGH
n,n,y,y,pc,home,240.0,ArraysTest7,test,F,HIGH
n,n,y,y,pc,home,240.0,ArraysRptaLibre1,short_text,F,LOW
n,n,y,y,pc,home,240.0,ArraysRptaLibre2,short_text,F,LOW
n,n,y,y,pc,home,240.0,ArraysRptaLibre3,short_text,F,LOW
n,n,y,y,pc,home,240.0,MatricesRptaLibre,short_text,F,LOW
n,n,y,y,pc,home,240.0,PunterosTest1,test,F,HIGH
n,n,y,y,pc,home,240.0,PunterosTest2,test,F,HIGH
n,n,y,y,pc,home,240.0,PunterosTest3,test,F,LOW
n,n,y,y,pc,home,240.0,PunterosTest4,test,F,HIGH
y,y,n,y,pda,home,60.0,EnumTest1,test,F,LOW
y,y,n,y,pda,home,60.0,OperadoresTest3,test,F,LOW
y,y,n,y,pda,home,60.0,OperadoresRptaLibre1,short_text,F,LOW
y,y,n,y,pda,home,60.0,OperadoresRptaLibre2,short_text,F,LOW
y,y,n,y,pda,home,60.0,OperadoresRptaLibre3,short_text,F,LOW
y,y,n,y,pda,home,60.0,OperadoresRptaLibre4,short_text,F,LOW
y,y,n,y,pda,home,60.0,OperadoresRptaLibre5,short_text,F,LOW
y,y,n,y,pda,home,30.0,SwitchRptaLibre1,short_text,F,LOW
y,y,n,y,pda,home,30.0,SwitchRptaLibre2,short_text,F,LOW
y,y,n,y,pda,home,30.0,IfTest,test,F,LOW
y,y,n,y,pda,home,30.0,CondReview,review,F,HIGH
y,y,n,y,pda,home,30.0,IfTestRC,review,F,HIGH
y,y,n,y,pda,home,30.0,SwitchRptaLibre1RC,review,F,HIGH
y,y,n,y,pda,home,10.0,SwitchEjemRC,review,F,HIGH
y,y,n,y,pda,home,10.0,SwitchRptaLibre2RC,review,F,HIGH
y,y,n,y,pc,home,60.0,DoWhileRptaLibre,short_text,F,LOW
y,y,n,y,pc,home,60.0,ForTest,test,F,LOW

```

A.6. Reglas de decisión

Esta sección muestra las salidas de la ejecución del programa *c4.5rules* con los ficheros de logs de la sección anterior.

A.6.1. Reglas de decisión para el curso “Introducción a la programación en el lenguaje C” en los sistemas *QuizGuide* y *QuizPACK*

C4.5 [release 8] rule generator Thu Mar 5 19:53:17 2009

```
-----  
  
Options:  
File stem <um2QP2007-gid-activity>  
Pruning confidence level 100%  
  
Read 52734 cases (2 attributes) from um2QP2007-gid-activity  
  
-----  
Processing tree 0  
  
Final rules from tree 0:  
  
Rule 21:  
    groupid = gid_441  
    activity in {character_array2, function1}  
-> class yes [100.0%]  
  
Rule 20:  
    groupid = gid_441  
    activity = if_else1  
-> class yes [80.8%]  
  
Rule 13:  
    groupid = gid_373  
    activity = function1  
-> class yes [77.5%]  
  
Rule 9:  
    groupid = gid_373  
    activity in {character_processing1, for2, function3, if_else1}  
-> class yes [74.4%]  
  
Rule 25:  
    groupid in {gid_72, gid_442, gid_67}  
-> class yes [67.5%]  
  
Rule 16:  
    groupid = gid_441
```

```
    activity in {2dimensional_array1, character_processing3, while2}
-> class yes [64.5%]
```

Rule 40:

```
    activity in {complex_conditional1, function2, printing1, switch1,
variable3, complex_conditional2, compound_assignment1, constant1, constant2,
increment_decrement1, variable1, while1, globvar_simplefunc1}
-> class yes [61.1%]
```

Rule 17:

```
    groupid = gid_441
    activity in {arithmetic_expression1, character_processing1, if_else2}
-> class yes [53.6%]
```

Rule 3:

```
    groupid = gid_373
    activity in {arithmetic_expression1, logical_expression1, pointer1}
-> class yes [52.7%]
```

Rule 48:

```
    groupid = gid_72
    activity in {constant1, increment_decrement1}
-> class no [100.0%]
```

Rule 19:

```
    groupid = gid_441
    activity = pointer2
-> class no [82.4%]
```

Rule 23:

```
    groupid in {gid_190, gid_441, gid_373, gid_394}
    activity in {2dimensional_array2, do2, array1, array3}
-> class no [81.8%]
```

Rule 8:

```
    groupid = gid_373
    activity = character_processing3
-> class no [79.1%]
```

Rule 33:

```
    activity = variable2
-> class no [70.2%]
```

Rule 7:

```

    groupid = gid_373
    activity in {character_array3, nested_loop1, pointer3}
-> class no [67.9%]

```

Rule 6:

```

    groupid in {gid_190, gid_373}
    activity in {character_array2, printing2, while2}
-> class no [67.6%]

```

Rule 12:

```

    groupid = gid_190
    activity in {2dimensional_array1, 2dimensional_array2, 2dimensional_array3,
arithmetic_expression1, arithmetic_expression2, character_array1, character_array2,
character_array3, character_processing1, character_processing2, character_processing3,
conditional_operator1, do2, function1, logical_expression1, nested_loop1,
pointer2, pointer3, printing2, variable2, array1, array2, array3, do1, for1, for2,
function3, if_else1, if_else2, logical_operator1, pointer1, while2,
globvar_simplefunc2}
-> class no [65.8%]

```

Rule 35:

```

    groupid in {gid_72, gid_442, gid_67}
    activity in {pointer3, for2, pointer1}
-> class no [65.0%]

```

Rule 14:

```

    groupid = gid_373
    activity in {do1, logical_operator1}
-> class no [60.3%]

```

Default class: no

Evaluation on training data (52734 items):

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
21	2	0.0%	74	0 (0.0%)	74 (74 0) yes
20	2	19.2%	159	30 (18.9%)	99 (129 30) yes
13	2	22.5%	60	13 (21.7%)	34 (47 13) yes
9	2	25.6%	236	60 (25.4%)	116 (176 60) yes
25	1	32.5%	3657	1189 (32.5%)	391 (1297 906) yes
16	2	35.5%	280	99 (35.4%)	82 (181 99) yes
40	1	38.9%	16061	6526 (40.6%)	3009 (9535 6526) yes

17	2	46.4%	738	342 (46.3%)	54 (396 342)	yes
3	2	47.3%	278	131 (47.1%)	16 (147 131)	yes
19	2	17.6%	71	12 (16.9%)	0 (0 0)	no
23	2	18.2%	2608	475 (18.2%)	0 (0 0)	no
8	2	20.9%	242	50 (20.7%)	0 (0 0)	no
33	1	29.8%	1771	494 (27.9%)	0 (0 0)	no
7	2	32.1%	344	110 (32.0%)	0 (0 0)	no
6	2	32.4%	2508	813 (32.4%)	0 (0 0)	no
12	2	34.2%	20419	7522 (36.8%)	0 (0 0)	no
14	2	39.7%	253	100 (39.5%)	0 (0 0)	no

Tested 52734, errors 19247 (36.5%) <<

```
(a) (b) <-classified as
---- ----
20334 8390 (a): class no
1085713153 (b): class yes
```

A.6.2. Reglas de decisión para el curso “Sistemas Operativos I” en el sistema *CoMoLE*

C4.5 [release 8] rule generator Mon May 25 16:48:41 2009

```
Options:
File stem <comole-0708-S01-logs-e1>

Read 3610 cases (8 attributes) from comole-0708-S01-logs-e1
```

```
-----
Processing tree 0
```

Final rules from tree 0:

```
Rule 112:
  tipo = collaborative
-> class HIGH [98.0%]
```

```
Rule 22:
  sequential = y
  sensitive = n
  nameAct = Mem_Test4
-> class HIGH [66.2%]
```

Rule 32:

```
    active = y
    sensitive = n
    nameAct = MV_Test2
-> class HIGH [63.0%]
```

Rule 56:

```
    device = pda
    nameAct = MV_Test9
-> class HIGH [63.0%]
```

Rule 88:

```
    sequential = n
    active = n
    sensitive = n
    location = home
    nameAct = MV_Test12
-> class HIGH [63.0%]
```

Rule 31:

```
    active = n
    device = pc
    nameAct = MV_Test2
-> class HIGH [57.8%]
```

Rule 46:

```
    sequential = n
    sensitive = n
    nameAct = MV_Test6
-> class HIGH [54.6%]
```

Rule 47:

```
    visual = n
    sensitive = n
    nameAct = MV_Test6
-> class HIGH [50.0%]
```

Rule 101:

```
    visual = n
    sensitive = n
    nameAct = MV_Test13
-> class HIGH [50.0%]
```

Rule 55:

```
    device = pc
    location = others
    nameAct = MV_Test8
-> class HIGH [48.7%]
```

Rule 72:

```
    sequential = n
    active = n
    nameAct = MV_Test16
-> class HIGH [47.5%]
```

Rule 74:

```
    nameAct = MV_Test17
-> class HIGH [46.1%]
```

Rule 16:

```
    visual = y
    sequential = y
    sensitive = n
    device = pc
    nameAct = Mem_Test1
-> class HIGH [45.3%]
```

Rule 5:

```
    visual = y
    sequential = y
    location = home
    nameAct = PagSimpleTest2
-> class HIGH [45.1%]
```

Rule 10:

```
    active = n
    nameAct = SegSimpleTest1
-> class HIGH [44.5%]
```

Rule 12:

```
    sensitive = n
    location = home
    nameAct = SegSimpleTest2
-> class HIGH [44.3%]
```

Rule 51:

```
    visual = n
```



```
    nameAct = MV_Test7  
-> class HIGH [44.1%]
```

Rule 79:

```
    device = pc  
    nameAct = MV_Test18  
-> class HIGH [41.5%]
```

Rule 36:

```
    active = n  
    device = pc  
    nameAct = MV_Test4  
-> class HIGH [40.5%]
```

Rule 52:

```
    visual = y  
    sequential = n  
    active = n  
    sensitive = y  
    location = home  
    nameAct = MV_Test8  
-> class HIGH [38.8%]
```

Rule 70:

```
    active = n  
    sensitive = n  
    device = pc  
    nameAct = MV_Test16  
-> class HIGH [38.8%]
```

Rule 68:

```
    visual = y  
    nameAct = MV_Test15  
-> class HIGH [38.7%]
```

Rule 48:

```
    active = n  
    sensitive = y  
    location = others  
    nameAct = MV_Test6  
-> class HIGH [37.3%]
```

Rule 81:

```
    sensitive = n
```

```
    nameAct = TablasTest2  
-> class HIGH [37.3%]
```

```
Rule 110:  
    sequential = y  
    location = others  
    nameAct = TablasRptaLibre1  
-> class HIGH [35.2%]
```

```
Rule 60:  
    nameAct = MV_Test11  
-> class HIGH [31.6%]
```

```
Rule 50:  
    sequential = n  
    nameAct = MV_Test7  
-> class HIGH [31.6%]
```

```
Rule 8:  
    location = lab  
    nameAct = PagSimpleTest3  
-> class HIGH [31.4%]
```

```
Rule 14:  
    location = lab  
    nameAct = SegSimpleTest2  
-> class HIGH [31.4%]
```

```
Rule 21:  
    device = pda  
    nameAct = Mem_Test3  
-> class HIGH [31.4%]
```

```
Rule 104:  
    tipo = short_text  
-> class LOW [91.9%]
```

```
Rule 19:  
    nameAct = Mem_Test2  
-> class LOW [91.5%]
```

```
Rule 30:  
    nameAct = MV_Test1  
-> class LOW [91.5%]
```

Rule 26:

```
    nameAct = Mem_Test5  
-> class LOW [91.0%]
```

Rule 15:

```
    location = others  
    nameAct = SegSimpleTest2  
-> class LOW [88.2%]
```

Rule 57:

```
    nameAct = MV_Test10  
-> class LOW [84.2%]
```

Rule 62:

```
    nameAct = MV_Test14  
-> class LOW [82.9%]
```

Rule 1:

```
    location = home  
    nameAct = PagSimpleTest1  
-> class LOW [82.8%]
```

Rule 54:

```
    active = y  
    location = home  
-> class LOW [82.2%]
```

Rule 37:

```
    sequential = y  
    active = y  
-> class LOW [81.6%]
```

Rule 7:

```
    visual = n  
-> class LOW [80.3%]
```

Rule 20:

```
    device = pc  
    nameAct = Mem_Test3  
-> class LOW [79.6%]
```

Rule 45:

```
    visual = y
```

```

sequential = y
nameAct = MV_Test6
-> class LOW [77.7%]

```

Default class: LOW

Evaluation on training data (3610 items):

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
112	1	2.0%	69	0 (0.0%)	69 (69 0) HIGH
22	3	33.8%	7	1 (14.3%)	5 (6 1) HIGH
32	3	37.0%	3	0 (0.0%)	3 (3 0) HIGH
56	2	37.0%	3	0 (0.0%)	3 (3 0) HIGH
88	5	37.0%	3	0 (0.0%)	3 (3 0) HIGH
31	3	42.2%	52	19 (36.5%)	14 (33 19) HIGH
46	3	45.4%	5	1 (20.0%)	3 (4 1) HIGH
47	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
101	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
55	3	51.3%	11	4 (36.4%)	3 (7 4) HIGH
72	3	52.5%	17	7 (41.2%)	3 (8 5) HIGH
74	1	53.9%	79	39 (49.4%)	1 (40 39) HIGH
16	5	54.7%	4	1 (25.0%)	2 (3 1) HIGH
5	4	54.9%	49	24 (49.0%)	1 (25 24) HIGH
10	2	55.5%	56	28 (50.0%)	0 (28 28) HIGH
12	3	55.7%	6	2 (33.3%)	2 (4 2) HIGH
51	2	55.9%	10	4 (40.0%)	2 (5 3) HIGH
79	2	58.5%	76	41 (53.9%)	-6 (35 41) HIGH
36	3	59.5%	52	28 (53.8%)	-4 (24 28) HIGH
52	6	61.2%	9	4 (44.4%)	1 (5 4) HIGH
70	4	61.2%	5	2 (40.0%)	1 (3 2) HIGH
68	2	61.3%	69	39 (56.5%)	-9 (30 39) HIGH
48	4	62.7%	7	3 (42.9%)	1 (4 3) HIGH
81	2	62.7%	7	3 (42.9%)	1 (4 3) HIGH
110	3	64.8%	5	2 (40.0%)	1 (3 2) HIGH
60	1	68.4%	81	52 (64.2%)	-23 (29 52) HIGH
50	2	68.4%	23	14 (60.9%)	-5 (9 14) HIGH
8	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH
14	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH
21	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH
104	1	8.1%	1225	85 (6.9%)	0 (0 0) LOW
19	1	8.5%	99	6 (6.1%)	0 (0 0) LOW
30	1	8.5%	99	6 (6.1%)	0 (0 0) LOW

26	1	9.0%	93	6 (6.5%)	0 (0 0)	LOW
15	2	11.8%	11	0 (0.0%)	0 (0 0)	LOW
57	1	15.8%	81	10 (12.3%)	0 (0 0)	LOW
62	1	17.1%	81	11 (13.6%)	0 (0 0)	LOW
1	2	17.2%	74	10 (13.5%)	0 (0 0)	LOW
54	2	17.8%	418	93 (22.2%)	0 (0 0)	LOW
37	2	18.4%	56	9 (16.1%)	0 (0 0)	LOW
7	1	19.7%	95	17 (17.9%)	0 (0 0)	LOW
20	2	20.4%	54	12 (22.2%)	0 (0 0)	LOW
45	3	22.3%	27	4 (14.8%)	0 (0 0)	LOW

Drop rule 79

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
112	1	2.0%	69	0 (0.0%)	69 (69 0) HIGH
22	3	33.8%	7	1 (14.3%)	5 (6 1) HIGH
32	3	37.0%	3	0 (0.0%)	3 (3 0) HIGH
56	2	37.0%	3	0 (0.0%)	3 (3 0) HIGH
88	5	37.0%	3	0 (0.0%)	3 (3 0) HIGH
31	3	42.2%	52	19 (36.5%)	14 (33 19) HIGH
46	3	45.4%	5	1 (20.0%)	3 (4 1) HIGH
47	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
101	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
55	3	51.3%	11	4 (36.4%)	3 (7 4) HIGH
72	3	52.5%	17	7 (41.2%)	3 (8 5) HIGH
74	1	53.9%	79	39 (49.4%)	1 (40 39) HIGH
16	5	54.7%	4	1 (25.0%)	2 (3 1) HIGH
5	4	54.9%	49	24 (49.0%)	1 (25 24) HIGH
10	2	55.5%	56	28 (50.0%)	0 (28 28) HIGH
12	3	55.7%	6	2 (33.3%)	2 (4 2) HIGH
51	2	55.9%	10	4 (40.0%)	2 (5 3) HIGH
36	3	59.5%	52	28 (53.8%)	-4 (24 28) HIGH
52	6	61.2%	9	4 (44.4%)	1 (5 4) HIGH
70	4	61.2%	5	2 (40.0%)	1 (3 2) HIGH
68	2	61.3%	69	39 (56.5%)	-9 (30 39) HIGH
48	4	62.7%	7	3 (42.9%)	1 (4 3) HIGH
81	2	62.7%	7	3 (42.9%)	1 (4 3) HIGH
110	3	64.8%	5	2 (40.0%)	1 (3 2) HIGH
60	1	68.4%	81	52 (64.2%)	-23 (29 52) HIGH
50	2	68.4%	23	14 (60.9%)	-5 (9 14) HIGH
8	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH
14	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH
21	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH

104	1	8.1%	1225	85 (6.9%)	0 (0 0)	LOW
19	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
30	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
26	1	9.0%	93	6 (6.5%)	0 (0 0)	LOW
15	2	11.8%	11	0 (0.0%)	0 (0 0)	LOW
57	1	15.8%	81	10 (12.3%)	0 (0 0)	LOW
62	1	17.1%	81	11 (13.6%)	0 (0 0)	LOW
1	2	17.2%	74	10 (13.5%)	0 (0 0)	LOW
54	2	17.8%	440	100 (22.7%)	0 (0 0)	LOW
37	2	18.4%	58	11 (19.0%)	0 (0 0)	LOW
7	1	19.7%	102	19 (18.6%)	0 (0 0)	LOW
20	2	20.4%	54	12 (22.2%)	0 (0 0)	LOW
45	3	22.3%	27	4 (14.8%)	0 (0 0)	LOW

Drop rule 36

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
112	1	2.0%	69	0 (0.0%)	69 (69 0) HIGH
22	3	33.8%	7	1 (14.3%)	5 (6 1) HIGH
32	3	37.0%	3	0 (0.0%)	3 (3 0) HIGH
56	2	37.0%	3	0 (0.0%)	3 (3 0) HIGH
88	5	37.0%	3	0 (0.0%)	3 (3 0) HIGH
31	3	42.2%	52	19 (36.5%)	14 (33 19) HIGH
46	3	45.4%	5	1 (20.0%)	3 (4 1) HIGH
47	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
101	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
55	3	51.3%	11	4 (36.4%)	3 (7 4) HIGH
72	3	52.5%	17	7 (41.2%)	3 (8 5) HIGH
74	1	53.9%	79	39 (49.4%)	1 (40 39) HIGH
16	5	54.7%	4	1 (25.0%)	2 (3 1) HIGH
5	4	54.9%	49	24 (49.0%)	1 (25 24) HIGH
10	2	55.5%	56	28 (50.0%)	0 (28 28) HIGH
12	3	55.7%	6	2 (33.3%)	2 (4 2) HIGH
51	2	55.9%	10	4 (40.0%)	2 (5 3) HIGH
52	6	61.2%	9	4 (44.4%)	1 (5 4) HIGH
70	4	61.2%	5	2 (40.0%)	1 (3 2) HIGH
68	2	61.3%	69	39 (56.5%)	-9 (30 39) HIGH
48	4	62.7%	7	3 (42.9%)	1 (4 3) HIGH
81	2	62.7%	7	3 (42.9%)	1 (4 3) HIGH
110	3	64.8%	5	2 (40.0%)	1 (3 2) HIGH
60	1	68.4%	81	52 (64.2%)	-23 (29 52) HIGH
50	2	68.4%	23	14 (60.9%)	-5 (9 14) HIGH
8	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH

14	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
21	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
104	1	8.1%	1225	85 (6.9%)	0 (0 0)	LOW
19	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
30	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
26	1	9.0%	93	6 (6.5%)	0 (0 0)	LOW
15	2	11.8%	11	0 (0.0%)	0 (0 0)	LOW
57	1	15.8%	81	10 (12.3%)	0 (0 0)	LOW
62	1	17.1%	81	11 (13.6%)	0 (0 0)	LOW
1	2	17.2%	74	10 (13.5%)	0 (0 0)	LOW
54	2	17.8%	440	100 (22.7%)	0 (0 0)	LOW
37	2	18.4%	58	11 (19.0%)	0 (0 0)	LOW
7	1	19.7%	109	23 (21.1%)	0 (0 0)	LOW
20	2	20.4%	54	12 (22.2%)	0 (0 0)	LOW
45	3	22.3%	27	4 (14.8%)	0 (0 0)	LOW

Drop rule 68

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
112	1	2.0%	69	0 (0.0%)	69 (69 0) HIGH
22	3	33.8%	7	1 (14.3%)	5 (6 1) HIGH
32	3	37.0%	3	0 (0.0%)	3 (3 0) HIGH
56	2	37.0%	3	0 (0.0%)	3 (3 0) HIGH
88	5	37.0%	3	0 (0.0%)	3 (3 0) HIGH
31	3	42.2%	52	19 (36.5%)	14 (33 19) HIGH
46	3	45.4%	5	1 (20.0%)	3 (4 1) HIGH
47	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
101	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
55	3	51.3%	11	4 (36.4%)	3 (7 4) HIGH
72	3	52.5%	17	7 (41.2%)	3 (8 5) HIGH
74	1	53.9%	79	39 (49.4%)	1 (40 39) HIGH
16	5	54.7%	4	1 (25.0%)	2 (3 1) HIGH
5	4	54.9%	49	24 (49.0%)	1 (25 24) HIGH
10	2	55.5%	56	28 (50.0%)	0 (28 28) HIGH
12	3	55.7%	6	2 (33.3%)	2 (4 2) HIGH
51	2	55.9%	10	4 (40.0%)	2 (5 3) HIGH
52	6	61.2%	9	4 (44.4%)	1 (5 4) HIGH
70	4	61.2%	5	2 (40.0%)	1 (3 2) HIGH
48	4	62.7%	7	3 (42.9%)	1 (4 3) HIGH
81	2	62.7%	7	3 (42.9%)	1 (4 3) HIGH
110	3	64.8%	5	2 (40.0%)	1 (3 2) HIGH
60	1	68.4%	81	52 (64.2%)	-23 (29 52) HIGH
50	2	68.4%	23	14 (60.9%)	-5 (9 14) HIGH

8	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
14	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
21	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
104	1	8.1%	1225	85 (6.9%)	0 (0 0)	LOW
19	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
30	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
26	1	9.0%	93	6 (6.5%)	0 (0 0)	LOW
15	2	11.8%	11	0 (0.0%)	0 (0 0)	LOW
57	1	15.8%	81	10 (12.3%)	0 (0 0)	LOW
62	1	17.1%	81	11 (13.6%)	0 (0 0)	LOW
1	2	17.2%	74	10 (13.5%)	0 (0 0)	LOW
54	2	17.8%	459	106 (23.1%)	0 (0 0)	LOW
37	2	18.4%	60	11 (18.3%)	0 (0 0)	LOW
7	1	19.7%	109	23 (21.1%)	0 (0 0)	LOW
20	2	20.4%	54	12 (22.2%)	0 (0 0)	LOW
45	3	22.3%	27	4 (14.8%)	0 (0 0)	LOW

Drop rule 60

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
112	1	2.0%	69	0 (0.0%)	69 (69 0) HIGH
22	3	33.8%	7	1 (14.3%)	5 (6 1) HIGH
32	3	37.0%	3	0 (0.0%)	3 (3 0) HIGH
56	2	37.0%	3	0 (0.0%)	3 (3 0) HIGH
88	5	37.0%	3	0 (0.0%)	3 (3 0) HIGH
31	3	42.2%	52	19 (36.5%)	14 (33 19) HIGH
46	3	45.4%	5	1 (20.0%)	3 (4 1) HIGH
47	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
101	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
55	3	51.3%	11	4 (36.4%)	3 (7 4) HIGH
72	3	52.5%	17	7 (41.2%)	3 (8 5) HIGH
74	1	53.9%	79	39 (49.4%)	1 (40 39) HIGH
16	5	54.7%	4	1 (25.0%)	2 (3 1) HIGH
5	4	54.9%	49	24 (49.0%)	1 (25 24) HIGH
10	2	55.5%	56	28 (50.0%)	0 (28 28) HIGH
12	3	55.7%	6	2 (33.3%)	2 (4 2) HIGH
51	2	55.9%	10	4 (40.0%)	2 (5 3) HIGH
52	6	61.2%	9	4 (44.4%)	1 (5 4) HIGH
70	4	61.2%	5	2 (40.0%)	1 (3 2) HIGH
48	4	62.7%	7	3 (42.9%)	1 (4 3) HIGH
81	2	62.7%	7	3 (42.9%)	1 (4 3) HIGH
110	3	64.8%	5	2 (40.0%)	1 (3 2) HIGH
50	2	68.4%	23	14 (60.9%)	-5 (9 14) HIGH

8	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
14	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
21	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
104	1	8.1%	1225	85 (6.9%)	0 (0 0)	LOW
19	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
30	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
26	1	9.0%	93	6 (6.5%)	0 (0 0)	LOW
15	2	11.8%	11	0 (0.0%)	0 (0 0)	LOW
57	1	15.8%	81	10 (12.3%)	0 (0 0)	LOW
62	1	17.1%	81	11 (13.6%)	0 (0 0)	LOW
1	2	17.2%	74	10 (13.5%)	0 (0 0)	LOW
54	2	17.8%	481	112 (23.3%)	0 (0 0)	LOW
37	2	18.4%	62	11 (17.7%)	0 (0 0)	LOW
7	1	19.7%	116	24 (20.7%)	0 (0 0)	LOW
20	2	20.4%	54	12 (22.2%)	0 (0 0)	LOW
45	3	22.3%	27	4 (14.8%)	0 (0 0)	LOW

Drop rule 50

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
112	1	2.0%	69	0 (0.0%)	69 (69 0) HIGH
22	3	33.8%	7	1 (14.3%)	5 (6 1) HIGH
32	3	37.0%	3	0 (0.0%)	3 (3 0) HIGH
56	2	37.0%	3	0 (0.0%)	3 (3 0) HIGH
88	5	37.0%	3	0 (0.0%)	3 (3 0) HIGH
31	3	42.2%	52	19 (36.5%)	14 (33 19) HIGH
46	3	45.4%	5	1 (20.0%)	3 (4 1) HIGH
47	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
101	3	50.0%	2	0 (0.0%)	2 (2 0) HIGH
55	3	51.3%	11	4 (36.4%)	3 (7 4) HIGH
72	3	52.5%	17	7 (41.2%)	3 (8 5) HIGH
74	1	53.9%	79	39 (49.4%)	1 (40 39) HIGH
16	5	54.7%	4	1 (25.0%)	2 (3 1) HIGH
5	4	54.9%	49	24 (49.0%)	1 (25 24) HIGH
10	2	55.5%	56	28 (50.0%)	0 (28 28) HIGH
12	3	55.7%	6	2 (33.3%)	2 (4 2) HIGH
51	2	55.9%	10	4 (40.0%)	2 (6 4) HIGH
52	6	61.2%	9	4 (44.4%)	1 (5 4) HIGH
70	4	61.2%	5	2 (40.0%)	1 (3 2) HIGH
48	4	62.7%	7	3 (42.9%)	1 (4 3) HIGH
81	2	62.7%	7	3 (42.9%)	1 (4 3) HIGH
110	3	64.8%	5	2 (40.0%)	1 (3 2) HIGH
8	2	68.6%	3	1 (33.3%)	1 (2 1) HIGH

14	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
21	2	68.6%	3	1 (33.3%)	1 (2 1)	HIGH
104	1	8.1%	1225	85 (6.9%)	0 (0 0)	LOW
19	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
30	1	8.5%	99	6 (6.1%)	0 (0 0)	LOW
26	1	9.0%	93	6 (6.5%)	0 (0 0)	LOW
15	2	11.8%	11	0 (0.0%)	0 (0 0)	LOW
57	1	15.8%	81	10 (12.3%)	0 (0 0)	LOW
62	1	17.1%	81	11 (13.6%)	0 (0 0)	LOW
1	2	17.2%	74	10 (13.5%)	0 (0 0)	LOW
54	2	17.8%	486	113 (23.3%)	0 (0 0)	LOW
37	2	18.4%	62	11 (17.7%)	0 (0 0)	LOW
7	1	19.7%	116	24 (20.7%)	0 (0 0)	LOW
20	2	20.4%	54	12 (22.2%)	0 (0 0)	LOW
45	3	22.3%	27	4 (14.8%)	0 (0 0)	LOW

Tested 3610, errors 685 (19.0%) <<

(a) (b) <-classified as

---- ----

2652 147 (a): class LOW

538 273 (b): class HIGH

A.6.3. Reglas de decisión para el curso “Estructura de Datos y de la Información I” en el sistema *CoMoLE*

C4.5 [release 8] rule generator Tue Apr 28 16:11:26 2009

Options:

File stem <comole-0708-EDI1-logs-e1>

Read 2309 cases (10 attributes) from comole-0708-EDI1-logs-e1

Processing tree 0

Final rules from tree 0:

Rule 232:

tipo = review

-> class HIGH [99.5%]

Rule 110:

```
location = home
timeUser <= 60
nameAct = ArraysTest7
-> class HIGH [89.9%]
```

Rule 113:

```
active = y
nameAct = ArraysTest7
-> class HIGH [89.9%]
```

Rule 19:

```
sequential = y
active = n
sensitive = y
timeUser > 60
nameAct = OperadoresTest1
-> class HIGH [82.0%]
```

Rule 26:

```
sequential = y
active = n
sensitive = y
timeUser > 60
nameAct = OperadoresTest2
-> class HIGH [82.0%]
```

Rule 64:

```
sequential = n
active = n
timeUser <= 60
nameAct = ESTest2
-> class HIGH [75.8%]
```

Rule 112:

```
visual = n
nameAct = ArraysTest7
-> class HIGH [75.8%]
```

Rule 172:

```
active = y
timeUser > 60
nameAct = OperadoresRptaLibre4
-> class HIGH [75.6%]
```

Rule 6:

```
location = home
timeUser > 20
timeUser <= 60
nameAct = EnumTest1
-> class HIGH [75.1%]
```

Rule 36:

```
sensitive = n
nameAct = OperadoresTest3
-> class HIGH [73.1%]
```

Rule 154:

```
location = lab
tipo = test
-> class HIGH [72.4%]
```

Rule 83:

```
sensitive = y
location = home
timeUser <= 60
nameAct = FuncTest3
-> class HIGH [72.2%]
```

Rule 11:

```
sequential = n
active = y
location = home
nameAct = EnumTest2
timeUser <= 60
-> class HIGH [70.7%]
```

Rule 119:

```
sequential = y
timeUser > 30
nameAct = PunterosTest2
-> class HIGH [70.7%]
```

Rule 75:

```
nameAct = NuevosDatosTest
-> class HIGH [70.4%]
```

Rule 200:

```
    nameAct = EstructurasRptaLibre  
-> class HIGH [70.4%]
```

Rule 86:

```
    active = n  
    sensitive = y  
    timeUser <= 60  
    nameAct = ArraysTest1  
-> class HIGH [70.0%]
```

Rule 23:

```
    visual = n  
    nameAct = OperadoresTest1  
-> class HIGH [70.0%]
```

Rule 45:

```
    visual = y  
    active = y  
    location = home  
    nameAct = WhileTest  
-> class HIGH [70.0%]
```

Rule 147:

```
    nameAct = FuncTest2  
-> class HIGH [68.6%]
```

Rule 148:

```
    nameAct = ArraysTest2  
-> class HIGH [68.1%]
```

Rule 213:

```
    nameAct = PasoArgumentosRptaLibre2  
-> class HIGH [67.1%]
```

Rule 31:

```
    active = y  
    location = home  
    nameAct = OperadoresTest2  
-> class HIGH [66.7%]
```

Rule 27:

```
    visual = n  
    sequential = y  
    active = n
```

```
    location = home
    tipo = test
-> class HIGH [66.2%]
```

```
Rule 197:
    visual = y
    timeUser > 120
    nameAct = DoWhileRptaLibre
-> class HIGH [66.2%]
```

```
Rule 133:
    sequential = y
    nameAct = EnumTest1
-> class HIGH [64.9%]
```

```
Rule 50:
    sequential = y
    timeUser > 60
    nameAct = ForTest
-> class HIGH [64.5%]
```

```
Rule 168:
    sequential = n
    active = n
    sensitive = y
    location = home
    timeUser <= 60
    nameAct = OperadoresRptaLibre4
-> class HIGH [64.5%]
```

```
Rule 107:
    location = home
    timeUser <= 60
    nameAct = ArraysTest6
-> class HIGH [63.8%]
```

```
Rule 209:
    active = y
    nameAct = PasoArgumentosRptaLibre1
-> class HIGH [63.8%]
```

```
Rule 204:
    visual = y
    timeUser <= 60
```

```
    nameAct = PasoArgumentosRptaLibre1  
-> class HIGH [63.5%]
```

```
Rule 124:  
    visual = n  
    timeUser <= 120  
    nameAct = PunterosTest3  
-> class HIGH [63.0%]
```

```
Rule 129:  
    sensitive = n  
    nameAct = PunterosTest4  
-> class HIGH [63.0%]
```

```
Rule 145:  
    location = others  
    nameAct = ESTest3  
-> class HIGH [63.0%]
```

```
Rule 220:  
    timeUser <= 60  
    nameAct = ArraysRptaLibre2  
-> class HIGH [61.5%]
```

```
Rule 4:  
    visual = n  
    timeUser <= 20  
    nameAct = EnumTest1  
-> class HIGH [61.2%]
```

```
Rule 194:  
    visual = n  
    active = n  
    nameAct = SwitchRptaLibre2  
-> class HIGH [61.2%]
```

```
Rule 192:  
    visual = y  
    sensitive = y  
    nameAct = SwitchRptaLibre2  
-> class HIGH [60.7%]
```

```
Rule 167:  
    visual = y
```

```
    sequential = y
    nameAct = OperadoresRptaLibre4
-> class HIGH [60.2%]
```

```
Rule 41:
    active = n
    timeUser <= 60
    nameAct = WhileTest
-> class HIGH [59.9%]
```

```
Rule 104:
    sensitive = y
    location = home
    timeUser <= 60
    nameAct = ArraysTest5
-> class HIGH [57.9%]
```

```
Rule 52:
    visual = y
    active = n
    sensitive = y
    location = home
    timeUser <= 60
    nameAct = ForTest
-> class HIGH [56.6%]
```

```
Rule 136:
    location = others
    nameAct = EnumTest2
-> class HIGH [56.6%]
```

```
Rule 225:
    sequential = y
    nameAct = MatricesRptaLibre
-> class HIGH [55.8%]
```

```
Rule 160:
    active = n
    sensitive = y
    timeUser <= 60
    nameAct = OperadoresRptaLibre2
-> class HIGH [54.8%]
```

```
Rule 90:
```



```
    visual = n
    nameAct = ArraysTest1
-> class HIGH [54.6%]
```

```
Rule 181:
    sequential = y
    sensitive = n
    nameAct = OperadoresRptaLibre5
-> class HIGH [54.6%]
```

```
Rule 184:
    visual = n
    active = n
    location = home
    nameAct = SwitchRptaLibre1
-> class HIGH [54.6%]
```

```
Rule 217:
    sensitive = y
    timeUser <= 60
    nameAct = ArraysRptaLibre1
-> class HIGH [54.4%]
```

```
Rule 231:
    timeUser <= 120
    nameAct = PunterosRptaLibre4
-> class HIGH [54.4%]
```

```
Rule 122:
    active = y
    nameAct = PunterosTest2
-> class HIGH [54.1%]
```

```
Rule 8:
    sequential = y
    sensitive = y
    nameAct = EnumTest2
-> class HIGH [53.5%]
```

```
Rule 166:
    nameAct = OperadoresRptaLibre3
-> class LOW [95.0%]
```

```
Rule 47:
```

```
    nameAct = DoWhileTest
-> class LOW [94.8%]
```

```
Rule 230:
    nameAct = PunterosRptaLibre3
-> class LOW [86.7%]
```

```
Rule 132:
    sequential = n
    location = home
    nameAct = PunterosTest5
-> class LOW [84.1%]
```

```
Rule 157:
    nameAct = OperadoresRptaLibre1
-> class LOW [83.6%]
```

```
Rule 78:
    sequential = n
    timeUser <= 60
    nameAct = FuncTest1
-> class LOW [82.0%]
```

```
Rule 130:
    location = home
    timeUser <= 30
    nameAct = PunterosTest5
-> class LOW [82.0%]
```

```
Rule 229:
    nameAct = PunterosRptaLibre2
-> class LOW [81.5%]
```

```
Rule 142:
    nameAct = IfTest
-> class LOW [79.8%]
```

```
Rule 196:
    location = home
    timeUser <= 120
    nameAct = DoWhileRptaLibre
-> class LOW [79.0%]
```

```
Rule 205:
```

```
visual = y
sequential = n
location = home
timeUser > 60
tipo = short_text
-> class LOW [77.8%]
```

Rule 85:

```
timeUser > 60
nameAct = FuncTest3
-> class LOW [77.7%]
```

Rule 218:

```
timeUser > 60
nameAct = ArraysRptaLibre1
-> class LOW [77.7%]
```

Rule 65:

```
sequential = n
active = n
timeUser > 60
tipo = test
-> class LOW [76.7%]
```

Rule 182:

```
sequential = n
sensitive = n
tipo = short_text
-> class LOW [76.1%]
```

Rule 25:

```
sequential = y
location = home
timeUser > 30
nameAct = OperadoresTest2
timeUser <= 60
-> class LOW [75.8%]
```

Rule 103:

```
visual = n
nameAct = ArraysTest4
-> class LOW [75.8%]
```

Rule 179:

```
    visual = y
    sensitive = y
    nameAct = OperadoresRptaLibre5
-> class LOW [71.2%]
```

```
Rule 224:
    nameAct = ArraysRptaLibre3
-> class LOW [70.8%]
```

```
Rule 60:
    sequential = y
    location = home
    timeUser <= 60
    nameAct = ESTest2
-> class LOW [70.0%]
```

```
Rule 114:
    active = n
    location = home
    nameAct = PunterosTest1
-> class LOW [70.0%]
```

```
Rule 149:
    nameAct = ArraysTest3
-> class LOW [68.1%]
```

```
Rule 170:
    visual = n
    timeUser > 20
    tipo = short_text
-> class LOW [67.6%]
```

```
Rule 228:
    nameAct = PunterosRptaLibre1
-> class LOW [66.7%]
```

```
Rule 139:
    location = others
    timeUser <= 30
    nameAct = OperadoresTest2
-> class LOW [63.0%]
```

```
Rule 219:
    sensitive = n
```

```
    nameAct = ArraysRptaLibre1  
-> class LOW [63.0%]
```

Rule 69:

```
    location = home  
    nameAct = ESTest3  
-> class LOW [62.3%]
```

Rule 22:

```
    visual = y  
    active = y  
    location = home  
    nameAct = OperadoresTest1  
-> class LOW [61.5%]
```

Rule 1:

```
    visual = y  
    sensitive = y  
    location = home  
    tipo = test  
    timeUser <= 20  
-> class LOW [60.7%]
```

Rule 152:

```
    nameAct = PunterosTest3  
-> class LOW [59.2%]
```

Rule 33:

```
    sequential = y  
    active = y  
    nameAct = OperadoresTest3  
-> class LOW [54.4%]
```

Rule 59:

```
    sensitive = y  
    location = home  
    timeUser > 30  
    nameAct = ESTest1  
-> class LOW [54.0%]
```

Default class: LOW

Evaluation on training data (2309 items):

Rule	Size	Error	Used	Wrong	Advantage
----	----	-----	----	-----	-----
232	1	0.5%	275	0 (0.0%)	275 (275 0) HIGH
110	3	10.1%	13	0 (0.0%)	6 (6 0) HIGH
113	2	10.1%	6	0 (0.0%)	4 (4 0) HIGH
19	5	18.0%	7	0 (0.0%)	6 (6 0) HIGH
26	5	18.0%	7	0 (0.0%)	6 (6 0) HIGH
64	4	24.2%	5	0 (0.0%)	5 (5 0) HIGH
112	2	24.2%	2	0 (0.0%)	2 (2 0) HIGH
172	3	24.4%	10	1 (10.0%)	5 (5 0) HIGH
6	4	24.9%	24	4 (16.7%)	5 (6 1) HIGH
36	2	26.9%	9	1 (11.1%)	6 (7 1) HIGH
154	2	27.6%	100	25 (25.0%)	36 (56 20) HIGH
83	4	27.8%	13	2 (15.4%)	9 (11 2) HIGH
11	5	29.3%	4	0 (0.0%)	4 (4 0) HIGH
119	3	29.3%	3	0 (0.0%)	3 (3 0) HIGH
75	1	29.6%	31	8 (25.8%)	15 (23 8) HIGH
200	1	29.6%	35	8 (22.9%)	19 (27 8) HIGH
86	4	30.0%	6	1 (16.7%)	4 (5 1) HIGH
23	2	30.0%	11	2 (18.2%)	7 (8 1) HIGH
45	4	30.0%	12	2 (16.7%)	8 (10 2) HIGH
147	1	31.4%	29	7 (24.1%)	15 (22 7) HIGH
148	1	31.9%	25	5 (20.0%)	15 (20 5) HIGH
213	1	32.9%	28	7 (25.0%)	14 (21 7) HIGH
31	3	33.3%	21	5 (23.8%)	11 (16 5) HIGH
27	5	33.8%	11	2 (18.2%)	5 (6 1) HIGH
197	3	33.8%	7	1 (14.3%)	5 (6 1) HIGH
133	2	35.1%	33	12 (36.4%)	9 (21 12) HIGH
50	3	35.5%	10	2 (20.0%)	6 (8 2) HIGH
168	6	35.5%	10	2 (20.0%)	6 (8 2) HIGH
107	3	36.2%	13	3 (23.1%)	7 (10 3) HIGH
209	2	36.2%	13	3 (23.1%)	5 (6 1) HIGH
204	3	36.5%	10	2 (20.0%)	6 (8 2) HIGH
124	3	37.0%	3	0 (0.0%)	3 (3 0) HIGH
129	2	37.0%	2	0 (0.0%)	2 (2 0) HIGH
145	2	37.0%	3	0 (0.0%)	3 (3 0) HIGH
220	2	38.5%	18	5 (27.8%)	8 (13 5) HIGH
4	3	38.8%	5	1 (20.0%)	3 (4 1) HIGH
194	3	38.8%	6	1 (16.7%)	4 (5 1) HIGH
192	3	39.3%	48	16 (33.3%)	16 (32 16) HIGH
167	3	39.8%	37	13 (35.1%)	11 (24 13) HIGH
41	3	40.1%	17	6 (35.3%)	5 (11 6) HIGH
104	4	42.1%	11	3 (27.3%)	5 (8 3) HIGH

52	6	43.4%	8	2 (25.0%)	4 (6 2)	HIGH
136	2	43.4%	8	2 (25.0%)	4 (4 0)	HIGH
225	2	44.2%	13	4 (30.8%)	5 (9 4)	HIGH
160	4	45.2%	32	12 (37.5%)	8 (20 12)	HIGH
90	2	45.4%	5	1 (20.0%)	3 (4 1)	HIGH
181	3	45.4%	5	1 (20.0%)	3 (4 1)	HIGH
184	4	45.4%	5	1 (20.0%)	3 (4 1)	HIGH
217	3	45.6%	15	5 (33.3%)	5 (10 5)	HIGH
231	2	45.6%	15	5 (33.3%)	5 (10 5)	HIGH
122	2	45.9%	9	2 (22.2%)	5 (7 2)	HIGH
8	3	46.5%	39	16 (41.0%)	7 (23 16)	HIGH
166	1	5.0%	76	2 (2.6%)	0 (0 0)	LOW
47	1	5.2%	46	1 (2.2%)	0 (0 0)	LOW
230	1	13.3%	19	1 (5.3%)	0 (0 0)	LOW
132	3	15.9%	8	0 (0.0%)	0 (0 0)	LOW
157	1	16.4%	78	10 (12.8%)	0 (0 0)	LOW
78	3	18.0%	7	0 (0.0%)	0 (0 0)	LOW
130	3	18.0%	6	0 (0.0%)	0 (0 0)	LOW
229	1	18.5%	20	2 (10.0%)	0 (0 0)	LOW
142	1	20.2%	58	7 (12.1%)	0 (0 0)	LOW
196	3	21.0%	34	5 (14.7%)	0 (0 0)	LOW
205	5	22.2%	50	8 (16.0%)	0 (0 0)	LOW
85	2	22.3%	10	1 (10.0%)	0 (0 0)	LOW
218	2	22.3%	8	1 (12.5%)	0 (0 0)	LOW
65	4	23.3%	114	20 (17.5%)	0 (0 0)	LOW
182	3	23.9%	15	4 (26.7%)	0 (0 0)	LOW
25	5	24.2%	3	0 (0.0%)	0 (0 0)	LOW
103	2	24.2%	3	0 (0.0%)	0 (0 0)	LOW
179	3	28.8%	46	11 (23.9%)	0 (0 0)	LOW
224	1	29.2%	26	6 (23.1%)	0 (0 0)	LOW
60	4	30.0%	12	2 (16.7%)	0 (0 0)	LOW
114	3	30.0%	9	2 (22.2%)	0 (0 0)	LOW
149	1	31.9%	20	4 (20.0%)	0 (0 0)	LOW
170	3	32.4%	38	6 (15.8%)	0 (0 0)	LOW
228	1	33.3%	15	5 (33.3%)	0 (0 0)	LOW
139	3	37.0%	3	0 (0.0%)	0 (0 0)	LOW
219	2	37.0%	3	0 (0.0%)	0 (0 0)	LOW
69	2	37.7%	27	8 (29.6%)	0 (0 0)	LOW
22	4	38.5%	18	5 (27.8%)	0 (0 0)	LOW
1	5	39.3%	67	20 (29.9%)	0 (0 0)	LOW
152	1	40.8%	12	3 (25.0%)	0 (0 0)	LOW
33	3	45.6%	10	3 (30.0%)	0 (0 0)	LOW
59	4	46.0%	16	6 (37.5%)	0 (0 0)	LOW

Tested 2309, errors 505 (21.9%) <<

```
(a) (b) <-classified as
---- ----
886 304 (a): class HIGH
201 918 (b): class LOW
```

A.7. Método *GeSES* para distintos tamaños de datos

Esta sección muestra las reglas obtenidas a partir de la aplicación del método *GeSES*, y el *ranking* final de reglas para los datos simulados de la sección 6.5 en el capítulo 6. En concreto, para los casos en los que se obtengan más de 3 reglas en la fase 3 del método, se muestra además el *ranking* obtenido en la fase 5.

A.7.1. Resultados para $N = 20$ **A.7.1.1. Reglas obtenidas en la fase 3**

Rule 4:
dim3 = v31
->class no [37,9 %]

Tabla A.1: Reglas obtenidas en la fase 3 para $N = 20$ **A.7.2. Resultados para $N = 50$** **A.7.2.1. Reglas obtenidas en la fase 3**

Rule 1:
dim1 = v11
activity = act1
->class no [93,0 %]

Rule 3:
dim1 = v11
activity = act5
->class no [42,2 %]

Tabla A.2: Reglas obtenidas en la fase 3 para $N = 50$ **A.7.3. Resultados para $N = 80$** **A.7.3.1. Reglas obtenidas en la fase 3**

Rule 5:
dim1 = v12
dim3 = v33
activity = act6
->class no [70,7 %]

Tabla A.3: Reglas obtenidas en la fase 3 para $N = 80$

A.7.4. Resultados para $N = 100$

A.7.4.1. Reglas obtenidas en la fase 3

Rule 11:
dim2 = v21
dim3 = v31
activity = act1
->class no [82,2 %]
Rule 18:
dim2 = v21
dim3 = v31
activity = act5
->class no [58,7 %]
Rule 14:
dim1 = v12
dim2 = v21
activity = act3
->class no [48,1 %]

Tabla A.4: Reglas obtenidas en la fase 3 para $N = 100$

A.7.5. Resultados para $N = 150$

A.7.5.1. Reglas obtenidas en la fase 3

Rule 3:
dim1 = v11
activity = act1
->class no [95,4 %]
Rule 6:
dim1 = v12
dim3 = v31
activity = act3
->class no [79,4 %]

Tabla A.5: Reglas obtenidas en la fase 3 para $N = 150$

A.7.6. Resultados para $N = 300$

A.7.6.1. Reglas obtenidas en la fase 3

Rule 5:
dim1 = v11
dim3 = v31
activity = act1
->class no [98,8 %]
Rule 12:
dim1 = v12
dim2 = v21
dim3 = v31
activity = act3
->class no [66,2 %]
Rule 7:
dim1 = v13
dim3 = v32
activity = act2
->class no [50,0 %]

Tabla A.6: Reglas obtenidas en la fase 3 para $N = 300$ A.7.7. Resultados para $N = 600$

A.7.7.1. Reglas obtenidas en la fase 3

Rule 1:
dim1 = v11
dim3 = v31
activity = act1
->class no [99,2 %]
Rule 15:
dim1 = v11
dim3 = v31
activity = act5
->class no [49,6 %]
Rule 9:
dim1 = v12
dim2 = v21
activity = act3
->class no [49,0 %]

Tabla A.7: Reglas obtenidas en la fase 3 para $N = 600$

A.7.8. Resultados para $N = 900$

A.7.8.1. Reglas obtenidas en la fase 3

Rule 2:	dim1 = v11 activity = act1 ->class no [98,0 %]
Rule 19:	dim1 = v11 dim3 = v31 activity = act5 ->class no [49,2 %]
Rule 10:	dim1 = v12 dim2 = v21 activity = act3 ->class no [45,0 %]
Rule 21:	dim1 = v11 dim2 = v22 dim3 = v32 ->class no [44,1 %]
Rule 8:	dim1 = v13 dim2 = v21 activity = act3 ->class no [35,2 %]

Tabla A.8: Reglas obtenidas en la fase 3 para $N = 900$

A.7.8.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
19	0,80	1,00	0,80	2,60
10	0,63	1,00	0,44	2,06
2	0,81	0,00	1,00	1,81
21	0,35	1,00	0,35	1,70
8	0,35	1,00	0,02	1,36

Tabla A.9: *Ranking* de reglas en la fase 5 para $N = 900$

A.7.9. Resultados para $N = 1200$

A.7.9.1. Reglas obtenidas en la fase 3

Rule 5:				
	dim1 = v11			
	dim2 = v21			
	activity = act1			
	->class no [98,4 %]			
Rule 10:				
	dim1 = v12			
	dim3 = v31			
	activity = act3			
	->class no [88,5 %]			
Rule 20:				
	dim1 = v11			
	dim3 = v31			
	activity = act5			
	->class no [52,4 %]			
Rule 9:				
	dim1 = v12			
	dim3 = v33			
	activity = act3			
	->class no [48,1 %]			

Tabla A.10: Reglas obtenidas en la fase 3 para $N = 1200$ A.7.9.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
5	0,81	1,00	0,79	2,60
10	0,35	1,00	0,83	2,18
20	0,80	1,00	0,37	2,16
9	0,34	1,00	0,31	1,65

Tabla A.11: *Ranking* de reglas en la fase 5 para $N = 1200$

A.7.10. Resultados para $N = 1500$

A.7.10.1. Reglas obtenidas en la fase 3

Rule 2:
dim1 = v11
dim2 = v21
activity = act1
->class no [98,9 %]
Rule 12:
dim1 = v12
dim3 = v31
activity = act3
->class no [92,6 %]
Rule 18:
dim1 = v11
dim2 = v31
activity = act5
->class no [54,5 %]

Tabla A.12: Reglas obtenidas en la fase 3 para $N = 1500$

A.7.11. Resultados para $N = 1800$

A.7.11.1. Reglas obtenidas en la fase 3

Rule 4:
dim1 = v11
activity = act1
->class no [97,7 %]
Rule 10:
dim1 = v12
dim3 = v31
activity = act3
->class no [87,1 %]
Rule 14:
dim1 = v11
dim2 = v31
activity = act5
->class no [49,7 %]

Tabla A.13: Reglas obtenidas en la fase 3 para $N = 1800$

A.7.12. Resultados para $N = 2100$

A.7.12.1. Reglas obtenidas en la fase 3

Rule 3:	
dim1 = v11	
activity = act1	
->class no [96,7 %]	
Rule 12:	
dim1 = v12	
dim2 = v21	
dim3 = v31	
activity = act3	
->class no [86,7 %]	
Rule 19:	
dim1 = v11	
dim3 = v31	
activity = act5	
->class no [53,3 %]	
Rule 24:	
dim1 = v13	
dim2 = v21	
dim3 = v31	
activity = act7	
->class no [44,1 %]	

Tabla A.14: Reglas obtenidas en la fase 3 para $N = 2100$ A.7.12.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
12	0,36	0,80	0,78	1,93
3	0,82	0,14	0,86	1,81
19	0,79	0,45	0,38	1,62
24	0,34	0,80	0,27	1,41

Tabla A.15: *Ranking* de reglas en la fase 5 para $N = 2100$

A.7.13. Resultados para $N = 2400$

A.7.13.1. Reglas obtenidas en la fase 3

Rule 3:
dim1 = v11
dim2 = v21
activity = act1
->class no [96,1 %]
Rule 10:
dim1 = v12
dim3 = v31
activity = act3
->class no [81,8 %]
Rule 20:
dim1 = v11
dim2 = v22
dim3 = v32
activity = act5
->class no [61,0 %]
Rule 16:
dim1 = v11
dim3 = v31
activity = act5
->class no [49,4 %]

Tabla A.16: Reglas obtenidas en la fase 3 para $N = 2400$

A.7.13.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
3	0,83	0,35	0,90	2,08
20	0,32	1,00	0,40	1,72
10	0,36	0,35	0,75	1,47
16	0,79	0,35	0,22	1,36

Tabla A.17: *Ranking* de reglas en la fase 5 para $N = 2400$

A.7.14. Resultados para $N = 2700$

A.7.14.1. Reglas obtenidas en la fase 3

Rule 18:
dim1 = v11
dim2 = v21
activity = act1
->class no [99,1 %]
Rule 7:
dim1 = v12
dim3 = v31
activity = act3
->class no [84,6 %]
Rule 17:
dim1 = v12
dim2 = v22
dim3 = v31
activity = act8
->class no [63,0 %]
Rule 23:
dim1 = v11
activity = act5
->class no [49,3 %]
Rule 30:
dim2 = v21
dim3 = v32
activity = act1
->class no [81,1 %]

Tabla A.18: Reglas obtenidas en la fase 3 para $N = 2700$ A.7.14.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
18	0,80	1,00	0,90	2,70
7	0,37	1,00	0,80	2,17
17	0,35	1,00	0,54	1,89
23	0,80	0,00	0,35	1,15
30	0,35	1,00	0,20	1,55

Tabla A.19: *Ranking* de reglas en la fase 5 para $N = 2700$

A.7.15. Resultados para $N = 3100$

A.7.15.1. Reglas obtenidas en la fase 3

Rule 2:	dim1 = v11 activity = act1 ->class no [97,2 %]
Rule 11:	dim1 = v12 dim2 = v21 dim3 = v31 activity = act3 ->class no [91,5 %]
Rule 15:	dim1 = v11 dim3 = v31 activity = act5 ->class no [49,2 %]
Rule 20:	dim1 = v11 dim2 = v22 dim3 = v32 activity = act5 ->class no [45,3 %]
Rule 8:	dim1 = v13 dim2 = v22 dim3 = v31 activity = act2 ->class no [38,8 %]

Tabla A.20: Reglas obtenidas en la fase 3 para $N = 3100$ A.7.15.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
11	0,36	0,80	0,80	1,96
2	0,82	0,07	0,84	1,72
15	0,80	0,35	0,39	1,54
20	0,35	0,80	0,35	1,50
8	0,35	0,80	0,29	1,44

Tabla A.21: *Ranking* de reglas en la fase 5 para $N = 3100$

A.7.16. Resultados para $N = 3500$

A.7.16.1. Reglas obtenidas en la fase 3

Rule 4:				
	dim1 = v11			
	dim2 = v21			
	activity = act1			
	->class no [98,2 %]			
Rule 11:				
	dim1 = v12			
	dim2 = v21			
	dim3 = v31			
	activity = act3			
	->class no [87,0 %]			
Rule 18:				
	dim1 = v11			
	dim3 = v31			
	activity = act5			
	->class no [52,6 %]			
Rule 22:				
	dim1 = v11			
	dim2 = v22			
	dim3 = v32			
	activity = act7			
	->class no [45,3 %]			

Tabla A.22: Reglas obtenidas en la fase 3 para $N = 3500$ A.7.16.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
4	0,81	0,35	0,86	2,02
11	0,35	0,80	0,78	1,93
18	0,80	0,35	0,37	1,52
22	0,34	0,80	0,29	1,43

Tabla A.23: *Ranking* de reglas en la fase 5 para $N = 3500$

A.7.17. Resultados para $N = 4000$

A.7.17.1. Reglas obtenidas en la fase 3

Rule 23:
dim1 = v12
dim3 = v31
activity = act3
->class no [97,0 %]
Rule 13:
dim1 = v11
activity = act1
->class no [97,0 %]
Rule 29:
dim1 = v11
dim3 = v31
activity = act5
->class no [52,1 %]
Rule 9:
dim1 = v12
dim2 = v21
dim3 = v33
activity = act7
->class no [45,3 %]
Rule 26:
dim1 = v13
dim2 = v21
dim3 = v31
activity = act4
->class no [38,8 %]

Tabla A.24: Reglas obtenidas en la fase 3 para $N = 4000$ A.7.17.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
13	0,82	0,07	0,80	1,69
29	0,80	0,35	0,41	1,56
23	0,37	0,35	0,80	1,52
9	0,35	0,80	0,35	1,50
26	0,35	0,80	0,30	1,45

Tabla A.25: *Ranking* de reglas en la fase 5 para $N = 4000$

A.7.18. Resultados para $N = 5000$

A.7.18.1. Reglas obtenidas en la fase 3

Rule 2:	
dim1 = v11	
activity = act1	
->class no [97,6 %]	
Rule 10:	
dim1 = v12	
dim3 = v31	
activity = act3	
->class no [92,1 %]	
Rule 17:	
dim1 = v11	
dim3 = v31	
activity = act5	
->class no [52,8 %]	
Rule 15:	
dim1 = v11	
dim2 = v22	
dim3 = v32	
activity = act5	
->class no [44,1 %]	

Tabla A.26: Reglas obtenidas en la fase 3 para $N = 5000$ A.7.18.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
17	0,79	0,60	0,37	1,77
10	0,36	0,60	0,79	1,74
2	0,81	0,03	0,83	1,67
15	0,34	0,99	0,29	1,61

Tabla A.27: *Ranking* de reglas en la fase 5 para $N = 5000$

A.7.19. Resultados para $N = 6000$

A.7.19.1. Reglas obtenidas en la fase 3

Rule 5:				
	dim1 = v11			
	dim3 = v31			
	activity = act1			
	->class no [99,9 %]			
Rule 4:				
	dim1 = v11			
	dim2 = v21			
	activity = act1			
	->class no [95,5 %]			
Rule 10:				
	dim1 = v12			
	dim3 = v31			
	activity = act3			
	->class no [85,5 %]			
Rule 23:				
	dim1 = v12			
	dim2 = v22			
	dim3 = v32			
	activity = act5			
	->class no [58,8 %]			
Rule 22:				
	dim1 = v11			
	dim2 = v21			
	dim3 = v31			
	activity = act5			
	->class no [51,4 %]			

Tabla A.28: Reglas obtenidas en la fase 3 para $N = 6000$ A.7.19.2. *Ranking* de reglas de la fase 5

Rule	w_{used}	w_{size}	$w_{precision}$	W_r
5	0,80	0,35	0,84	1,99
22	0,80	0,80	0,27	1,87
4	0,35	0,35	0,80	1,50
23	0,28	0,80	0,35	1,43
10	0,35	0,35	0,70	1,40

Tabla A.29: *Ranking* de reglas en la fase 5 para $N = 6000$

Bibliografía

- [Abraham 2001] David Abraham, Liz Crawford, Leanna Lesta, Agathe Merceron and Kalina Yacef. *The Logic Tutor: A Multimedia Presentation*. Interactive Multimedia Electronic Journal of Computer-Enhanced learning, vol. 3, no. 2, nov 2001. <http://imej.wfu.edu/articles/2001/2/03/index.asp>. 50
- [ADL 2004] ADL - Advanced Distributed Learning. *Sharable Content Object Reference Model (SCORM)*, 4th edition, 2004. <http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/20044thEdition/>. 49
- [Agrawal 1993] Rakesh Agrawal, Tomasz Imielinski and Arun Swami. *Mining association rules between sets of items in large databases*. En Peter Buneman and Sushil Jajodia, editores, Proceedings of ACM SIGMOD Conference on Management of Data, páginas 207–216, Washington, USA, may 1993. ACM Press. 40
- [Agrawal 1995] Rakesh Agrawal and Ramakrishnan Srikant. *Mining Sequential Patterns*. En Philip S. Yu and Arbee L.P. Chen, editores, Proceedings of the 11th International Conference on Data Engineering - ICDE, páginas 3–14, Taipei, Taiwan, mar 1995. IEEE Computer Society. https://www.qbic.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/icde95_rj.pdf. 48
- [Arabie 1996] Phipps Arabie, Lawrence J. Hubert and Geert de Soete. *Clustering and Classification*. World Scientific Publishers, 1996. <http://books.google.es/books?id=AgovP9eJP5UC>. 40
- [Azevedo 2008] Roger Azevedo, Amy Witherspoon, Arthur Graesser, Danielle McNamara, Vasile Rus, Zhiqiang Cai and Mihai Lintean. *MetaTutor: An adaptive hypermedia system for training and fostering self-regulated learning about complex science topics*. En John Krantz, editor, Proceedings of the 38th Annual Meeting of Society for Computers in Psychology - SCiP, Chicago, USA, nov 2008. http://home.scip.ws/file.php/1/Past_SCIP_Conference_Programs/2008/Final_SCiP2008_Program.pdf. 12
- [Baker 2004a] Ryan S.J.D. Baker, Albert T. Corbett and Kenneth R. Koedinger. *Detecting Student Misuse of Intelligent Tutoring Systems*. En C. James Lester, Rosa M. Vicari and Fábio Paraquacu, editores, Proceedings of the 7th International Conference on Intelligent Tutoring Systems, vol. 3220, páginas 531–540, Brazil, 2004. Springer. <http://www.springerlink.com/content/lwbcj451817qxfwm/fulltext.pdf>. 41, 42, 52, 80
- [Baker 2004b] Ryan S.J.D. Baker, Albert T. Corbett, Kenneth R. Koedinger and Angela Z. Wagner. *Off-task behavior in the cognitive tutor classroom: when*

- students game the system*. En Elizabeth Dykstra-Erickson and Manfred Tscheligi, editores, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, páginas 383–390, Vienna, Austria, apr 2004. ACM Press. http://portal.acm.org/ft_gateway.cfm?id=985741&type=pdf&coll=GUIDE&dl=GUIDE&CFID=62294797&CFTOKEN=95992074. 52
- [Baker 2008] Ryan S.J.D. Baker, Tiffani Barnes and Joe E. Beck, editores. Proceedings of the First International Conference on Educational Data Mining - EDM08, Montréal, Canada, jun 2008. <http://www.educationaldatamining.org/EDM2008/uploads/proc/fulldings.pdf>. 40, 293, 294
- [Baker 2009] Ryan S.J.D. Baker and Kalina Yacef. *The State of Educational Data Mining in 2009: A Review and Future Visions*. Journal of Educational Data Mining, vol. 1, no. 1, páginas 1–15, oct 2009. http://www.educationaldatamining.org/JEDM/images/articles/vol1/issue1/JEDMVol1Issue1_BakerYacef.pdf. 40
- [Barnett 1994] Vic Barnett and Toby Lewis. *Outliers in statistical data*. Wiley, third edition, 1994. 38
- [Ben-Naim 2007] Dror Ben-Naim, Nadine Marcus and Michael Bain. *Virtual Apparatus Framework Approach to Constructing Adaptive Tutorials*. En Hamid R. Arabnia and Azita Bahrami, editores, Proceedings of the 2007 International Conference on E-Learning , E-Business, Enterprise Information Systems, and E-Government - EEE, páginas 3–10, Las Vegas, USA, jun 2007. CSREA Press. <http://www.adaptiveelearning.com/papers/vaf4at.pdf>. 54
- [Ben-Naim 2009] Dror Ben-Naim, Michael Bain and Nadine Marcus. *A User-Driven and Data-Driven Approach for Supporting Teachers in Reflection and Adaptation of Adaptive Tutorials*. En Romero et al. [Romero 2009], páginas 21–30. <http://www.educationaldatamining.org/EDM2009/uploads/proceedings/edm-proceedings-2009.pdf>. 42, 54
- [BMW-AG 1987] BMW-AG. *Instrucciones de Servicio BMW*. Bayerische Motoren Werke (BMW AG), Munich, Germany, 1987. 77
- [Bravo 2006a] Javier Bravo. *Mecanismos de integración y prueba de herramientas automáticas de evaluación de calidad de cursos adaptativos (tii)*. Master's thesis, EPS, UAM, sep 2006. http://arantxa.ii.uam.es/~jbravo/tii/memoria-dea_final.pdf. 22, 173, 207
- [Bravo 2006b] Javier Bravo and Alvaro Ortigosa. *Integración y Prueba de Herramientas de Evaluación en un Entorno Hipermedia Adaptativo*. En Proceedings of 8th International Symposium on Computers in Education (SIIE2006), páginas 253–261. Universidad de León, León, Spain, oct 2006. (ISBN: 84-9773-302-9). 22, 206

- [Bravo 2006c] Javier Bravo and Alvaro Ortigosa. *Validating the Evaluation of Adaptive Systems by User Profile Simulation*. En Proceedings of Fifth Workshop User-Centred Design and Evaluation of Adaptive Systems held at the Fourth International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006), páginas 479–483. National College of Ireland, Dublin, Ireland, jun 2006. (ISSN: 1649-8623). 21, 173, 205
- [Bravo 2007] Javier Bravo, César Vialardi and Alvaro Ortigosa. *A Problem-Oriented Method for Supporting AEH Authors through Data Mining*. En Proceedings of International Workshop on Applying Data Mining in e-Learning (ADML07) held at the Second European Conference on Technology Enhanced Learning (EC-TEL2007), páginas 53–62. Crete, Greece, sep 2007. (ISSN: 1613-0073). 23, 206
- [Bravo 2008a] Javier Bravo, César Vialardi and Alvaro Ortigosa. *ASquare: A Powerful Evaluation Tool for Adaptive Hypermedia Course System*. En Proceedings of Hypertext 2008 Conference, páginas 219–220. University of Pittsburgh, Pittsburgh, USA, jun 2008. (ISBN: 978-1-59593-998-2). 23, 206
- [Bravo 2008b] Javier Bravo, César Vialardi and Alvaro Ortigosa. *Using Decision Trees for Discovering Problems on Adaptive Courses*. En Proceedings of E-Learn 2008: World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education, páginas 268–277. Las Vegas, USA, nov 2008. (ISBN: 1-880094-66-5). 23, 206
- [Bravo 2009a] Javier Bravo, Estefanía Martín, Alvaro Ortigosa and Rosa M. Carro. *Checking the Reliability of GeSES: Method for Detecting Symptoms of Low Performance*. En Proceedings of workshop on Educational Data Mining held at the 9th International Conference on Intelligent System Design and Applications (ISDA09), páginas 1108–1113. Pisa, Italy, IEEE press, nov 2009. (ISBN: 978-1-4244-4735-0). 24, 206
- [Bravo 2009b] Javier Bravo and Alvaro Ortigosa. *Detecting Symptoms of Low Performance Using Production Rules*. En Romero et al. [Romero 2009], páginas 31–40. <http://www.educationaldatamining.org/EDM2009/uploads/proceedings/edm-proceedings-2009.pdf>. 24, 206
- [Bravo 2010] Javier Bravo, César Vialardi and Alvaro Ortigosa. Handbook of educational data mining, chapter: 26: Using decision trees for improving AEH courses. Taylor and Francis, 2010. 25, 207
- [Browne 1990] Dermot Browne, Peter Totterdell and Mike Norman, editores. Adaptive User Interfaces. Academic Press, London, 1990. 297
- [Brusilovsky 1996] Peter Brusilovsky. *Methods and Techniques of Adaptive Hypermedia*. User Modeling and User-Adapted Interaction: The Journal of Personalization Research, vol. 6, no. 2-3, páginas 87–129, 1996. 12

- [Brusilovsky 2001] Peter Brusilovsky. *Adaptive Hypermedia*. User Modeling and User-Adapted Interaction, vol. 11, no. 1-2, páginas 87–110, 2001. 12
- [Brusilovsky 2004] Peter Brusilovsky, Sergey Sosnovsky and Olena Shcherbini-na. *QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support*. En Janice Nall and Robby Robson, editores, Proceedings of E-Learn, páginas 1806–1813. Washington, DC, USA, AACE, 2004. <http://www.sis.pitt.edu/~peterb/papers/ELearnQP04.pdf>. 12, 116, 218, 219
- [Brusilovsky 2005] Peter Brusilovsky and Sergey Sosnovsky. *Individualized Exercises for Self-Assessment of Programming Knowledge: An Evaluation of Quiz-PACK*. Journal of Educational Resources in Computing (JERIC), vol. 5, no. 3, páginas 6.1–6.22, sep 2005. 116, 219
- [Buendía 2006a] Félix Buendía and Antonio Hervás. *An Evaluation Framework for e-Learning Platforms Based on Educational Standard Specifications*. En Kinshuk, Rob Koper, Piet Kommers, Paul Kirschner, Demetrios Sampson and Wim Didderen, editores, Proceedings of the 6th International Conference on Advanced Learning Technologies (ICALT), páginas 184–186, Kerkrade, The Netherlands, jul 2006. IEEE Computer Society. 32
- [Buendía 2006b] Félix Buendía and Antonio Hervás. *Evaluating e-Learning Platforms Through SCORM Specifications*. En Proceedings of the IADIS Virtual Multi Conference on Computer Science and Information Systems, páginas 53–58, Virtual location, may 2006. 20, 32
- [Camacho 2009] David Camacho, Estrella Pulido, Maria D. Rodríguez-Moreno, Rosa M. Carro, Alvaro Ortigosa and Javier Bravo. *Automatic Course Redesign: global vs. individual adaptation*. International Journal of Engineering Education, vol. 25, no. 6, páginas 1270–1282, dec 2009. (ISSN: 0949-149X/91). 25, 205
- [Carro 1999] Rosa M. Carro, Estrella Pulido and Pilar Rodríguez. *Dynamic generation of adaptive Internet-based courses*. Journal of Network and Computer Applications, vol. 22, páginas 249–257, 1999. 12, 91
- [Carro 2001] Rosa M. Carro. *Un mecanismo basado en tareas y reglas para la creación de sistemas hipermedia adaptativos: aplicación a la educación a través de Internet*. PhD thesis, EPS, UAM, jun 2001. 217
- [Cendrowska 1987] Jadzia Cendrowska. *PRISM: An algorithm for inducing modular rules*. International Journal of Man-Machine Studies, vol. 27, no. 4, páginas 349–370, oct 1987. 47
- [Chen 2000] Franklin Chen, Brad Myers and David Yaron. *Using Handheld Devices for Tests in Classes*. Informe técnico CMU-CS-00-152, CMU-HCII-00-101, Carnegie Mellon University - School of Computer Science and Human

- Computer Interaction Institute, 2000. <http://www.cs.cmu.edu/~pebbles/papers/CMU-CS-00-152.pdf>. 7
- [Clancey 1979] William John Clancey. *Transfer of Rule-Based Expertise through a Tutorial Dialogue*. PhD thesis, Computer Science, University of Stanford, sep 1979. 10
- [Clancey 1987] William John Clancey. Knowledge-based tutoring: the guidon program. MIT Press, 1987. 10
- [Clark 2008] Ruth Colvin Clark and Richard E. Mayer. E-Learning and the Science of Instruction. Pfeiffer, an Imprint of Willey, second edition, 2008. 9
- [de Bra 1999] Paul de Bra, Geert-jan Houben and Hongjing Wu. *AHAM: A Dexter-based Reference Model for Adaptive Hypermedia*. En K. Tochtermann, J. Westbomke, U.K. Wiil and J. Leggett, editores, Proceedings of the 10th ACM Conference on Hypertext and Hypermedia, páginas 147–156, Darmstadt, Germany, feb 1999. ACM Press. <http://wwwis.win.tue.nl/~debra/ht99/ht99.ps>. 36
- [de Bra 2001] Paul de Bra and Jan-Peter Ruiter. *AHA! Adaptive Hypermedia for All*. En Proceedings of the WebNet Conference, páginas 262–268, oct 2001. 48
- [de Bra 2002a] Paul de Bra, Ad Aerts, David Smits and Natalia Stash. *AHA! The Next Generation*. En Proceedings of the ACM Conference on Hypertext and Hypermedia, may 2002. 48
- [de Bra 2002b] Paul de Bra and Natalia Stash. *AHA! Adaptive Hypermedia for All*. En Proceedings of the SANE Conference, páginas 411–412, Maastrich, may 2002. 12, 48
- [DeGroot 1986] Morris H. DeGroot. Probability and statistics. Addison-Wesley Publishing Company, 1986. 180, 181
- [Dix 2004] Alan Dix, Janet E. Finlay, Gregory D. Abowd and Russell Beale. Human-Computer Interaction. Prentice Hall, tercera edition, 2004. 30
- [Dodds 2004a] Philip Dodds. *SCORM Content Agregation Model*. ADL, 2004. 173
- [Dodds 2004b] Philip Dodds. *SCORM Sequencing and Navigation*. ADL, 2004. 173
- [Eckhardt 2009] Clint Eckhardt. *ProLobe*, 2009. <http://www.prolobe.com>. 12
- [edm 2010] Proceedings of the Third International Conference on Educational Data Mining - EDM10 (in press), Pittsburgh, USA, 2010. 40
- [EHEA 1999] EHEA. *The Bologna Declaration of 19 June 1999*, 1999. http://www.ond.vlaanderen.be/hogeronderwijs/bologna/documents/MDC/BOLOGNA_DECLARATION1.pdf. 10

- [EHEA 2010] EHEA. *About the Bologna Process*, 2010. <http://www.ond.vlaanderen.be/hogeronderwijs/bologna/>. 10
- [Fayyad 1997] Usama Fayyad, Gregory Piatetsky-Shapiri and Padhraic Smyth. *From Data Mining to Knowledge Discovery in Databases*. AI Magazine, vol. 17, páginas 37–54, jul 1997. <http://www.kdnuggets.com//gpspubs/aimag-kdd-overview-1996-Fayyad.pdf>. 39
- [Felder 1988] R. M. Felder and L. K. Silverman. *Learning styles and teaching styles in engineering education*. Engineering Education, vol. 78, no. 7, páginas 674–681, 1988. 128, 232
- [Ferl 2009] Ferl. *Evaluating Learning Platforms*. Further Education Resources for Learning, Quality Improvement Agency, 2009. <http://www.excellencegateway.org.uk/page.aspx?o=ferl.aclearn.page.id461>. 32
- [Freedman 2000] Reva Freedman. *What is an Intelligent Tutoring System?* Intelligence, vol. 11, no. 3, páginas 15–16, 2000. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.26.6543&rep=rep1&type=pdf>. 11
- [Freire 2007] Manuel Freire. *An Approach to the Visualization of Adaptive Hypermedia Structures and other Small-World Networks based on Hierarchically Clustered Graphs*. PhD thesis, EPS, UAM, sep 2007. 217, 218
- [García Ferrando 1995] Manuel García Ferrando. *Socioestadística: Introducción a la estadística en sociología*. Alianza Editorial, second edition, 1995. 99, 180
- [Karat 1997] John Karat. *Handbook of Human-Computer Interaction*, chapter: Part IV: Evaluation of HCI. Elsevier Science B. V., second edition, 1997. <http://books.google.es/books?id=WuQbERgXR10C>. 30
- [KDnuggets 2009] Data Mining Community KDnuggets. *Software for Data Mining, Analytics and Knowledge Discovery*. KDnuggets Data Mining Community, 2009. <http://www.kdnuggets.com/software/>. 64
- [Khribi 2008] Mohamed K. Khribi, Mohamed Jemni and Olfa Nasraoui. *Automatic Recommendations for e-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval*. En Paloma Díaz, Kinshuk, Ignacio Aedo and Eduardo Mora, editores, Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies - ICALT, páginas 241–245, Santander, Spain, jul 2008. IEEE Computer Society. 41, 42, 44
- [Koch 2002] Nora Koch and Martin Wirsing. *The Munich Reference Model for Adaptive Hypermedia Applications*. En Paul de Bra, Peter Brusilovsky and Ricardo Conejo, editores, Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems

- AH (LNCS-2347), páginas 213–222, Malaga, Spain, may 2002. Springer. <http://www.pst.informatik.uni-muenchen.de/personen/kochn/munich-koch-wirsing-final.pdf>. 36
- [Koedinger 2008] Kenneth R. Koedinger, Kyle Cunningham, Alida Skogsholm and Brett Leber. *An open repository and analysis tools for fine-grained, longitudinal learner data*. En Baker et al. [Baker 2008], páginas 157–166. <http://www.educationaldatamining.org/EDM2008/uploads/proc/fulldings.pdf>. 40
- [Kolb 2006] David Kolb. *Kolb Learning Styles*, mar 2006. <http://www.businessballs.com/kolblearningstyles.htm>. 49
- [lar 1977] Gran Enciclopedia Larousse. Editorial Planeta, 1977. 76, 87
- [Lesta 2002] Leanna Lesta and Kalina Yacef. *An Intelligent Teaching-Assistant System for Logic*. En Stefano Cerri, Guy Gouardères and Fábio Paraguaçu, editores, Proceedings of the 6th International Conference on Intelligent Tutoring Systems - ITS, páginas 421–431, Biarritz, France, jun 2002. Springer-Verlag. <http://www.springerlink.com/content/5j4raytm3veqe1jg/fulltext.pdf>. 50
- [Li 2004] Jia Li and Osmar R. Zaiane. *Combining Usage, Content, and Structure Data to Improve Web Site Recommendation*. En Proceedings of the 5th International Conference on Electronic Commerce and Web Technologies (EC-Web), páginas 305–315, 2004. <http://www.cs.ualberta.ca/~zaiane/postscript/ecweb04.pdf>. 41, 44
- [Maris 1995] Eric Maris. *Psychometric latent response models*. Psychometrika, vol. 60, no. 4, páginas 523–547, dec 1995. <http://www.springerlink.com/content/2031453k038p5031/fulltext.pdf>. 53
- [Martín 2006] Estefanía Martín, Rosa M. Carro and Pilar Rodríguez. *A Mechanism to Support Context-based Adaptation in M-Learning*. Innovative Approaches for Learning and Knowledge Sharing - Lecture Notes in Computer Science, vol. 4227, páginas 302–315, 2006. 220
- [Martín 2008] Estefanía Martín. *Creación de entornos adaptativos móviles: recomendación de actividades y generación dinámica de espacios de trabajo basadas en información sobre usuarios, grupos y contextos*. PhD thesis, Escuela Politécnica Superior (Universidad Autónoma de Madrid), 2008. <http://www.escet.urjc.es/~emartin/>. 12, 221, 222, 230
- [Martín 2009] Estefanía Martín and Rosa M. Carro. *Supporting the Development of Mobile Adaptive Learning Environments: A case study*. IEEE Transactions on Learning Technologies, vol. 2, no. 1, páginas 23–36, 2009. 127, 220
- [Merceron 2003] Agathe Merceron and Kalina Yacef. *A web-based tutoring tool with mining facilities to improve learning and teaching*. En Ulrich Hoppe, Felisa

- Verdejo and Judy Kay, editores, Proceedings of the 11th International Conference on Artificial Intelligence in Education - AIED, páginas 201–208, Sydney, Australia, jul 2003. IOS Press. http://www.it.usyd.edu.au/~kalina/publis/merceron_yacef_aied03.pdf. 21, 42, 50
- [Merceron 2008] Agathe Merceron and Kalina Yacef. *Interestingness Measures for Association Rules in Educational Data*. En Baker et al. [Baker 2008], páginas 57–66. <http://www.educationaldatamining.org/EDM2008/uploads/proc/fulldings.pdf>. 54
- [Min 2009] Ministerio de Sanidad y Política Social. *Información general sobre la Gripe A*, nov 2009. <http://www.informaciongripe.es/>. 76
- [Mitchell 1997] Tom M. Mitchell. Machine learning. McGraw-Hill, 1997. 103, 104
- [Mobasher 2000] Bamshad Mobasher, Robert Cooley and Jaideep Srivastava. *Automatic personalization based on Web usage mining*. Communication of the Association of Computing Machinery (ACM), vol. 43, no. 8, páginas 142–151, 2000. 39, 41
- [Moodle-Community 2010] Moodle-Community. *Moodle Documentation*, 2010. http://docs.moodle.org/en/About_Moodle. 10
- [OpenACS.org 2009] OpenACS.org. *OpenACS Core Documentation*. OpenACS.org, 2009. <http://openacs.org/doc/>. 42
- [OpenOffice.org 2009] OpenOffice.org. *Documentation: How To Use Calc: PERCENTILE function*. OpenOffice.org, 2009. http://wiki.services.openoffice.org/wiki/Documentation/How_Tos/Calc:_PERCENTILE_function. 99
- [Oppermann 1994] Reinhard Oppermann. *Adaptively supported adaptability*. International Journal of Human Computer Studies, vol. 40, no. 3, páginas 455–472, mar 1994. ISSN: 1071-5819. 36
- [Owen 1962] Donald B. Owen. Handbook of statistical tables. Addison-Wesley Publishing Company, 1962. 181
- [Paramythis 2001] Alexandros Paramythis, Alexandra Totter and Constantine Stephanidis. *A Modular Approach to the Evaluation of Adaptive User Interfaces*. En Stephan Weibelzahl, David Chin and Gerhard Weber, editores, Proceedings of the First Workshop on Empirical Evaluations of Adaptive Systems held at the 8th International Conference on User Modeling - UM, páginas 9–24, Sonthofen, Germany, jul 2001. <http://www.easy-hub.org/workshops/um2001/documents/paramythis.pdf>. 35

- [Paramythis 2005] Alexandros Paramythis and Stephan Weibelzahl. *A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems*. En Proceedings of the 10th International Conference on User Modeling, UM2005, Lecture Notes in Artificial Intelligence LNAI 3538, vol. 3538, páginas 438–442, Edinburgh, UK, jul 2005. Springer. <http://www.springerlink.com/content/94w41q8clp15d0c7/fulltext.pdf>. 36
- [Pei 2001] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl and Helen Pinto. *PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth*. En Proceedings of the 17th International Conference on Data Engineering - ICDE, Heidelberg, Germany, apr 2001. <http://www.cs.uiuc.edu/homes/hanj/pdf/span01.pdf>. 48
- [Perkowitz 1998] Mike Perkowitz and Oren Etzioni. *Adaptive Web Sites: Automatically Synthesizing Web Pages*. En Proceedings of the Fifteenth National Conference on Artificial Intelligence, páginas 727–732, 1998. <http://www.cs.washington.edu/homes/etzioni/papers/aaai98.pdf>. 41
- [Quinlan 1986] J. Ross Quinlan. *Induction of decision trees*. Machine Learning, vol. 1, no. 1, páginas 81–106, 1986. 47, 49
- [Quinlan 1993] J. Ross Quinlan. C4.5: programs for machine learning. Morgan Kaufmann Publishers, Inc., 17th edition, 1993. 63, 65, 67, 68, 70, 71, 72, 88, 200
- [RAE 2001] Real Academia Española RAE. Diccionario de la lengua española. 22th edition, 2001. <http://www.rae.es/rae.html>. 75, 76
- [Romero 2002] Cristóbal Romero, Sebastián Ventura, Carlos de Castro, Wendy Hall and Muan Hong Ng. *Using Genetic Algorithms for Data Mining in Web-based Educational Hypermedia Systems*. En Proceedings of Workshop of the Second International Conference on Adaptive Systems for Web-based Education - AH, Málaga, Spain, may 2002. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.6831&rep=rep1&type=url&i=0>. 46, 47, 172
- [Romero 2003] Cristóbal Romero, Sebastián Ventura, Paul de Bra and Carlos de Castro. *Discovering Prediction Rules in AHA! Courses*. En Proceedings of the International Conference on User Modeling, páginas 25–34, 2003. 47, 48
- [Romero 2004] Cristóbal Romero, Sebastián Ventura and Paul de Bra. *Knowledge Discovery with Genetic Programming for Providing Feedback to Courseware Authors*. User Modeling and User-Adapted Interaction, vol. 14, no. 5, páginas 425–464, jan 2004. 41, 42, 47, 48
- [Romero 2005] Cristóbal Romero, Sebastián Ventura and César Hervás. *Estado actual de la aplicación de la minería de datos a los sistemas de enseñanza*

- basada en web*. En Actas del III Taller Nacional de Minería de Datos y Aprendizaje - TAMIDA, páginas 49–46. Thomson, 2005. <http://www.lsi.us.es/redmidas/CEDI/papers/189.pdf>. 39
- [Romero 2006a] Cristobal Romero and Sebastián Ventura, editores. Data Mining in e-Learning. WIT Press, Southampton, UK, 2006. 298, 300
- [Romero 2006b] Cristóbal Romero, A. R. Porras, Sebastián Ventura, César Hervás and Amelia Zafra. *Using sequential pattern mining for links recommendation in adaptive hypermedia educational systems*. En A. Méndez-Vilas, A. Solano, J. A. Mesa and J. Mesa, editores, Proceedings of the 4th International Conference on Multimedia and Information and Communication Technologies in Education - Current Developments in Technology-Assisted Education (Technological Issues), vol. 2, páginas 1016–1020, Sevilla, Spain, nov 2006. <http://www.formatex.org/micte2006/pdf/1016-1020.pdf>. 41, 42, 48
- [Romero 2007] Cristóbal Romero and Sebastián Ventura. *Educational Data Mining: a Survey from 1995 to 2005*. Expert Systems with Applications, vol. 33, no. 1, páginas 135–146, 2007. 40
- [Romero 2008] Cristóbal Romero, Mykola Pechenizkiy, Toon Calders, Silvia R. Viola and Frans Van Assche, editores. Proceedings of International Workshop on Applying Data Mining in e-Learning, vol. 305, Crete, Greece, 2008. CEUR-WS.org. <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-305/>. 39
- [Romero 2009] Cristóbal Romero, Sebastián Ventura, Tiffani Barnes and Michael Desmarais, editores. Proceedings of the Second International Conference on Educational Data Mining - EDM09, Córdoba, Spain, jul 2009. Universidad de Córdoba. <http://www.educationaldatamining.org/EDM2009/uploads/proceedings/edm-proceedings-2009.pdf>. 40, 288, 289, 298
- [Rosenberg 2001] Mark J. Rosenberg. E-learning: Strategies for delivering knowledge in the digital age. McGraw-Hill, 2001. 9
- [Rosenberg 2005] Mark J. Rosenberg. *The Future of e-Learning*. ASTD's Learning Circuits Webzine, páginas 1–5, 2005. http://www.marcrosenberg.com/images/Interview_ASTD_Learning_Circuits_0505.pdf. 9
- [Rubio 2003] María José Rubio. *Enfoques y Modelos de Evaluación del e-Learning*. Revista ELección de Investigación y Evaluación Educativa (RELIEVE), vol. 9, no. 2, 2003. http://www.uv.es/RELIEVE/v9n2/RELIEVEv9n2_1.htm. 20, 31, 38
- [Sancerni 2008] María Dolores Sancerni and Paz Villar. *Evaluación de la plataforma Illuminate Live!: un estudio piloto en la Universidad de Valencia*. @tic. Revista de Innovación Educativa, vol. 1, 2008. <http://ojs.uv.es/index.php/attic/article/view/38/48>. 10

- [Srikant 1996] Ramakrishnan Srikant and Rakesh Agrawal. *Mining Sequential Patterns: Generalizations and Performance Improvements*. En Peter Apers, Mokrane Bouzeghoub and Georges Gardarin, editores, Proceedings of the 5th International Conference on Extending Database Technology - EDBT, vol. 1057, páginas 3–17, Avignon, France, mar 1996. Springer-Verlag. <http://www.rsrikant.com/papers/edbt96.pdf>. 48, 49
- [Srivastava 2000] Jaideep Srivastava, Robert Cooley, Mukund Deshpande and Pang-Ning Tan. *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data*. SIGKDD Explorations, vol. 1, no. 2, páginas 12–23, jan 2000. <http://www.sigkdd.org/explorations/issues/1-2-2000-01/srivastava.pdf>. 40, 81, 88
- [Sta 2009] Stanford Center for Biomedical Informatics Research. *Protégé*, 2009. <http://protege.stanford.edu/>. 84
- [Su 2006] Jun-Ming Su, Shian-Shyong Tseng, Wei Wang and Jui-Feng Weng. *Learning Portfolio Analysis and Mining for SCORM Compliant Environment*. Educational Technology and Society, vol. 9, no. 1, páginas 262–275, jan 2006. http://www.ifets.info/journals/9_1/21.pdf. 41, 42, 49
- [Talavera 2004] Luis Talavera and Elena Gaudioso. *Mining Student Data to Characterize Similar Behavior Groups in Unstructured Collaboration Spaces*. En Proceedings of the Workshop on Artificial Intelligence Methods in Computer-Supported Collaborative Learning (CSCL) held in conjunction with the 16th European Conference on Artificial Intelligence - ECAI, páginas 17–23, Valencia, Spain, aug 2004. <http://www.ia.uned.es/~elena/ecai04-ws/papers/TalaveraGaudiosoECAI04ws.pdf>. 41, 42
- [Tan 2006] Pang-Ning Tan, Michael Steinbach and Vipin Kumar. Introduction to Data Mining. Pearson-Addison Wesley publishers, Boston, USA, 2006. 40, 58, 60, 61, 62
- [Totterdell 1990a] Peter Totterdell and Elizabeth Boyle. Adaptive User Interfaces, chapter: The Evaluation of Adaptive Systems, páginas 161–194. En Browne et al. [Browne 1990], 1990. 31
- [Totterdell 1990b] Peter Totterdell and Paul Rautenbach. Adaptive User Interfaces, chapter: Adaptation as a Problem of Design, páginas 61–84. En Browne et al. [Browne 1990], 1990. 36
- [Trosby 2004] Finn Trosby. *SMS, the strange duckling of GSM*. Teletronikk, vol. 3, páginas 187–194, 2004. http://www.telenor.com/teletronikk/volumes/pdf/3.2004/Page_187-194.pdf. 203
- [UC 2010] UC. *Comprende Bolonia*, 2010. <http://planbolonia.universia.es/>. 10

- [Ueno 2002] Maomi Ueno and Keizo Nagaoka. *Learning log database and data mining system for e-Learning - On-Line Statistical Outlier Detection of Irregular Learning Processes*. En Piet Kommers, Valery Petrushin, Kins-huk and Ildar Galeev, editores, Proceedings of the Second IEEE International Conference on Advanced Learning Technologies - ICALT, páginas 436–438, Kazan, Russia, sep 2002. IEEE Computer Society. http://lutf.ieee.org/icalt2002/proceedings/t1402_icalt050_End.pdf. 45
- [Ueno 2003] Maomi Ueno. *LMS with irregular learning processes detection system*. En Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare and Higher Education - E-Learn, páginas 2486–2493, Phoenix, USA, nov 2003. http://www.editlib.org/d/14506/proceeding_14506.pdf. 45
- [Ueno 2005] Maomi Ueno. *Animated Pedagogical Agent based on Decision Tree for e-Learning*. En Peter Goodyear, Demetrios Sampson, David Yang, Kins-huk, Toshio Okamoto, Roger Hartley and Nian-Shing Chen, editores, Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies - ICALT, páginas 188–192, Kaohsiung, Taiwan, jul 2005. IEEE Computer Society. <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1508768&isnumber=32317>. 45, 46
- [Ueno 2006] Maomi Ueno. Data Mining in e-Learning, chapter: Online outlier detection of learners' irregular learning processes, páginas 261–277. En Romero & Ventura [Romero 2006a], 2006. 39, 41, 42, 45, 78, 81
- [Vialardi 2007] César Vialardi, Javier Bravo and Alvaro Ortigosa. *Empowering AEH Authors Using Data Mining Techniques*. En Proceedings of Fifth International Workshop on Authoring of Adaptive and Adaptable Hypermedia (A3H2007) held at the 11th International Conference on User Modeling (UM2007), páginas 33–43. Corfu, Greece, jun 2007. 22, 205
- [Vialardi 2009a] César Vialardi, Javier Bravo and Alvaro Ortigosa. *Improving AEH Courses through Log Analysis*. Journal of Universal Computer Science - J.UCS, vol. 14, no. 17, páginas 2777–2798, feb 2009. 23, 205
- [Vialardi 2009b] César Vialardi, Javier Bravo, Leila Shafti and Alvaro Ortigosa. *Recommendation in Higher Education Using Data Mining Techniques*. En Romero et al. [Romero 2009], páginas 190–199. <http://www.educationaldatamining.org/EDM2009/uploads/proceedings/edm-proceedings-2009.pdf>. 24, 206
- [W3C 2004a] W3C (World Wide Web Consortium). *RDF Vocabulary Description Language 1.0: RDF Schema*, 2004. <http://www.w3.org/TR/rdf-schema/>. 82

- [W3C 2004b] W3C (World Wide Web Consortium). *Resource Description Framework (RDF)*, 2004. <http://www.w3.org/RDF/>. 82
- [Weber 2001] Gerhard Weber and Peter Brusilovsky. *ELM-ART: An Adaptive Versatile System for Web-based Instruction*. International Journal of Artificial Intelligence in Education, vol. 12, páginas 351–384, 2001. 12
- [Weibelzahl 2002a] Stephan Weibelzahl. *Evaluation of Adaptive Systems*. PhD thesis, Faculty I of the University of Trier, oct 2002. 17, 33
- [Weibelzahl 2002b] Stephan Weibelzahl and Gerhard Weber. *Advantages, Opportunities, and Limits of Empirical Evaluations: Evaluating Adaptive Systems*. Künstliche Intelligenz - KI, vol. 3, no. 2, páginas 17–20, 2002. Extended version in <http://www.easy-hub.org/stephan/weibelzahl-ki02.pdf>. 33
- [Weibelzahl 2006] Stephan Weibelzahl, Alexandros Paramythis and Judith Masthoff, editores. Proceedings of 5th Workshop on User-Centred Design and Evaluation of Adaptive Systems held in conjunction with the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems - AH, Dublin, Irland, jun 2006. National College of Irland. http://www.easy-hub.org/workshops/ah2006/doc/UCDEAS06_Proceedings_Final.pdf. 39
- [Witten 2000] Ian H. Witten and Eibe Frank. Data Mining: practical machine learning tools and techniques with java implementations. Morgan Kaufmann Publishers - Academic Press, 2000. 9, 57, 58, 62
- [Witten 2005] Ian H. Witten and Eibe Frank. Data Mining: practical machine learning tools and techniques. Morgan Kaufmann Publishers - Elsevier, second edition, 2005. 64
- [Woodhead 1991] Nigel Woodhead. Hypertext and hypermedia theory and applications. Addison-Wesley Publishing Company, 1991. 11
- [Woolf 2009] Beverly Park Woolf. Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. Morgan Kaufmann Publishers - Elsevier, 2009. 10, 11
- [Worthen 1996] Blaine Worthen, James Sanders and Jody Fitzpatrick. Program Evaluation: Alternative Approaches and Practical Guidelines. Longman Pub Group, second edition, 1996. 30
- [Zaiane 2001a] Osmar R. Zaiane. *Building a recommender agent for e-learning systems*. En Proceedings of International Conference on Computers in Education, páginas 55–59, 2001. 44
- [Zaiane 2001b] Osmar R. Zaiane. *Web usage mining for a better web-based learning environment*. En Proceedings of the 4th IASTED International Conference on Computers and Advanced Technology in Education - CATE, páginas

- 60–64, Banff, Canada, jun 2001. <http://www.cs.ualberta.ca/~zaiane/postscript/CATE2001.pdf>. 44
- [Zaïane 2006] Osmar R. Zaïane. Data Mining in e-Learning, chapter: Recommender System for e-Learning: Towards Non-Instructive Web Mining, páginas 79–96. En Romero & Ventura [Romero 2006a], 2006. 21, 41, 42, 43
- [Zupan 2008] Blad Zupan and Janez Demsar. *Open-Source Tools for Data Mining*. Clinics in Laboratory Medicine, vol. 28, no. 1, páginas 37–54, 2008. <http://eprints.fri.uni-lj.si/893/1/2008-OpenSourceDataMining.pdf>. 63

Lista de abreviaturas

A3H	Authoring of Adaptive and Adaptable Hypermedia
ACS	ArsDigita Community System
ADAPT2	Advanced Distributed Architecture for Personalized Teaching and Training
ADL	Advanced Distributed Learning
ADML	Applying Data Mining in e-Learning
AEH	Adaptive Educational Hypermedia
AEHS	Adaptive Educational Hypermedia Systems
AeLP	Adaptive eLearning Platform
AH	Adaptive Hypermedia
AHS	Adaptive Hypermedia Systems
AI	Artificial Intellingence
ASquare	Author Assistant
BOE	Boletín Oficial del Estado
cap	capítulo
CF	Confidence factor
CLOVER	Cluster-Oriented Visualization Environment
CoMoLE	Context-based Adaptive Mobile Learning Environments
CSV	Comma Separated Values
CUMULATE	Centralized User Modeling Architecture for Teaching
DM	Data Mining
DMS	Data Mining System
DOI	Documment Object Identifier
DR	Decision Rules
DT	Decision Trees

- DVD-ROM Digital Versatile Disc - Read Only Memory
- EC-TEL European Conference on Technology Enhanced Learning
- EDM Educational Data Mining
- EHEA European Higher Education Area
- EPRules Education Prediction Rules
- EPS Escuela Politécnica Superior
- FAQ Frequently Asked Questions
- FPI Formación de Personal Investigador
- GA Genetic Algorithms
- GeSES Generate Select Establish Select
- GSP Generalized Sequential Pattern
- HADA Hipermedia Adaptativa para la atención a la Diversidad en entornos de inteligencia Ambiental
- HTML HyperText Markup Language
- IAS Interactive Adaptive Systems
- IQ Interquartile Range
- ISDA Intelligent System Design and Applications
- ITA Intelligent Teaching Assistant
- ITS Intellingent Tutoring Systems
- LED Light-Emitting Diode
- Logic-ITA Intelligent Teaching Assistant of logic exercises
- LRM Latent Response Model
- MC Moodle Community
- MDL Minimum Description Length
- Moodle Modular Object-Oriented Dynamic Learning Environment
- MP Modus Ponens
- MT Modus Tollens
- OpenACS Open Architecture Community System

p/pp	página/páginas
PAWS	Personalized Adaptive Web Systems
PDA	Portable Digital Assistant
PER	Pessimistic Error Rate
PSLC	Pittsburgh Science of Learning Center
RAE	Real Academia Española
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
ROI	Return On Investment
SCORM	Sharable Content Object Reference Model
SimuLog	Simulation of User Logs
SMS	Short Message Service
SO1	Sistemas Operativos I
SQL	Structured Query Language
TANGOW	Task-based Adaptive Learner Guidance On the Web
TII	Trabajo de Iniciación a la Investigación
U-CAT	Ubiquitous Collaborative Adaptive Training
UAM	Universidad Autónoma de Madrid
UC	Universia y CRUE
UM	User Modeling
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
Weka	Waikato Environment for Knowledge Analysis
Wotan	Web-based Online Task-based Adaptive Learning
WotEd	Wotan Editor
YALE	Yet Another Learning Environment