

**DIVISIÓN DE ESTUDIOS DE POSGRADO E
INVESTIGACIÓN**



Tesis:

**“Agente selector de estrategias de enseñanza-aprendizaje
para la educación basada en competencias”**

Para Obtener el Grado de:

Maestro en Ciencias en Ciencias de la Computación

Presenta:

ISC. Julia Patricia Sánchez Solís

Director de Tesis

Dra. María Lucila Morales Rodríguez

Co-Director de Tesis

M.C. José Apolinar Ramírez Saldívar



SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

SECRETARÍA DE
EDUCACIÓN PÚBLICA

Cd. Madero, Tamps; a **17 de Agosto de 2012.**

OFICIO No.: U5.155/12
AREA: DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN
DE TESIS

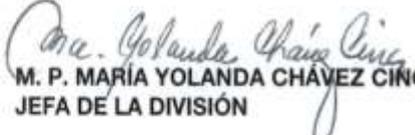
**C. ING. JULIA PATRICIA SÁNCHEZ SOLÍS
PRESENTE**

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestría en Ciencias en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

**"AGENTE SELECTOR DE ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE PARA LA EDUCACIÓN BASADA EN
COMPETENCIAS"**

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

ATENTAMENTE
"Por mi patria y por mi bien"


M. P. MARÍA YOLANDA CHÁVEZ CÍNCRO
JEFA DE LA DIVISIÓN



S.E.P.
DIVISIÓN DE ESTUDIOS
DE POSGRADO E
INVESTIGACIÓN
I.T.C.M.

c.c.p.- Archivo
Minuta



Ave. 1ª de Mayo y Sor Juana I. de la Cruz, Col. Los Mangos, C.P. 89440 Cd. Madero, Tam.
Tels. (833) 3 57 48 20, Fax: (833) 357 48 20, Ext. 1002, email: itcm@itcm.edu.mx
www.itcm.edu.mx



DECLARACIÓN DE ORIGINALIDAD

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y publicaciones.

Además, en caso de infracción de los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Ciudad Madero y sus autoridades.

27 de Agosto del 2012, Cd. Madero, Tamps.

ISC. Julia Patricia Sánchez Solís

DEDICATORIAS

A Dios Todopoderoso por iluminarme el camino a seguir y porque siempre está conmigo en los buenos y sobre todo en los malos momentos.

A mis padres *María Solís Zúñiga* y *Juan Manuel Sánchez Banda* por comprenderme y alentarme en cada uno de mis proyectos de vida.

A mi esposo *Rogelio Florencia Juárez* quien me ha brindado su amor y su apoyo incondicional en todo momento.

AGRADECIMIENTOS

En primer lugar deseo expresar un especial agradecimiento a la directora de esta tesis de Maestría, Dra. María Lucila Morales Rodríguez, por la dedicación y apoyo brindado a este trabajo, por guiarme de forma rigurosa, por enmarcar mi trabajo en el suyo y gracias principalmente por ser parte en mi formación como investigadora y profesional del área de las Ciencias Computacionales.

Por su orientación y disposición para la culminación de este trabajo, mi sincero agradecimiento al Co-Director de esta tesis de Maestría, M.C. Apolinar Ramírez Saldívar quien atendió cada una de mis consultas sobre el desarrollo del proyecto.

También, gracias al Dr. Arturo Hernández y Dr. José Antonio Martínez Flores, por la revisión cuidadosa que han realizado de este texto y sus valiosas sugerencias.

Asimismo quisiera expresar mi gratitud a la M.C. Graciela Mora Otero, a la M.C. Euri Salgado y al Ing. Carlos Arturo Aguilar Díaz, por haberme brindado su apoyo para realizar la experimentación de este proyecto de tesis.

Quiero extender mi agradecimiento a mis compañeros de maestría con quienes he compartido espacio, proyectos e ilusiones durante estos dos años.

Para todos los miembros del Departamento de Posgrado de la Maestría en Ciencias en Ciencias de la Computación: profesores y personal administrativo, vayan también mis más sinceros agradecimientos.

Finalmente, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT), a la Dirección General de Educación Superior Tecnológica (DGEST), y sobre todo al Instituto Tecnológico de Ciudad Madero (ITCM), por todas las facilidades y apoyo otorgado para la realización de este trabajo de investigación.

RESUMEN

El desarrollo de las tecnologías de la información y la comunicación, aunado al desarrollo de la Inteligencia Artificial, han propiciado la creación de plataformas para la gestión del aprendizaje, tal es el caso de los Sistemas Tutores Inteligentes (STI). Los STI fueron creados con el objetivo de proporcionar aprendizaje a los alumnos. Para obtener mejores beneficios de estos sistemas, es necesario implantar elementos que garanticen la calidad de la formación virtual. Fue por esta razón, que en este proyecto de tesis, se consideró integrar en un STI el concepto de estilos de aprendizaje y la educación basada en competencias.

En este trabajo de investigación se plantea la problemática de incorporar en el módulo tutor de un STI, un agente capaz de seleccionar la estrategia de enseñanza más adecuada al estilo de aprendizaje del alumno, para mostrarle los materiales didácticos de forma personalizada. También se abordó la problemática de integrar un proceso que diagnostique si un alumno posee las competencias específicas de una asignatura, para determinar si es necesario cambiar la estrategia de enseñanza, con el fin de apoyar el aprendizaje del alumno. Estos dos procesos fueron implementados en la plataforma Moodle, el cual es un STI que los profesores pueden utilizar para crear cursos en línea.

Para comprobar que mostrar los contenidos al alumno de manera personalizada, ayuda a que éste obtenga mejores calificaciones, se crearon dos cursos en Moodle y se realizó un experimento con dos grupos de alumnos. A cada grupo de alumnos se le pidió registrarse en uno de los cursos para que revisara un tema y contestara el examen correspondiente. En uno de los cursos se presentó a los alumnos el material didáctico de forma personalizada. En el otro curso no se visualizó el contenido de forma personalizada.

A pesar de observar una tendencia en los resultados de la experimentación a favor de las teorías aquí planteadas, los elementos utilizados, no fueron los idóneos para *asegurar* que se puede diagnosticar que un alumno posee una competencia dada, así como tampoco fue posible *aseverar* que mostrarle al alumno la información de acuerdo a su estilo de aprendizaje, conlleva a que obtenga un mejor desempeño que otro alumno, al cual no se le enseña conforme a su estilo de aprendizaje. Dado estas situaciones, se identificaron factores y se realizaron algunas recomendaciones que pueden mejorar el escenario de la experimentación, esperando obtener resultados más confiables.

SUMMARY

The development of technologies of information and communication, coupled with the development of artificial intelligence, have brought on to the creation of platforms to learning's management, such is the case of Intelligent Tutoring Systems (ITS). The ITS's were created in order to guide students in their learning process in a personalized way. For best benefits of these systems, in this thesis, were integrated in a STI, the concept of learning styles and competency-based education.

In this research raises the problematical to incorporating in the tutor module of a STI, an agent capable of selecting the most appropriate teaching strategy to the student's learning style, to show him teaching materials in a personalized way. Also addressed the problematical to incorporating a process to diagnose whether a student has the specific competencies of a subject, to determine whether to change the teaching strategy, in order to support student learning. These two processes were implemented in the Moodle platform, a system for creating online courses.

To verify that, show to the student learning materials of a personalized way, it helps you get better grades, an experiment was conducted with two groups of students. Each group of students was registered, in one of two courses, to review a topic and answer the examination. One course shows to the students the learning material in a personalized way.

As no significant difference, in the superiority of score achieved by students who received personalized material, can not be said to show the student the information according to your learning style, entails to the student gets to outperform another student, whom is not taught according to their learning style. Considering this, some recommendations were realized, that can improve the stage of experimentation, hoping to obtain more reliable results.

TABLA DE CONTENIDOS

RESUMEN.....	i
SUMMARY.....	ii
LISTADO DE FIGURAS.....	vi
LISTADO DE TABLAS.....	viii
CAPÍTULO 1 INTRODUCCION.....	1
1.1 Definición del problema	1
1.2 Justificación y Beneficios	2
1.3 Objetivo	3
1.3.1 Objetivo General.....	3
1.3.2 Objetivos Específicos	3
1.4 Hipótesis	4
1.5 Alcances y Limitaciones.....	4
1.6 Organización del documento	5
CAPÍTULO 2 MARCO CONCEPTUAL	7
2.1 Sistemas Tutores Inteligentes	7
2.2 Sistemas Gestores del Aprendizaje.....	10
2.2.1. Moodle.....	11
2.2.2 Estándares de e-learning	12
2.3 Estilos de Aprendizaje	14
2.3.1 Modelos de estilos de aprendizaje	14
2.3.2 Modelo VARK	19
2.4 Redes Bayesianas para la representación de conocimiento.....	21
2.4.1 Sistema Elvira.....	29
2.5 Educación Basada en Competencias	30
CAPÍTULO 3 ESTADO DEL ARTE	34
3.1 Trabajos que siguen la arquitectura de los STI.....	34
3.1.1 ActiveMath	34
3.1.2 Sistema Tutor Inteligente basado en Web, BITS	35
3.1.3 Tutor Inteligente para aprendizaje basado en ejemplos.....	36
3.2 Trabajos que consideran Estilos de Aprendizaje.....	36
3.2.1 Diseño de un Sistema Tutor Inteligente basado en estilos cognitivos.....	37
3.2.2 Learning styles, Personalisation and Adaptable e-learning.....	38
3.2.3 Modelado del Estudiante en Sistemas Tutores Inteligentes	41
3.2.4 Integrando Estilos de Aprendizaje y Comportamiento afectivo en un Ambiente de Aprendizaje Inteligente	43
3.2.5 A Web-Based Intelligent Tutoring System on teaching and learning Electrical Project.....	44
3.3 Trabajos que consideran Competencias	46
3.3.1 Arquitectura para un Sistema Tutor Inteligente basado en Competencias (STI-C)	46
3.3.2 Intelligent Tutoring System Framework for the Acquisition of Knowledge and Competences	47

3.4 Trabajos implementados sobre la plataforma Moodle	49
3.4.1 Personalización de Itinerarios Formativos en Moodle adaptados a los Estilos de Aprendizaje de Honey-Alonso.....	49
3.4.2 Modificación del módulo SCORM para el soporte básico del IMS-LD	51
3.5 Comparación de Trabajos	53
CAPÍTULO 4 PROPUESTA DE SOLUCIÓN	56
4.1 Propuesta de arquitectura.....	56
4.2 Módulo Conocimiento	58
4.3 Módulo Interfaz	59
4.4 Módulo Tutor.....	60
4.5 Módulo estudiante	60
CAPÍTULO 5 ANÁLISIS Y CONFIGURACIÓN DE MOODLE	62
5.1 Análisis de los componentes de Moodle	62
5.1.1 Archivos PHP de Moodle	63
5.1.2 Funciones para acceder a la base de datos de Moodle	63
5.1.3 Objetos globales de Moodle	64
5.2 Configuración de Moodle.....	64
5.2.1 Creación de un curso	64
5.2.2 Elaboración, empaquetado e importación de contenidos	66
5.2.3 Elaboración de exámenes para evaluar temas del curso.....	67
5.2.4 Configuración de los campos de perfil del usuario	69
5.2.5 Modificación del menú: Administración del sitio	70
5.2.6 Módulos añadidos a Moodle.....	73
CAPÍTULO 6 IMPLEMENTACIÓN DEL PROTOTIPO	95
6.1 Configuración de la Red Bayesiana.....	95
6.1.1 Modelado	95
6.1.2 Obtención de las probabilidades.....	97
6.2 Implementación del proceso para el diagnóstico de las competencias.....	100
6.2.1 Ubicación de nuevas evidencias	101
6.2.2 Inferencia sobre la red bayesiana.....	102
6.2.3 Recuperación de los resultados generados por Elvira	115
6.3 Implementación del Agente Selector.....	120
6.3.1 Proceso de selección del agente selector	120
6.3.2 Proceso para el filtrado de los contenidos	123
CAPÍTULO 7 EXPERIMENTACIÓN Y RESULTADOS.....	127
7.1 Características de la muestra examinada y del material didáctico utilizado.	127
7.2 Proceso realizado en la experimentación.....	129
7.3 Resultados obtenidos	130
7.4 Discusión de resultados	137
CAPÍTULO 8 CONCLUSIONES Y TRABAJOS FUTUROS.....	139
8.1 Conclusiones.....	139
8.2 Aportaciones	140
8.3 Trabajos Futuros	141

ANEXOS	142
Anexo A. Creación y configuración de cursos en Moodle.....	142
Anexo B. Creación de paquetes SCORM y su empaquetamiento utilizando el software Reload Editor	146
Anexo C. Importar paquetes SCORM a Moodle.....	152
Anexo D. Creación de exámenes utilizando Moodle	157
Anexo E: Configuración de campos de perfil del usuario.....	164
Anexo F: Script para añadir las tablas a la base de datos de Moodle.....	167
<i>Anexo G: Cuestionario VARK</i>	170
Anexo H: Interfaces para los catálogos de estilos, estrategias y competencias ...	173
Anexo I: Interfaces para realizar la vinculación entre temas, estilos de aprendizaje, estrategias de enseñanza y competencias.	174
Anexo J: Interfaces para capturar las probabilidades	175
Anexo K: Formatos Elvira.....	176
Anexo L: Publicaciones.....	178
 REFERENCIAS BIBLIOGRÁFICAS	 179

LISTADO DE FIGURAS

Figura 1. Arquitectura básica de un STI (Butz C.J., Hua S., Maguire R.B, 2006)	9
Figura 2. Cuadrantes Cerebrales de Herrmann	15
Figura 3. Una red bayesiana representando influencias causales entre cinco variables.....	22
Figura 4. Una red bayesiana con dos estados y sus tablas de probabilidad condicional	26
Figura 5. Arquitectura de ActiveMath	35
Figura 6. Arquitectura de BITS.....	35
Figura 7. Arquitectura de ExBL.....	36
Figura 8. Arquitectura de STI-SWS	37
Figura 9. El motor de procesamiento semántico de <i>iLearn</i>	39
Figura 10. Diagrama de flujo de la interacción del alumno en <i>iLearn</i>	41
Figura 11. Componentes básicos del Módulo del Estudiante.....	42
Figura 12. Diagrama general del modelo de comportamiento afectivo	43
Figura 13. Principales características del editor de contenidos	45
Figura 14. Arquitectura de STI-C	46
Figura 15. Representación tradicional del árbol del conocimiento	48
Figura 16. Representación de los datos en el framework.....	48
Figura 17. Arquitectura propuesta.....	57
Figura 18. Diagrama de secuencia del STI	58
Figura 19. Organización de las carpetas de Moodle.	62
Figura 20. Diagrama entidad-relación de las tablas donde se guarda la información de los cursos	65
Figura 21. Diagrama entidad-relación de las tablas donde se guarda la información relacionada con los paquetes SCORM	67
Figura 22. Diagrama entidad-relación de las tablas donde se guarda la información relativa a los exámenes creados en Moodle.....	68
Figura 23. Diagrama entidad-relación de las tablas donde se guarda la información de los usuarios y de los campos añadidos al perfil del usuario.....	70
Figura 24. Menú <i>Administración del sitio</i> con la opción <i>Configuración</i> incluida	71
Figura 25. Fragmento de código del archivo <i>top.php</i>	72
Figura 26. Código del archivo <i>itcmConfiguracion.php</i>	73
Figura 27. Estructura y dominio de la tabla <i>mdl_historial_vark</i>	76
Figura 28. Cuestionario VARK incorporado a Moodle	77
Figura 29. Estructura y dominio de la tabla <i>mdl_estilos</i>	78
Figura 30. Diagrama entidad-relación de las tablas relacionadas con el catálogo de estrategias	79
Figura 31. Estructura de la tabla del catálogo de competencias	80
Figura 32. Diagrama entidad-relación de las tablas utilizadas en Vincular tema, estilo y estrategia	83
Figura 33. Diagrama entidad-relación de las tablas utilizadas para Vincular un tema a un examen	84
Figura 34. Diagrama entidad-relación de las tablas utilizadas en Vincular un tema a una competencia.....	86
Figura 35. Diagrama entidad-relación de las tablas utilizadas en la capturar de las probabilidades de los nodos competencia.....	88
Figura 36. Estructura de la tabla <i>mdl_rb_temas_competencias_comb</i>	90

Figura 37. Estructura de la tabla <i>mdl_rb_temas_competencias_porc</i>	92
Figura 38. Diagrama entidad-relación de las tablas utilizadas para almacenar las probabilidades de un nodo tema	93
Figura 39. Ejemplo de la estructura que puede tener una red bayesiana en este proyecto de tesis	95
Figura 40. Representación gráfica de la red bayesiana utilizada en la experimentación de esta tesis.....	97
Figura 41. Red bayesiana	100
Figura 42. Proceso de comunicación entre Moodle y Elvira.....	113
Figura 43. Página que muestra la retroalimentación del examen y los porcentajes de competencia del alumno.....	118
Figura 44. Promedio general de calificaciones de los alumnos y de intentos realizados para cada tipo de curso	130
Figura 45. Promedio general de calificaciones obtenido en cada intento por curso	131
Figura 46. Distribución de estilos de aprendizaje detectados en el curso <i>Personalizado</i>	132
Figura 47. Distribución de estilos de aprendizaje detectados en el curso <i>No Personalizado</i>	132
Figura 48. Promedio de calificaciones y de intentos por curso, reubicando del curso <i>No Personalizado</i> al curso <i>Personalizado</i> , a los dos alumnos con estilo de <i>Lectura/Escritura</i>	133
Figura 49. Promedio general de calificaciones obtenido en cada intento por curso	133
Figura 50. Promedio general de calificación obtenido por cada uno de los estilos de aprendizaje	135
Figura 51. Promedios generales presentados por cada uno de los estilos de aprendizaje	136
Figura 52. Cantidad de alumnos diagnosticados con competencia, en cada intento realizado .	136

LISTADO DE TABLAS

Tabla 1. Estándares de e-learning	13
Tabla 2. Características de los estudiantes según cada cuadrante	15
Tabla 3. Características de los estudiantes de acuerdo a las cinco dimensiones de Felder y Silverman	16
Tabla 4. Características de los estudiantes en la clasificación de Kolb.....	17
Tabla 5. Características de los estudiantes en el Modelo de la Programación Neurolingüística	17
Tabla 6. Características de los estudiantes del Modelo de los hemisferios cerebrales.....	18
Tabla 7. Características de los estudiantes del Modelo de las Inteligencias Múltiples	19
Tabla 8. Tabla de respuestas predefinidas.....	20
Tabla 9. Identificación de respuestas	20
Tabla 10. Estrategias de enseñanza-aprendizaje para el modelo VARK	21
Tabla 11. Conceptos de redes bayesianas	23
Tabla 12. Conceptos de probabilidad.....	25
Tabla 13. Algoritmos de inferencia.....	28
Tabla 14. Representación del estilo de aprendizaje VARK dentro de iLearn.....	39
Tabla 15. Tabla comparativa del Estado del Arte	55
Tabla 16. Elementos añadidos al módulo interfaz	59
Tabla 17. Archivos PHP de Moodle analizados.....	63
Tabla 18. Funciones analizadas de Moodle.....	63
Tabla 19. Objetos globales de Moodle analizados	64
Tabla 20. Tablas donde se guarda la información de los cursos	65
Tabla 21. Tablas donde se guarda la información relacionada con los paquetes SCORMS	67
Tabla 22. Tablas donde se guarda la información relativa a los exámenes creados en Moodle .	68
Tabla 23. Tablas donde se guarda la información de los usuarios y de los campos añadidos al perfil del usuario	69
Tabla 24. Archivos utilizados para añadir un elemento al menú Administración del sitio	71
Tabla 25. Tabla donde se almacenan las respuestas de los alumnos al contestar el cuestionario VARK	76
Tabla 26. Archivos utilizados para incorporar el cuestionario VARK en Moodle	76
Tabla 27. Tabla de estilos de aprendizaje VARK	78
Tabla 28. Tabla relacionada con el catálogo de estrategias de enseñanza-aprendizaje.....	79
Tabla 29. Archivos que forman el catálogo de estrategias de enseñanza-aprendizaje	79
Tabla 30. Función de los archivos del catálogo de estrategias de enseñanza.....	80
Tabla 31. Tabla relacionada con el catálogo de Competencias.....	80
Tabla 32. Archivos que forman el catálogo de competencias específicas.....	81
Tabla 33. Función de los archivos del catálogo de competencias específicas	81
Tabla 34. Tablas relacionadas con la opción Vincular tema, estilo y estrategia	82
Tabla 35. Archivos que realizan la Vinculación entre tema, estilo y estrategia.....	83
Tabla 36. Función que desempeña cada archivo al Vincular tema, estilo y estrategia	83
Tabla 37. Tablas relacionadas con la opción Vincular un tema a un examen.....	84
Tabla 38. Archivos creados para Vincular un tema con un examen	84
Tabla 39. Función que desempeña cada archivo al Vincular un tema a un examen	85
Tabla 40. Tablas relacionadas con la opción Vincular un tema a una competencia	86
Tabla 41. Archivos que realizan la Vinculación entre un tema y una competencia.....	86

Tabla 42. Función que desempeña cada archivo para Vincular un tema a una competencia.....	86
Tabla 43. Tablas relacionadas con la captura de probabilidades para un nodo competencia.....	87
Tabla 44. Archivos añadidos a Moodle para editar las probabilidades de los nodos competencia	88
Tabla 45. Función que desempeña cada archivo en la edición de las probabilidades de nodos competencia.....	88
Tabla 46. Tabla de probabilidad condicional para un nodo tema relacionado a dos competencias	89
Tabla 47. Información que almacenan los campos de la tabla <i>mdl_rb_temas_competencias_comb</i>	91
Tabla 48. Información almacenada en la tabla <i>mdl_rb_temas_competencias_comb</i> , considerando la combinación de los estados de dos nodos competencias	91
Tabla 49. Almacenamiento de las probabilidades de los nodos tema	92
Tabla 50. Tablas relacionadas con la captura de las probabilidades de un nodo tema.....	92
Tabla 51. Archivos añadidos a Moodle para editar las probabilidades de los nodos tema	93
Tabla 52. Función que realiza cada archivo para editar las probabilidades de un nodo tema.....	93
Tabla 53. Información considerada para crear la red bayesiana para la experimentación	97
Tabla 54. Funciones para formar el archivo que contiene la estructura de la red bayesiana	105
Tabla 55. Librerías y clases de <i>Elvira</i> utilizadas para la inferencia	115
Tabla 56. Características del grupo muestra utilizado en la experimentación	128
Tabla 57. Elementos utilizados en la experimentación	128
Tabla 58. Alumnos del curso <i>Personalizado</i> , categorizados por estilo de aprendizaje y background.....	134
Tabla 59. Alumnos del curso <i>No Personalizado</i> , categorizados por estilo de aprendizaje y background.....	134

Capítulo 1

INTRODUCCION

En la actualidad, es evidente que en las universidades y centros de enseñanza, se ha incrementado el uso de las tecnologías de la información y comunicación en los procesos de enseñanza-aprendizaje, lo cual ha dado origen a la creación de plataformas para la gestión del aprendizaje, tales como los *Sistemas Tutores Inteligentes*.

Un Sistema Tutor Inteligente (STI) es un sistema de software que tiene como objetivo guiar de forma personalizada a los estudiantes en su proceso de aprendizaje y/o formación. Para lograr su objetivo, el STI debe modelar la enseñanza, el dominio del conocimiento del experto, el entendimiento del estudiante sobre ese dominio, y realizar la comunicación entre estos elementos.

Para obtener los mejores beneficios que ofrece esta nueva forma de enseñanza, es necesario considerar algunos elementos que garanticen la calidad de la formación virtual, siendo ésta la razón por la cual se propone que, el diseño de un STI también debe considerar el *estilo de aprendizaje del alumno* así como el *enfoque basado en competencias*.

Este trabajo busca crear un agente selector de estrategias de enseñanza-aprendizaje en el módulo tutor del *Sistema Gestor de Aprendizaje Moodle*, para lo cual el agente integra en su proceso cognitivo el concepto de estilo de aprendizaje. El agente basará su ejecución de acuerdo al resultado generado por un proceso que confirma si un alumno posee las competencias específicas de la materia que le será enseñada. Este proceso, está basado en una red bayesiana.

1.1 Definición del problema

Con el avance de la tecnología computacional se ha incrementado el uso de las herramientas informáticas en la educación. En particular, gracias a las técnicas de la Inteligencia Artificial, surgieron los llamados Sistemas Tutores Inteligentes, los cuales tienen como objetivo ayudar y guiar a los estudiantes en su proceso de aprendizaje.

Sin embargo, la mayoría de estos sistemas presentan a los estudiantes la información de forma estática y carecen tanto de capacidad para adaptarse a las necesidades del estudiante así como de diagnosticar si el estudiante tiene las competencias propias de la asignatura, las cuales hoy en día forman parte de los programas educativos. Por lo tanto, estos sistemas no consideran los siguientes aspectos:

- Una estructura para almacenar el contenido educativo con características multimedia.
- Un proceso que seleccione una estrategia de enseñanza-aprendizaje acorde al estilo de aprendizaje del alumno.
- Un proceso para seleccionar el contenido educativo que se mostrará al estudiante, según su estilo de aprendizaje y la estrategia de enseñanza asociada a él.
- Un proceso que diagnostique si el alumno posee las competencias asociadas a la asignatura que le será enseñada.

En conclusión, las problemáticas abordadas en este proyecto de tesis consistieron en incluir en un Sistema Tutor: *a)* un modelo para detectar el estilo de aprendizaje de cada estudiante, *b)* un proceso que, basado en el desempeño del estudiante, diagnostique si el alumno posee las competencias asociadas a la asignatura que le será enseñada, y *c)* un agente selector que, de acuerdo al resultado obtenido por el proceso del diagnóstico de las competencias, decida si cambia de estrategia de enseñanza-aprendizaje, la cual es utilizada para seleccionar el formato en el cual se le presentará la información al alumno.

1.2 Justificación y Beneficios

El uso de software educativo hoy en día, constituye una opción formativa que facilita el aprendizaje, tal es el caso de los Sistemas Tutores Inteligentes. Estos sistemas permiten crear cursos sobre alguna asignatura en particular, en donde el estudiante puede autoevaluarse, retroalimentarse y ver los temas las veces que lo desee. Por estas razones, los STI's se están volviendo cada vez más atractivos y útiles.

Sin embargo, la mayoría de estos sistemas no están desarrollados bajo estructuras que les permitan tanto llevar a cabo la enseñanza en una forma adaptativa para cada estudiante, como el diagnosticar si el estudiante posee las competencias relacionadas con la asignatura.

Por lo que, el considerar el estilo de aprendizaje del estudiante en la implementación de un STI, puede ser utilizado para seleccionar la estrategia de enseñanza más apropiada al perfil del alumno, y así poder proporcionar una educación individualizada, lo cual de acuerdo con un estudio realizado en (Bloom, 1984) se avala como la forma más efectiva de aprendizaje.

En éste trabajo se implementó, en el módulo tutor de un STI, un agente que tiene como función, seleccionar la estrategia de enseñanza-aprendizaje más adecuada al perfil del alumno, para que de acuerdo a ésta, se muestre la información al estudiante y así apoyarlo en la adquisición de conocimientos, destrezas y finalmente competencias. La ejecución del agente selector estará determinada por el resultado obtenido del proceso de diagnóstico de las competencias. El implementar el enfoque de competencias en un STI servirá como herramienta para sustentar a los programas de enseñanza del ITCM que adoptaron el planteamiento de la educación basada en competencias a partir del año 2010.

1.3 Objetivo

A continuación, se presentan los objetivos generales y particulares del proyecto.

1.3.1 Objetivo General

Crear un agente inteligente en el módulo tutor de un STI, que incorpore el concepto de estilos de aprendizaje en la selección de estrategias de enseñanza-aprendizaje, para apoyar la educación basada en competencias.

1.3.2 Objetivos Específicos

- Incorporar en un STI un modelo para detectar el estilo de aprendizaje de cada alumno.

- Diseñar en el módulo tutor de un STI, un agente que realice un proceso de selección de una estrategia de enseñanza-aprendizaje, que favorezca al estilo de aprendizaje del estudiante.
- Realizar la comunicación entre el STI y el agente selector, para que de acuerdo a la estrategia de enseñanza-aprendizaje seleccionada por el agente, se muestre la información al alumno.
- Incorporar en el módulo de conocimiento de un STI el concepto de la educación basada en competencias.
- Diseñar en el módulo tutor de un STI, un proceso para diagnosticar si el alumno posee las competencias de la asignatura que le será enseñada.

1.4 Hipótesis

La implementación de un agente en el módulo tutor de un STI, que seleccione una estrategia de enseñanza-aprendizaje, la cual favorezca al estilo de aprendizaje del alumno mostrando los contenidos de forma personalizada, permite al alumno comprender de mejor forma los temas de una asignatura, mejorando su desempeño.

La integración de un proceso que diagnostique si un alumno posee las competencias específicas asociadas a una asignatura, permite identificar si es necesario cambiar la estrategia de enseñanza, utilizada en mostrar los contenidos al alumno.

1.5 Alcances y Limitaciones

Existen varias versiones del Sistema Gestor del Aprendizaje Moodle, de las cuales se decidió trabajar con la versión 1.9.12. Dentro de Moodle, se pueden desarrollar diversos tipos de actividades, para llevar a cabo este trabajo de tesis sólo utilizamos las actividades denominadas *SCORM* y *Cuestionario*. Los archivos utilizados en la actividad *SCORM*, serán elaborados bajo el estándar de e-learning 1.2 utilizando el programa Reload Editor 1.3.

Dentro de la literatura se encontraron distintos modelos que proponen una clasificación de los estilos de aprendizaje que pueden presentar los alumnos, dentro de éstos se encuentra el modelo VARK, el cual fue elegido para ser implementado dentro

de Moodle. El modelo VARK propone varias estrategias de enseñanza-aprendizaje que ayudan a cada uno de los estilos, de las cuales, para llevar a cabo la elaboración de los materiales didácticos, sólo se consideró una estrategia de enseñanza asociada a cada uno de los estilos de aprendizaje. La asignatura elegida para ser enseñada mediante Moodle, fue Fundamentos de Programación, de la cual sólo se desarrollaron dos temas correspondientes a la unidad 5.

Dentro de la educación basada en competencias, existen las competencias genéricas y específicas, siendo estas últimas las utilizadas en la implementación de este proyecto. El proceso integrado en Moodle, para realizar el diagnóstico de las competencias, está diseñado para determinar, si un alumno obtuvo cierta calificación debido a que contaba con el nivel requerido de la(s) competencia(s) específicas asociada(s) al tema evaluado. Para realizar el diagnóstico de las competencias, se utilizó el software *Elvira* en su versión 0.162.

La implementación en Moodle del agente selector, está diseñada para cambiar, en caso de ser necesario, la estrategia de enseñanza-aprendizaje, la cual indica el formato en el cual se le debe presentar al alumno la información. Para que el agente selector realice el cambio de estrategia, dependerá del resultado obtenido por el diagnóstico de las competencias.

1.6 Organización del documento

En esta sección se describe cómo está organizado este documento, así como la información que contiene cada uno de los capítulos.

En el capítulo 2, se presenta el marco teórico que da fundamento a este trabajo. Se revisan los conceptos relacionados con Sistemas Tutores Inteligentes, Sistemas Gestores del Aprendizaje, Estilos de Aprendizaje, Educación Basada en Competencias y Redes Bayesianas.

El capítulo 3, muestra los trabajos relacionados que involucran elementos considerados en esta investigación.

El capítulo 4, describe la arquitectura propuesta para la solución de esta tesis, detallando los módulos que componen al STI, así como los elementos que forman a cada uno de ellos. En el capítulo 5, se describe el análisis y la configuración realizados a

Moodle para cumplir los objetivos de este proyecto. En el capítulo 6, se detalla la implementación del agente selector y del proceso para el diagnóstico de las competencias.

El capítulo 7, presenta los resultados obtenidos relacionados con la evaluación del prototipo.

Por último, en el capítulo 8, se describen las conclusiones de este trabajo, así como las líneas de investigación identificadas durante la implementación del mismo.

Capítulo 2

MARCO CONCEPTUAL

La Inteligencia Artificial (IA) es un campo perteneciente a las Ciencias Computacionales, que tiene como objetivo lograr que las computadoras actúen con inteligencia y razonamiento humano. La IA puede ser aplicada en una gran cantidad de áreas, dentro de las cuales se encuentra la educación. El utilizar las técnicas de la IA en la educación, dio origen a los Sistemas Tutores Inteligentes (STI), los cuales actualmente son utilizados en labores de enseñanza-aprendizaje, dentro de los centros educativos.

En esta sección se revisan los conceptos relacionados con Sistemas Tutores Inteligentes y Sistemas Gestores del Aprendizaje. También se revisan los conceptos de Estilos de Aprendizaje, Estrategias de Enseñanza y Educación Basada en Competencias, los cuales están siendo considerados en el campo educativo para promover el aprendizaje. Por último se revisan los conceptos relacionados con las técnicas de la IA utilizadas para el diagnóstico de situaciones.

2.1 Sistemas Tutores Inteligentes

Actualmente la aplicación de la IA en la Educación constituye un campo de creciente interés, donde se tratan de aplicar las técnicas de la IA al desarrollo de sistemas de enseñanza asistida por computadora, esto con el propósito de construir sistemas de enseñanza inteligentes (González, 2004).

Los sistemas de enseñanza asistida por computadora tuvieron su origen a finales de los años 50, pero fue hasta finales de los años 70 dónde, estos sistemas evolucionaron para convertirse en lo que hoy conocemos como Sistemas Tutores Inteligentes (STI's) (González, 2004).

Salgueiro (Salgueiro, 2005) cita a Guardia Robles (1993), quién presentó la siguiente definición para los tutores inteligentes: *Un STI es un sistema de enseñanza asistida por computadora, que utiliza técnicas de Inteligencia Artificial, principalmente*

para representar el conocimiento y dirigir una estrategia de enseñanza; y es capaz de comportarse como un experto, tanto en el dominio del conocimiento que enseña (mostrando al alumno cómo aplicar dicho conocimiento), como en el dominio pedagógico, donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje.

Guardia Robles (1993) sintetizó el siguiente conjunto de características que deben cumplir todos los STI's:

- Deben ser “inteligentes” en comparación con los sistemas tradicionales de enseñanza asistida por computadora, siendo el diferencial de inteligencia los métodos de la rama de la Inteligencia Artificial.
- Deben poseer la capacidad tanto para resolver el problema que se le presenta a un estudiante como también la capacidad de explicar como lo resolvió.
- Utilizar técnicas de IA para planeación, optimización y búsquedas, dejando que el sistema decida el orden de presentación del contenido al alumno.
- La interacción puede ser muy variada en un STI: desde sistemas pasivos (que esperan para que el alumno realice una acción), hasta los que constantemente presentan nueva información (tutor oportunista), con casos intermedios en los que se enseña un concepto en un momento determinado o sólo cuando el alumno lo pide.
- Utilizar nuevas tecnologías, por ejemplo, interfaces que hacen uso de la multimedia y del WWW.
- No basta con indicarle un error al estudiante, el sistema debe hacer hipótesis basadas en el historial de errores del alumno y detectar la fuente del problema.

Los STI's pueden ser vistos bajo el concepto de Agentes, es decir son sistemas computacionales capaces de actuar de manera autónoma para satisfacer sus objetivos y metas, mientras se encuentra situado persistentemente en un medio ambiente (Guerra, 2010). Russell y Norvig (Russell & Norvig, 2004) señalan que: un agente es un sistema capaz de percibir a través de sensores la información que proviene del ambiente donde

está insertado y reaccionar a través de efectores. Los STI's pueden ser diseñados como un agente inteligente o un sistema multiagente dependiendo del grado de especialización que lo conforme. La arquitectura básica de un STI está compuesta por cuatro módulos principales, los cuales se describen a continuación.

Arquitectura de los Sistemas Tutores Inteligentes

La arquitectura clásica de un STI consiste en un módulo de *dominio del conocimiento*, un módulo del *estudiante* y un módulo *tutor* también llamado módulo de estrategias de enseñanza. Estos módulos operan de forma interactiva y se comunican a través de un módulo central que se suele denominar *Interfaz de usuario* (Millán, 2000). Esta arquitectura se representa en la Figura 1 y a continuación se describen cada uno de estos módulos con más detalle.



Figura 1. Arquitectura básica de un STI (Butz C.J., Hua S., Maguire R.B, 2006)

Módulo de Dominio del Conocimiento

Este módulo almacena el material de la asignatura que le será enseñada al estudiante (Butz *et al*, 2006), esto con el objetivo de almacenar todos los conocimientos dependientes e independientes del campo de aplicación del STI. Básicamente el dominio del conocimiento está compuesto de (Cataldi & Lage, 2009):

- *Conocimientos*: son los contenidos que deben cargarse en el sistema, a través de los conceptos, las preguntas, los ejercicios, los problemas y sus relaciones.
- *Elementos didácticos*: son las imágenes, videos, sonidos, es decir material multimedia que facilita al alumno la obtención de conocimiento durante la sesión pedagógica.

Módulo del estudiante

Este módulo almacena la información que refleja el estado de entendimiento que tiene cada estudiante, también puede seguir el entendimiento y necesidades particulares de cada alumno. Sin este componente el módulo tutor sería incapaz de tomar decisiones para adaptar la enseñanza a cada estudiante y estaría forzado a tratar a todos los alumnos de la misma manera (Butz et al, 2006).

Módulo Tutor

Este componente se refiere a las técnicas de enseñanza que se utilizarán durante la interacción con el estudiante. Por ejemplo, el componente debe decidir cuando presentar un tema nuevo, proveer asistencia, monitorear el desempeño, seleccionar el tema a presentar, etc. Este módulo tiene como entrada la evaluación resultante del módulo del estudiante, que utiliza para tomar decisiones apropiadas y así dirigir una tutoría personalizada, utilizando y cambiando las estrategias de enseñanza de acuerdo a las necesidades del alumno (Butz *et al*, 2006).

Interfaz de usuario

La interfaz decide como el sistema interactúa con el usuario. Una interfaz bien diseñada puede incrementar las capacidades de un STI, permitiendo al sistema presentar indicaciones y retroalimentación hacia el estudiante, en una forma clara y directa (Butz *et al*, 2006). La interfaz debe ser fácil de utilizar y atractiva, de forma que el alumno pierda el mínimo tiempo posible en aprender a utilizarla y pueda centrar toda su atención en el proceso de aprendizaje de la materia (Millán, 2000).

De acuerdo a Cárdenas (Cárdenas, 2006) en la actualidad a los Sistemas Tutores Inteligentes se les conoce como *Sistemas de Gestión del Aprendizaje*, uno de ellos es Moodle, el cual fue utilizado en la realización de este proyecto.

2.2 Sistemas Gestores del Aprendizaje

Un Sistema Gestor del aprendizaje (SGA) es una herramienta informática y telemática organizada en función de objetivos formativos. Esta herramienta posee una serie de requisitos básicos, como son permitir el acceso a profesores y a alumnos a través de Internet, con protocolos TCP/IP, y con navegadores estándares, mediante acceso personal, diferenciado y selectivo. También incluye una interfaz gráfica común, con un único punto de acceso, permite utilizar páginas elaboradas con un estándar aceptado por

el protocolo http (normalmente HTML o XML), permitir establecer diferentes niveles de usuarios con distintos privilegios de acceso y uso, etc (Itmazi, 2005).

Algunos SGA son software comercial, mientras que otros son software de código fuente abierto. Entre los SGA's de código abierto se encuentran: Dokeos, Claroline, Ilias, Atutor, Sakai y Moodle, el cual es el SGA utilizado en la implementación de éste proyecto de tesis.

2.2.1. Moodle

De acuerdo a (Dougiamas, 2011) Moodle es un paquete de software para la creación de cursos y sitios Web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educación social constructivista.

Moodle se distribuye gratuitamente como Software libre bajo la Licencia Pública GNU. Básicamente esto significa que Moodle tiene derechos de autor, pero se puede copiar, usar y modificar siempre que se acepten los siguientes términos: proporcionar el código fuente a otros, no modificar o eliminar la licencia original y los derechos de autor, y aplicar esta misma licencia a cualquier trabajo derivado de él.

Moodle puede funcionar en cualquier computadora, donde pueda correr PHP, y soporta varios tipos de bases de datos, en especial MySQL.

Algunas de las características que presenta Moodle son:

- Permite crear cursos en diferentes formatos tales como: semanal, por temas o el formato social basado en debates.
- Ofrece una serie de actividades para los cursos: foros, glosarios, cuestionarios, recursos, consultas, encuestas, tareas, chats y talleres.
- Todas las calificaciones para los foros, cuestionarios y tareas pueden verse en una única página y descargarse como un archivo con formato de hoja de cálculo.
- Los profesores pueden definir sus propias escalas para calificar los cuestionarios, foros, tareas y glosarios.

- Permite incorporar materiales empaquetados según el estándar SCORM. Un paquete SCORM puede incluir páginas Web, vídeos, programas Javascript, presentaciones Flash y cualquier material que funcione en un navegador Web.

2.2.2 Estándares de e-learning

Los estándares de e-learning permiten que un curso desarrollado con cualquier herramienta pueda ser visualizado en cualquier plataforma SGA, siempre y cuando el curso y la plataforma cumplan con el mismo estándar (García, 2011).

La meta de la estandarización, consiste en etiquetar y empaquetar los llamados objetos de aprendizaje (OA) o LO por sus siglas en inglés (Learning Objects), los OA son los elementos más pequeños que forman un curso en línea, y pueden ser desde una imagen en formato JPG, una presentación Power Point o cualquier documento de Word. Las primeras especificaciones o pasos para la estandarización surgen de la necesidad de etiquetar “poner datos” a estos OA, con el fin de poder localizarlos fácilmente cuando se requieran (Castro, 2002).

Dentro de los estándares comunes se encuentran: los metadatos, empaquetamiento de contenidos, secuenciación de contenidos, interoperabilidad, perfiles de los estudiantes, tiempo de interacción, etc., que son requeridos para que haya éxito en la economía del conocimiento y obviamente para el futuro del e-learning. La forma en la que estos estándares pretenden etiquetar a los contenidos, es a partir de los metadatos, que es información acerca de la información. El empaquetamiento se basa en el empleo del lenguaje XML (Extensible Markup Language) (Castro, 2002).

La aplicación de estándares ayuda a asegurar cinco habilidades dentro de los SGA (Castro, 2002).

- **Interoperabilidad.** La capacidad que tiene un sistema de trabajar en otro.
- **Reusabilidad.** La facilidad de reutilizar los objetos de aprendizaje.
- **Gestión.** La facilidad que brinda el sistema para tener la información concreta y correcta acerca del estudiante y de los contenidos.

- **Accesibilidad.** La facilidad de que el estudiante pueda acceder fácilmente a los contenidos apropiados, en el tiempo apropiado.
- **Durabilidad.** La capacidad que tendrá la tecnología de eliminar la obsolescencia.

Antes de que apareciera la frase e-learning, organizaciones alrededor del mundo comenzaron a trabajar en la creación de especificaciones o estándares para las tecnologías relacionadas con el aprendizaje (Castro, 2002).

Los principales estándares de e-learning que se utilizan en la actualidad son los que se muestran en la Tabla 1 (García, 2011):

Tabla 1. Estándares de e-learning

ESTÁNDAR	DESCRIPCIÓN
<i>AICC</i>	Fue el primer estándar de e-learning que se adoptó de manera generalizada. Creado en 1988 por la Aviation Industry CBT Committee, para facilitar la formación de los pilotos, basándose en los entrenamientos en línea, pruebas y lecciones.
<i>IMS Global Consortium</i>	Consortio formado por entidades educacionales, públicas y privadas. Su misión es desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje online
<i>IEEE LTSA</i>	Dentro del IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), el LTSA (Learning Technology Systems Architecture) se encarga de preparar normas técnicas, prácticas y guías recomendadas para el uso informático de componentes y sistemas de educación y de formación.
<i>ADL SCORM</i>	Es el más utilizado a nivel mundial. Creado por la administración estadounidense para facilitar la formación de sus funcionarios a lo largo de todo el país. Tomó parte de otros estándares ya existentes y creó una base sólida (la versión 1.2) sobre la que han ido apareciendo actualizaciones. Las versiones de paquetes SCORM son 1.2 y 2004.

Para el propósito de este proyecto utilizaremos el estándar SCORM 1.2, donde su principal característica es responder a la posibilidad de cargar y trasladar contenidos (cursos) entre diferentes plataformas (García, 2011). El programa editor del SCORM utilizado en este trabajo es Reload Editor.

Haciendo uso de estas herramientas, podremos elaborar los contenidos aplicando las *estrategias de enseñanza* de acuerdo al *estilo de aprendizaje* de cada alumno.

2.3 Estilos de Aprendizaje

Los conceptos que se describen en esta sección, fueron tomados del Manual de Estilos de Aprendizaje, diseñado en el año 2004, por la Dirección de Coordinación Académica, de la Dirección General del Bachillerato (Gómez, 2004).

En (Gómez, 2004), definen que el término “estilo de aprendizaje” se refiere al hecho de que cada persona utiliza su propio método o estrategias para aprender. Aunque las estrategias varían según lo que se quiera aprender, cada persona tiende a desarrollar ciertas preferencias o tendencias globales, tendencias que definen un estilo de aprendizaje. Son los rasgos *cognitivos*, *afectivos* y *fisiológicos* que sirven como indicadores relativamente estables de cómo los alumnos perciben interacciones y responden a sus ambientes de aprendizaje.

Los rasgos *cognitivos* tienen que ver con la forma en que los estudiantes estructuran los contenidos, forman y utilizan conceptos, interpretan la información, resuelven los problemas, seleccionan medios de representación (visual, auditivo, kinestésico), etc. Los rasgos *afectivos* se vinculan con la motivación y las experiencias que intervienen en el proceso de aprendizaje, mientras que los rasgos *fisiológicos* están relacionados con el género y ritmos biológicos.

La noción de que cada persona aprende de manera distinta a las demás, permite buscar las vías más adecuadas para facilitar el aprendizaje, sin embargo se debe tener cuidado de no “etiquetar”, ya que los estilos de aprendizaje, aunque son relativamente estables, pueden cambiar; pueden ser diferentes en situaciones diferentes; son susceptibles de mejorarse; y cuando a los estudiantes se les enseña según su propio estilo de aprendizaje, aprenden con más efectividad.

2.3.1 Modelos de estilos de aprendizaje

A continuación se presentan algunos de los modelos más conocidos en cuanto a Estilos de aprendizaje.

El Modelo de los Cuadrantes Cerebrales de Herrmann

Este modelo se inspira en los conocimientos del funcionamiento cerebral. Lo describen como una analogía de nuestro cerebro con el globo terrestre con sus cuatro puntos

cardinales. Lo representan como una esfera dividida en cuatro cuadrantes (ver Figura 2), los cuales representan cuatro formas distintas de operar, de pensar, de crear, de aprender y, en suma, de convivir con el mundo.

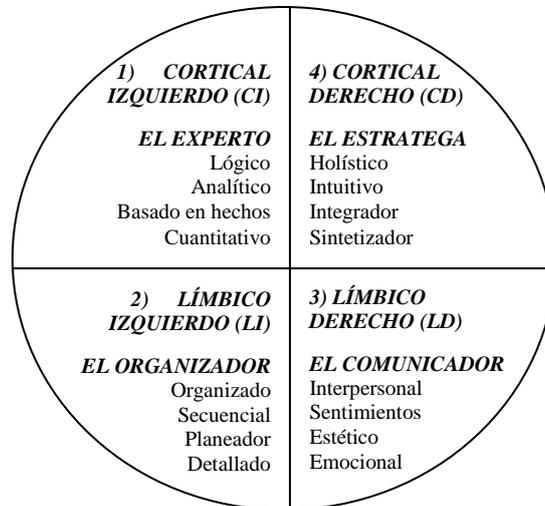


Figura 2. Cuadrantes Cerebrales de Herrmann

En la Tabla 2 se muestran algunas características que presentan los estudiantes, de acuerdo al uso de cada cuadrante.

Tabla 2. Características de los estudiantes según cada cuadrante

CUADRANTE	CARACTERÍSTICAS
Cortical izquierdo (el experto)	<ul style="list-style-type: none"> • Competitivo. • Individualista. • Intellectualmente brillante. • Le agradan las clases sólidas apoyadas en hechos y pruebas. • Toma apuntes.
Límbico izquierdo (el organizador)	<ul style="list-style-type: none"> • Introverso. • Minucioso. • Metódico. • Organizado.
Límbico derecho (el comunicador)	<ul style="list-style-type: none"> • Extroverso. • Idealista. • Espontáneo. • Aprecia las actividades que no parezcan clase.
Cortical derecho (el estratega)	<ul style="list-style-type: none"> • Intuitivo. • Futurista. • Selecciona lo esencial. • Le gustan las discusiones.

Modelo de estilos de aprendizaje de Felder y Silverman

En el modelo de Felder y Silverman clasifican a los estilos de aprendizaje a partir de cinco dimensiones: *activo-reflexivo*, *sensitivo-intuitivo*, *visual-verbal*, *secuencial-global* e *inductivo-deductivo*.

A continuación **Tabla 3** se muestran algunas características que presentan los alumnos dependiendo de la dimensión en la que clasifiquen (ver Tabla 3).

Tabla 3. Características de los estudiantes de acuerdo a las cinco dimensiones de Felder y Silverman

DIMENSIÓN	CARACTERÍSTICAS
Activo	<ul style="list-style-type: none"> • Activo. • Práctico. • Le gusta trabajar en equipo.
Reflexivo	<ul style="list-style-type: none"> • Reflexivo. • Juicioso. • Le gusta trabajar solo.
Sensitivo	<ul style="list-style-type: none"> • Concreto. • Práctico. • Detallista. • Orientado hacia hechos y procedimientos. • Memoriza hechos con facilidad.
Intuitivo	<ul style="list-style-type: none"> • Conceptual. • Innovador. • Orientado hacia las teorías y los significados. • Trabaja bien con abstracciones y formulaciones matemáticas.
Visual	<ul style="list-style-type: none"> • Recuerda mejor lo que ve. • Prefiere representaciones visuales como diagramas, diagramas de flujo, etc.
Verbal	<ul style="list-style-type: none"> • Recuerda mejor lo que lee o lo que oye. • Prefiere obtener la información en forma escrita o hablada.
Secuencial	<ul style="list-style-type: none"> • Aprende en pequeños pasos lógicos, incrementales, ordenados y lineales.
Global	<ul style="list-style-type: none"> • Aprende visualizando la totalidad. • Resuelve problemas complejos rápidamente.
Inductivo	<ul style="list-style-type: none"> • Entiende mejor la información cuando se le presentan hechos y observaciones y luego se infieren los principios o generalizaciones.
Deductivo	<ul style="list-style-type: none"> • Prefiere deducir él mismo las consecuencias y aplicaciones a partir de los fundamentos o generalizaciones.

Modelo de Kolb

El modelo propuesto por Kolb supone que para aprender algo debemos trabajar o procesar la información que recibimos. Por lo que Kolb identifica cuatro tipos de alumnos: activo, teórico, reflexivo y pragmático (ver Tabla 4).

De acuerdo a Kolb cuando el alumno parte de una experiencia directa y concreta, se clasifica como *activo*, o si parte de una experiencia abstracta, por ejemplo cuando leemos acerca de algo o cuando alguien nos lo cuenta, se cataloga como *teórico*.

En esta propuesta un alumno se clasifica como *reflexivo* cuando transforma la experiencia en conocimiento reflexionando y pensando; por el contrario cuando lo hace experimentando de forma activa la información se cataloga como *pragmático*.

En la Tabla 4 se presentan algunas características que presentan los diferentes tipos de estudiantes.

Tabla 4. Características de los estudiantes en la clasificación de Kolb

TIPO DE ALUMNO	CARACTERÍSTICAS
Activo	<ul style="list-style-type: none"> • Se involucra sin prejuicios en situaciones nuevas. • Entusiasta. • Le gusta trabajar en equipo.
Reflexivo	<ul style="list-style-type: none"> • Observador. • Conciencioso. • Precavido.
Teórico	<ul style="list-style-type: none"> • Piensa en forma secuencial. • Analiza y sintetiza la información. • Le gustan las teorías complejas y lógicas.
Pragmático	<ul style="list-style-type: none"> • Práctico. • Le gusta poner en práctica lo aprendido. • Le gusta tomar decisiones y resolver problemas.

Modelo de la Programación Neurolingüística de Bandler y Grinder

Este modelo, también llamado visual-auditivo-kinestésico (VAK), toma en cuenta que tenemos tres grandes sistemas para representar mentalmente la información, el visual, el auditivo y el kinestésico. En la Tabla 5 se mencionan ciertas características que presentan los alumnos de acuerdo a cada sistema. La mayoría de nosotros utilizamos los sistemas de representación de forma desigual, potenciando unos y utilizando en menos grado otros. Los sistemas de representación se desarrollan más cuanto más los utilizemos.

Tabla 5. Características de los estudiantes en el Modelo de la Programación Neurolingüística

SISTEMA	CARACTERÍSTICAS
Visual	<ul style="list-style-type: none"> • Organizado • Ordenado • Observador • Aprende lo que ve.
Auditivo	<ul style="list-style-type: none"> • Aprende mejor cuando recibe las explicaciones oralmente y cuando puede hablar y explicar esa información a otra persona.
Kinestésico	<ul style="list-style-type: none"> • Aprende lo que experimenta directamente, aquello que involucra movimiento.

Modelo de los hemisferios cerebrales

Este modelo propone que cada hemisferio es el responsable de la mitad del cuerpo situada en el lado opuesto: es decir, el hemisferio derecho dirige la parte izquierda del cuerpo, mientras que el hemisferio izquierdo dirige la parte derecha. La idea de que cada hemisferio está especializado en una modalidad distinta de pensamiento ha llevado al concepto de uso diferencial de hemisferios. Esto significa que existen personas que son dominantes en su hemisferio derecho y otras dominantes en su hemisferio izquierdo.

La utilización diferencial se refleja en la forma de pensar y actuar de cada persona. Aunque cada persona utiliza permanentemente todo su cerebro, existen interacciones continuas entre los dos hemisferios, y generalmente uno es más activo que el otro.

En la Tabla 6 se mencionan algunas características que presentan los alumnos de acuerdo a cual hemisferio utilicen.

Tabla 6. Características de los estudiantes del Modelo de los hemisferios cerebrales

HEMISFERIO	CARACTERÍSTICAS DEL MODO DE PENSAMIENTO
Izquierdo	<ul style="list-style-type: none"> • Lógico y analítico. • Abstracto. • Secuencias (de la parte al todo). • Verbal.
Derecho	<ul style="list-style-type: none"> • Holístico e intuitivo. • Global (del todo a la parte). • No verbal. • Analógico.

Modelo de las Inteligencias Múltiples de Gardner

Gardner propuso en su libro “Estructuras de la mente” la existencia de por lo menos siete inteligencias o categorías básicas. Donde los individuos se diferencian es en la intensidad con que recurren a esas inteligencias y la forma en que las combinan para llevar a cabo diferentes labores, para solucionar problemas diversos y progresar en distintos ámbitos.

A continuación se muestra en la Tabla 7 las categorías y algunas características que presentan los estudiantes que las emplean.

Tabla 7. Características de los estudiantes del Modelo de las Inteligencias Múltiples

CATEGORÍAS	CARACTERÍSTICAS
Lógico-matemática	<ul style="list-style-type: none"> • Le gusta resolver problemas, cuestionar, trabajar con números y experimentar.
Lingüístico-verbal	<ul style="list-style-type: none"> • Le gusta leer, escribir, contar cuentos, hablar, memorizar y hacer rompecabezas.
Corporal-kinestésica	<ul style="list-style-type: none"> • Le gusta moverse, tocar, hablar y el lenguaje corporal.
Espacial	<ul style="list-style-type: none"> • Le gusta diseñar, dibujar, construir, crear y observar dibujos.
Musical	<ul style="list-style-type: none"> • Le gusta cantar, tocar instrumentos y escuchar música.
Interpersonal	<ul style="list-style-type: none"> • Le gustan las actividades grupales.
Intrapersonal	<ul style="list-style-type: none"> • Le gusta reflexionar y trabajar solo.

Modelo VARK

Este modelo fue desarrollado por Neil Fleming (Fleming, 2011) en 1987, el cual es un instrumento para determinar las preferencias de las personas al procesar información. Considera 4 distintos estilos de aprendizaje: *Visual*, *Auditivo*, *Lectura/Escritura* y *Kinestésico*. Los cuales dan origen a su nombre por sus iniciales en inglés: **V**isual, **A**ural, **R**ead/Write y **K**inesthetic.

Dentro de las actividades de algunas asignaturas del ITCM se encuentra el solicitar que los estudiantes contesten el cuestionario correspondiente al Modelo de Estilo de Aprendizaje de Neil Fleming, llamado VARK (Visual, Auditivo, Lectura/Escritura y Kinestésico). Por tal razón este modelo fue el utilizado en la implementación de este proyecto.

2.3.2 Modelo VARK

Este modelo utiliza un cuestionario, el cual debe ser contestado por aquella persona a la cual se desea determinar su estilo de aprendizaje predominante. El cuestionario consta de 16 preguntas y cada pregunta tiene 4 opciones de respuesta que corresponden a cada uno de los estilos de aprendizaje. Para contestar el cuestionario es necesario que la persona que lo conteste, seleccione la respuesta que mejor se adapte a sus preferencias, y debe marcar la letra que represente su elección. Puede seleccionar más de una respuesta a una pregunta si una sola no encaja con su predilección y dejar en blanco toda pregunta que no aplique a sus preferencias. Una restricción es, que al menos se deben contestar 12 preguntas.

Evaluación y almacenamiento de los resultados obtenidos del cuestionario

El proceso para obtener los resultados del cuestionario VARK considera tres pasos, los cuales se detallan a continuación:

1. Tener la tabla de respuestas predefinidas por el modelo de (Fleming, 2011). Un fragmento de ésta se aprecia en la Tabla 8.

Tabla 8. Tabla de respuestas predefinidas

PREGUNTA	LETRA a	LETRA b	LETRA c	LETRA d
1	K	A	R	V
2	V	A	R	K
.
.
.
15	K	A	R	V
16	V	A	R	K

2. Identificar y marcar en la tabla la(s) letra(s) que corresponda(n) a la(s) respuesta(s) seleccionada(s). Por ejemplo, si se eligieron las letras **b** y **c** en la pregunta 3, debemos circular las letras **V** y **R** en la fila de la pregunta 3, ver Tabla 9.

Tabla 9. Identificación de respuestas

PREGUNTA	LETRA a	LETRA b	LETRA c	LETRA d
3	K	Ⓟ	Ⓟ	A

Contar por cada pregunta cada una de las letras V, A, R, K elegidas. La letra que haya obtenido el mayor número de puntos, será la que indique el estilo de aprendizaje sobre el cual se tiene preferencia. En caso de que haya empate entre 2 o más letras, se considera que se tiene un estilo de aprendizaje *Multimodal*, el cual no será abordado en este proyecto de tesis.

Al mismo tiempo que estudiamos las posturas de los estudiantes frente al aprendizaje, podemos examinar las diferentes estrategias de enseñanza-aprendizaje que pueden ser aplicadas de acuerdo a los estilos de aprendizaje.

Se puede definir a las estrategias de enseñanza como los procedimientos o recursos utilizados por el agente de enseñanza para promover aprendizajes significativos (Díaz, 1999).

En la Tabla 10 se muestran algunas de las propuestas de Neil Fleming para las estrategias de enseñanza-aprendizaje para cada uno de los estilos de aprendizaje.

Tabla 10. Estrategias de enseñanza-aprendizaje para el modelo VARK

ESTILO DE APRENDIZAJE	ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE
Visual	<ul style="list-style-type: none"> ▪ Imágenes. ▪ Videos. ▪ Carteles, letreros.
Auditivo	<ul style="list-style-type: none"> ▪ Discutir los temas con los maestros. ▪ Exponer ideas novedosas a otros. ▪ Utilizar grabadora.
Lectura/Escritura	<ul style="list-style-type: none"> ▪ Utilizar diccionarios, glosarios, definiciones, manuales. ▪ Libros de texto. ▪ Lecturas.
Kinestésico	<ul style="list-style-type: none"> ▪ Viajes de estudio. ▪ Proporcionar ejemplos de la vida real. ▪ Aplicaciones computacionales.

2.4 Redes Bayesianas para la representación de conocimiento

La Redes Bayesianas (RB), también conocidas como redes de creencia, pertenecen a la familia de los modelos gráficos probabilísticos. Estas estructuras gráficas son utilizadas para representar conocimiento acerca de un cierto dominio. Las RB combinan principios de: teoría de grafos, teoría de probabilidad, teorema de Bayes, estadística y ciencias computacionales (Ben-Gal, 2007; Stephenson, 2000).

Pearl (Pearl, 1999) define a una Red Bayesiana como un grafo acíclico dirigido (GAD), en el cual los nodos representan variables y las aristas, representan dependencias causales entre dichas variables.

Una red bayesiana está definida por un par $B = (G, P)$, donde G es un grafo acíclico dirigido y P es un conjunto de parámetros de la red.

El grafo G está compuesto por un número finito de nodos y aristas. Los nodos X, Y, \dots, Z representan variables aleatorias que pueden ser tipo discreto o continuo, y se

identifican mediante círculos etiquetados con los nombres de las variables. Las aristas, representan dependencias causales dirigidas entre las variables, se identifican con flechas, la variable a la que apunta la flecha depende de la que está en el origen de ésta (Ben-Gal, 2007; Felgaer, 2005).

El conjunto P caracteriza tres elementos esenciales de una red: el concepto de probabilidad como un grado de creencia subjetiva relativa a la ocurrencia de un evento, un conjunto de funciones de probabilidad condicionada que definen a cada variable en el modelo y el teorema de Bayes como herramienta básica para actualizar probabilidades con base en experiencia (López & García, 2008).

En la Figura 3 se muestra el ejemplo de una red bayesiana (Pearl & Russell, 2000), ésta describe la relación causal entre: la temporada del año (X1) puede causar lluvia (X2) o causar que se utilice el aspersor (X3), a su vez, si el aspersor se encuentra encendido o si está lloviendo, pueden causar que haya humedad (X4), y por lo tanto, si hay humedad, esto puede causar que el pavimento esté resbaloso (X5).

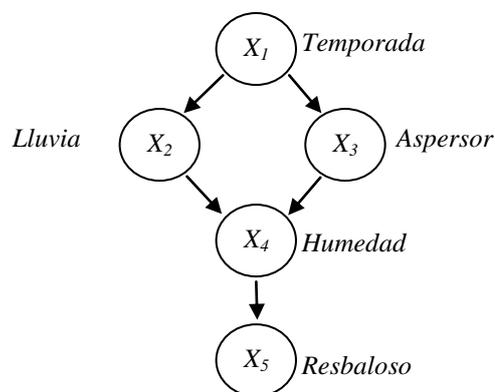


Figura 3. Una red bayesiana representando influencias causales entre cinco variables

Las redes bayesianas son utilizadas en diversas áreas de aplicación, como por ejemplo en medicina (Beinlich, Suermondt, Chavez, & Cooper, 1988), ciencia (Bickmore, 1994) y economía (Ezawa & Schuermann, 1995). En el ámbito de la educación, particularmente en la tutorización electrónica, se utilizan para modelar la incertidumbre asociada al estudiante y a su nivel de conocimiento (López & García, 2008). Algunos STI que se han desarrollado utilizando redes bayesianas son los siguientes: BITS (Butz *et al*, 2006), EXBL (Conati, 2009), STI-C (Badaracco & Martínez, 2010), etc.

A continuación en la Tabla 11, se presentan algunas definiciones de la terminología de las redes bayesianas y en la

algunos conceptos de probabilidad.

Tabla 11. Conceptos de redes bayesianas

TÉRMINO	DEFINICIÓN
Nodo o vértice	Un nodo X es una variable aleatoria que puede tener varios estados x_i .
Arista o arco	Es la unión entre dos nodos y representa la dependencia entre dos variables del modelo. Un arco queda definido por un par ordenado de nodos (X, Y) . En la representación gráfica, un arco (X, Y) viene dado por una flecha desde X hasta Y .
Padre	El nodo X es un padre del nodo Y , si existe un arco (X, Y) entre los dos nodos.
Hijo	El nodo Y es un hijo del nodo X , si existe un arco (X, Y) entre los dos nodos.
Grafo dirigido	Es un par $G = (N, A)$ donde N es un conjunto de nodos y A un conjunto de arcos definidos sobre los nodos.

Una red bayesiana representa relaciones causales en el dominio del conocimiento a través de una estructura gráfica y de las tablas de probabilidad condicional, por lo tanto la construcción de una red bayesiana implica, básicamente, tres tareas (Druzel & Van Der Gaag, 2000):

1. Identificación de las variables (nodos) y de sus estados.
2. Identificación de las relaciones (aristas) entre las variables.
3. Obtención de las probabilidades asociadas a cada nodo del grafo.

Se determina una tabla con las *probabilidades a priori* de cada una de las variables que no tiene padres, y para cada una de las variables restantes, se determina una tabla de *probabilidad condicional*, indicando la probabilidad de sus estados para cada combinación de los estados de sus padres. Posiblemente el aspecto más importante de las RB, es que son representaciones directas del mundo, no sólo procesos de razonamiento (Pearl & Russell, 2000).

En la Figura 4 se ilustra el mismo ejemplo de red bayesiana, expuesto anteriormente, donde se considera que cada una de sus variables tendrá dos estados (verdadero y falso), así como también se muestran las tablas de probabilidad para cada una de las variables.

Tabla 12. Conceptos de probabilidad

TÉRMINO	DEFINICIÓN
Probabilidad conjunta completa	<p>Dado un conjunto de variables (X, Y, \dots, Z), la probabilidad conjunta completa, especifica la probabilidad de cada combinación posible de estados de cada variable $P(x_i, y_j, \dots, z_k) \forall i, j, \dots, k$, de manera que se cumple que:</p> $\sum_{i,j,\dots,k} P(x_i, y_j, \dots, z_k) = 1$
Probabilidad conjunta	<p>Dado un conjunto de variables $X = X_1, \dots, X_n$ la probabilidad conjunta, es el producto de las probabilidades de cada variable, dado el valor de sus padres. La probabilidad conjunta para cualquier red bayesiana es:</p> $P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \text{padres}(x_i))$
Probabilidad condicional	<p>Dadas dos variables X e Y, la probabilidad de que ocurra y_j dado que ocurrió el evento x_i, es la probabilidad condicional de Y dado X y se denota como $P(y_j x_i)$.</p> <p>La probabilidad condicional por definición es:</p> $P(y_j x_i) = \frac{P(x_i, y_j)}{P(x_i)} \quad \text{donde } P(x_i) > 0$ <p>Análogamente, si se intercambia el orden de las variables:</p> $P(x_i y_j) = \frac{P(y_j, x_i)}{P(y_j)} \quad \text{donde } P(y_j) > 0$
Teorema de Bayes	<p>A partir de las dos fórmulas anteriores se obtiene:</p> $P(y_j x_i) = \frac{P(x_i y_j)P(y_j)}{P(x_i)}$ <p>Esta expresión se conoce como el <i>Teorema de Bayes</i> que en su forma más general es:</p> $P(y_j x_i) = \frac{P(x_i y_j)P(y_j)}{\sum_j P(x_i y_j)P(y_j)}$ <p>Al denominador se lo conoce como el <i>Teorema de la Probabilidad Total</i>.</p> <p>En las redes bayesianas, el conjunto de valores que componen la probabilidad condicional de un hijo dados sus padres, se representa en las llamadas tablas de probabilidad condicional.</p>
Independencia condicional	<p>Una variable X es condicionalmente independiente de otra variable Y dada una tercer variable Z, si al conocer Z hace que X e Y sean independientes. Es decir, si conozco Z, Y no tiene influencia en X. Esto es: $P(X Y, Z) = P(X Z)$.</p>
Observación	<p>Es la determinación del estado de un nodo ($X = x$) a partir de un dato obtenido en el exterior del modelo.</p>
Evidencia	<p>Es el conjunto de observaciones $e = \{X = x, Y = y, \dots, Z = z\}$ en un momento dado.</p>
Probabilidad a priori	<p>Es la probabilidad de una variable en ausencia de evidencia.</p>
Probabilidad a posteriori	<p>Es la probabilidad de una variable condicionada a la existencia de una determinada evidencia.</p>

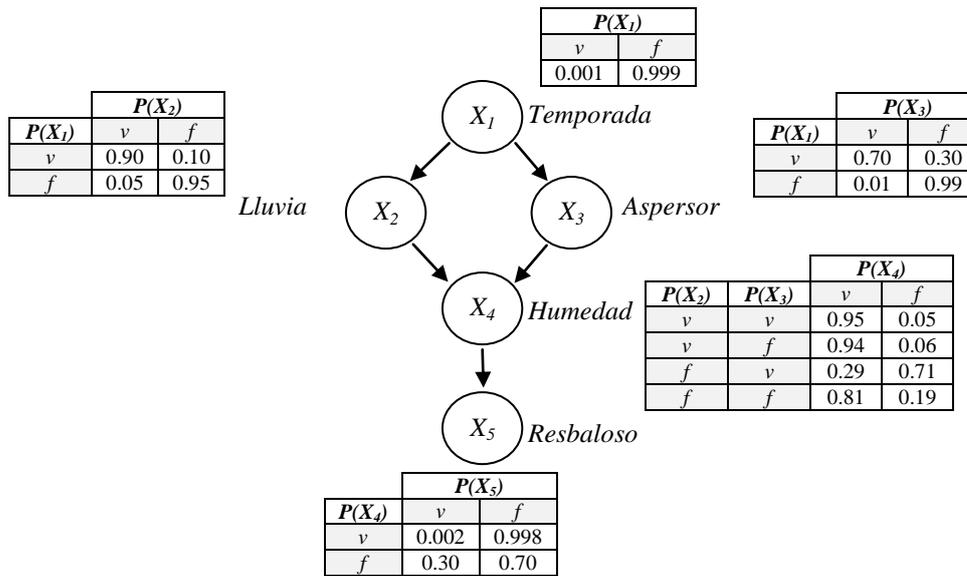


Figura 4. Una red bayesiana con dos estados y sus tablas de probabilidad condicional

Las RB se utilizan en problemas en los que interviene la incertidumbre, es decir, donde no se tiene un completo conocimiento del estado del sistema, pero que sin embargo, podemos realizar observaciones (obtener evidencia) y entonces, actualizar las probabilidades del resto del sistema (Rodríguez & Dolado, 2007). Las RB utilizan el teorema de Bayes como herramienta básica para actualizar las probabilidades con base en la evidencia aportada sobre el modelo (López & García, 2008).

Una vez que se tiene evidencia sobre el estado de ciertas variables, es decir, cuando tenemos conocimiento o podemos observar su estado, se está en condiciones de conocer la probabilidad de una determinada variable, para ello se debe calcular la probabilidad *a posteriori* de cada variable condicionada a la evidencia observada. Para lo cual se debe compilar la red para que se genere una representación interna de las probabilidades del modelo. De esta manera se estima lo que se conoce como probabilidad previa (o probabilidad *a priori*) del modelo. Posteriormente, se van añadiendo evidencias sobre el estado de las variables del modelo y se van obteniendo sucesivamente lo que se conoce como probabilidad posterior (o *a posteriori*).

Estas probabilidades *a posteriori* se podrán obtener a partir de la *probabilidad conjunta* de todas las variables (Felgaer, 2005), la cual esta dada por el producto de:

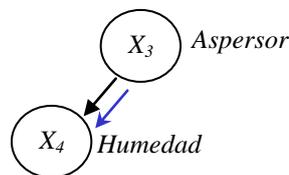
$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \mid \text{padres}(x_i))$$

Para el ejemplo de red bayesiana que hemos venido manejando, la *probabilidad conjunta* es la siguiente:

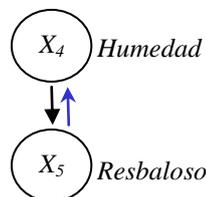
$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1)P(x_4 \mid x_2, x_3)P(x_5 \mid x_4)$$

El cálculo de la probabilidad *a posteriori* de una variable es utilizado para realizar *razonamiento probabilístico* o *inferencia* en una RB. El proceso de *razonamiento* o *inferencia* puede funcionar propagando la información, es decir la(s) evidencia(s), en cualquier dirección. De acuerdo a Pearl y Russell (Pearl & Russell, 2000), existen los siguientes tres tipos de inferencia:

- **Predicción:** este tipo de inferencia se realiza desde la causa a los efectos, es decir, en la dirección de las aristas. Por ejemplo, si el *Aspersor* está encendido, entonces probablemente el pavimento este *húmedo*.

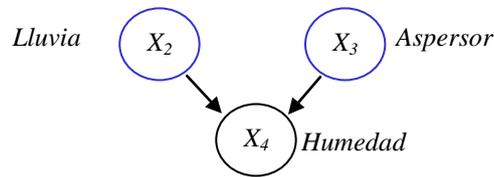


- **Abducción o Diagnóstico:** este tipo de inferencia va desde los efectos a las causas, es decir, en el sentido opuesto al de las aristas. Por ejemplo, si alguien resbala en el pavimento proporciona evidencia de que éste está húmedo.



- **Intercausal:** este tipo de inferencia determina el impacto cualitativo de la evidencia para una variable X_1 sobre otra variable X_2 , cuando ambas son antecesoras de una tercera variable X_3 sobre la que existe evidencia independiente. Por ejemplo, si sabemos que hay humedad, la explicación en este

caso es que la *Lluvia* y el *Aspersor* generan *Humedad*, y si se observa que el *Aspersor* está encendido, esto reduce la probabilidad de que este lloviendo.



Debido al tipo de problemática que se abordó en éste proyecto de tesis, se decidió utilizar el tipo de *inferencia abductiva* en el proceso del *diagnóstico de las competencias*.

Algoritmos para la propagación de evidencia

Existen varios métodos computacionales que aprovechan la estructura gráfica de la red bayesiana, para *propagar* los efectos que las observaciones tienen, sobre el resto de las variables de la red. El primer algoritmo propuesto para cálculos probabilísticos en redes bayesianas utiliza una arquitectura de paso de mensajes distribuidos (Kim & Pearl, 1983; Pearl, 1982). Inicialmente este enfoque estuvo limitado a redes con estructura de árbol, pero fue extendido más tarde a redes generales en el método llamado *Propagación en Árbol de unión* propuesto por Lauritzen y Spiegelhalter en 1988 (Pearl & Russell, 2000). Las principales diferencias entre los distintos algoritmos se basan principalmente en la precisión de los resultados y en el consumo de recursos durante el tiempo de ejecución (Felgaer, 2005).

Los algoritmos de inferencia, de acuerdo a la precisión de sus cálculos, se dividen en *exactos* y *aproximados*, en la Tabla 13 se mencionan algunos de estos algoritmos (Pearl & Russell, 2000; Russell & Norvig, 2004).

Tabla 13. Algoritmos de inferencia

ALGORITMOS DE INFERENCIA	
Exactos	Aproximados
<ul style="list-style-type: none"> • Paso de mensaje • Propagación en árboles de unión • Cycle-cutset conditioning • Eliminación de variables • Enumeración 	<ul style="list-style-type: none"> • Monte Carlo para cadenas de Markov(MCCM) • Ponderación de la verosimilitud • Muestreador de Metrópolis-Hastings • Simulación de Monte Carlo. • Métodos de muestreo directo • Método de muestreo por rechazo en RB

Herramientas para construir y realizar inferencia en redes bayesianas

Durante la investigación de este proyecto, se encontraron herramientas que permiten modelar redes bayesianas. Algunas de estas herramientas están elaboradas para uso comercial y otras como proyectos de investigación.

De estas herramientas, el software elegido para modelar la red bayesiana utilizada en el diagnóstico de las competencias, fue el programa *Elvira*¹. Además, se utilizó una versión de prueba del software comercial *AgenaRisk*², para validar que en ambos programas se obtuvieran resultados similares.

2.4.1 Sistema Elvira

El programa *Elvira* está destinado a la edición y evaluación de modelos gráficos probabilísticos, concretamente redes bayesianas y diagramas de influencia. El programa *Elvira* es producto de un proyecto de investigación realizado en España, el cual fue financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT) y el Ministerio de Ciencia y Tecnología, en el que participan investigadores de varias universidades españolas y de otros centros.

Elvira está escrito y compilado en Java, lo cual permite que pueda funcionar en diferentes plataformas y sistemas operativos (Linux, MS-DOS/Windows, Solaris, etc.). Contiene cerca de 115,000 líneas de código y está públicamente disponible en Internet, tanto su versión compilada como su código fuente.

Elvira cuenta con un formato propio para la codificación de los modelos, un lector-intérprete para los modelos codificados, una interfaz gráfica para la construcción de redes bayesiana, algunos algoritmos exactos y aproximados de razonamiento tanto para variables discretas como continuas, métodos de explicación del razonamiento, algoritmos de toma de decisiones, aprendizaje de modelos a partir de bases de datos, fusión de redes, etc.

Algunas de las principales clases de *Elvira* son las siguientes:

- Clases que implementan métodos de propagación por el método de eliminación de variables en redes Bayesianas y diagramas de influencia (*elvira.inference.elimination*).

¹ <http://www.ia.uned.es/~elvira/>

² <http://www.agenarisk.com/>

- Clases para almacenar en memoria un árbol de grupos o cliques, así como métodos de triangulación necesarios para construirlo (*elvira.inference.clustering*).
- Clases que implementan métodos aproximados de propagación en redes Bayesianas mediante algoritmos de MonteCarlo (*elvira.inference.approximate*).

En particular, para el desarrollo de este proyecto de tesis, se utilizó la clase *VariableElimination*, la cual realiza la propagación en redes bayesianas mediante el método exacto *Eliminación de variables*.

2.5 Educación Basada en Competencias

La Educación por Competencias, en el marco de la formación, pretende ser un enfoque integral que busca vincular el sector educativo con el productivo y elevar el potencial de los individuos, enfrentando las transformaciones que sufren el mundo y la sociedad actual (Cejas, 2004).

Bajo éste enfoque, es necesario realizar cambios metodológicos, didácticos, actitudinales y curriculares en la formación de profesionales, con el fin de promover la participación y cooperación del alumno, apostando por un estudiante que aprenda a aprender, con una actitud crítica y capacidad de responder y actuar ante el cambio (Larraín & González, 2005).

En comparación con los modelos de diseño curricular anteriores, el enfoque por competencias ofrece claras ventajas (Andrade, 2005), tales como:

- Vincular la educación y el mercado laboral. Sin descuidar una educación integral que abarca los aspectos cognitivos, procedimentales y actitudinales.
- Las fuentes de aprendizaje son múltiples, no se reducen al aula y al trabajo con el profesor en clase.
- Estimula la actualización continua de los programas educativos, para poder responder a las necesidades reales de la sociedad globalizada y del avance de la ciencia y la tecnología.

Actualmente, en México está cobrando gran auge el desarrollo curricular por competencias, tal es el caso del ITCM que adoptó el enfoque basado en competencias en sus programas de enseñanza a partir del año 2010.

El concepto de competencia es diverso, pero en este trabajo de tesis consideramos el propuesto por el *Proyecto Tuning Educational Structure in Europe* que define competencia como: una combinación dinámica de atributos, en relación con conocimientos, habilidades, actitudes y responsabilidades, que describen el nivel o grado de suficiencia con que una persona es capaz de desempeñarlos como producto final de un proceso educativo [<http://www.unideusto.org/tuning/>].

El proyecto *Tuning*, como se le conoce actualmente, es uno de los proyectos más importantes en la educación superior en Europa, tuvo sus inicios en el año 1998 y se ha ido expandiendo hacia América Latina y México. Este proyecto surge en un contexto de reflexión sobre la educación superior ante los acelerados cambios de la sociedad (Ramírez & Medina, 2011). El proyecto *Tuning* clasifica a las competencias en *específicas* y *genéricas*.

Las *competencias genéricas* se refieren a atributos generales de los sujetos y que pueden ser manifestadas en sus diferentes desempeños en diferentes contextos. Estas a su vez se clasifican en los siguientes tres grupos: instrumentales, interpersonales y sistémicas.

Las *competencias específicas* se relacionan con el área de conocimiento específico de un campo de formación en particular, y tienen que ver con los atributos cognitivos que deben desplegar los estudiantes. Son de gran importancia por que están específicamente relacionadas con el conocimiento concreto de un área temática.

Tanto las *competencias genéricas* como *específicas*, se adquieren a través de distintas unidades del programa, por lo que pueden estar relacionadas no sólo con una unidad, sino con más unidades de estudio. Por tanto, es necesario identificar las unidades que abordan las distintas competencias para asegurar su evaluación y el cumplimiento de los criterios de calidad (Ramírez & Medina, 2011).

Para llevar acabo el proceso de diagnóstico de las competencias, se creó una red bayesiana, en la cual, cada *competencia específica* de cada unidad, está representada por

un nodo, así como también cada tema de cada unidad, representa un nodo. Y las relaciones causales entre ellos se definió de acuerdo a nuestro criterio, y en dirección de los nodos competencias hacía los nodos temas. El diseño de la red bayesiana será detallado más adelante.

A continuación se presenta un ejemplo del formato de *Instrumentación didáctica para la formación y desarrollo de competencias*, en donde se describen las actividades de aprendizaje y de enseñanza a realizar en una unidad de la asignatura de Fundamentos de Programación del ITCM para el desarrollo de competencias. En éste esquema se especifican las *competencias específicas* y *genéricas* que el alumno debe desarrollar. Para la implementación de éste proyecto de tesis, se trabajó sólo con las *competencias específicas*

Unidad: 5 Tema: Arreglos

Competencia específica de la unidad

Construir programas que utilicen arreglos unidimensionales y multidimensionales para solucionar problemas.

Criterios de evaluación de la Unidad

40 % examen teórico
 30% prácticas
 10% participaciones
 10% tareas
 10% portafolio de evidencias

Actividades de aprendizaje	Actividades de enseñanza	Desarrollo de competencias genéricas	Horas teórico-prácticas
Investigar las características de los arreglos, así como sus operaciones básicas. Buscar cómo se declaran, leen y escriben los arreglos. Resolver problemas utilizando arreglos unidimensionales. Resolver problemas utilizando arreglos bidimensionales	Dar una breve introducción a los tipos de arreglos. Mostrar en el pizarrón las operaciones básicas de los arreglos. Aplicar arreglos unidimensionales en la solución de un problema. Mostrar un ejemplo de implementación de las operaciones básicas de los arreglos bidimensionales. Proponer problemas.	<p>Competencias instrumentales Capacidad de comprender y manipular ideas y pensamientos. Conocimientos básicos de la carrera. Capacidad para organizar el tiempo. Destreza para la búsqueda y manejo de la información. Comunicación oral y escrita Conocimiento de una segunda lengua. Capacidad de análisis y síntesis. Solución de problemas</p> <p>Competencias interpersonales Trabajo en equipo. Habilidades interpersonales.</p> <p>Competencias sistémicas Capacidad de aplicar los conocimientos en la práctica. Capacidad de aprender.</p>	
Resolver examen de la Unidad 5			

Capítulo 3

ESTADO DEL ARTE

3.1 Trabajos que siguen la arquitectura de los STI

A continuación se describen algunos trabajos encontrados en la literatura que consideran la arquitectura básica de un STI.

3.1.1 ActiveMath

La investigación realizada por Melis y Siekmann (Melis & Siekmann, 2004), es acerca de ActiveMath, un sistema de tutoría inteligente basado en Web para la asignatura de Matemáticas. Posee reglas pedagógicas, contenido educativo y el conocimiento del usuario almacenado en el modelo del estudiante. La arquitectura de ActiveMath se muestra en la Figura 5.

Cuando el usuario elige los conceptos a estudiar y el escenario de aprendizaje, se envía una solicitud al Generador de Curso, el cual es el responsable de elegir y organizar el contenido por aprender. Éste accede a la base de conocimientos con el fin de obtener los identificadores (ID) de los conceptos matemáticos requeridos, para la comprensión de los conceptos solicitados. Consulta al modelo del estudiante con el fin de obtener información sobre el conocimiento previo y las preferencias del usuario. También hace uso de las reglas pedagógicas para seleccionar, anotar y organizar el contenido (incluyendo ejemplos y ejercicios) de una manera conveniente para el usuario. La lista de IDs, se envía al motor de presentación, el cual recupera el contenido real matemático correspondiente a los IDs y transforma los datos en páginas de salida presentadas al usuario a través de un navegador. El Generador de Curso y el mecanismo de sugerencias trabaja con el sistema basado en reglas Jess, el cual evalúa las reglas pedagógicas para decidir qué adaptaciones y qué contenido seleccionar y cuales acciones sugerir.

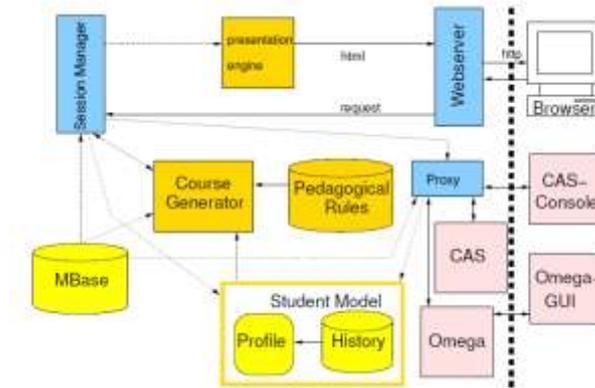


Figura 5. Arquitectura de ActiveMath

3.1.2 Sistema Tutor Inteligente basado en Web, BITS

La investigación que presenta Butz y Maguire (Butz *et al*, 2006), es acerca de BITS, el cual es un STI que tiene el propósito de apoyar a los estudiantes en el aprendizaje de programación de computadoras. El proceso de llevar a cabo la toma de decisiones en el sistema inteligente es guiado por una red bayesiana. Esto viene a solucionar el problema de *Diagnóstico*, que es una de las cuestiones fundamentales en el problema del modelado del alumno. BITS puede ayudar al estudiante en la navegación a través de los materiales en línea.

BITS puede recomendar metas de aprendizaje, y generar secuencias apropiadas de lectura. Por ejemplo, un estudiante puede querer aprender el tema "Archivos de E/S" sin tener que aprender todos los conceptos de los temas anteriores. BITS puede determinar el requisito mínimo de conocimiento necesario para entender el tema elegido y mostrar los vínculos de estos conceptos en el orden correcto de aprendizaje. La arquitectura de BITS se ilustra en la Figura 6.

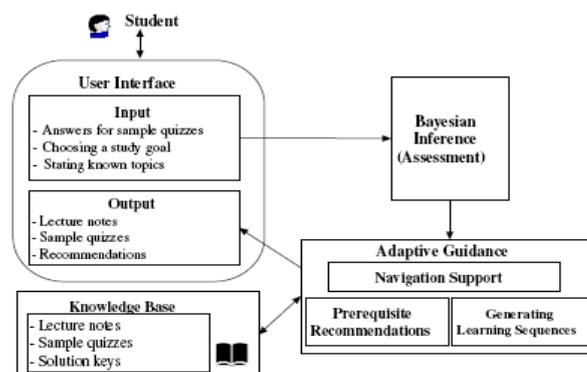


Figura 6. Arquitectura de BITS

3.1.3 Tutor Inteligente para aprendizaje basado en ejemplos

En el trabajo de Conati (Conati, 2009) presenta el STI llamado ExBL, un tutor inteligente diseñado para dar apoyo en la solución de problemas de física, está constituido principalmente por dos componentes. El *primer* componente es conocido como SE (Self-Explanation: Auto-Explicación), el cual apoya a los estudiante con ejemplos antes de la resolución de los problemas. Genera la solución de los ejemplos con diferentes niveles de detalle y ayuda a los estudiantes a generar el paso faltante de la solución. El *segundo* componente, es conocido como el EA (Example-Analogy: Ejemplo-Analogía), el cual ayuda al estudiante con ejemplos durante la resolución de problemas. Selecciona un ejemplo que ayude a solucionar satisfactoriamente el problema actual y maximice el aprendizaje del estudiante. Esto es realizado, calculando la probabilidad de que el ejemplo pueda ayudar al estudiante a resolver el problema y aprender en el proceso.

ExBL se adapta las necesidades de los estudiantes, supervisa y evalúa el desempeño de cada estudiante con respecto al objetivo pedagógico perseguido. Posee una representación interna de las tareas para comparar las soluciones del alumno y del ejemplo. Incorpora en el modelo del estudiante la evaluación de los conocimientos del estudiante y sus niveles cognitivos. En la Figura 7, se muestra la arquitectura de ExBL.

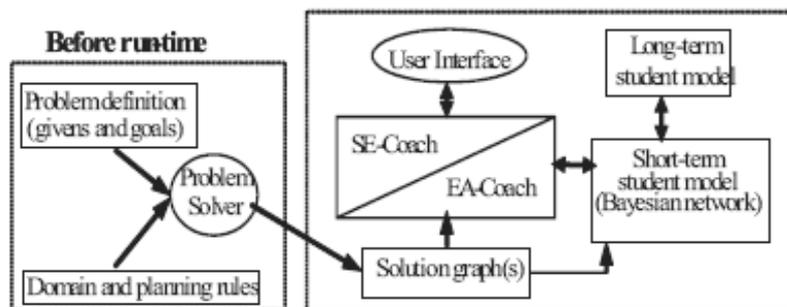


Figura 7. Arquitectura de ExBL.

3.2 Trabajos que consideran Estilos de Aprendizaje

En esta sección se describen algunos trabajos encontrados en la literatura que consideran en su arquitectura algún modelo de estilo de aprendizaje.

3.2.1 Diseño de un Sistema Tutor Inteligente basado en estilos cognitivos

El trabajo que presenta Caviedes (Caviedes, Medina, & García, 2009) se enfoca en el uso de los estilos cognitivos de la psicología para el diseño de un Sistema Tutor Inteligente (STI), caracterizado principalmente por tener la capacidad de adaptarse a las necesidades del estudiante, mostrando la información de forma apropiada según el perfil del usuario y las metas del curso.

Otro de sus aportes es el planteamiento de como se pueden fusionar la tecnologías de los sistemas multi-agentes y la Web semántica para potenciar las características de un Sistema Tutor Inteligente.

Para el diseño del agente de software inteligente que maneja la estructura pedagógica del Entorno Virtual de Aprendizaje (EVA) inteligente, decidieron utilizar el modelo de Felder y Silverman (FSLSM), ya que mencionan que está diseñado específicamente para la utilización en cursos de ingeniería. En 1999 fue utilizado para implementar un curso virtual, en 2004 fue utilizado como parte de una tesis doctoral, y con esto demuestran que el modelo funciona y puede ser utilizado exitosamente en el desarrollo de un EVA.

En la Figura 8 se muestra la arquitectura del sistema propuesto por (Caviedes *et al.*, 2009).

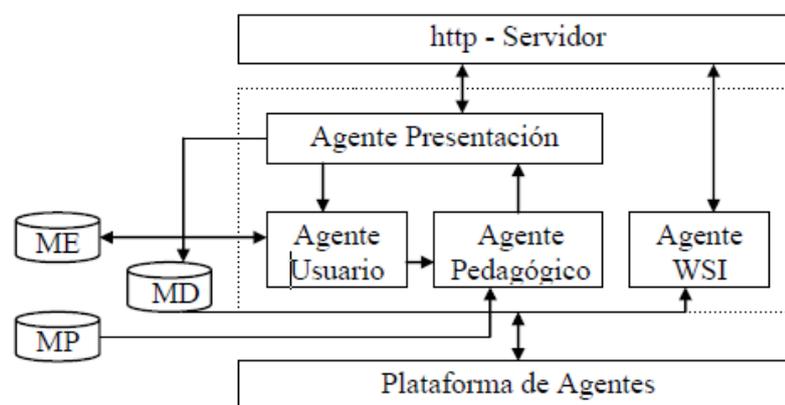


Figura 8. Arquitectura de STI-SWS

3.2.2 Learning styles, Personalisation and Adaptable e-learning

El trabajo presentado por (Peter, Bacon, & Dastbaz, 2009) describe el diseño de una plataforma para e-learning llamada *iLearn*, que utiliza el estilo de aprendizaje VARK, para mejorar la personalización y la adaptación de la plataforma hacia el estudiante.

La plataforma iLearn

iLearn es un sistema basado en una ontología que facilita al estudiante, basado en sus necesidades pedagógicas, recursos de aprendizaje mediante un *paquete e-learning personalizado*. El paquete e-learning estará integrado por recursos digitales que puede incluir: texto, vídeo, audio o podcast. Los recursos digitales pueden ser tanto un *objeto de aprendizaje* como un *objeto de evaluación*.

En el diseño de la plataforma utilizaron tecnologías de la Web semántica, tales como XML, RDF y un lenguaje ontológico OWL. Estas tecnologías añadirían intención y razonamiento al *motor de procesamiento semántico*. El *motor de procesamiento semántico* administra y genera la personalización para cada alumno específico. Este motor será capaz de interactuar con el alumno para determinar su estilo de aprendizaje y sus preferencias en cuanto a los temas, y también interactuará con los *objetos de aprendizaje*, para entonces desarrollar un *paquete e-learning personalizado* diseñado en torno a los requerimientos específicos del estudiante.

iLearn contiene un *perfil del estudiante* el cual está integrado por una variedad de información *específica del estudiante* tales como: el tipo de alumno basado en el estilo de aprendizaje, la meta actual de aprendizaje del alumno y el comportamiento del alumno (incluyendo qué tipo de material está viendo y cómo).

En la Figura 9 se muestra a detalle el *motor de procesamiento semántico* de *iLearn*, donde se puede observar claramente que el motor es el corazón del sistema y que éste interactúa con los *objetos de aprendizaje*, el usuario, el *perfil del estudiante* (contiene el tipo de estilo de aprendizaje) para entonces generar el *paquete e-learning personalizado*.

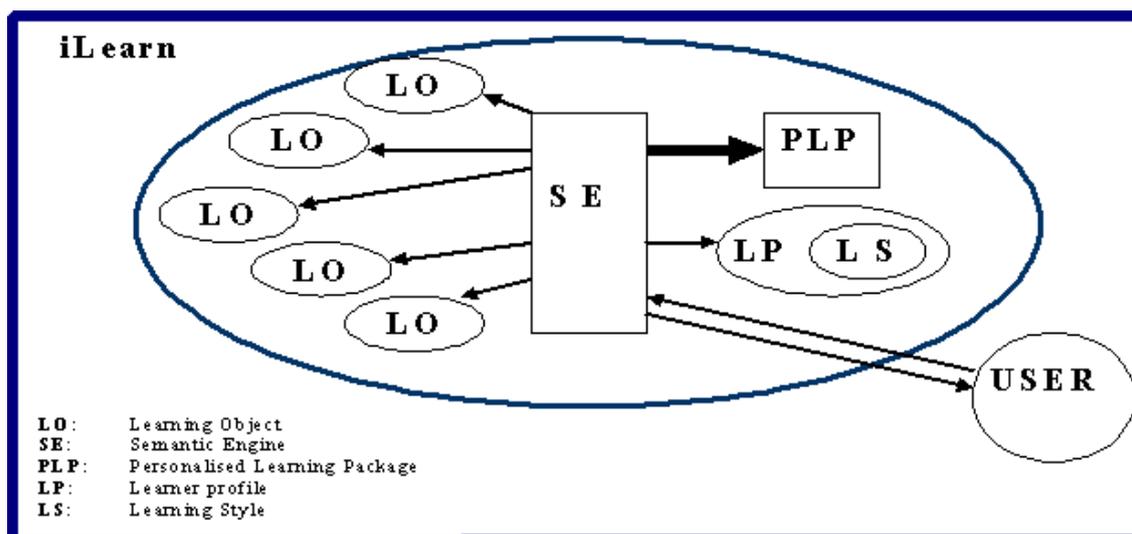


Figura 9. El motor de procesamiento semántico de *iLearn*

El estilo de aprendizaje

Evaluaron distintos modelos de estilos de aprendizaje, de los cuales decidieron trabajar con el modelo VARK. Fue elegido debido a que consideraron que es la herramienta más concisa y que tiene las preguntas más relevantes, además de que éste modelo en comparación con otros, tiene claramente definido el tipo de material de aprendizaje para cada uno de los estilos. El estilo de aprendizaje VARK es utilizado para proveer el material relevante para el alumno, basado sobre su tipo de aprendizaje, en la Tabla 14 se muestra como es utilizado y representado dentro de *iLearn*.

Tabla 14. Representación del estilo de aprendizaje VARK dentro de *iLearn*

ESTILO DE APRENDIZAJE VARK	RECOMENDACIONES DE FLEMING SOBRE ESTRATEGIAS DE ENSEÑANZA	OBJETOS DE APRENDIZAJE A SER PRESENTADOS EN ILEARN
Visual	Imágenes, videos, carteles, diapositivas, diagramas de flujo, gráficas, diagramas.	Video Vodcast Diapositivas
Auditivo	Debatir temas e ideas.	Diapositivas con audio Multimedia Podcast
Lectura/Escritura	Listas, encabezados, diccionarios, definiciones, libros de texto, manuales.	Diapositivas Documentos de texto
Kinestesico	Interesado en hacer, práctico, real, relevante.	Multimedia
Multimodal	Mezcla de los estilos y objetos de aprendizaje anteriores.	

Diseño de iLearn y de un framework semántico

Durante la *primera etapa* de diseño de *iLearn* desarrollaron las clases y sus relaciones. Una de las clases principales es la clase *Persona*, la cual tiene las subclases *Instructor* y *Alumno* que representan a los usuarios del sistema. El *Instructor* puede añadir y administrar los objetos de aprendizaje. El *Alumno* puede generar el paquete de aprendizaje y tendrá asignado un *perfil de estudiante*, para guardar detalles específicos de sus propios requerimientos (incluyendo su estilo de aprendizaje).

Para que el alumno pueda generar el paquete de aprendizaje, le serán presentados grupos de objetos de aprendizaje, que incluyen *objetos de aprendizaje* y *objetos de evaluación*, de los cuales el alumno podrá seleccionar los que vayan de acuerdo a sus necesidades pedagógicas, creando con esto el *paquete e-learning personalizado*.

Ejemplifican este proceso considerando que el alumno desea aprender *programación básica en java* y que tiene un estilo de aprendizaje *visual*. Por lo que, a partir de esas especificaciones, el motor de procesamiento buscará en todos los grupos de objetos de aprendizaje de java (materiales de aprendizaje y de evaluación) los que cumplan con el estilo de aprendizaje del alumno para que le sean mostrados al alumno, en este caso serán archivos de tipo video, vodcast o diapositivas. El alumno entonces, podrá seleccionar de esos objetos, los que sean de su preferencia, formando así el *paquete e-learning personalizado* con el que interactuará.

La *siguiente etapa* de diseño de la plataforma fue realizar las reglas semánticas para el sistema. Estas reglas se desarrollaron para permitir que las reglas de inferencia fueran utilizadas para formar el framework semántico del sistema.

En el diagrama de flujo de la Figura 10, se puede apreciar la interacción entre el alumno, el *perfil del estudiante* y el *motor de procesamiento semántico*. El proceso es el siguiente: inicia cuando el alumno contesta el cuestionario para detectar su estilo de aprendizaje, el resultado obtenido es almacenado en el *perfil del estudiante*, donde el alumno también podrá añadir alguna información específica. Posteriormente el alumno solicita el paquete de aprendizaje personalizado, para lo cual, el *motor de procesamiento semántico* procesa las reglas para proporcionar al alumno una selección de grupos de objetos de aprendizaje. El alumno procederá a elegir los grupos de objetos de

aprendizaje que satisfagan sus necesidades. Una vez que los haya elegido estos serán procesados para formar el *paquete e-learning personalizado* el cual será presentado y utilizado por el alumno.

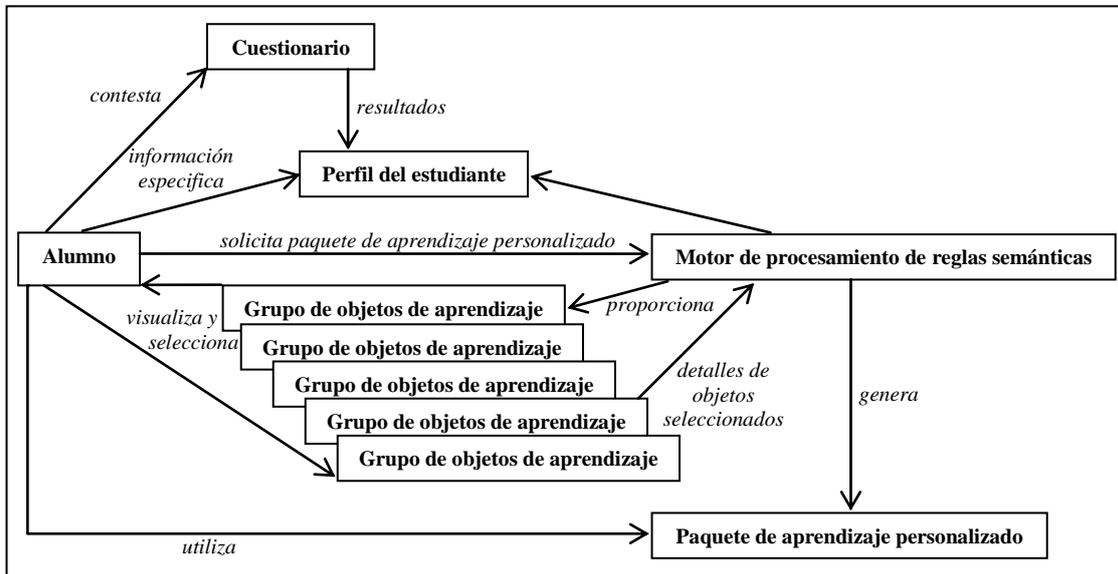


Figura 10. Diagrama de flujo de la interacción del alumno en *iLearn*

3.2.3 Modelado del Estudiante en Sistemas Tutores Inteligentes

La investigación realizada por Cataldi (Cataldi & Lage, 2010), tuvo como finalidad, de acuerdo a la estructura clásica de los STI, redefinir los componentes básicos y las interfaces que posee el Módulo del Estudiante para llevar a cabo la tarea de representar el estado de conocimiento del estudiante real en forma efectiva, es decir con funcionalidades independientes.

De acuerdo a lo planteado, se deben considerar sólo las características propias del sujeto relativas a sus aprendizajes que quedan delimitadas al tener en cuenta: los estilos de aprendizaje y el perfil psico-sociológico.

Propuso que para una enseñanza centrada en los Estilos de Aprendizaje, se puede aplicar el modelo de Felder y Silverman, y en el perfil psico-sociológico se puede aplicar los tipos de inteligencias que el estudiante posee más desarrolladas de acuerdo con la teoría de las Inteligencias Múltiples de Gardner.

En la Figura 11 se puede observar la redefinición del módulo del estudiante, el cual contiene dos grandes sub-módulos y cuyas características se resumen a continuación:

a) Módulos de estilos de aprendizaje y perfil psico-sociológico.

Está compuesto por una base de datos de Estilos de Aprendizaje, los métodos de selección y las características de cada estudiante.

b) Módulo de estado de conocimientos.

Este contiene el mapa de conocimientos obtenido inicialmente respecto del módulo del dominio, que progresivamente irá modificando el actualizador de conocimientos, a través de los resultados obtenidos de las evaluaciones efectuadas por el módulo del tutor, quien enviará dichos resultados procesados. Estos datos también se proporcionaran al módulo tutor para que éste pueda decidir cómo y qué temas impartir en cada una de las sesiones con el estudiante. La idea de este módulo es contener una representación del estado de conocimientos instantáneo del estudiante. Deberá ser modificado por sesiones tuteladas fuera del sistema, como pueden ser las clases particulares o magistrales con tutores humanos.

En la Figura 11 se pueden apreciar estos sub-módulos, los cuales interactúan entre sí y con el resto de los módulos para componer el módulo del estudiante de un STI.

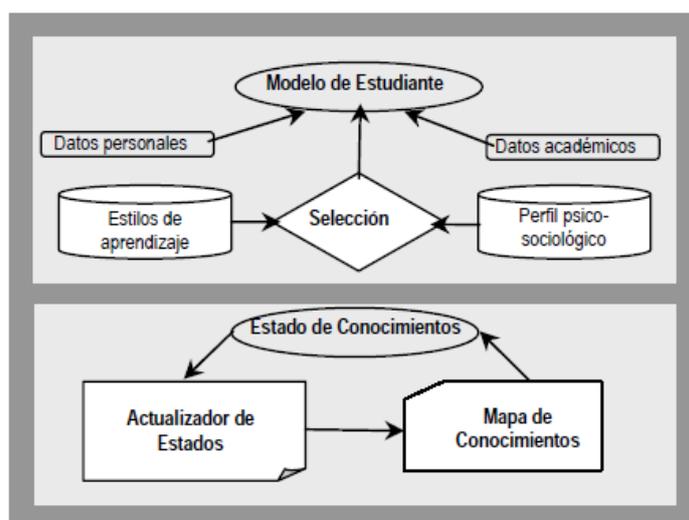


Figura 11. Componentes básicos del Módulo del Estudiante

3.2.4 Integrando Estilos de Aprendizaje y Comportamiento afectivo en un Ambiente de Aprendizaje Inteligente

En el trabajo de Hernández (Hernández, Rodríguez, & Arroyo, 2010) se describe un ambiente de aprendizaje inteligente personalizado que selecciona los objetivos de aprendizaje en base al conocimiento del alumno.

Presentan el desarrollo de un modelo que incluye conocimiento, afecto y estilos de aprendizaje de los alumnos. En la

Figura 12 se presenta el diagrama del modelo.

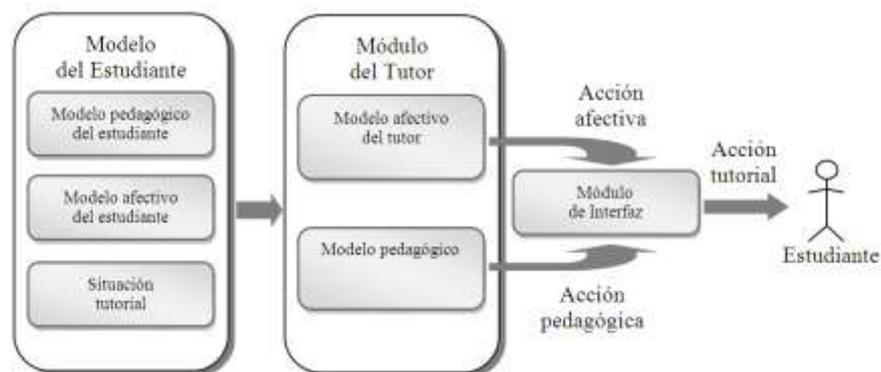


Figura 12. Diagrama general del modelo de comportamiento afectivo

El modelo del estudiante consta de una red bayesiana dinámica (DBN: Dynamic Bayesian Network) que probabilísticamente relaciona la personalidad del estudiante, los objetivos y los eventos con los estados afectivos de los alumnos.

El proceso de decisión está representado por una Red de Decisión Dinámica (DDN: Dynamic Decision Network). El DBN incluido en el modelo del DDN es utilizado para predecir como las acciones tutoriales disponibles influyen y afectan en el conocimiento del alumno, dado su estado actual.

Para incorporar estilos de aprendizaje en la adaptación lógica de sistemas tutoriales, se requiere un enfoque para identificar la mejor acción tutorial dado el estilo de aprendizaje del estudiante. Para esto se basaron en el modelo de Felder-Silverman.

3.2.5 A Web-Based Intelligent Tutoring System on teaching and learning Electrical Project

En el trabajo realizado por (Gomes, Miranda, Vale, & Faria, 2010) presentan un STI el cual tiene como objetivo, ser una herramienta de apoyo para la enseñanza y aprendizaje en el campo de la Ingeniería Eléctrica.

Para realizar el proyecto utilizaron como base la plataforma AHA!, que es un sistema hipermedia adaptativo de código abierto desarrollado en la Universidad Tecnológica de Eindhoven. La plataforma AHA! tiene como objetivo facilitar el diseño de aplicaciones Web como hipermedia adaptativo y cuenta con algunas herramientas de creación que ayudan al desarrollo de procesos.

Dentro de las funciones que el sistema pretende desarrollar se encuentran las siguientes:

- Proporcionar un curso interactivo que vaya dirigido hacia dos tipos de usuarios: estudiantes y especialistas, ambos dentro del campo de diseño de instalaciones eléctricas.
- Aplicar una prueba a los usuarios, para identificar su estilo de aprendizaje.
- Ajustar el nivel de detalle de los contenidos al nivel del conocimiento del usuario.
- Adaptar el contenido de acuerdo al estilo de aprendizaje del usuario, y si el usuario no entiende bien los conceptos (es decir cuando falla la validación del test de aprendizaje), el sistema debe ser capaz de mostrar el contenido en otro formato.
- Permitir a los autores de contenidos consultar los hábitos de uso de los usuarios registrados en el sistema.

Procesos diseñados

Para identificar el estilo de aprendizaje de los usuarios del sistema, crearon una prueba de diagnóstico basada en el modelo de estilo de aprendizaje VARK, cuya resolución esta propuesta en la primera interacción del usuario con el sistema.

Debido a que, generalmente los alumnos tienen un nivel de conocimiento bajo acerca del área de instalaciones eléctricas, se implementó en el sistema, una prueba de diagnóstico para evaluar el nivel de conocimiento del estudiante, esto con la intención de no enseñarle conocimiento que ya posee y ofrecerle una educación personalizada. Esta prueba sólo será aplicada a los estudiantes, ya que consideraron que los expertos tienen un alto conocimiento en la materia en comparación con los estudiantes, por lo tanto no requieren de la aplicación de una prueba de diagnóstico.

Realizaron reglas para la navegación y la adaptación de los contenidos de acuerdo al estilo de aprendizaje de los usuarios.

La plataforma AHA! requiere que los contenidos sean páginas con formato XHTML. La creación de éste tipo de páginas es una tarea difícil y requiere de mucho tiempo, lo que hace que la creación y actualización de contenidos sea una tarea complicada, afectando directamente a la calidad del contenido la cual puede influir en el éxito del STI. Dado que no existe una herramienta que permita la creación y edición de contenidos para AHA!, desarrollaron una herramienta cuyas características desde el punto de vista del administrador/autor de contenidos son presentadas en la Figura 13.

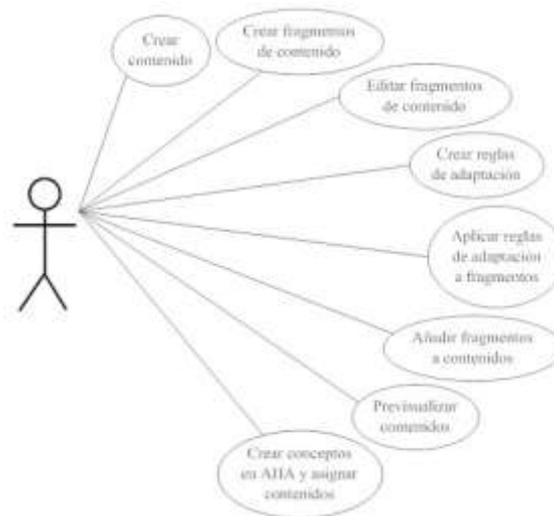


Figura 13. Principales características del editor de contenidos

3.3 Trabajos que consideran Competencias

A continuación se describen algunos trabajos encontrados en la literatura, que consideran en su arquitectura, el concepto de educación basada en competencias.

3.3.1 Arquitectura para un Sistema Tutor Inteligente basado en Competencias (STI-C)

En el trabajo realizado por Badaracco y Martínez (Badaracco & Martínez, 2010), proponen una nueva arquitectura de STI cuyo modelo pedagógico está basado en la Enseñanza Basada en Competencias (EBC). Se centran en la representación del modelo de dominio y del modelo del alumno, así como en el proceso de diagnóstico acorde con la EBC. En la Figura 14 se presenta la arquitectura propuesta del STI basado en Competencias.

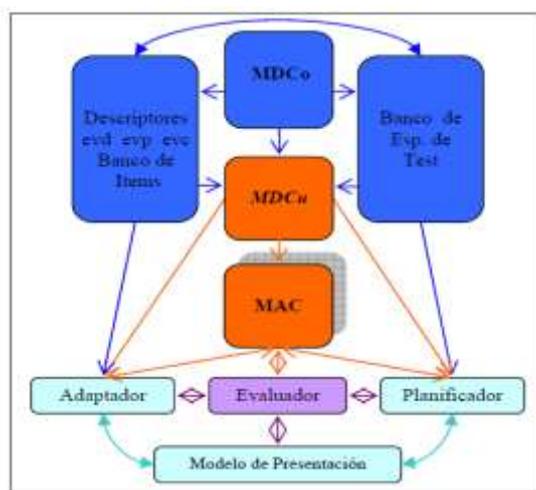


Figura 14. Arquitectura de STI-C

La arquitectura contiene los siguientes componentes:

Modelo de dominio: Contiene el perfil de competencias definido en un dominio de conocimiento. Se compone de cuatro componentes:

- Modelo de dominio de competencia (*MDCo*), derivado del mapa funcional de las normas de competencias, se representa mediante una red semántica que incluye unidades de competencia, elementos de competencia (*ec*), descriptores y las relaciones entre estos.

- Modelo de dominio curricular (*MDCu*) organizado en módulos didácticos M_i que elabora el profesor en base al *DCBC* (Diseño Curricular Basado en Competencias) sobre el *MDCo*.
- Un conjunto de descriptores que incluyen las evidencias de producto (*evp*), las evidencias de desempeño (*evd*) y las evidencias de conocimiento (*evc*). Cada uno tiene asociado un banco de ítems.
- Un banco de especificaciones de test asociados a los ec_i (elementos de competencia íésimo), definidos por el docente.

El modelo del alumno de competencia (MAC): Se encarga de almacenar la información referente al alumno y su representación dentro del modelo de diagnóstico.

3.3.2 Intelligent Tutoring System Framework for the Acquisition of Knowledge and Competences

El trabajo realizado por (Vernet, Nicolas, Golobardes, Fornells, & Garcia-Piquer, 2010) es un proyecto en desarrollo, donde presentan un framework para desarrollar un STI basado en competencias. El sistema propone monitorear la adquisición de conocimiento teórico, evaluar las competencias relacionadas con los temas y efectuar acciones correctivas cuando sea necesario para solventar una evolución negativa del estudiante. El framework está dividido en cuatro fases principales basadas en técnicas de inteligencia artificial: construcción, ubicación, predicción y reforzamiento.

Definición de los datos

La adquisición de una competencia es calculada a la par con las áreas de conocimiento, las cuales evalúan las competencias. La calificación final no depende de un tema en particular, si no, de un conjunto de áreas de conocimiento, las cuales evalúan a las competencias. Los resultados obtenidos correlacionan con la adquisición de esas competencias, las cuales forman la calificación final de la asignatura. Un conjunto de competencias es asignado a cada área de conocimiento de la asignatura y un subsecuente conjunto de actividades está definido por cada competencia. Esto permite evaluar el progreso del estudiante quien está adquiriendo estas competencias.

Como se puede ver en la Figura 15, el conocimiento de la asignatura está representado de acuerdo al modelo tradicional de enseñanza, la cual, es la forma más utilizada en la mayoría de los sistemas actuales.

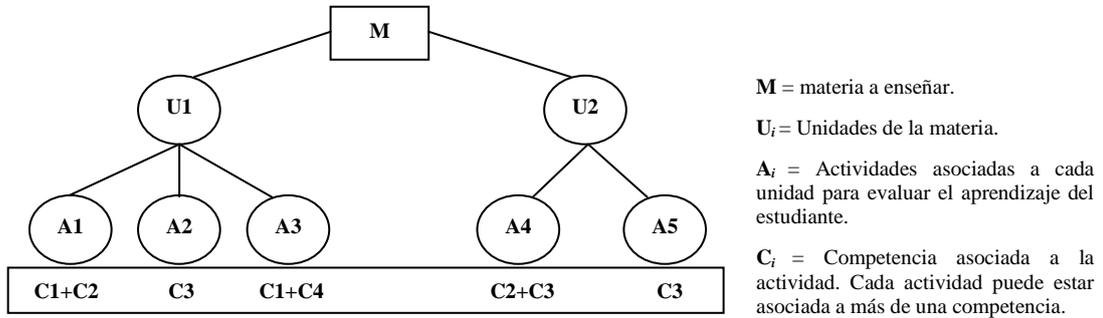


Figura 15. Representación tradicional del árbol del conocimiento

El enfoque de éste trabajo es colocar al modelo de las competencias como el centro del conocimiento a ser enseñado. La Figura 16 muestra como el árbol de conocimiento previamente mencionado, puede ser transformado en el nuevo modelo de enseñanza.

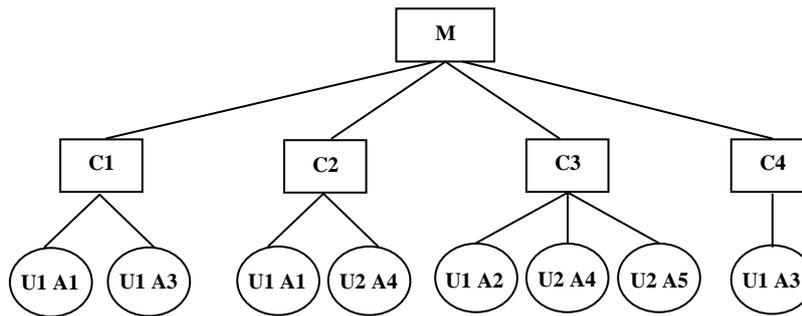


Figura 16. Representación de los datos en el framework

El diagrama muestra como el sistema tutor se enfoca sobre la competencia y como las actividades propuestas durante todo el proceso de aprendizaje del estudiante son usados para optimizar el potencial de cada competencia. Los datos más importantes introducidos al sistema son las calificaciones de las competencias de cada una de las actividades llevadas a cabo.

A continuación se describen cada una de las fases que componen al framework.

Construcción

En ésta fase se construye el árbol del conocimiento de la asignatura a ser enseñada. Los nodos de este árbol constituyen unidades de dominio de conocimiento las cuales al

mismo tiempo son divididas en diferentes actividades. Los maestros diseñan las actividades a resolver, asocian las competencias a ellas y las localizan en el árbol, dependiendo sobre la competencia en cuestión.

Ubicación

En ésta fase el sistema asigna al estudiante un estado inicial de conocimiento y de competencias utilizando un algoritmo automático. Los profesores proponen algunas actividades, una vez que son llevadas a cabo por el alumno, el algoritmo recopila la información de las evaluaciones de las actividades y genera la tupla (competencia-grado obtenido) para cada una de las competencias del curso. Esta evaluación y ubicación debe ser repetida más de una vez en conformidad con el nivel de competencias adquirido y conocimiento en los diferentes campos. En este punto el sistema puede activar su fase predictiva.

Predicción

Esta fase está basada sobre el desempeño del estudiante (calificación de los ejercicios y competencias adquiridas), se encarga de predecir la evolución del estudiante al final de curso utilizando técnicas de Inteligencia Artificial.

Reforzamiento

En esta fase el sistema proporciona al alumno una selección inteligente de actividades las cuales, no sólo mejoran el potencial de aprendizaje de una materia, si no que también desarrollan el árbol de conocimiento y competencias.

3.4 Trabajos implementados sobre la plataforma Moodle

En esta sección se describen algunos trabajos encontrados en la literatura donde utilizan a Moodle como plataforma de implementación.

3.4.1 Personalización de Itinerarios Formativos en Moodle adaptados a los Estilos de Aprendizaje de Honey-Alonso

En el trabajo de (Torres, Castillo, Morales, Jiménez, Llorca & Ortega, 2008) se describe el proyecto realizado por la empresa Conzepto, perteneciente a Grupo TADEL en colaboración con el Departamento de Inteligencia Artificial de la Facultad de

Informática de la Universidad de Granada, donde el objetivo principal del proyecto fue la creación de una herramienta para la presentación personalizada de contenidos a los alumnos, a través de Moodle, dependiendo de dos variables: el estilo de aprendizaje de los alumnos y su disponibilidad de tiempo.

Lo que pretendían lograr, era pasar de una única forma de aprender, a cuatro formas o estilos de aprendizaje diferentes, para alcanzar los objetivos específicos del curso que eran comunes para todos los alumnos.

Para el desarrollo del proyecto, establecieron las siguientes fases:

En la *primera fase* realizaron la elección de una plataforma que se ajustara a sus requerimientos técnicos y metodológicos, para lo cuál decidieron utilizar Moodle 1.9.

En la *segunda fase* realizaron la definición pedagógica del curso (según los 4 estilos de aprendizaje). El estilo de aprendizaje de cada alumno lo detectaron integrando el cuestionario de Honey-Alonso en Moodle, mediante el módulo Questionnaire.

A medida que los alumnos contestaban, el cuestionario registraba los resultados y eran trasladados posteriormente al estilo de aprendizaje determinado para cada alumno. A continuación, se creaban los grupos de alumnos en Moodle (Activos, Reflexivos, Teóricos y Pragmáticos), y se especificaba a qué grupo de estilo de aprendizaje pertenecía cada alumno.

Por otra parte, la variable “disponibilidad de tiempo” la obtuvieron a través de una sencilla consulta, en la que el alumno seleccionaba su disponibilidad.

En la *tercera fase* elaboraron los contenidos en formato HTML-SCORM 1.2, para poder ser integrados en Moodle como paquetes SCORM.

En la *cuarta fase* fueron etiquetados los metadatos siguiendo la especificación IMS/MD-LOM, donde utilizaron la herramienta Reload Editor. Estos metadatos serían los que señalarían al módulo desarrollado, qué contenidos mostrar y cuáles no, dependiendo del estilo de aprendizaje y de la disponibilidad de tiempo del alumno.

En la *quinta fase* se integró y creó el módulo PAIF en la plataforma. La Universidad de Granada integró en Moodle el sistema de planificación inteligente desarrollado, el cual extrae la información de los metadatos de los diferentes contenidos

y los devuelve en forma de contenidos personalizados, a través del módulo SCORM de Moodle.

En la *sexta y última fase* se desarrolló una experiencia piloto. Una vez que el trabajo se realizó, y tras clasificar a los alumnos en su estilo de aprendizaje y disponibilidad de tiempo, los alumnos accedieron a sus contenidos (en formato SCORM), mostrándolos de manera personalizada para cada alumno (según su estilo de aprendizaje y disponibilidad de tiempo).

Además de mostrar los contenidos personalizados a través del módulo SCORM, se presentaban también de forma personalizada para cada tipo de estilo de aprendizaje diferentes recursos, basándose en las posibilidades introducidas por Moodle sobre Grupos y Agrupamientos a partir de su versión 1.9 (las cuales permiten mostrar distintos tipos de recursos para diferentes tipos de grupos y subgrupos).

3.4.2 Modificación del módulo SCORM para el soporte básico del IMS-LD

El trabajo realizado por (Figuerola, Martínez, Morales & Castillo, 2008) surge de un tema ampliamente discutido donde se trata de adoptar el estándar IMS-LD dentro del sistema gestor de Moodle, como su implementación resulta inviable a corto plazo, propusieron una modificación del módulo SCORM para soportar los requerimientos básicos de un IMS Learning Design.

Dicha modificación permitiría adaptar el contenido de una actividad tipo SCORM/AICC, al perfil de cada alumno matriculado en el curso, que utilizara dicha actividad. Los objetos de aprendizaje deberían estar etiquetados en base al estándar IEEE/LOM.

El estándar IMS-LD implica la personalización pedagógica y técnica de los contenidos a cada uno de los estudiantes, además del uso de recursos para su interacción dados los diferentes colectivos y roles existentes.

Por la naturaleza del estándar, mencionaron que ha resultado difícil integrar un visor del mismo dentro de Moodle, pero como existe una colaboración directa entre IMS Global Consortium y Moodle, se ha logrado hasta el momento implementar

algunos visores de objetos de aprendizaje siguiendo los estándares IMS-CP y SCORM, cuyos contenidos son elaborados externamente en herramientas de diseño libres tales como RELOAD o eXe Editor.

Específicamente el módulo encargado del visor de las actividades tipo SCORM, es el que utilizaron para desarrollar su propuesta de integración de un LD generado automáticamente dentro de la plataforma Moodle.

Para la generación automática de un IMS-LD, utilizaron la herramienta PAIF, que les fue proporcionada por la Universidad de Granada.

Dicha herramienta requirió de la siguiente información para poder generar una serie de itinerarios formativos adaptados a las principales características de cada uno de los estudiantes de un curso.

Fue necesario etiquetar cada uno de los componentes de la actividad SCORM de acuerdo a ciertos metadatos que se basan en el estándar IEEE/LOM para su definición. Estos metadatos se pueden dividir en dos tipos.

- Metadatos que representan relaciones de orden y jerarquía: IsPartOf e IsBasedOn.
- Metadatos tipo atributos de los objetos de aprendizaje: Language, Learning Resource Type, Other Platform Requirements, Difficulty, Coverage, y Typical Learning Time.

También fue importante enviar al servidor de planificación la siguiente información del perfil del estudiante que se ha incluido dentro de nuevos campos del mismo: English Level, Equipment, Performance Level, Learning Style y Available Time.

Después de analizar el funcionamiento del módulo SCORM, vieron que la adaptación a los alumnos se podía conseguir reordenando o eliminando los items SCORM a mostrar para cada alumno.

Mencionan que los items SCORM están guardados en la tabla scorm_scoes, en donde se especifica su orden jerárquico y otros datos.

La manera más sencilla que encontraron para realizar la personalización, fue modificar la tabla `scorm_scoes` agregándole el campo `userid`. Este campo les permitió tener ítems ordenados para cada alumno, por lo cual, toda la información de los ítems es repetida tantas veces como el número de alumnos que puedan ver el SCORM.

Una vez que tenían la posibilidad de almacenar ítems ordenados para cada alumno, se presentó la tarea de seleccionarlos cuando el alumno estuviera visualizando el paquete SCORM. Esto lo realizaron modificando todos los accesos y modificaciones a los registros de la tabla `scorm_scoes`, agregando el parámetro `userid` a las sentencias utilizadas.

Experimentación

Mencionan que la versión beta de esta herramienta fue probada en un curso real. En el servidor Moodle de la empresa Conzepto se instaló la modificación del SCORM y ellos se encargaron de etiquetar la actividad SCORM “Conceptos básicos” para el curso de “Microsoft Power Point”, además de probar con el perfil de los alumnos de dicho curso que las actividades se mostraban de acuerdo a sus características individuales.

En base a una encuesta que realizaron a los estudiantes, después de probar las actividades adaptadas, informaron que un 80% de los alumnos piensa que ha aprendido bastante o mucho con esta nueva estrategia de formación que utilizó la empresa y que mostraron un 86% de conformidad, con la planificación temporal que se les dio para su curso, dada la disponibilidad horaria que habían dicho tener en un principio.

Como comentario final mencionan que estas dos estadísticas, dan una idea del éxito inicial que puede tener este símil de LD sobre la plataforma.

3.5 Comparación de Trabajos

A continuación se presenta una tabla comparativa (ver Tabla 15) de todos los trabajos arriba mencionados. En ella se puede apreciar que actualmente los desarrolladores de STI están optando por implementar, dentro de estas plataformas, mecanismos que ayuden a detectar el estilo de aprendizaje de los alumnos. Esto con el fin de utilizarlo para presentar al estudiante los materiales en un formato que vaya de acuerdo a su estilo de aprendizaje.

Otro elemento que se está considerando en la implementación de los STI es, el evaluar las competencias que desarrollan los estudiantes dentro de una asignatura, ya que es el nuevo enfoque de enseñanza que han adoptado las escuelas de nivel superior.

Debido a estas razones, se propuso implementar en éste proyecto de tesis, un STI que considera el estilo de aprendizaje del alumno para personalizar los contenidos a mostrar al alumno, así como también incorporar un proceso que ayude a diagnosticar las competencias que posee el alumno.

Tabla 15. Tabla comparativa del Estado del Arte

Trabajo	Dominio	Propósito	Técnicas aplicadas en la arquitectura de los STIs				Enseñanza por Competencias	Estilos de aprendizaje
			Módulo Interfaz	Módulo Dominio	Módulo Tutor	Módulo Estudiante		
Melis <i>et al.</i> , 2004 (ActiveMath)	Matemáticas	Utiliza reglas pedagógicas para seleccionar, anotar y organizar el contenido conveniente para el usuario.	No especificado	Red semántica	Basado en reglas	Basado en reglas	✗	✗
Butz <i>et al.</i> , 2006 (BITS)	Programación computadoras	Ayudar en la navegación de los materiales en línea.	Submódulo de entrada y de salida	Red bayesiana	Red bayesiana	Red Bayesiana	✗	✗
Conati, 2009 (ExBL)	Física	Apoyar al alumno con ejemplos antes y durante la interacción.	No especificado	Basado en reglas	No especificado	Red Bayesiana	✗	✗
Caviedes <i>et al.</i> , 2009	No especificado	Adaptarse a las necesidades del alumno, mostrando la información según el perfil del usuario	No especificado	Red semántica	Basado en reglas	No especificado	✗	✓ Felder
PETER 2009	Programación	Personalizar el contenido a mostrar al alumno, considerando el estilo de aprendizaje del alumno	No especificado	No especificado	Reglas semánticas	No especificado	✗	✓ VARK
Cataldi, 2010	No especificado	Redefinir el Módulo del Estudiante, para representar el conocimiento en funcionalidades independientes.	No especificado	No especificado	No especificado	No especificado	✗	✓ Felder-Inteligencias Múltiples
Hernández, 2010	No especificado	Proporcionar al estudiante una enseñanza de acuerdo a su estado afectivo y su estado de conocimiento.	No especificado	No especificado	Red dinámica	Red bayesiana	✗	✓ Felder
ARAUJO 2010	Instalaciones eléctricas	Ser una herramienta de apoyo para la enseñanza y aprendizaje en el campo de la Ingeniería Eléctrica	No especificado	Formato XHTML	Reglas	No especificado	✗	✓ VARK
Badaracco <i>et al.</i> , 2010 (STI-C)	No especificado	Proveer al docente de una plataforma que facilite la propuesta de la Educación Basada en Competencias	No especificado	Red semántica	No especificado	Modelo Bayesiano	✓	✗
VERNET 2010	No especificado	Diseñar un framework para implementar un STI basado en competencias	No especificado	Árbol	No especificado	No especificado	✓	✗
Torres, 2008	No especificado	Crear itinerarios formativos automatizados, adaptados a los estilos de aprendizaje de los alumnos	Moodle	No especificado	No especificado	No especificado	✗	✓
Figuroa, 2008	Matemáticas discretas	Adaptar el contenido de una actividad tipo SCORM/AICC, al perfil de cada alumno matriculado en un curso.	Moodle	SCORM	No especificado	No especificado	✗	✓
Este trabajo 2011	Fundamentos de Programación	Mostrar contenidos según el estilo de aprendizaje de los alumnos y diagnosticar las competencias específicas.	Moodle	SCORM Cuestionario	Agente y red bayesiana	BD	✓	✓

Capítulo 4

PROPUESTA DE SOLUCIÓN

En este capítulo se describe la arquitectura propuesta para el desarrollo de éste proyecto. Se describen cada unos de sus módulos y los componentes que lo conforman, así como la función que desempeñan.

4.1 Propuesta de arquitectura

Como se ha mencionado anteriormente, la arquitectura de un STI, está compuesta generalmente por los siguientes cuatro módulos: *tutor*, *estudiante*, *dominio de conocimiento e interfaz*. Entre las funciones que realiza cada módulo se encuentran las siguientes, el módulo *tutor* determina las estrategias pedagógicas de enseñanza y monitorear el desempeño del alumno, el módulo *estudiante* representa el estado cognitivo del alumno, el módulo de *dominio de conocimiento* almacena toda la información referente a la asignatura a enseñar por el STI, y el módulo *interfaz* da soporte a la interacción entre los tres módulos anteriores, así como también se encarga de administrar el proceso mediante el cual, el alumno interactúa con el STI.

El considerar que Moodle personalice los contenidos educativos para cada alumno y diagnostique si el alumno posee las competencias de la asignatura, implicó modificar y añadir funciones a desempeñar por los módulos de la arquitectura clásica de un STI. En la Figura 17 se presenta la arquitectura propuesta en éste trabajo de tesis. En la arquitectura se puede apreciar cada uno de los módulos y los elementos propuestos. La línea remarcada indica los elementos añadidos, con una línea pausada los elementos rediseñados y con una línea punteada los elementos utilizados del Sistema Gestor del Aprendizaje Moodle.

El proceso general llevado a cabo por el STI se puede apreciar mediante el diagrama de secuencia mostrado en la Figura 18.

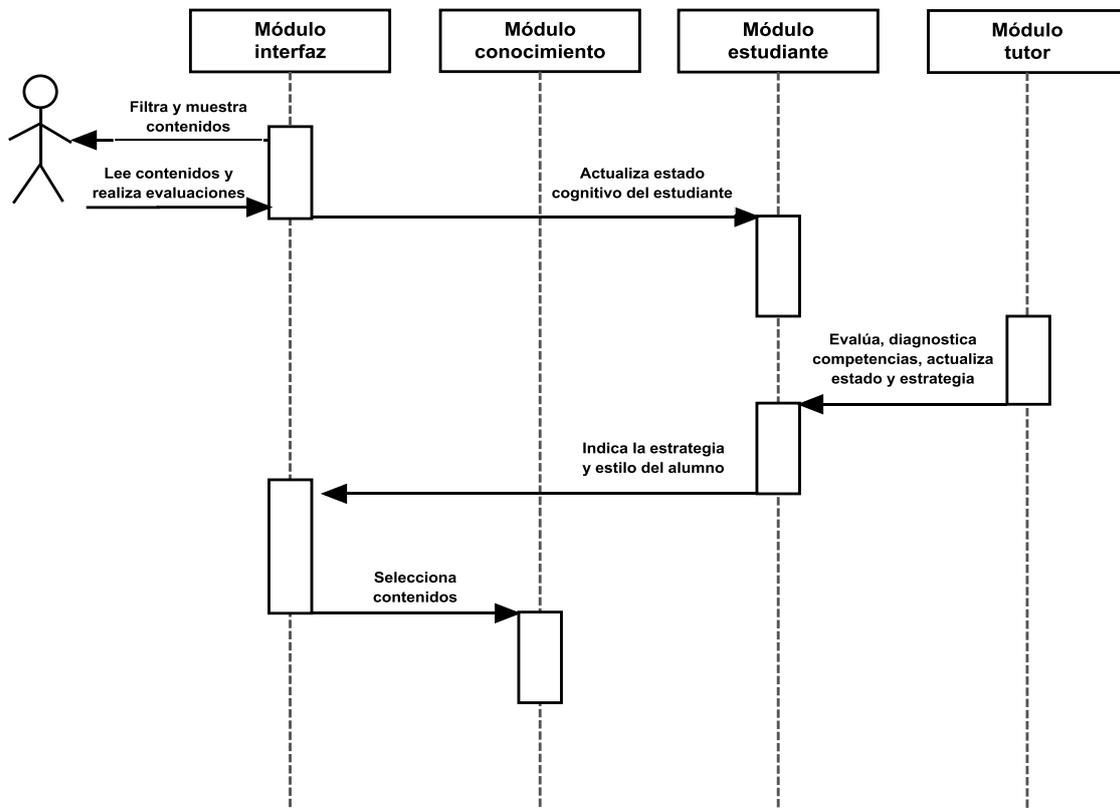


Figura 18. Diagrama de secuencia del STI

A continuación se describen cada uno de los módulos que componen a la arquitectura propuesta para el STI desarrollado en este proyecto, así como los elementos que los componen y las funciones que realizan.

4.2 Módulo Conocimiento

En este módulo se añadieron las siguientes tablas, una tabla para almacenar las competencias específicas de la materia, una tabla para almacenar los estilos de aprendizaje y una tabla para almacenar las estrategias de enseñanza-aprendizaje relacionadas a cada estilo de aprendizaje.

También como elementos de éste módulo, se consideraron las actividades *SCORM* y *Cuestionario* que se encuentran disponibles dentro de Moodle. La actividad *SCORM* fue utilizada para importar a Moodle los contenidos de la asignatura, los cuales están definidos por los paquetes SCORM. Cada paquete representa a un sólo tema de la

asignatura y está formado por varios *objetos de aprendizaje* que se encuentran diseñados en las diferentes estrategias de enseñanza que favorecen a cada uno de los estilos de aprendizaje. También en éste módulo, se lleva acabo el almacenamiento de los exámenes correspondientes a cada uno de los temas de la asignatura, los cuales son editados mediante la actividad de Moodle llamada *Cuestionario*.

Para mostrar tanto el contenido didáctico como los exámenes, éste módulo se comunica con el *módulo interfaz*.

4.3 Módulo Interfaz

Los elementos añadidos a éste módulo y la función que desempeña cada uno, se describe en la Tabla 16. Todos ellos utilizan la interfaz de Moodle para realizar su función.

Tabla 16. Elementos añadidos al módulo interfaz

ELEMENTO	FUNCIÓN
Gestor de estilos de aprendizaje y de estrategias de enseñanza	Permite registrar, eliminar, cambiar o consultar los estilos de aprendizaje del modelo VARK.
Gestor de competencias	Permite registrar, eliminar, cambiar o consultar las competencias de la asignatura y las tablas de probabilidad condicional que forman la red bayesiana.
Cuestionario VARK	Cuando un alumno se registra por primera vez en un curso, le muestra por única ocasión en la primera interacción, el cuestionario VARK, que es utilizado para identificar el estilo de aprendizaje del alumno.
Configurador de contenidos	Permite relacionar cada tema con una o varias competencias específicas y con el examen correspondiente. Así como también relacionar cada tema con el estilo de aprendizaje y la estrategia de enseñanza para el que fue elaborado.

El componente de Moodle encargado de mostrar los contenidos fue rediseñado y nombrado agente interfaz, éste personaliza los contenidos a mostrar, de acuerdo al estilo de aprendizaje y estrategia de enseñanza que favorezcan al alumno. Originalmente mostraba a todos los alumnos los contenidos de la misma forma, es decir, en el mismo formato. Una vez filtrados los contenidos son mostrados al alumno.

Utilizando la interfaz, se editan los exámenes de cada uno de los temas del curso, esto mediante la actividad *Cuestionario* disponible en Moodle.

4.4 Módulo Tutor

En este módulo se añadió un proceso para evaluar el cuestionario VARK una vez que fue contestado por el alumno. Éste proceso calcula la puntuación para cada uno de los cuatro estilos de aprendizaje del modelo. Almacena estos resultados en la tabla donde se encuentran los datos generales de los alumnos. También se encarga de determinar cual es el estilo de aprendizaje predominante del alumno, es decir, el que obtuvo el mayor número de puntos. Una vez que determina el estilo de aprendizaje, éste es almacenado en la misma tabla mencionada anteriormente, definiéndolo como el *estilo de aprendizaje actual*. Posteriormente, elige una estrategia de enseñanza que vaya de acuerdo con el estilo de aprendizaje del alumno, almacenándola y definiéndola como la *estrategia actual*. Este proceso es ejecutado sólo una vez cuando el alumno contesta el cuestionario VARK.

También se añadió un proceso para el diagnóstico de las competencias, el cual entra en función cada vez que un alumno contesta el examen de algún tema, diagnosticando si éste posee las competencias específicas relacionadas con el tema evaluado.

Por último, se añadió un proceso llamado agente selector de estrategias de enseñanza, el cual depende del resultado obtenido por el diagnóstico de las competencias. Cuando un alumno es evaluado y no posee la competencia, el agente selector entrará en función, eligiendo a una de las estrategias de enseñanza disponible, la cual debe ser diferente a la estrategia actual, pero que también se encuentre relacionada con el estilo de aprendizaje del alumno. En el caso de que se hayan utilizado todas las estrategias de enseñanza, el agente selector procederá a cambiar el estilo de aprendizaje actual del alumno, este proceso se detallará en la sección 6.3.1.

Para llevar a cabo la evaluación de los exámenes, éste módulo utilizó el proceso realizado en Moodle.

4.5 Módulo estudiante

En este módulo se utilizaron las tablas originales de Moodle. En la tabla donde se guardan los datos generales de cada alumno, se realiza también el almacenamiento de las puntuaciones de cada uno de los estilos de aprendizaje, así como del estilo de

aprendizaje y de la estrategia de enseñanza, considerados como las actuales del alumno. Estos dos últimos datos son asignados inicialmente cuando el alumno contesta el cuestionario VARK como se mencionó arriba, posteriormente son actualizados por el agente selector ubicado en el *módulo tutor*.

También utiliza las tablas de Moodle, para almacenar las calificaciones obtenidas por los alumnos al contestar algún examen.

Capítulo 5

ANÁLISIS Y CONFIGURACIÓN DE MOODLE

En este capítulo se describen algunos de los componentes principales de Moodle, el cómo se utilizó ciertas actividades permitidas por la plataforma, así como también se describen las tablas y archivos que se añadieron a la funcionalidad de Moodle. Todo esto, con el propósito de configurar a la plataforma, con los elementos necesarios para que el agente selector de estrategias de enseñanza y el proceso para el diagnóstico de las competencias, puedan desempeñarse de una forma adecuada.

5.1 Análisis de los componentes de Moodle

La versión de Moodle que se instaló localmente y con la que se trabajó es la 1.9.12, la cual está compuesta por más de 2,000 archivos PHP, 1,138 archivos HTML y una base de datos de MySQL que contiene 212 tablas. Los archivos PHP y HTML se encuentran organizados por carpetas según su función, como se muestra en la Figura 19.

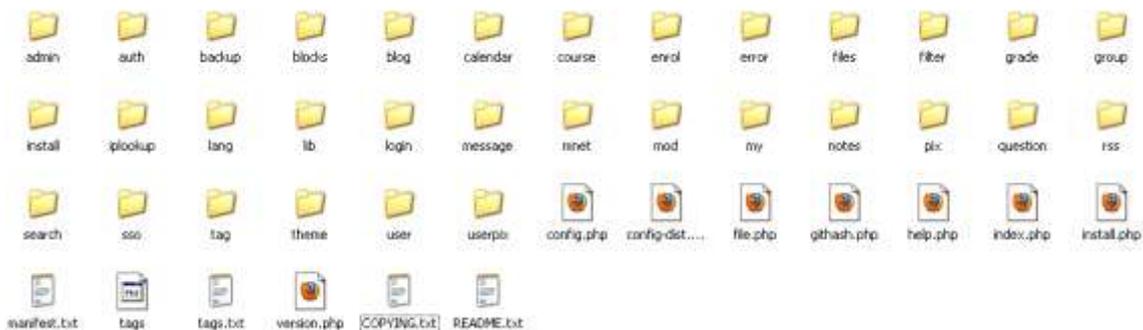


Figura 19. Organización de las carpetas de Moodle.

A continuación se describe la función de los *archivos*, *funciones* y *objetos* de Moodle que se analizaron, los cuales son utilizados para la implementación de nuestro proyecto.

5.1.1 Archivos PHP de Moodle

Los archivos PHP contienen los códigos de programación que dan soporte al funcionamiento de Moodle. Entre sus funciones se encuentran, el procesamiento de información, el acceso a la base de datos, la creación y el manejo de objetos en memoria, etc. En la

Tabla 17 se muestran los archivos analizados y/o utilizados.

Tabla 17. Archivos PHP de Moodle analizados

ARCHIVO	UBICADO EN CARPETA	FUNCION
<i>index.php</i>	Moodle	Muestra la pantalla inicial de Moodle, donde se despliegan los cursos disponibles.
<i>config.php</i>	Moodle	Inicializa las variables para la conexión con la BD y para almacenar la ubicación Web y local de Moodle.
<i>setup.php</i>	Moodle\lib\	Declara las variables globales, carga librerías básicas, configura variables para conectarse con la base de datos, etc.
<i>dmlib.php</i>	Moodle\lib\	Contiene las funciones para acceder a la información contenida en la base de datos.
<i>index.php</i>	Moodle\login\	Carga la pantalla de inicio de sesión de los usuarios para ingresar al sistema.
<i>moodleblock.class.php</i>	Moodle\blocks\	Muestra los bloques de Moodle, los cuales se encuentran ubicados en los laterales de la pantalla principal.
<i>scorm_12lib.php</i>	Moodle\mod\scorm\datamodels	Contiene las funciones utilizadas para seleccionar y presentar los temas.
<i>attempt.php</i>	Moodle\mod\quiz\	Contiene información de los resultados obtenidos por un alumno al intentar acreditar un examen.
<i>review.php</i>	Moodle\mod\quiz\	Muestra una retroalimentación al alumno después de que éste presenta un examen.

5.1.2 Funciones para acceder a la base de datos de Moodle

Para acceder a la base de datos de Moodle, la cual almacena información de la configuración de Moodle, de los contenidos didácticos y de los usuarios, se utilizan las funciones nativas de Moodle. En la Tabla 18, se muestran las funciones utilizadas en la implementación de esta tesis.

Tabla 18. Funciones analizadas de Moodle

FUNCION	DESCRIPCION
<i>get_field_sql</i> (consulta SQL)	Recibe como parámetro una consulta SQL y retorna un registro.
<i>get_recordset_sql</i> (consulta SQL)	Recibe como parámetro una consulta SQL y retorna un conjunto de registros.
<i>rs_fetch_next_record</i> (consulta SQL)	Recibe como parámetro una consulta SQL y devuelve el siguiente registro de un conjunto de registros.
<i>scorm_get_scoes</i> (identificador SCORM)	Recibe como parámetro el identificador de un paquete SCORM, y retorna los elementos del paquete.

<i>get_records_select</i> (Nombre Tabla, Cláusula Where)	Recibe como parámetros el nombre de la tabla a procesar y la cláusula Where, y retorna un conjunto de registros.
--	--

5.1.3 Objetos globales de Moodle

Al igual que otros lenguajes de programación, PHP también permite la programación orientada a objetos. Los desarrolladores de Moodle utilizaron esta técnica para crear objetos en memoria, los cuales almacenan información acerca de los usuarios, exámenes, datos del curso, etc. Estos objetos se encuentran disponibles para ser accedidos por otros códigos de programación durante la sesión activa de un alumno. En la Tabla 19 se muestran los objetos analizados y utilizados en la implementación de este proyecto.

Tabla 19. Objetos globales de Moodle analizados

OBJETO	FUNCIÓN
<i>\$user</i>	Contiene los datos del usuario activo, incluyendo los valores de los campos agregados al perfil del usuario.
<i>\$cfg</i>	Contiene los datos de la configuración actual de Moodle (nombres del servidor y base de datos, tipo de base de datos, directorio Web de trabajo, directorio local de trabajo, etc.)
<i>\$course</i>	Contiene información relacionada con el curso al que ingresa un alumno.
<i>\$quiz</i>	Contiene información relacionada de un examen realizado por un alumno.
<i>\$attempt</i>	Contiene información relacionada con el intento que realiza un alumno para acreditar un examen.

5.2 Configuración de Moodle

En esta sección se describen las herramientas utilizadas de Moodle, así como también se detallan los archivos que se añadieron a la plataforma. Dentro de los elementos utilizados de Moodle se encuentran: la creación de un curso, la preparación de los contenidos de un curso y su importación a Moodle, así como también la creación de los exámenes correspondientes a los temas de un curso. Cabe mencionar que todos estos elementos pueden considerar otra configuración, sin embargo, para efectos de éste trabajo de tesis se requieren seguir los lineamientos que se describen.

5.2.1 Creación de un curso

Los cursos son la parte más importante de Moodle, estos son creados por los administradores del sitio y dirigidos por los usuarios establecidos como profesores del curso. Para que un administrador pueda crear un curso, debe proceder a su

configuración mediante un formulario proporcionado por Moodle, en el que se establecen valores para distintos campos.

En particular, para éste proyecto de tesis, se requiere crear un curso por cada asignatura que se quiera enseñar. Para la configuración de un curso se debe considerar las siguientes especificaciones para los siguientes campos:

- En el campo **Formato** se debe elegir *Formato de temas*.
- En el campo **Número de semanas o temas** se selecciona el número de unidades que considera la asignatura para la cual se creó el curso.

El proceso para la creación y configuración de un curso en Moodle se describe en el Anexo A.

Una vez creado un curso, 2 tablas de Moodle guardan la información generada

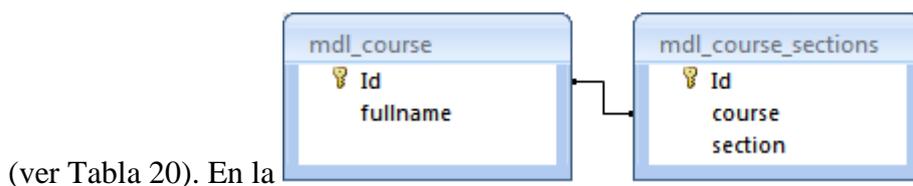


Figura 20 se muestra el diagrama entidad-relación de las mismas.

Tabla 20. Tablas donde se guarda la información de los cursos

TABLA	FUNCIÓN
<i>mdl_course</i>	Almacena los datos de encabezamiento de los cursos creados.
<i>mdl_course_section</i>	Almacena los datos de cada una de las secciones de un curso.

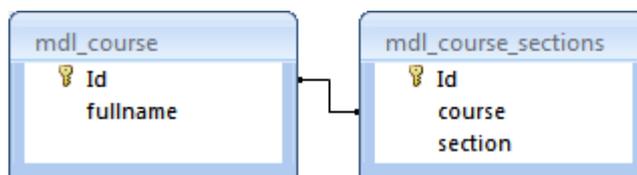


Figura 20. Diagrama entidad-relación de las tablas donde se guarda la información de los cursos

5.2.2 Elaboración, empaquetado e importación de contenidos

Para aplicar la arquitectura propuesta, la elaboración de los contenidos para un curso dentro de Moodle debe considerar lo siguiente:

- Un mismo tema debe ser elaborado utilizando en su diseño al menos una estrategia de enseñanza que favorezca a cada uno de los cuatro estilos de aprendizaje del modelo VARK. De esta forma se estará realizando el mismo tema en una forma adecuada para cada estilo de aprendizaje.
- Los temas deben ser elaborados en archivos de tipo pdf, html y avi.
- El nombre de cada archivo debe estar formado indicando los siguientes elementos: el nombre del estilo de aprendizaje para el cual está diseñado, el nombre de la estrategia de enseñanza en la cual fue diseñado, y por último el nombre del tema. El nombre de archivo debe ir separado por un guión entre cada elemento. Por ejemplo: Visual-Vídeo-Arreglos.pdf.
- Todos los archivos generados para un mismo tema deben estar guardados en una carpeta exclusiva para ese tema.
- Empaquetar en formato SCORM (ver sección 5.2.2) los archivos correspondientes a un mismo tema.

Para empaquetar los contenidos en formato SCORM, se debe realizar lo siguiente:

- Utilizar el programa *Reload Editor* versión 1.3 para formar los paquetes SCORM.
- Cada tema que forma parte de un curso debe ser un paquete SCORM.
- Cada paquete SCORM deberá estar formado únicamente por los archivos correspondientes a un mismo tema.

El proceso para crear los paquetes SCORM con el programa *Reload Editor* se encuentra en el Anexo B. El proceso para la importación de los paquetes SCORM a Moodle se describe en el Anexo C.

Las tablas donde Moodle guarda la información de los paquetes SCORM importados a Moodle, se muestran en la Tabla 21. En la Figura 21 se muestra el diagrama entidad-relación de las mismas.

Tabla 21. Tablas donde se guarda la información relacionada con los paquetes SCORMS

TABLA	FUNCIÓN
<i>mdl_scorm</i>	Almacena los datos de encabezamiento de los paquetes SCORM importados a Moodle, así como el identificador del curso al que pertenecen.
<i>mdl_scorm_scoes</i>	Almacena los datos de cada uno de los recursos contenidos dentro de los paquetes SCORM, así como también el identificador del SCORM al que pertenece el recurso.

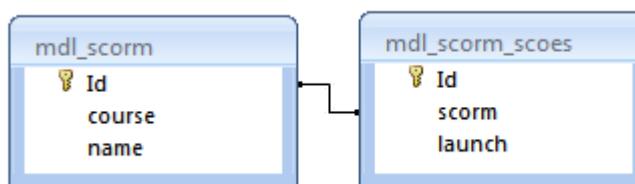


Figura 21. Diagrama entidad-relación de las tablas donde se guarda la información relacionada con los paquetes SCORM

5.2.3 Elaboración de exámenes para evaluar temas del curso

La elaboración de los exámenes utilizados en evaluar el conocimiento que tiene un alumno sobre algún tema, se realiza mediante la actividad *Cuestionario* de Moodle, la cual es una herramienta que permite realizar exámenes dentro de la misma plataforma. La actividad *Cuestionario* permite realizar exámenes de distintos tipos, tales como: respuesta múltiple, respuesta corta, verdadero/falso, etc.

La elaboración de los exámenes para un curso debe considerar lo siguiente:

- Realizar un sólo examen por cada tema del curso.
- Los exámenes deben ser de tipo respuesta múltiple.
- Cada pregunta del examen puede tener una puntuación diferente.
- La suma de las puntuaciones de las preguntas debe ser de 100 puntos.
- Cada examen puede ser contestado un número ilimitado de veces.
- El método de calificación para cada examen debe ser el del último intento.

En el Anexo D se describe el proceso para crear los exámenes utilizando la actividad *Cuestionario* de Moodle.

Las tablas donde Moodle almacena la información relativa a los exámenes creados a través de la actividad *Cuestionario*, se muestran en la

Tabla 22, y en la Figura 22 se muestra un fragmento de estas tablas y el diagrama entidad-relación de las mismas.

Tabla 22. Tablas donde se guarda la información relativa a los exámenes creados en Moodle

TABLA	FUNCIÓN
<i>mdl_quiz</i>	Almacena los datos de encabezamiento de los exámenes creados en Moodle, así como el identificador del curso al que pertenecen.
<i>mdl_quiz_grade</i>	Almacena la calificación que obtiene cada alumno al presentar un examen. La calificación guardada depende del método de calificación determinada por la configuración dada al examen. El método elegido en este proyecto de tesis es considerar la calificación obtenida en el último intento realizado en un examen por un alumno.
<i>mdl_quiz_attempts</i>	Almacena los datos de cada intento que realiza un alumno al contestar un mismo examen, así como también guarda la calificación obtenida en cada uno de los intentos.

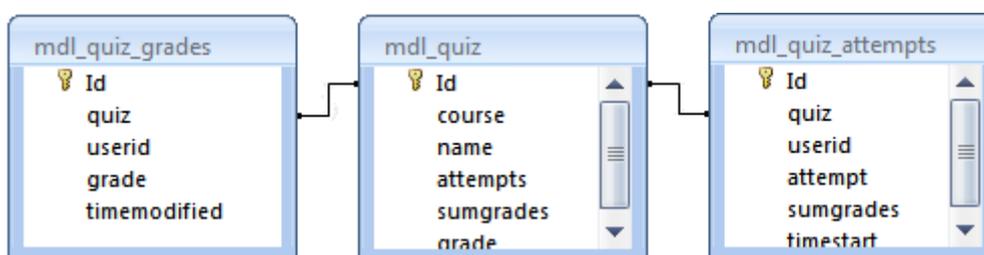


Figura 22. Diagrama entidad-relación de las tablas donde se guarda la información relativa a los exámenes creados en Moodle

5.2.4 Configuración de los campos de perfil del usuario

Durante la revisión de la funcionalidad de Moodle se encontró que éste permite añadir campos al perfil del usuario. Esta función fue utilizada para agregar los campos que identifiquen a cada uno de los cuatro estilos de aprendizaje del modelo VARK y almacenar en ellos, las puntuaciones obtenidas por cada uno de los usuarios al contestar el cuestionario VARK.

Los campos añadidos fueron: *Visual*, *Auditivo*, *LeerEscribir* y *Kinestesico*, los cuales almacenan la puntuación obtenida para cada uno de los estilos de aprendizaje. También se añadió el campo *Estilo_actual*, el cual almacena el identificador del estilo de aprendizaje actual del alumno. De la misma forma se agregó el campo *EstrategiaActual*, que almacena el identificador de una estrategia de aprendizaje acorde al *Estilo_actual*. Por último se agregó el campo *Edad*, para registrar la edad de cada uno de los usuarios. El procedimiento para añadir campos de perfil del usuario se encuentra en el Anexo E.

Las tablas donde Moodle guarda la información relativa a los campos añadidos al perfil del usuario se muestran en la Tabla 23, y en la Figura 28 se muestra un fragmento de estas tablas y el diagrama entidad-relación de las mismas.

Tabla 23. Tablas donde se guarda la información de los usuarios y de los campos añadidos al perfil del usuario

TABLA	FUNCIÓN
<i>mdl_user</i>	Almacena los datos generales de los usuarios. Esta tabla no guarda información de los campos de perfil, pero es utilizada para identificar al usuario y relacionarlo con sus respectivos campos de perfil añadidos en la tabla <i>mdl_user_info_data</i> .
<i>mdl_user_info_data</i>	Almacena los datos que el usuario especifica en los campos de perfil añadidos, cuando realiza su registro en la plataforma.
<i>mdl_user_info_field</i>	Almacena la configuración de cada uno de los campos que se añaden al perfil del usuario.

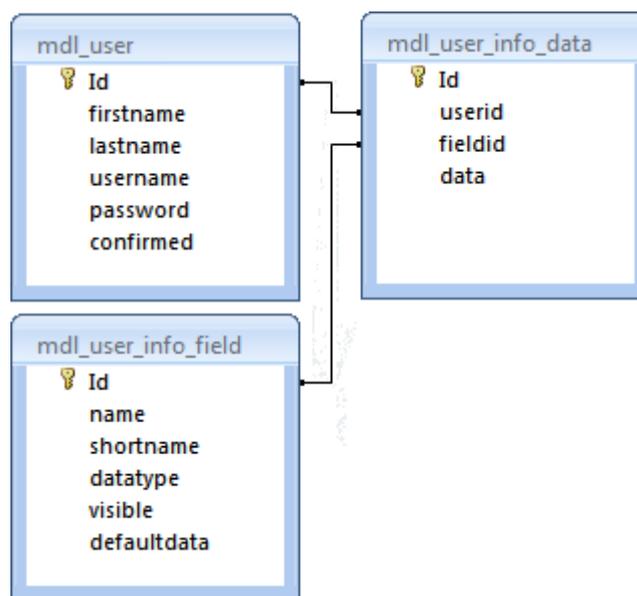


Figura 23. Diagrama entidad-relación de las tablas donde se guarda la información de los usuarios y de los campos añadidos al perfil del usuario

5.2.5 Modificación del menú: Administración del sitio

El menú *Administración del sitio* es utilizado por el administrador de la plataforma y se encuentra ubicado en la página principal de Moodle. Dentro de las diferentes acciones que puede realizar el administrador mediante éste menú se encuentran: autenticar a los usuarios, administrar los cursos, administrar la forma de realizar y mostrar las calificaciones en cada curso, etc.

Como se puede apreciar, en el menú *Administración del sitio* se localizan las opciones más importantes para la administración de Moodle, por tal motivo, se decidió incorporar a este menú un nuevo elemento llamado *Configuración*. La función de este nuevo componente es permitir el acceso a las interfaces que fue necesario añadir a Moodle, para realizar la implementación de este proyecto de tesis. Estas interfaces son sub-elementos del componente *Configuración*, y son explicadas en las secciones 5.2.6.2 – 5.2.6.4. En la Figura 24 se puede observar al menú *Administración del sitio* integrando al elemento *Configuración* y a sus respectivos sub-elementos.



Figura 24. Menú *Administración del sitio* con la opción *Configuración* incluida

Para incorporar al elemento *Configuración* en el menú *Administración del sitio* fue necesario modificar un archivo nativo de Moodle y crear un archivo, los cuales se describen en la Tabla 24.

Tabla 24. Archivos utilizados para añadir un elemento al menú *Administración del sitio*

ARCHIVO	FUNCIÓN
<i>top.php</i> (nativo de Moodle)	Almacenar los elementos del menú <i>Administración del sitio</i> .
<i>itcmConfiguracion.php</i> (añadido a Moodle)	Almacenar los sub-elementos del elemento <i>Configuración</i> incluido en el menú <i>Administración del sitio</i> .

En la Figura 25 se presenta un fragmento de código del archivo *top.php*, en donde al final de éste se encuentra la línea de código añadida. Este archivo *top.php*, nativo de Moodle, fue modificado con el propósito de incluir en el menú *Administración del sitio* al elemento *Configuración*. Para añadir un elemento al menú *Administración del sitio*, se realiza a través del método *add* del objeto *\$ADMIN*, el cual requiere de dos parámetros. El primer parámetro indica que la opción a agregar, formará parte del menú principal, ya que se está especificando que depende de *root*. El segundo parámetro es un objeto, el cual a su vez requiere de dos parámetros. El primer parámetro es el nombre del archivo PHP (en este caso *itcmConfiguracion* sin indicar la extensión del archivo) el cual contiene a los sub-elementos del elemento a añadir. El segundo parámetro es el nombre del elemento a añadir, en este caso *Configuración*. Cabe

mencionar que el archivo *itcmConfiguracion.php* debe ser creado en la misma ubicación del archivo *top.php*, la cual es Moodle\admin\settings.

```

$ADMIN->add('root', new admin_category('users', get_string('users','admin')));
$ADMIN->add('root', new admin_category('courses', get_string('courses','admin')));
$ADMIN->add('root', new admin_category('grades', get_string('grades')));
$ADMIN->add('root', new admin_category('location', get_string('location','admin')));
$ADMIN->add('root', new admin_category('language', get_string('language')));
$ADMIN->add('root', new admin_category('modules', get_string('plugins', 'admin')));
$ADMIN->add('root', new admin_category('security', get_string('security','admin')));
$ADMIN->add('root', new admin_category('appearance', get_string('appearance','admin')));
$ADMIN->add('root', new admin_category('frontpage', get_string('frontpage','admin')));
$ADMIN->add('root', new admin_category('server', get_string('server','admin')));
$ADMIN->add('root', new admin_category('mnet', get_string('net','mnet')));
$ADMIN->add('root', new admin_category('reports', get_string('reports')));
$ADMIN->add('root', new admin_category('misc', get_string('miscellaneous')));

//TESIS. Agrega el nodo de Configuración en el ARBOL de ADMINISTRACION
$ADMIN->add('root', new admin_category('itcmConfiguracion', 'Configuraci&osacute;n'));

```

Figura 25. Fragmento de código del archivo *top.php*

En la

Figura 26, se muestra el código del archivo *itcmConfiguración.php*. Este archivo fue creado con el propósito de incluir los sub-elementos al elemento *Configuración*. Cada sub-elemento fue añadido a través de una línea de código utilizando el método *add* del objeto *\$ADMIN* de forma similar a cómo se agregó el elemento *Configuración*. La diferencia radica en el segundo parámetro, ya que es un objeto distinto, el cual requiere de tres parámetros. El primer parámetro indica quién es el padre del sub-elemento, en este caso es *itcmConfiguracion*. El segundo parámetro es el nombre del sub-elemento que será mostrado como dependiente del elemento *Configuración* en el menú *Administración del sitio*, por ejemplo *Catálogo Estilos* muestra el título *Catálogo Estilos*. Por último, el tercer parámetro es el que indica el archivo a ejecutar cuando algún sub-elemento sea elegido. Cabe mencionar que cada sub-elemento será descrito en la siguiente sección.

```

if (!$baseconfig) {
$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Cat&aacute;logo Estilos', "%CFG->wwwroot./login/cat_estilos.php"));
$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Cat&aacute;logo Estrategias', "%CFG->wwwroot./login/cat_estrategias.php"));
$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Cat&aacute;logo Competencias', "%CFG->wwwroot./login/cat_competencias.php"));

$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Vincular Tema-Estilo-Estrategia', "%CFG->wwwroot./login/p_scorra_ee.php"));
$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Vincular Tema-Examen', "%CFG->wwwroot./login/p_unidad_tema_examen_ee.php"));
$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Vincular Tema-Competencias', "%CFG->wwwroot./login/p_competencias_temas.php"));

$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Captura prob. Competencias', "%CFG->wwwroot./login/p_rb_competencias.php"));
$ADMIN->add('itcmConfiguracion', new admin_externalpage('itcmConfiguracion','Captura prob. Tem&as', "%CFG->wwwroot./login/p_rb_competencias_temas.php"));
}

```

Figura 26. Código del archivo *itcmConfiguracion.php*

5.2.6 Módulos añadidos a Moodle

Para implementar la arquitectura propuesta, fue necesario añadir a Moodle algunos módulos, los cuales formaron parte imprescindible para la realización de este proyecto de tesis. Estos módulos están compuestos por archivos y tablas, la creación de las tablas se realizó mediante el *Script* descrito en el Anexo F. A continuación se presenta un panorama general de los módulos añadidos y posteriormente se describe cada uno de ellos.

- Incorporación del Cuestionario VARK.
- Incorporación de catálogos (estilos, estrategias y competencias).
- Vinculación de temas, estilos, estrategias, exámenes y competencias.
- Captura de las tablas de probabilidad.

Cabe mencionar que estos módulos, excepto la Incorporación del Cuestionario VARK, pueden ser accedidos desde el menú *Configuración*, descrito en la sección anterior.

5.2.6.1 Incorporación del Cuestionario VARK

El cuestionario VARK se implementó e incorporó a Moodle con el propósito de determinar la preferencia que tiene cada alumno al procesar la información desde un punto de vista sensorial. La implementación del cuestionario consistió en realizar las siguientes actividades:

1. Crear un archivo con las preguntas y respuestas del cuestionario.
2. Implementar un proceso para examinar las respuestas del alumno.
3. Guardar las puntuaciones obtenidas de cada uno de los estilos de aprendizaje.
4. Determinar y guardar el estilo de aprendizaje predominante del alumno.
5. Asignar al alumno una estrategia de enseñanza que vaya de acuerdo a su estilo de aprendizaje predominante y almacenar dicha información.

6. Incluir el cuestionario a Moodle.

Para realizar el almacenamiento de los resultados obtenidos del cuestionario al ser contestado por un alumno, se utilizaron los campos de perfil del usuario añadidos, descritos en la sección 5.2.4. Considerando dichos campos para almacenar la puntuación de cada uno de los cuatro estilos de aprendizaje, el estilo de aprendizaje y la estrategia de enseñanza, se procedió a realizar la incorporación en Moodle del cuestionario VARK.

El *primer paso* para la incorporación del cuestionario fue diseñar la página Web llamada *p_vark.html*, la cual contiene las preguntas y respuestas del cuestionario. En la Figura 28 se muestra un fragmento de la página Web que contiene el cuestionario VARK. Las preguntas y respuestas que componen al cuestionario se encuentran en el Anexo G.

El *segundo paso* consistió en diseñar el archivo *p_vark.php*, el cual realiza las siguientes funciones:

- a) Revisa cada una de las respuestas elegidas por el usuario y determina a cuál de los estilos de aprendizaje corresponde. Esto con la intención de calcular cuantas respuestas pertenecen a cada uno de los cuatro estilos de aprendizaje. Una vez realizado esto, almacena las puntuaciones correspondientes en los campos *Visual*, *Auditivo*, *LeerEscribir* y *Kinestesico*.
- b) Determina cuál fue el estilo de aprendizaje que obtuvo la mayor puntuación y lo asigna como el estilo actual, guardando en el campo *Estilo_actual* el número que identifica a ese estilo de aprendizaje. En el caso donde dos o más estilos tienen la misma puntuación, los dispone en el siguiente orden: Visual, Auditivo, LeerEscribir y Kinestesico. Una vez que se encuentran ordenados los estilos, se elige de forma ascendente al que encabeza la lista y se almacena su identificador en el campo *Estilo_actual*. Por ejemplo, considerar que un alumno obtiene las siguientes puntuaciones: Visual 2, Auditivo 6, LeerEscribir 4 y Kinestesico 6, en este caso, se observa que el mayor número de puntos es de 6 y que los estilos Auditivo y Kinestesico tienen esa misma puntuación, por lo que será elegido como estilo actual al *Auditivo* ya que respetando el orden de los estilos, se encuentra antes que el Kinestesico.

- c) Selecciona a las estrategias de enseñanza que están relacionadas al estilo de aprendizaje actual del usuario, ordenándolas de acuerdo a como fueron capturadas, y elige a la estrategia que encabeza la lista guardando su identificador en el campo *EstrategiaActual*.
- d) Guarda en la tabla *mdl_historial_vark* cada una de las respuestas elegidas por cada usuario al contestar el cuestionario.

El *tercer* y último paso fue incorporar el cuestionario en Moodle, es decir, lograr que después que un alumno se registra en la plataforma, a continuación y por única ocasión, le sea presentado el cuestionario VARK. Cabe mencionar que un usuario al registrarse, posteriormente puede matricularse en los cursos existentes en la plataforma siempre y cuando estos se lo permitan, sin necesidad de volver a contestar el cuestionario.

Debido a que el cuestionario VARK se debe aplicar sólo una vez, fue necesario detectar en el código de Moodle, el punto donde se valida el inicio de sesión del alumno.

El archivo *index.php* ubicado en la carpeta *moodle\login*, es el que se encarga de validar el inicio de sesión de los usuarios en su ingreso a Moodle. Por lo que se modificó dicho archivo para lograr que consultara los valores de los campos *Visual*, *Auditivo*, *LeerEscribir* y *Kinestésico*. Los cuales, de encontrarse vacíos, indican que no ha sido aplicado el cuestionario VARK y le será mostrado el cuestionario al alumno para que sea contestado. A continuación se presenta el código agregado al archivo *index.php*:

```
if ($USER->Visual == "" and $USER->Auditivo == "" and $USER->LeerEscribir == "" and
    $USER->Kinestesico == "")
{
    redirect($CFG->httpswwwroot.'/login/p_vark_preguntas.php');
}
```

Los campos *Visual*, *Auditivo*, *LeerEscribir*, *Kinestésico*, *Estilo_Actual* y *EstrategiaActual* son cargados automáticamente durante toda la sesión del alumno en el objeto global *\$USER*, el cual permite tener acceso a ellos en cualquier momento.

Cuando los valores de los campos arriba mencionados son consultados por el *if*, estos al encontrarse vacíos, significa que no ha sido contestado el cuestionario, por lo que se ejecutará la línea *redirect(\$CFG->httpswwwroot.'/login/p_vark_preguntas.php')*, la cual llama al archivo *p_vark_preguntas.php* que a su vez llama a la página que contiene el

cuestionario VARK, el cual es mostrado al usuario para ser contestado. Por el contrario, sí los campos al ser consultados ya tienen información, significa que ya fue contestado el cuestionario y no será mostrado al alumno.

En la Tabla 25 se indica la tabla utilizada en almacenar cada una de las respuestas elegidas por los alumnos al contestar el cuestionario y en la Figura 27 se ilustra su estructura y dominio.

Tabla 25. Tabla donde se almacenan las respuestas de los alumnos al contestar el cuestionario VARK

TABLA	FUNCIÓN
<i>mdl_historial_vark</i>	Almacenar cada una de las respuestas elegidas por los alumnos al contestar el cuestionario VARK.



id	userid	respuestas
5	78	K...I...A...I...R...I...R...I...A...V...K...R...I...
6	78	K...I...A...I...R...I...R...I...A...V...K...R...I...
7	79	...A...I...A...I...A...I...A...I...K...R...I...
8	80	...R...I...R...I...R...A...I...K...A...I...R...I...
9	81	...V...I...V...I...V...I...V...I...V...I...V...
10	82	...A...I...A...I...A...I...A...I...A...I...A...

Figura 27. Estructura y dominio de la tabla *mdl_historial_vark*

En la Tabla 26 se muestra un resumen de los archivos creados y agregados a Moodle, así como también los archivos nativos de Moodle que fueron modificados y utilizados para la incorporación del cuestionario VARK a la plataforma Moodle. Estos archivos están listados de acuerdo a como son llamados después de que un alumno se registra en la plataforma.

Tabla 26. Archivos utilizados para incorporar el cuestionario VARK en Moodle

ARCHIVO	FUNCIÓN
<i>index.php</i> (nativo de Moodle, ubicado en Moodle\login)	<ul style="list-style-type: none"> Valida el inicio de sesión de los usuarios. Verifica si se le presentará o no el cuestionario VARK al usuario que está iniciando sesión. Si se le presenta el cuestionario llama al archivo <i>p_vark_preguntas.php</i>.
<i>p_vark_preguntas.php</i> (añadido a Moodle)	<ul style="list-style-type: none"> Muestra el encabezado y pie de página estándar de Moodle y llama a la página <i>p_vark.html</i>.
<i>p_vark.html</i> (añadido a Moodle)	<ul style="list-style-type: none"> Muestra el cuestionario VARK al alumno. Envía las respuestas elegidas por el alumno en el cuestionario, al archivo <i>p_vark.php</i>.

<p><i>p_vark.php</i> (añadido a Moodle)</p>	<ul style="list-style-type: none"> • Calcula y almacena en los campos correspondientes, la puntuación para cada uno de los estilos de aprendizaje de acuerdo a las respuestas elegidas por el alumno. • Determina el estilo de aprendizaje predominante del alumno y lo almacena en el campo <i>Estilo_actual</i>. • Asigna una estrategia de enseñanza que vaya de acuerdo al estilo de aprendizaje actual del alumno y lo almacena en el campo <i>EstrategiaActual</i>. • Nota: Los campos arriba mencionados corresponden a la tabla <i>mdl_user_info_data</i> descrita en la sección 5.2.4. • Guarda todas las respuestas elegidas por el usuario en la tabla <i>mdl_historial_vark</i>.
---	---

CUESTIONARIO VARK

Con este cuestionario se tiene el propósito de saber acerca de sus preferencias para trabajar con información. Elija las respuestas que mejor expliquen su preferencia. Puede seleccionar más de una respuesta a una pregunta si una sola no encaja con su percepción. Deje en blanco toda pregunta que no se aplique a sus preferencias.

Favor de contestar al menos 12 preguntas.

1. Está ayudando a una persona que desea ir al aeropuerto, al centro de la ciudad o a la estación del ferrocarril. Ud.:

- iría con ella.
- le diría cómo llegar.
- le daría las indicaciones por escrito (sin un mapa).
- le daría un mapa.

2. No está seguro si una palabra se escribe como: trascendente o tracentente, Ud.:

- venía las palabras en su mente y elegiría la que mejor luce.
- pensaría en cómo suena cada palabra y elegiría una.
- las buscaría en un diccionario.
- escribiría ambas palabras y elegiría una.

Figura 28. Cuestionario VARK incorporado a Moodle

5.2.6.2 Incorporación de catálogos

Para lograr el propósito de esta tesis, fue indispensable incorporar interfaces que le permitan al administrador de Moodle, capturar información, la cual será utilizada por el agente selector y la red bayesiana. Entre las interfaces añadidas se encuentran los catálogos para estilos, estrategias y competencias. Dichos catálogos se pueden acceder a través de la opción *Configuración* localizada en el menú *Administración del sitio*, ver

sección 5.2.5. El procedimiento para utilizar estas interfaces se describe en el Anexo H. A continuación se describe cada uno de los catálogos.

Catálogo de estilos

Este catálogo contiene la información acerca de los estilos de aprendizaje del modelo VARK. Este catálogo no tiene una interfaz de captura, ya que por motivos de alcance de esta tesis, no se contempla algún otro modelo de estilos de aprendizaje, es decir, sólo se considera el modelo VARK.

La tabla añadida a Moodle donde se almacena la información de los estilos de aprendizaje del modelo VARK se presenta en la

Tabla 27. En la parte izquierda de la

Figura 29, se muestra la estructura de la tabla *mdl_estilos* y en la parte derecha, se presenta el dominio de la misma.

Tabla 27. Tabla de estilos de aprendizaje VARK

TABLA	FUNCIÓN
<i>mdl_estilos</i>	Almacena los estilos de aprendizaje del modelo VARK.



Figura 29. Estructura y dominio de la tabla *mdl_estilos*

Catálogo de Estrategias

Este catálogo es el encargado de permitir la captura de las estrategias de enseñanza-aprendizaje que se encuentren relacionadas a cada uno de los estilos de aprendizaje.

La tabla incorporada a Moodle relacionada con éste catálogo se presenta en la Tabla 28. En la Figura 30 se muestra un fragmento de esta tabla y la relación que tiene con la tabla que almacena los estilos de aprendizaje.

Tabla 28. Tabla relacionada con el catálogo de estrategias de enseñanza-aprendizaje

TABLA	FUNCIÓN
<i>mdl_estrategias</i>	Almacena las estrategias de enseñanza, así como el identificador del estilo de aprendizaje al que favorece cada estrategia.

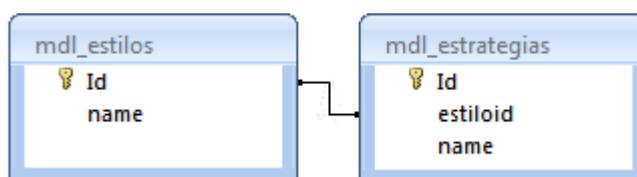


Figura 30. Diagrama entidad-relación de las tablas relacionadas con el catálogo de estrategias

El código de programación agregado a Moodle, relacionado con el funcionamiento de éste catálogo, se encuentra distribuido en los archivos indicados en la Tabla 29:

Tabla 29. Archivos que forman el catálogo de estrategias de enseñanza-aprendizaje

ARCHIVOS	FUNCIÓN
<i>cat_estrategias.php</i> <i>cat_estrategias_captura.html</i> <i>cat_estrategias_bd.php</i>	Realizar el registro de nuevas estrategias de enseñanza.
<i>cat_estrategias_modificar.php</i> <i>cat_estrategias_modificar_captura.html</i> <i>cat_estrategias_modificar_bd.php</i>	Realizar modificaciones a las estrategias de enseñanza existentes.
<i>cat_estrategias_eliminar.php</i>	Realizar la eliminación de estrategias de enseñanza existentes.

En la

Tabla 30 se indica la función que desempeña cada uno de éstos archivos dentro de Moodle.

Tabla 30. Función de los archivos del catálogo de estrategias de enseñanza

ARCHIVO	UBICADO EN CARPETA	FUNCIÓN
<i>cat_estrategias.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>cat_estrategias_captura.html</i> .
<i>cat_estrategias_captura.html</i>	Moodle\login	Capturar una nueva estrategia de enseñanza.
<i>cat_estrategias_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para registrar a la nueva estrategia de enseñanza capturada.
<i>cat_estrategias_modificar.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>cat_estrategias_modificar_captura.html</i> .
<i>cat_estrategias_modificar_captura.html</i>	Moodle\login	Actualizar alguna estrategia de enseñanza existente.
<i>cat_estrategias_modificar_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para actualizar la estrategia de enseñanza modificada.
<i>cat_estrategias_eliminar.php</i>	Moodle\login	Realizar la conexión con la base de datos para eliminar la estrategia de enseñanza seleccionada.

Catálogo de competencias

Este catálogo es el encargado de facilitar la captura de la información de las competencias específicas relacionadas con la asignatura a enseñar. La tabla integrada a Moodle vinculada con éste catálogo se presenta en la Tabla 31, y en la Figura 31 se muestra su estructura.

Tabla 31. Tabla relacionada con el catálogo de Competencias

TABLA	FUNCIÓN
<i>mdl_competencias</i>	Almacena las competencias específicas de la asignatura que será enseñada mediante un curso en Moodle.

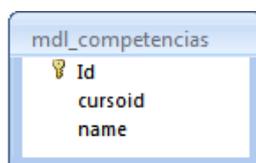


Figura 31. Estructura de la tabla del catálogo de competencias

El código de programación agregado a Moodle, relacionado con el funcionamiento de éste catálogo, se encuentra distribuido en los archivos mencionados en la

Tabla 32. En la Tabla 33 se indica la función que desempeña cada uno de éstos archivos:

Tabla 32. Archivos que forman el catálogo de competencias específicas

ARCHIVOS	FUNCIÓN
<i>cat_competencias.php</i> <i>cat_competencias_captura.html</i> <i>cat_competencia_bd.php</i>	Realizar el registro de nuevas competencias específicas.
<i>cat_competencias_modificar.php</i> <i>cat_competencias_modificar_captura.html</i> <i>cat_competencias_modificar_bd.php</i>	Realizar modificaciones a las competencias específicas existentes.
<i>cat_competencias_eliminar.php</i>	Realizar la eliminación de competencias específicas existentes.

Tabla 33. Función de los archivos del catálogo de competencias específicas

ARCHIVO	UBICADO EN CARPETA	FUNCIÓN
<i>cat_competencias.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>cat_competencias_captura.html</i> .
<i>cat_competencias_captura.html</i>	Moodle\login	Capturar una nueva competencia específica.
<i>cat_competencia_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para registrar a la nueva competencia específica capturada.
<i>cat_competencias_modificar.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>cat_competencias_modificar_captura.html</i> .
<i>cat_competencias_modificar_captura.html</i>	Moodle\login	Actualizar alguna competencia específica existente.
<i>cat_competencias_modificar_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para actualizar la competencia específica existente.
<i>cat_competencias_eliminar.php</i>	Moodle\login	Realizar la conexión con la base de datos para eliminar la competencia específica seleccionada.

5.2.6.3 Vinculación de temas, estilos, estrategias, exámenes y competencias

Además de contar con interfaces que permitan la captura de información necesaria para la integración del Agente Selector y la Red Bayesiana a Moodle, también fue necesario implementar interfaces que permitan relacionar dicha información. Debido a esto, se agregaron a Moodle interfaces para vincular los siguientes elementos:

- Vincular cada tema a un estilo y a una estrategia.

- Vincular cada tema a su respectivo examen.
- Vincular cada tema a la competencia con la que se encuentra relacionada.

Estas interfaces se pueden acceder a través de la opción *Configuración* localizada en el menú *Administración del sitio*, ver sección 5.2.5. El procedimiento para utilizar estas interfaces se describe en el Anexo I. A continuación se describen cada una de ellas.

Vincular Tema, Estilo y Estrategia

Esta interfaz se realizó con el propósito de vincular a cada tema con: *a)* el estilo de aprendizaje para el cual fue diseñado y *b)* la estrategia de enseñanza que favorece a ese estilo de aprendizaje y que fue utilizada en elaborar el tema. En la Tabla 34 se ilustra la tabla añadida a Moodle en la cual se almacena la información configurada en esta interfaz, así como también se muestran las tablas nativas de Moodle o añadidas a Moodle (descritas en secciones anteriores), utilizadas para realizar el enlace. En la Figura 32 se muestra el diagrama entidad-relación de las mismas.

Tabla 34. Tablas relacionadas con la opción Vincular tema, estilo y estrategia

TABLA	FUNCIÓN
<i>mdl_scorm_ee</i> (añadida a Moodle)	Almacena la relación existente entre un tema con un estilo de aprendizaje y con una estrategia de enseñanza.
<i>mdl_scorm</i> (nativa de Moodle)	De ésta tabla toma los datos del paquete SCORM que representa al tema que se desea vincular a un estilo y a una estrategia.
<i>mdl_scorm_scoes</i> (nativa de Moodle)	De ésta tabla toma el nombre del recurso en particular del paquete SCORM, el cual será relacionado a un estilo y a una estrategia.
<i>mdl_course</i> (nativa de Moodle)	De ésta tabla toman los datos del curso en el cual se está realizando la vinculación.
<i>mdl_estilos</i> (añadida a Moodle)	De ésta tabla toma los datos del estilo de aprendizaje al que se va a vincular el tema elegido.
<i>mdl_estrategias</i> (añadida a Moodle)	De ésta tabla toma los datos de la estrategia de enseñanza a la que se va a vincular el tema elegido.

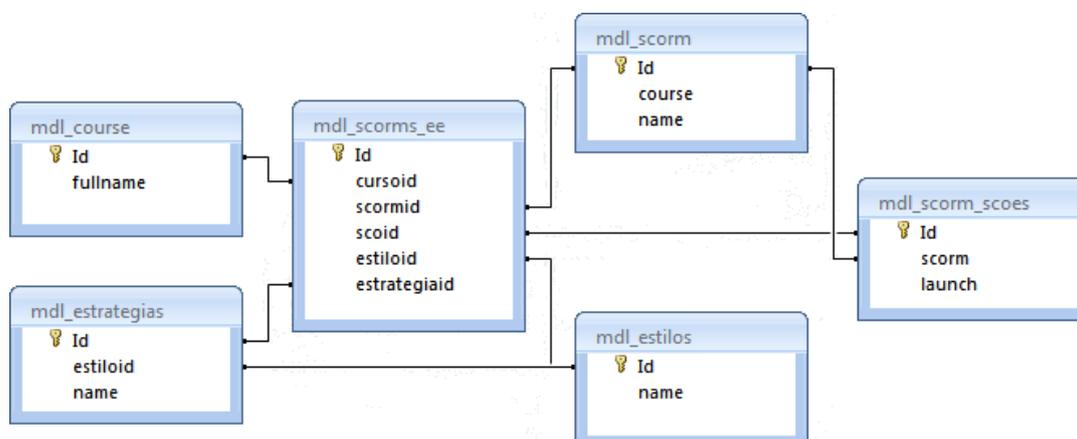


Figura 32. Diagrama entidad-relación de las tablas utilizadas en Vincular tema, estilo y estrategia

El código de programación agregado a Moodle, relacionado con el funcionamiento de esta interfaz, se encuentra distribuido en los archivos incluidos en la Tabla 35.

Tabla 35. Archivos que realizan la Vinculación entre tema, estilo y estrategia

ARCHIVOS	FUNCIÓN
<i>p_scorm_ee.php</i> <i>p_scorm_ee_captura.html</i> <i>p_scorm_ee_bd.php</i>	Realizar el registro de la relación definida mediante la interfaz, entre un tema con un estilo de aprendizaje y una estrategia de enseñanza.
<i>p_scorm_ee_modificar.php</i> <i>p_scorm_ee_modificar_captura.html</i> <i>p_scorm_ee_modificar_bd.php</i>	Realizar alguna modificación en una relación existente entre un tema con un estilo y con una estrategia.
<i>p_scorm_ee_eliminar.php</i>	Realizar la eliminación de la relación existente entre un tema con un estilo y con una estrategia.

En la Tabla 36 se indica la función que desempeña dentro de Moodle, cada uno de estos archivos.

Tabla 36. Función que desempeña cada archivo al Vincular tema, estilo y estrategia

ARCHIVO	UBICADO EN CARPETA	FUNCION
<i>p_scorm_ee.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_scorm_ee_captura.html</i> .
<i>p_scorm_ee_captura.html</i>	Moodle\login	Definir una vinculación entre un tema con un estilo y con una estrategia.
<i>p_scorm_ee_bd.php</i>	Moodle\login	Realiza la conexión con la base de datos para almacenar la información definida.
<i>p_scorm_ee_modificar.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_scorm_ee_modificar_captura.html</i> .
<i>p_scorm_ee_modificar_captura.html</i>	Moodle\login	Actualizar alguna vinculación existente entre un tema, estilo y estrategia.
<i>p_scorm_ee_modificar_bd.php</i>	Moodle\login	Realiza la conexión con la base de datos para actualizar la información modificada.
<i>p_scorm_ee_eliminar.php</i>	Moodle\login	Realiza la conexión con la base de datos para eliminar los datos seleccionados.

Vincular Tema a Examen

Esta interfaz se implementó para vincular a cada uno de los temas de un curso con el respectivo examen que lo evalúa. En la Tabla 37 se muestra la tabla añadida a Moodle, en la cual se almacena la información configurada mediante esta opción, también se

muestran las tablas nativas de Moodle o añadidas a Moodle, utilizadas en realizar la vinculación. En la Figura 33 se muestra el diagrama entidad-relación de las mismas.

Tabla 37. Tablas relacionadas con la opción Vincular un tema a un examen

TABLA	FUNCIÓN
<i>mdl_tema_examen</i>	Almacena la relación establecida, a través de la interfaz de vinculación, entre un tema y un examen, así como el curso y la unidad a la que estos pertenecen.
<i>mdl_quiz</i>	De esta tabla toma los datos del examen que se desea vincular con un tema.
<i>mdl_course_sections</i>	De esta tabla toma los datos de la unidad seleccionada.
<i>mdl_course_modules</i>	De esta tabla toma los datos del recurso (tema) seleccionado para ser vinculado a un examen.
<i>mdl_scorm</i>	De esta tabla toma los datos del paquete SCORM, al cual pertenece el tema seleccionado.

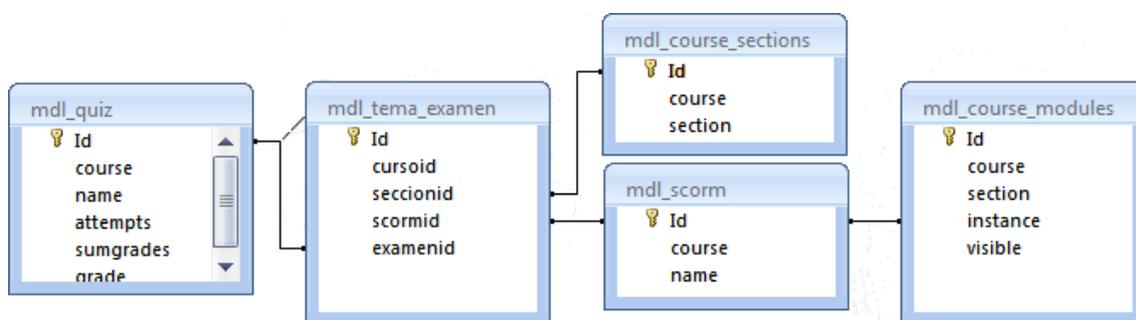


Figura 33. Diagrama entidad-relación de las tablas utilizadas para Vincular un tema a un examen

El código de programación agregado a Moodle, relacionado con el funcionamiento de esta interfaz, se encuentra distribuido en los archivos incluidos en la Tabla 38:

Tabla 38. Archivos creados para Vincular un tema con un examen

ARCHIVOS	FUNCIÓN
<i>p_unidad_tema_examen_ee.php</i> <i>p_unidad_tema_examen_captura.html</i> <i>p_unidad_tema_examen_ee_bd.php</i>	Permite relacionar a un tema con un examen.
<i>p_unidad_tema_examen_ee_modificar.php</i> <i>p_unidad_tema_examen_modificar_captura.html</i> <i>p_unidad_tema_examen_ee_modificar_bd.php</i>	Permite modificar una relación existente entre un tema y un examen.
<i>p_unidad_tema_examen_ee_eliminar.php</i>	Permite eliminar una relación existente entre un tema y un examen.

En la

Tabla 39 se indica la función que desempeña dentro de Moodle, cada uno de estos archivos.

Tabla 39. Función que desempeña cada archivo al Vincular un tema a un examen

ARCHIVO	UBICADO EN CARPETA	FUNCION
<i>p_unidad_tema_examen_ee.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_unidad_tema_examen_captura.html</i> .
<i>p_unidad_tema_examen_captura.html</i>	Moodle\login	Capturar la relación entre un tema y un examen.
<i>p_unidad_tema_examen_ee_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para almacenar la información capturada.
<i>p_unidad_tema_examen_ee_modificar.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_unidad_tema_examen_modificar_captura.html</i> .
<i>p_unidad_tema_examen_modificar_captura.html</i>	Moodle\login	Modificar alguna relación existente entre un tema y un examen.
<i>p_unidad_tema_examen_ee_modificar_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para actualizar la relación modificada.
<i>p_unidad_tema_examen_ee_eliminar.php</i>	Moodle\login	Realizar la conexión con la base de datos para eliminar una relación existente entre un tema y un examen.

Vincular Tema a Competencia

Esta interfaz se desarrolló con el objetivo de vincular a cada uno de los temas de un curso con una o varias competencias específicas. Esta relación existe debido a que las competencias, tienen una influencia causal sobre la calificación que obtenga un alumno al realizar un examen de dicho tema. En la Tabla 40 se muestra la tabla añadida a Moodle, en la cual se almacena la información configurada mediante esta interfaz, también se muestran las tablas nativas de Moodle o añadidas a Moodle, utilizadas en realizar la vinculación. En la Figura 34 se muestra el diagrama entidad-relación de las mismas.

Tabla 40. Tablas relacionadas con la opción Vincular un tema a una competencia

TABLA	FUNCIÓN
<i>mdl_competencias_temas</i>	Almacena la relación de un tema con una o varias competencias, así como el curso al que pertenecen.
<i>mdl_competencias</i>	De esta tabla toma los datos de la competencia a vincular.
<i>mdl_scorm</i>	De esta tabla toma los datos del tema a vincular.

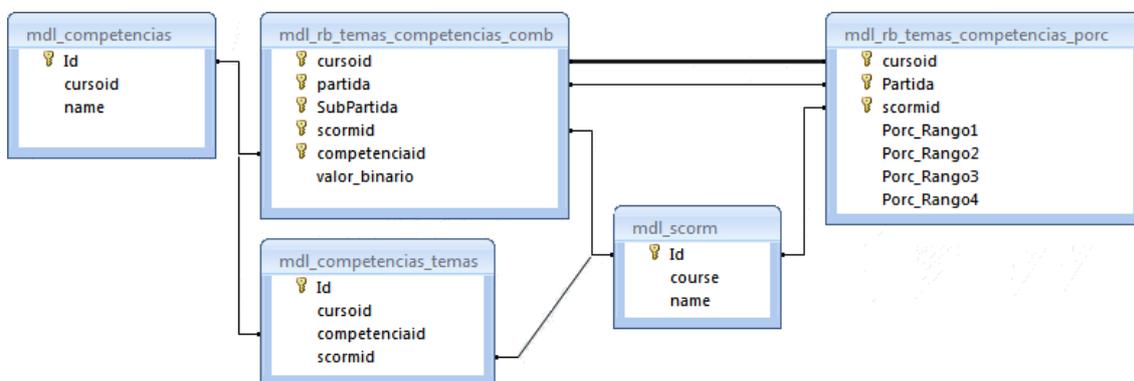


Figura 34. Diagrama entidad-relación de las tablas utilizadas en Vincular un tema a una competencia

El código de programación agregado a Moodle, relacionado con el funcionamiento de este catálogo, se encuentra distribuido en los archivos descritos en la Tabla 41:

Tabla 41. Archivos que realizan la Vinculación entre un tema y una competencia

ARCHIVOS	FUNCIÓN
<i>p_competencias_temas.php</i> <i>p_competencias_temas_captura.html</i> <i>p_competencias_temas_bd.php</i>	Permite relacionar a un tema con una competencia.
<i>p_competencias_temas_modificar.php</i> <i>p_competencias_temas_modificar_captura.html</i> <i>p_competencias_temas_modificar_bd.php</i>	Permite modificar una relación existente entre un tema y una competencia.
<i>p_competencias_temas_eliminar.php</i>	Permite eliminar una relación existente entre un tema y una competencia.

En la Tabla 42 se indica la función que desempeña dentro de Moodle, cada uno de éstos archivos.

Tabla 42. Función que desempeña cada archivo para Vincular un tema a una competencia

ARCHIVO	UBICADO EN CARPETA	FUNCIÓN
<i>p_competencias_temas.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_competencias_temas_captura.html</i> .
<i>p_competencias_temas_captura.html</i>	Moodle\login	Capturar la relación entre un tema y una competencia.
<i>p_competencias_temas_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para almacenar la información capturada.
<i>p_competencias_temas_modificar.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo

		<i>p_competencias_temas_modificar_captura.html</i>
<i>p_competencias_temas_modificar_captura.html</i>	Moodle\login	Modificar alguna relación existente entre un tema y una competencia.
<i>p_competencias_temas_modificar_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para actualizar la relación modificada.
<i>p_competencias_temas_eliminar.php</i>	Moodle\login	Realizar la conexión con la base de datos para eliminar una relación existente entre un tema y una competencia.

5.2.6.4 Captura de las tablas de probabilidad

El proceso para el diagnóstico de las competencias basa su funcionamiento en la representación de una red bayesiana. Como se mencionó en la sección 2.4 del Capítulo 2, una red bayesiana se compone de nodos y de sus respectivas tablas de probabilidad. Los nodos para la red bayesiana de este proyecto de tesis, son representados por las competencias y los temas correspondientes a un curso, y la relación existente entre ellos. Las competencias representa a los nodos padres y los temas representan a los nodos hijos. Considerando lo anterior, fue necesario desarrollar interfaces que le permitan al administrador de Moodle, capturar las probabilidades para cada uno de los nodos que forman a la red bayesiana. Por tal motivo, se agregaron a Moodle las siguientes interfaces: Configurar probabilidades de las competencias y Configurar probabilidades de los temas. Estas interfaces se pueden acceder a través de la opción *Configuración* localizada en el menú *Administración del sitio*, ver sección 5.2.5. El procedimiento para utilizar estas interfaces se describe en el Anexo J. A continuación se describen cada una de estas interfaces.

Capturar las probabilidades de las competencias

Esta interfaz se desarrolló con el objetivo de permitir al administrador de Moodle, realizar la captura de las probabilidades correspondientes a cada uno de los nodos competencia. Cada nodo competencia será de tipo *booleano*, es decir, tendrá dos estados: *Si* y *No*, que serán representados por *1* y *0* respectivamente. El estado *Si* significa la probabilidad de que un alumno posea la competencia, el estado *No* representa lo contrario. La tabla añadida a Moodle, relacionada con esta interfaz, se presenta en la Tabla 43, así como también se muestra la tabla con la que se relaciona. En la Figura 35 se muestra el diagrama entidad-relación de las mismas.

Tabla 43. Tablas relacionadas con la captura de probabilidades para un nodo competencia

TABLA	FUNCIÓN
-------	---------

<i>mdl_rb_competencias_porc</i> (añadida a Moodle)	Almacena las probabilidades de cada una de las competencias que forma parte de un curso determinado.
<i>mdl_competencias</i> (añadida a Moodle)	De esta tabla toma los datos de la competencia de la cual se desea capturar su tabla de probabilidad.

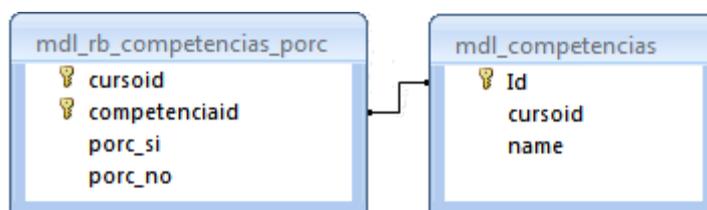


Figura 35. Diagrama entidad-relación de las tablas utilizadas en la captura de las probabilidades de los nodos competencia

El código de programación agregado a Moodle, relacionado con la edición de las tablas de probabilidad de cada nodo competencia, se encuentra distribuido en los archivos descritos en la Tabla 44.

Tabla 44. Archivos añadidos a Moodle para editar las probabilidades de los nodos competencia

ARCHIVOS	FUNCIÓN
<i>p_rb_competencias.php</i> <i>p_rb_competencias_captura.html</i> <i>p_rb_competencias_bd.php</i>	Permite la captura de las probabilidades para los nodos competencia de un curso específico.
<i>p_rb_competencias_modificar.php</i> <i>p_rb_competencias_modificar_captura.html</i> <i>p_rb_competencias_modificar_bd.php</i>	Permite modificar el valor de las probabilidades de un nodo competencia.

En la Tabla 45 se indica la función que desempeña dentro de Moodle, cada uno de éstos archivos. Cabe mencionar que la opción de eliminar alguna probabilidad no se consideró en esta interfaz, ya que se asume que todas las competencias del catálogo deben estar asociadas mínimo a un tema y deben tener asignadas una probabilidad.

Tabla 45. Función que desempeña cada archivo en la edición de las probabilidades de nodos competencia

ARCHIVO	UBICADO EN CARPETA	FUNCIÓN
<i>p_rb_competencias.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_rb_competencias_captura.html</i> .
<i>p_rb_competencias_captura.html</i>	Moodle\login	Permitir la captura de las probabilidades para cada uno de los nodos competencia de un curso específico.
<i>p_rb_competencias_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para almacenar las probabilidades capturadas.
<i>p_rb_competencias_modificar.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_rb_competencias_modificar_captura.html</i> .

<i>p_rb_competencias_modificar_captura.html</i>	Moodle\login	Permitir la modificación de las probabilidades asignadas a un nodo competencia.
<i>p_rb_competencias_modificar_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para actualizar las probabilidades modificadas.

Capturar las probabilidades de los temas

Esta interfaz se desarrolló con el objetivo de permitir al administrador de Moodle, realizar la captura de las probabilidades correspondientes a cada uno de los nodos tema. Cada nodo tema será de tipo *discreto* y tendrá cuatro estados: *Reprobado*, *Suficiente*, *Bueno*, *Excelente*, los cuales son utilizados para ubicar la calificación obtenida por un alumno al contestar un examen. Los rangos de calificación para cada estado son los siguientes: *Reprobado (0-69)*, *Suficiente (70-80)*, *Bueno (81-90)*, *Excelente (91-100)*.

El número de probabilidades a capturar para cada nodo tema, depende del número de competencias con el cual se encuentre relacionado. Por ejemplo:

Considerar que el tema T41 se encuentra relacionado con las competencias C11 y C12, por consiguiente, su tabla de probabilidad condicional para el nodo tema se formará como se muestra en la Tabla 46.

Tabla 46. Tabla de probabilidad condicional para un nodo tema relacionado a dos competencias

C11	C12	REPROBADO	SUFICIENTE	BUENO	EXCELENTE
0	0	0.05	0.07	0.3	0.58
1	0	0.1	0.08	0.35	0.47
0	1	0.3	0.06	0.1	0.54
1	1	0.25	0.44	0.23	0.08

Esta tabla de probabilidad condicional considera:

1. A todas las posibles combinaciones de los estados (*0* y *1*) de las competencias, donde el número de combinaciones se calcula mediante un 2^n , donde el 2 representa a los estados (*0* y *1*) de la competencia y n representa al número de competencias asociadas al nodo tema. En este ejemplo, el número de combinaciones posibles es de $2^2 = 4$.
2. A las probabilidades asignadas a cada estado del nodo tema, relacionado a cada una de las combinaciones del punto 1.

Esto implica que se debe almacenar a cada uno de los valores que forman las combinaciones de los estados de las competencias. El número de valores se calcula como $2^n n = 2^2 2$, por lo tanto se almacenan 8 valores. Así como también se debe almacenar cada una de las probabilidades asignadas a los estados del nodo tema (16 valores), en total se almacenarán en la base de datos, 24 valores.

Ahora, considerando que el tema T41 está relacionado con las competencias C11, C12, C13, se tendrían un total de $2^3 = 8$ combinaciones posibles, dando un total de $2^3 3 = 24$ elementos que forman a las combinaciones, y se tendrían 32 valores correspondientes a las probabilidades asignadas a los estados del nodo tema, dando un total de 56 valores a almacenar.

Como se puede apreciar, la cantidad de información a guardar es variante, ya que está en función de las combinaciones que se pueden realizar, entre los estados de las competencias relacionadas a un tema, y el número de competencias relacionadas a un tema es variable. Por tal motivo, se dificulta la forma de almacenar ésta información en la base de datos.

Para resolver esta problemática, se utilizó el siguiente esquema para almacenar la información:

- a) Se creó la tabla *mdl_rb_temas_competencias_comb*, en la Figura 36 se muestra su estructura.

Column Name	Datatype
cursoid	INT(10)
partida	INT(10)
scormid	INT(10)
competenciaid	INT(10)
valor_binario	INT(10)
SubPartida	INT(10)

Figura 36. Estructura de la tabla *mdl_rb_temas_competencias_comb*

En esta tabla, sólo se almacenan los valores que forman las combinaciones de las competencias. Por ejemplo, para almacenar los valores que componen a las cuatro combinaciones de las competencias de la Tabla 46, se debe insertar en la tabla, “*m*” registros por cada combinación, donde “*m*” es el número de competencias. Por lo tanto, el valor de $m = 2$ porque son dos competencias

involucradas y el número de combinaciones posibles entre los estados de esas competencias es 4, de ésta manera se insertarían $2 \times 4 = 8$ registros en la tabla.

En la Tabla 47 se describe la información que almacena cada uno de los campos de la tabla *mdl_rb_temas_competencias_comb*.

Tabla 47. Información que almacenan los campos de la tabla *mdl_rb_temas_competencias_comb*

CAMPO	CONTENIDO
cursoid	Almacena el identificador del curso al que pertenecen las competencias.
partida	Almacena el número que identifica a una combinación, tantas veces como elementos tenga la combinación.
scormid	Almacena el identificador del tema al que están relacionadas las competencias.
competenciaid	Almacena el identificador de cada competencia, según el orden de aparición de izquierda a derecha.
valor_binario	Almacena los valores de las combinaciones de las competencias en el orden de izquierda a derecha y de arriba hacia abajo.
subpartida	Almacena el número consecutivo que le corresponde a cada elemento de la combinación, comenzando de izquierda a derecha.

En la Tabla 48 se muestra como se almacenaría la información de las competencias del ejemplo de la Tabla 46, considerando lo siguiente: el identificador del curso es 3, el identificador del tema es 41 y el identificador de las competencias C11 y C12, sean 11 y 12 respectivamente.

Tabla 48. Información almacenada en la tabla *mdl_rb_temas_competencias_comb*, considerando la combinación de los estados de dos nodos competencias

cursoid	partida	scormid	competenciaid	valor_binario	subpartida
3	1	41	11	0	1
3	1	41	12	0	2
3	2	41	11	1	1
3	2	41	12	0	2
3	3	41	11	0	1
3	3	41	12	1	2
3	4	41	11	1	1
3	4	41	12	1	2

- b) Se creó la tabla *mdl_rb_temas_competencias_porc*, en la Figura 37 se muestra su estructura.

Column Name	Datatype
cursoid	INT(10)
Partida	INT(10)
scormid	INT(10)
Porc_Rango1	DECIMAL(8,2)
Porc_Rango2	DECIMAL(8,2)
Porc_Rango3	DECIMAL(8,2)
Porc_Rango4	DECIMAL(8,2)

Figura 37. Estructura de la tabla *mdl_rb_temas_competencias_porc*

Esta tabla almacenará la información correspondiente a las probabilidades asignadas por el administrador de Moodle, a cada uno de los estados del nodo tema relacionado a las combinaciones de los nodos competencias. El almacenamiento de la información en esta tabla se realiza de forma normal, ya que sólo basta con saber el número de combinaciones posibles, según el número de competencias involucradas. El almacenamiento de la información de la Tabla 46, se muestra en la Tabla 49. Cabe mencionar que el campo *partida* almacena el número correspondiente a la combinación con la cual se están relacionando los estados del nodo tema.

Tabla 49. Almacenamiento de las probabilidades de los nodos tema

cursoid	partida	scormid	porc_rango1	porc_rango2	porc_rango3	porc_rango4
3	1	41	0.05	0.07	0.3	0.58
3	2	41	0.1	0.08	0.35	0.47
3	3	41	0.3	0.06	0.1	0.54
3	4	41	0.25	0.44	0.23	0.08

En resumen, las tablas agregadas a Moodle relacionadas con la captura de las probabilidades para los nodos tema, se presentan en la Tabla 50 y en la Figura 38 se muestra el diagrama entidad-relación de las mismas.

Tabla 50. Tablas relacionadas con la captura de las probabilidades de un nodo tema

TABLA	FUNCIÓN
<i>mdl_rb_temas_competencias_comb</i> (añadida a Moodle)	Almacena los valores que forman las combinaciones de los estados de las competencias, relacionadas con un tema.
<i>mdl_rb_temas_competencias_porc</i> (añadida a Moodle)	Almacena las probabilidades de los cuatro estados asociados a cada tema.
<i>mdl_competencias_temas</i> (añadida a Moodle)	De esta tabla se obtienen cuáles temas y competencias se encuentran relacionados.
<i>mdl_competencias</i> (añadida a Moodle)	De esta tabla se obtienen los datos de las competencias relacionadas a un tema.
<i>mdl_scorm</i> (nativa de Moodle)	De esta tabla se obtienen los datos del tema asociado a ciertas competencias.

El código de programación agregado a Moodle, relacionado con la captura y almacenamiento de las probabilidades de los nodos tema, se encuentra distribuido en los archivos descritos en la Tabla 51.

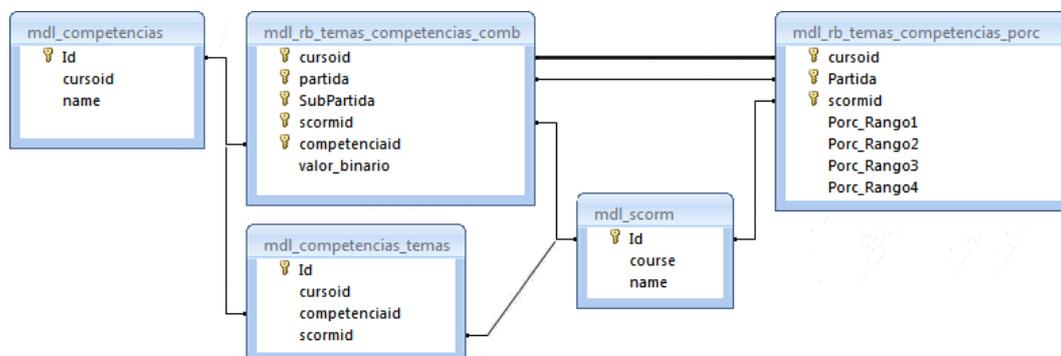


Figura 38. Diagrama entidad-relación de las tablas utilizadas para almacenar las probabilidades de un nodo tema

Tabla 51. Archivos añadidos a Moodle para editar las probabilidades de los nodos tema

ARCHIVOS	FUNCIÓN
<p><i>p_rb_competencias_temas.php</i> <i>p_rb_competencias_temas_captura.html</i> <i>p_rb_competencias_temas_bd.php</i></p>	Permite la captura de las probabilidades para los nodos tema de un curso específico.
<p><i>p_rb_competencias_temas_modificar.php</i> <i>p_rb_competencias_temas_modificar_captura.html</i> <i>p_rb_competencias_temas_modificar_bd.php</i></p>	Permite modificar el valor de las probabilidades de un nodo tema.

En la Tabla 52 se indica la función que desempeña cada uno de éstos archivos. Cabe mencionar que la opción de eliminar alguna probabilidad no se consideró en esta interfaz, ya que se asume que todos los temas del catálogo deben estar asociados mínimo a una competencia y tener asignadas las probabilidades.

Tabla 52. Función que realiza cada archivo para editar las probabilidades de un nodo tema

ARCHIVO	UBICADO EN CARPETA	FUNCIÓN
<i>p_rb_competencias_temas.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_rb_competencias_temas_captura.html</i> .
<i>p_rb_competencias_temas_captura.html</i>	Moodle\login	Permitir la captura de las probabilidades para cada uno de los nodos tema de un curso específico.
<i>p_rb_competencias_temas_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para almacenar las probabilidades capturadas.
<i>p_rb_competencias_temas_modificar.php</i>	Moodle\login	Mostrar el encabezado y pie de página estándar de Moodle y llamar al archivo <i>p_rb_competencias_temas_modificar_captura.html</i> .
<i>p_rb_competencias_temas_mod</i>	Moodle\login	Permitir la modificación de las probabilidades

<i>ificar_captura.html</i>		asignadas a un nodo tema.
<i>p_rb_competencias_temas_modificar_bd.php</i>	Moodle\login	Realizar la conexión con la base de datos para actualizar las probabilidades modificadas.

Capítulo 6

IMPLEMENTACIÓN DEL PROTOTIPO

Después de haber descrito en los capítulos anteriores: *a)* las adaptaciones realizadas a la base de datos de Moodle, *b)* la integración de nuevas interfaces de captura de información, y *c)* la captura de información necesaria, en este capítulo se detalla cómo se utilizaron estos elementos para realizar: *1)* la configuración de una red bayesiana, *2)* la implementación del proceso para el diagnóstico de las competencias y *3)* la implementación del agente selector.

6.1 Configuración de la Red Bayesiana

En esta sección se describe el modelado y la obtención de probabilidades para configurar una red bayesiana dentro de la plataforma Moodle.

6.1.1 Modelado

El modelado de una red bayesiana puede ser realizado en Moodle debido a las adaptaciones implementadas en este proyecto de tesis. Las características que debe seguir la creación de una red son las siguientes: la red sólo puede tener dos niveles, el primer nivel está compuesto por los nodos padre que representan las competencias del curso y el segundo nivel está compuesto por los nodos hijo que representan los temas del curso (ver Figura 39).

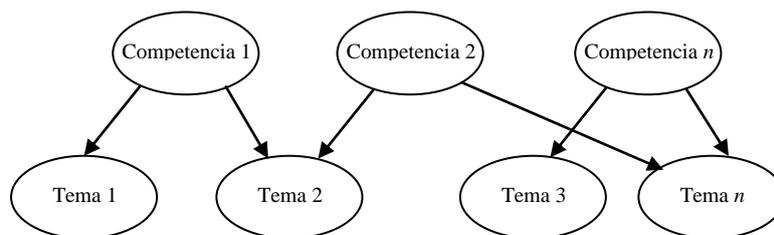


Figura 39. Ejemplo de la estructura que puede tener una red bayesiana en este proyecto de tesis

Las competencias de un curso, representadas por los Nodos Padre tienen dos estados *Si* y *No*. El estado *Si* significa que un alumno tiene desarrollada dicha competencia. El estado *No* significa lo contrario. Estos nodos padre tienen una influencia causal sobre los nodos hijo, los cuales son representados por los temas del curso, ya que si un alumno tiene desarrollada cierta competencia, la cual está asociada a un tema, el alumno al presentar un examen relacionado con dicho tema, debería obtener una calificación aprobatoria.

Los nodos hijo, como ya se mencionó, son representados por cada uno de los temas de un curso, estos tienen cuatro estados *Reprobado*, *Suficiente*, *Bueno*, *Excelente*. Cada estado representa un rango, en el cual se puede ubicar la calificación obtenida por un alumno al contestar un examen de un tema. Los rangos de calificación que considera cada estado son los siguientes: *Reprobado (0-69)*, *Suficiente (70-80)*, *Bueno (81-90)*, *Excelente (91-100)*.

En cuanto a las relaciones que pueden existir en la configuración de una red bayesiana en Moodle se tiene, que una competencia puede estar relacionada a varios temas, o visto desde otra perspectiva, un mismo tema puede estar influenciado por varias competencias.

Para definir y configurar una red bayesiana en Moodle, se deben realizar las siguientes acciones:

1. Crear un curso.
2. Capturar las competencias específicas correspondientes al curso.
3. Importar al curso todos los temas que se desean enseñar (paquetes SCORM).
4. Crear los exámenes correspondientes a cada tema.
5. Realizar la vinculación de cada tema con el estilo y la estrategia correspondiente.
6. Realizar la vinculación entre cada tema y su respectivo examen.
7. Realizar la vinculación entre cada tema y la(s) competencias asociadas a él.
8. Capturar las probabilidades de los nodos competencia.
9. Capturar las probabilidades de los nodos tema.

Estas mismas acciones fueron realizadas para configurar la red bayesiana utilizada en la experimentación de esta tesis. La información considerada en la definición de la red se muestra en la Tabla 53.

Tabla 53. Información considerada para crear la red bayesiana para la experimentación

CURSO	UNIDAD#	TEMAS	COMPETENCIA ASOCIADA A TEMAS	ESTRATEGIAS UTILIZADAS PARA CADA ESTILO
Fundamentos de Programación	5	Arreglos	Construir programas que utilicen arreglos unidimensionales y multidimensionales para solucionar problemas.	Textos subrayados-Visual Vídeo con sonido-Auditivo Texto-Lectura/Escritura Ejemplos-Kinestesico
		Matrices		Textos subrayados-Visual Vídeo con sonido-Auditivo Texto-Lectura/Escritura Ejemplos-Kinestesico

La representación gráfica de la información anterior mediante una red bayesiana se muestra en la Figura 40, donde el *nodo C* representa a la competencia, la cual se encuentra relacionada con el *nodo A* que representa al tema *Arreglos* y con el *nodo M* que representa al tema *Matrices*.

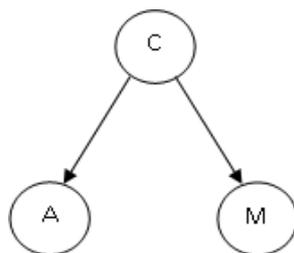


Figura 40. Representación gráfica de la red bayesiana utilizada en la experimentación de esta tesis

En la siguiente sección se detalla el cálculo realizado para obtener los parámetros (probabilidades) para definir las tablas de probabilidad de cada nodo de la red.

6.1.2 Obtención de las probabilidades

Los valores para definir las tablas de probabilidad de la red bayesiana se obtuvieron mediante una actividad realizada con un grupo de alumnos. La actividad consistió en concentrar a los alumnos en el centro de cómputo del ITCM, para que interactuaran con la plataforma Moodle, revisando los temas *Arreglos* y *Matrices* y presentando el examen correspondiente a cada tema. Estos temas corresponden a la unidad 5 del programa para la materia de Fundamentos de Programación. Los alumnos que formaron parte de esta actividad cursaban el primer semestre de la carrera de Ingeniería en

Sistemas Computacionales en el ITCM. Participaron 11 alumnos con una edad promedio de 19 años.

En las siguientes tablas se muestran las calificaciones obtenidas por los alumnos al contestar el examen de cada tema. Los alumnos que se encuentran remarcados en negrita, son aquellos que, de acuerdo al criterio proporcionado por el profesor encargado del grupo, tienen desarrollada la competencia asociada a estos temas.

TEMA "ARREGLOS"	
ALUMNO	CALIFICACIÓN
1	75
2	70
3	100
4	80
5	95
6	80
7	90
8	45
9	70
10	75
11	85

TEMA "MATRICES"	
ALUMNO	CALIFICACIÓN
1	100
2	80
3	90
4	100
5	90
6	100
7	100
8	40
9	90
10	90
11	90

Utilizando la información obtenida de los alumnos, se procedió a realizar una tabla de frecuencias para cada tema. Cada tabla considera ubicar la calificación obtenida por cada alumno, de acuerdo a los estados de los nodos tema y competencia. Como se ha mencionado anteriormente, los estados de los nodos tema son representados por los siguientes rangos *Reprobado (0-69)*, *Suficiente (70-80)*, *Bueno (81-90)*, *Excelente (91-100)*, los cuales son representados en la tabla por las siguientes letras *R*, *S*, *B* y *E* respectivamente y los estados para los nodos competencia son *SI* y *No*. Considerando lo anterior, la tabla de frecuencias para cada tema es la que se muestra a continuación.

COMPETENCIA	TEMA "ARREGLOS"				
	R	S	B	E	TOTAL
SI	0	2	1	1	4
NO	1	4	1	1	7
TOTAL	1	6	2	2	11

COMPETENCIA	TEMA "MATRICES"				
	R	S	B	E	TOTAL
SI	0	0	3	1	4
NO	1	1	2	3	7
TOTAL	1	1	5	4	11

A partir de la información proporcionada por el profesor encargado del grupo, se obtuvieron las probabilidades *a priori* para el nodo competencia, y para los nodos tema se calcularon las probabilidades a partir de la información contenida en las tablas de frecuencia.

Las probabilidades para el nodo competencia se obtuvieron considerando que 4 de los 11 alumnos sí tiene desarrollada la competencia, y los 7 alumnos restantes no. Por lo que la tabla de probabilidad *a priori* para el nodo competencia quedó definida de la siguiente forma:

COMPETENCIA	
SI	0.36
NO	0.64

Las probabilidades condicionales para cada nodo tema se calcularon utilizando: la información de su tabla de frecuencia correspondiente y la información de la tabla de probabilidad *a priori* de la competencia asociadas al nodo tema. Una vez que se tiene identificada la información, se utiliza la fórmula de probabilidad condicional, la cual se muestra a continuación:

$$P(y_j | x_i) = \frac{P(x_i, y_j)}{P(x_i)} \quad \text{donde} \quad P(x_i) > 0$$

Un ejemplo del empleo de esta fórmula sería: considerar que un alumno haya obtenido una calificación entre 0 y 69 (rango R) al presentar el examen del tema *Arreglos* y que además sí tenga desarrollada la competencia, esto se representa de la siguiente forma:

$$P(R | C) = \frac{P(R \cap C)}{P(C)}$$

Donde $P(R \cap C) = \frac{0}{11}$ y $P(C) = 0.36$, por lo tanto $P(R | C) = \frac{\left(\frac{0}{11}\right)}{0.36} = 0$

Otro ejemplo sería, obtener la probabilidad condicional de que un alumno haya obtenido una calificación entre 0 y 69 (rango R) al presentar el examen del tema *Arreglos* y que además *no* tenga desarrollada la competencia, aplicando la fórmula el cálculo sería:

$$P(R | \neg C) = \frac{P(R \cap \neg C)}{P(\neg C)} = \frac{\left(\frac{1}{11}\right)}{0.64} = 0.14$$

De igual forma se calcularon las probabilidades para los dos temas, quedando definidas las tablas de probabilidad condicional para cada nodo, de la siguiente forma:

TEMA "ARREGLOS"				
COMPETENCIA	R	S	B	E
SI	0.00	0.50	0.25	0.25
NO	0.14	0.58	0.14	0.14

TEMA "MATRICES"				
COMPETENCIA	R	S	B	E
SI	0.00	0.00	0.75	0.25
NO	0.14	0.14	0.29	0.43

Como ya se había mencionado, estas probabilidades son las que se emplearon para configurar la red bayesiana utilizada en la experimentación de esta tesis, siendo capturadas mediante las interfaces descritas en la sección 5.2.6.4 del capítulo 5. En la Figura 41 se muestra la representación gráfica de la red bayesiana y las tablas de probabilidad para cada nodo. El nodo competencia está representado por la letra *C*, y sus estados *Si* y *No*, están representados por las letras *S* y *N*, respectivamente. Los nodos *Arreglos* y *Matrices*, están representado por las letras *A* y *M*, y sus estados *Reprobado*, *Suficiente*, *Bueno*, *Excelente*, están representados por las letras *R*, *S*, *B*, *E* respectivamente.

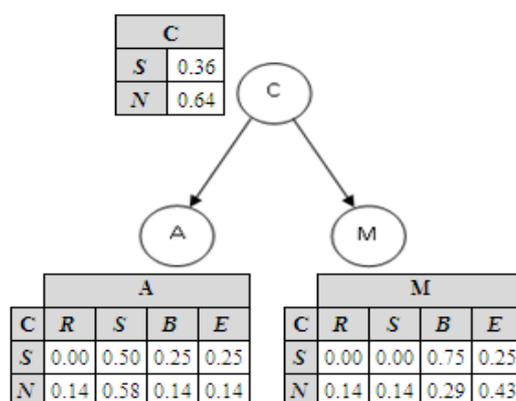


Figura 41. Red bayesiana

6.2 Implementación del proceso para el diagnóstico de las competencias

La implementación de éste proceso, tiene como fin el poder diagnosticar la probabilidad existente de que un alumno haya obtenido cierta calificación, a causa de que posee la competencia requerida. Una de las fases del proceso para el diagnóstico de las competencias, fue buscar un software que realizara inferencia sobre una red bayesiana, entre los cuales se revisaron: el software comercial AgenaRisk y el software Elvira. De

estos, se decidió utilizar el software Elvira, debido a que, es de código libre y posee varios métodos para realizar inferencia sobre una red bayesiana (ver sección 2.4.1 del capítulo 2). Entre los métodos con los que cuenta Elvira, se encuentra el método de Eliminación de Variables, el cual fue elegido para realizar la inferencia en este proyecto de tesis, ya que es un algoritmo exacto y de los más sencillos de entender.

Una vez definido, el modelado de la red bayesiana, sus probabilidades y el software para realizar inferencia sobre la red, se continuó con la implementación de las otras fases del proceso para el diagnóstico de las competencias. Cada una de estas fases se menciona y describe a continuación:

- Ubicar en Moodle, el punto donde se genera una nueva evidencia.
- Realizar la inferencia sobre la red bayesiana.
- Analizar el resultado obtenido de la inferencia.
- Determinar si el Agente Selector se ejecutará.

6.2.1 Ubicación de nuevas evidencias

Para realizar inferencia sobre una red bayesiana, es necesario conocer el estado en el que se encuentran algunas de sus variables (nodos), lo cual se conoce como *evidencia*. En este proyecto de tesis se considera como *evidencia*, a la *calificación obtenida* por un alumno al contestar un examen. Por tal motivo, fue necesario determinar en que punto de Moodle, se generan nuevas evidencias, es decir, en que lugar se genera la calificación de un examen contestado por un alumno. Después de analizar el funcionamiento de los exámenes, se determinó que el archivo *attempt.php* nativo de Moodle ubicado en la carpeta `\moodle\mod\quiz\`, se encarga de almacenar en la base de datos, la información relacionada con los exámenes presentados por los alumnos, es decir, en este archivo se generan nuevas evidencias. Habiendo detectado el punto donde se originan las evidencias, se determinó incluir código en este archivo, para utilizar algunos objetos contenidos en él. Estos objetos son utilizados para realizar el proceso de inferencia. El código añadido a este archivo es el siguiente:

```
if (CalculaProbabilidad($course->id, $USER->id, $quiz->id, $attempt->attempt))
{
    agente_selector($attempt->uniqueid, $attempt->attempt, $attempt->sumgrades,
    .....
    $course->id, $quiz->id);
}
```

El código comienza con la sentencia *If*, la cual llama a la función *CalculaProbabilidad*, si ésta devuelve un valor *verdadero*, se cumple la condición, por lo que se hará el llamado de la función *agente_selector*, la cual se explicará en la sección 6.3 de este capítulo.

Por otro lado, la función *CalculaProbabilidad* es donde se lleva a cabo el proceso de inferencia sobre la red bayesiana considerando las evidencias. Esta función requiere de ciertos parámetros, los cuales son: el identificador del curso donde se encuentra el alumno, el identificador del alumno, el identificador del examen contestado y el identificador del intento realizado por el alumno. Estos parámetros son los objetos obtenidos del archivo *attempt.php* y los cuales son enviados a la función *CalculaProbabilidad*. Cabe mencionar que estos elementos serán referidos a lo largo de este capítulo. En la siguiente sección se describe el funcionamiento de *CalculaProbabilidad*.

6.2.2 Inferencia sobre la red bayesiana

Como se mencionó en la sección anterior, la función *CalculaProbabilidad* es la que se encarga del proceso para realizar la inferencia sobre la red bayesiana. Cabe recordar que la inferencia es llevada a cabo mediante el software *Elvira*, el cual es un programa externo a la plataforma Moodle. Debido a esto, una de las acciones que se efectúan en la función *CalculaProbabilidad*, es realizar la comunicación entre Moodle y el software *Elvira*. Esta comunicación considera el intercambio de información entre ambas plataformas, para lo cual se utilizaron archivos de texto que siguen un cierto formato. Esto fue posible, debido a que *Elvira* puede leer la información necesaria para realizar la inferencia, a partir de archivos de texto, los cuales deben seguir una cierta estructura para que puedan ser interpretados por *Elvira*. De igual forma, una vez que *Elvira* interpreta y procesa la información, genera un archivo de texto con un determinado formato, en el cual guarda los resultados obtenidos de la inferencia. Fue por estas razones que se decidió utilizar archivos de texto, para realizar la comunicación entre las dos plataformas.

La información que Moodle prepara y envía a *Elvira* son dos archivos de texto, un archivo contiene la *estructura de la red* (los nodos que forman la red, las relaciones

existentes entre ellos y las tablas de probabilidad de cada nodo), y el otro archivo contiene la información de las *evidencias observadas*, es decir las calificaciones de todos los exámenes correspondientes a un curso, que haya contestado un alumno.

Por otra parte, una vez que *Elvira* procesa la información de los archivos enviados por Moodle y realiza la inferencia sobre la red bayesiana, genera un archivo de texto, en el cual almacena los *resultados obtenidos* y el cual será decodificado por Moodle para extraer los resultados.

En total serán tres archivos con los que se realizará el intercambio de información entre Moodle y *Elvira*.

En esta sección sólo se detalla la implementación de los dos archivos generados por Moodle y enviados a *Elvira*, en la sección 6.2.3 se abordará la implementación del archivo generado por *Elvira* en donde almacena los resultados de la inferencia.

La primera acción que realiza la función *CalculaProbabilidad* para formar a los archivos con los que trabajarán Moodle y *Elvira*, es declarar una variable para cada archivo, la cual almacenará la ruta donde estos serán creados, así como el nombre que le será asignado a cada uno, el cual se formará de acuerdo a los parámetros recibidos en esta función. A continuación se muestra el código que realiza esta acción:

```

$NombreArchivoElvira      = "c:\\appserv\\www\\moodle\\C_" . $CursoID . $UserID . ".elv";
$NombreArchivoEvidencias = "c:\\appserv\\www\\moodle\\C_" . $CursoID . $UserID . ".evi";
$NombreArchivoResultados = "c:\\appserv\\www\\moodle\\C" . $CursoID . $UserID . ".txt";

```

La primera variable tendrá asignada la ruta y nombre del archivo donde Moodle creará y almacenará la información de *la estructura de la red bayesiana*. El nombre que tendrá este archivo se formará iniciando con la letra *C* seguida de un guío bajo, al cual se le concatenará el identificador del curso, el identificador del alumno y la extensión *elv*, ya que *Elvira* requiere que este archivo tenga dicha extensión. Por ejemplo, suponer que el identificador del curso es *3* y que el identificador del alumno es *15*, por lo tanto el nombre asignado al archivo será *C_315.elv*.

La segunda variable tendrá asignada la ruta y nombre del archivo donde Moodle creará y almacenará la información correspondiente a todas *las evidencias observadas*. El nombre que se asignará a este archivo se formará igual que el del archivo anterior,

sólo cambia la extensión, para este archivo debe ser *evi*, ya que *Elvira* requiere que este archivo tenga dicha extensión. Considerando los datos anteriores para formar el nombre del archivo, éste sería *C_315.evi*.

La tercera variable tendrá asignada la ruta y nombre del archivo que le indican a *Elvira* en donde debe crear y almacenar la información correspondiente a *los resultados obtenidos de la inferencia*. El nombre que se asignará a este archivo se formará igual que los anteriores, sólo cambia la extensión, la cual debe ser *txt*, ya que Moodle requiere que este archivo tenga dicha extensión. Considerando los datos anteriores para formar el nombre del archivo, éste sería *C_315.txt*.

Posterior a la creación de las variables, en donde se almacena la ruta y nombre de los archivos, se hace un llamado a las funciones:

```
GeneraFormatoElvira($NombreArchivoElvira, $CursoID, $UserID);
GeneraEvidenciasFormatoElvira($NombreArchivoEvidencias, $CursoID, $UserID);
```

La primera función se encarga de recopilar la información para crear el archivo de texto con la estructura de la red (ver sección 6.2.2.1). La segunda función se encarga de recopilar la información para crear el archivo de texto con las evidencias de la red (ver sección 6.2.2.2). A continuación se describen las acciones realizadas por cada una de estas funciones.

6.2.2.1 Creación del archivo que contiene la estructura de la red bayesiana

La creación del archivo que contendrá la estructura de la red bayesiana, se realizó mediante la implementación de la función:

```
GeneraFormatoElvira($NombreArchivoElvira, $CursoID, $UserID);
```

Esta función recibe como parámetros el nombre del archivo a generar, el identificador del curso y el identificador del alumno. Dentro de esta función se hace el llamado a otras funciones, las cuales obtendrán de la base de datos de Moodle, la información necesaria para generar a cada uno de los elementos que formará la estructura de la red bayesiana que se guardará en el archivo de texto *elv*. En la Tabla 54

se muestran estas funciones y la actividad que desempeña cada una. Posteriormente se describe a detalle cada una de ellas.

Tabla 54. Funciones para formar el archivo que contiene la estructura de la red bayesiana

FUNCIÓN	ACTIVIDAD DESEMPEÑADA
<i>GeneraFormatoElvira_Encabezado</i>	Escribe en el archivo de texto, el encabezado requerido por el formato <i>Elvira</i> .
<i>GeneraFormatoElvira_Variable</i>	Recupera de la base de datos la información correspondiente a las variables que forman la red (nodos competencia y nodos tema) y escribirla en el archivo.
<i>GeneraFormatoElvira_Relacion</i>	Recuperar de la base de datos, las relaciones existentes entre los nodos competencia y los nodos tema, y escribirlas en el archivo.
<i>GeneraFormatoElvira_RelacionValores</i>	Recuperar de la base de datos, los valores de las probabilidades de cada nodo competencia y cada nodo tema, y escribirlas en el archivo.

Función *GeneraFormatoElvira_Encabezado*

Esta función es la única que no necesita acceder a la información de la base de datos, ya que sólo se encarga de escribir en el archivo de texto, el encabezado requerido por el formato *Elvira*. El código de esta función se muestra a continuación:

```
function GeneraFormatoElvira_Encabezado($file) {
    fputs($file, "// Bayesian Network \n");
    fputs($file, "// Elvira format \n\n");
    fputs($file, "bnet \"Moodle\" { \n\n");
    fputs($file, "// Network Properties \n\n");
    fputs($file, "kindofgraph = \"directed\"; \n");
    fputs($file, "visualprecision = \"0.00\"; \n");
    fputs($file, "version = 1.0; \n");
    fputs($file, "default node states = (presente , ausente); \n\n");
    fputs($file, "// Variables \n\n");
}
```

Función *GeneraFormatoElvira_Variable*

Mediante la función *GeneraFormatoElvira_Variable*, se escribe en el archivo de texto el nombre de cada nodo (variable) que forma a la red bayesiana, el número de estados de cada nodo y el nombre de cada uno de los estados de cada nodo. Esto tanto para los nodos competencia como para los nodos tema.

Antes de llamar a esta función, se realizan consultas a la base de datos de Moodle para obtener la información acerca de los nodos competencia y de los nodos tema que forman a la red bayesiana. Enseguida se detallan las consultas a la base de datos:

a) Consulta para obtener a todas las competencias relacionadas con el curso actual

```
// COMPETENCIAS DEL CURSO SELECCIONADO
$rs_Competencias = get_recordset_sql("SELECT cp.CompetenciaId
                                     FROM mdl_rb_competencias_porc cp
                                     WHERE cpCursoId = $CursoID");

while ($row = rs_fetch_next_record($rs_Competencias)) {
    GeneraFormatoElvira_Variable($file, "C".$row->CompetenciaId, 2, "\\no\\ \\si\\");
}
rs_close($rs_Competencias);
```

Con la función *get_recordset_sql*, nativa de Moodle, se ejecuta la consulta adecuada para traer el identificador de las competencias que tienen definidos valores de probabilidad y que pertenecen al identificador del curso actual. Mediante la función *GeneraFormatoElvira_Variable*, se escribe en el archivo de texto el nombre de la variable que va a representar a la competencia en la red bayesiana, el número de estados definidos para las competencias y el nombre de cada uno de sus estados. El nombre que identificará a cada una de las competencias correspondientes al curso, se formará concatenando la letra “C” con el identificador de la competencia correspondiente. Por ejemplo, si existiera una competencia en la base de datos relacionada con el curso y su identificador fuera 11, el nombre que identificará a esta competencia y que será escrito en el archivo de texto sería C11. El número de estados para las competencias está definido en dos y los nombres de cada uno de estos estados es “no” y “si”.

b) Consulta para obtener a todos los temas correspondientes al curso actual.

```
// TEMAS
$rs_Temas = get_recordset_sql("SELECT DISTINCT tc.ScormId
                              FROM mdl_rb_temas_competencias_porc tc
                              WHERE tcCursoId = $CursoID
                              ORDER BY tc.ScormId");

while ($row = rs_fetch_next_record($rs_Temas)) {
    GeneraFormatoElvira_Variable($file, "T".$row->ScormId, 4, "\\r0\\ \\r1\\ \\r2\\ \\r3\\");
}
rs_close($rs_Temas);
```

El funcionamiento del código anterior, es similar al descrito en el inciso a). La diferencia radica en que la consulta a ejecutar, ahora extrae de la base de datos, la información de los temas que tengan probabilidades asignadas y pertenezcan al

curso actual. El nombre que identificará a cada uno de los temas correspondientes al curso, se formará concatenando la letra “T” con el identificador del tema correspondiente. Por ejemplo, si existiera un tema en la base de datos relacionado con el curso y su identificador es 41, el nombre que identificará a este tema y que será escrito en el archivo de texto sería T41.

Como se puede observar, después de cada consulta realizada a la base de datos, se hace el llamado de la función *GeneraFormatoElvira_Variable*, la cual escribe en el archivo de texto la información correspondiente a cada uno de los nodos que forman a la red. A continuación se muestra el código de esta función.

```
function GeneraFormatoElvira_Variable($file, $NombreVariable, $NumStates, $States) {
    fputs($file, "node $NombreVariable(finite-states) { \n");
    fputs($file, "kind-of-node = chance; \n");
    fputs($file, "type-of-variable = finite-states; \n");
    fputs($file, "pos_x =459; \n");
    fputs($file, "pos_y =86; \n");
    fputs($file, "relevance = 7.0; \n");
    fputs($file, "purpose = \"\"; \n");
    fputs($file, "num-states = $NumStates; \n");
    fputs($file, "states = ($States); \n");
    fputs($file, "} \n\n");
}
```

Función GeneraFormatoElvira_Relacion

Mediante la función *GeneraFormatoElvira_Relacion*, se escribe en el archivo de texto las relaciones existentes entre los nodos competencia y tema, que forman a la red bayesiana. Antes de realizar en el archivo de texto la escritura de las relaciones, se realizan dos consultas a la base de datos de Moodle. Las cuales se ilustran en el siguiente código:

```
fputs($file, "// Links of the associated graph: \n\n");
$rs_Temas = get_recordset_sql("SELECT DISTINCT tc.ScormId
                                FROM mdl_rb_temas_competencias_porc tc
                                WHERE tcCursoId = $CursoID
                                ORDER BY tc.ScormId");

while ($row = rs_fetch_next_record($rs_Temas)) {

    $rs_Tmp = get_recordset_sql("SELECT DISTINCT CompetenciaId
                                FROM mdl_rb_temas_competencias_comb m
                                WHERE ScormId = $row->ScormId
                                ORDER BY Partida, SubPartida");

    while ($rowTmp = rs_fetch_next_record($rs_Tmp)) {
        GeneraFormatoElvira_Relacion($file, "C".$rowTmp->CompetenciaId, "T".$row->ScormId);
    }
    rs_close($rs_Tmp);
}
rs_close($rs_Temas);
```

La primera consulta extrae de la base de datos los identificadores de los temas y se recorren cada uno de ellos mediante el ciclo *While*. Dentro de este ciclo, se ejecuta la segunda consulta, la cual extrae las competencias que influyen en el tema que se está procesando en cada iteración del primer *While*. Cada una de las competencias extraídas, relacionadas con el tema que se procesa en cada iteración, se escribe en el archivo de texto mediante la función *GeneraFormatoElvira_Relacion*. El código de esta función mediante el cual realiza la escritura en el archivo de texto se muestra a continuación:

```
function GeneraFormatoElvira_Relacion($file, $VariableOrigen, $VariableDestino) {
    fputs($file, "link $VariableOrigen $VariableDestino; \n\n");
}
```

Función *GeneraFormatoElvira_RelacionValores*

Los últimos elementos a escribir en el archivo de texto, son las probabilidades de cada uno de los nodos (variables), esto se lleva a cabo mediante la función *GeneraFormatoElvira_RelacionValores*. El llamado de esta función se realiza dos veces, pero antes se efectúan las siguientes consultas a la base de datos:

- a) Consulta para obtener las probabilidades correspondientes a los nodos competencia a través del siguiente código:

```
fputs($file, "//Network Relationships: \n\n");
//COMPETENCIAS
$rs_Competencias = get_recordset_sql("SELECT cp.CompetenciaId, Porc_NO, Porc_SI
                                     FROM mdl_rb_competencias_porc cp
                                     WHERE cp.CursorId = $CursoID");

while ($row = rs_fetch_next_record($rs_Competencias)) {
    (GeneraFormatoElvira_RelacionValores)($file, "C".$row->CompetenciaId, $row->Porc_NO." ".$row->Porc_SI);
}
rs_close($rs_Competencias);
```

Mediante la función *get_recordset_sql*, se extraen los porcentajes *NO* y *SI* de las competencias definidas para el curso actual (las probabilidades de las competencias fueron capturadas utilizando la interfaz descrita en la sección 5.2.6.4). Dentro del ciclo *While*, la función *GeneraFormatoElvira_RelacionValores*, escribe en el archivo de texto el nombre de la variable que representa a las competencias y sus propiedades *NO* y *SI*.

- b) Consulta para obtener las probabilidades correspondientes a los temas mediante el siguiente código:

```

$rs_Temas = get_recordset_sql("SELECT DISTINCT tc.ScoreId
                             FROM mdl_rb_temas_competencias_porc tc
                             WHERE tc.CursoId = {CursoId}
                             ORDER BY tc.ScoreId");

while ($row = rs_fetch_next_record($rs_Temas)) {
    $CadenaVariables = "T".$row->ScoreId." ";

    $rs_Tap = get_recordset_sql("SELECT DISTINCT CompetenciaId
                              FROM mdl_rb_temas_competencias_comb m
                              WHERE ScoreId = $row->ScoreId
                              ORDER BY Partida, SubPartida DESC");

    while ($rowTap = rs_fetch_next_record($rs_Tap)) {
        $CadenaVariables .= "C".$rowTap->CompetenciaId." ";
    }
    rs_close($rs_Tap);

    $CadenaProbRango1 = "";
    $CadenaProbRango2 = "";
    $CadenaProbRango3 = "";
    $CadenaProbRango4 = "";

    $rs_Tap = get_recordset_sql("SELECT tc.Forc_Rango1, tc.Forc_Rango2, tc.Forc_Rango3, tc.Forc_Rango4
                              FROM mdl_rb_temas_competencias_porc tc
                              WHERE tc.ScoreId = $row->ScoreId
                              ORDER BY tc.Partida");

    while ($rowTap = rs_fetch_next_record($rs_Tap)) {
        $CadenaProbRango1 .= $rowTap->Forc_Rango1." ";
        $CadenaProbRango2 .= $rowTap->Forc_Rango2." ";
        $CadenaProbRango3 .= $rowTap->Forc_Rango3." ";
        $CadenaProbRango4 .= $rowTap->Forc_Rango4." ";
    }
    rs_close($rs_Tap);
    GeneraFormatoElvira_RelacionValores($file, $CadenaVariables, $CadenaProbRango1, $CadenaProbRango2, $CadenaProbRango3, $CadenaProbRango4);
}
rs_close($rs_Temas);

```

En el código anterior, se obtienen mediante la función *get_recordset_sql* los identificadores de los temas del curso actual. Dentro del ciclo *While* se realizan principalmente dos tareas:

- 1) Se debe generar una variable de tipo cadena, la cual debe contener las relaciones de cada tema con las respectivas competencias (por ejemplo: T41 C11 C12). Para esto, se procesa cada uno de los identificadores extraídos en la primera consulta, inicializando una variable de tipo cadena con el valor del identificador del tema. Posteriormente, se obtiene de la tabla *mdl_rb_temas_competencias_comb* los identificadores de las competencias que están relacionadas con el tema. Se recorre dentro del ciclo *While*, cada una de las competencias asociadas al tema, concatenando a la cadena previamente inicializada, el identificador de las competencias asociadas al tema.
- 2) Después de generar la cadena de relaciones, ahora se procede a generar una variable de tipo cadena por cada uno de los estados del tema (son 4 estados). En cada variable de tipo cadena, se concatenan las probabilidades de su respectivo rango, extrayendo las probabilidades de

la tabla *mdl_rb_temas_competencias_porc*. Habiendo generado la cadena con la relación del tema a sus competencias asociadas y las cadenas de las probabilidades de cada uno de los estados del tema, se escriben las cadenas en el archivo de texto mediante la función *GeneraFormatoElvira_RelacionValores*.

Como se puede observar, después de cada consulta realizada a la base de datos, la función *GeneraFormatoElvira_RelacionValores* es llamada para que realice la escritura de la información en el archivo de texto. El código de esta función se muestra a continuación:

```
function GeneraFormatoElvira_RelacionValores($file, $NombreVariable, $Valores) {
    fputs($file, "relation $NombreVariable { \n");
    fputs($file, "comment = \"new\"; \n");
    fputs($file, "kind-of-relation = potential; \n");
    fputs($file, "deterministic=false; \n");
    fputs($file, "values= table ($Valores); \n");
    fputs($file, "} \n\n");
}
```

De esta manera se forma el archivo de texto *elv* que contiene la estructura de la red bayesiana en el formato que la plataforma *Elvira* reconoce. En el Anexo K se puede ver un ejemplo de este formato. En la siguiente sección se detalla la forma en la que se genera el archivo que contiene la información de las evidencias.

6.2.2.2 Creación del a archivo que contienen a las evidencias

La creación del archivo que contendrá a las evidencias, se realizó mediante la implementación de la función:

```
GeneraEvidenciasFormatoElvira($NombreArchivoEvidencias, $CursoID, $UserID);
```

Esta función recibe como parámetros el nombre del archivo a generar, el identificador del curso y el identificador del alumno. El primer paso realizado en esta función, fue simplemente escribir en el archivo de texto, el encabezado requerido por el formato de *Elvira*. Para esto no fue necesario acceder a la información de la base de datos de Moodle. Esto se realizó mediante el siguiente código:

```
function GeneraEvidenciasFormatoElvira($NombreArchivoFormatoEvidencias, $CursoID, $UserID) {
    $file = fopen($NombreArchivoFormatoEvidencias, 'w');
    fputs($file, "// Evidence case \n");
    fputs($file, "// Elvira format \n\n");
    fputs($file, "evidence NoName { \n\n");
    fputs($file, "// Evidence Properties\n\n");
    fputs($file, "title = \"Untitled\";\n");
    fputs($file, "version = 1.0;\n\n");
}
```

El siguiente paso fue extraer las evidencias, es decir, las calificaciones de todos los exámenes que haya presentado un alumno en el curso actual. Esto se realizó mediante el siguiente código:

```
$rs_Examenes = get_recordset_sql("SELECT q.id, q.name, te.scormid
                                FROM mdl_quiz q, mdl_tema_examen te
                                WHERE (q.id = te.examenid) AND
                                       q.course = $CursoID");

while ($row = rs_fetch_next_record($rs_Examenes)) {
    $rs_CalificacionExamen = get_recordset_sql("SELECT qq.Grade
                                                FROM mdl_quiz_grades qq
                                                WHERE qq.UserId = $UserID AND qq.Quiz = $row->id");
    $rowCalExa = rs_fetch_next_record($rs_CalificacionExamen);

    if ($rowCalExa != "") {
        fputs($file, "T".$row->scormid." = ".DeterminaRangoELVIRA($rowCalExa->Grade).",\n");
    }
    rs_close($rs_CalificacionExamen);
}
rs_close($rs_Examenes);
```

La primera consulta del código anterior, enlaza la información de todos los exámenes del curso con la información de los temas relacionados a cada examen. Esto se realizó debido a que a través del identificador del examen, se obtiene el identificador del tema asociado a dicho examen.

Después de obtener los identificadores de los temas, a partir del identificador de los exámenes, se recorre cada uno de los identificadores de los temas mediante el ciclo *While*. En cada iteración, el identificador del tema que está siendo procesado y el identificador del usuario actual, son utilizados para formar la segunda consulta. Mediante la segunda consulta, se extrae la calificación del alumno, correspondiente al identificador del tema que está siendo procesado.

Se debe considerar que el alumno no siempre tiene calificación en todos los exámenes, ya que los puede ir presentando en diferentes momentos, por lo cual, si la

calificación extraída mediante la segunda consulta, trae un valor diferente de “”, indica que el alumno ya realizó el examen. Si la calificación es “”, indica que el alumno no ha presentado ese examen, por lo cual, ese examen no es considerado como una evidencia.

Cuando el alumno tiene calificación en algún examen, ésta es ubicada en los rangos de los nodos tema, mediante la función *DeterminaRangoELVIRA*, es decir, se determina el estado del nodo, que representará en la red bayesiana al tema asociado al examen que tiene calificación.

El código que determina los estados de los nodos tema de la red bayesiana, se muestra a continuación:

```
function DeterminaRangoELVIRA($Calificacion) {  
  
    $Rango = "";  
  
    if ((int) $Calificacion) >= 0 && ((int) $Calificacion) <= 69) {  
        $Rango = "0";  
    }  
    else  
        if ((int) $Calificacion) >= 70 && ((int) $Calificacion) <= 80) {  
            $Rango = "1";  
        }  
        else  
            if ((int) $Calificacion) >= 81 && ((int) $Calificacion) <= 90) {  
                $Rango = "2";  
            }  
            else  
                if ((int) $Calificacion) >= 91 && ((int) $Calificacion) <= 100) {  
                    $Rango = "3";  
                }  
  
    return $Rango;  
}
```

Cabe señalar que aunque los rangos se hayan definido con el nombre de *Reprobado*, *Suficiente*, *Bueno* y *Excelente*, para el manejo lógico en *Elvira* se utiliza el número de rango, es decir los valores 0, 1, 2 y 3 respectivamente.

De ésta forma se genera el archivo de texto *evi* que contiene las evidencias de la red bayesiana utilizada en esta tesis. En el Anexo K se puede ver un ejemplo de este formato.

6.2.2.3 Ejecución de la inferencia utilizando Elvira

Después de haber generado el archivo de texto que contiene la *estructura de la red bayesiana* y el archivo de texto que contiene las *evidencias observadas*, el siguiente paso fue enviarle a *Elvira* las variables que contienen los nombres y ubicación de estos archivos de texto. Así como también la variable que contiene el nombre y ubicación del archivo de texto donde se le especifica a *Elvira* que guarde los *resultados obtenidos al realizar la inferencia*.

Para lograr enviar el nombre de los archivos de Moodle a *Elvira*, se realiza el proceso descrito en la **Figura 42**

Figura 42:

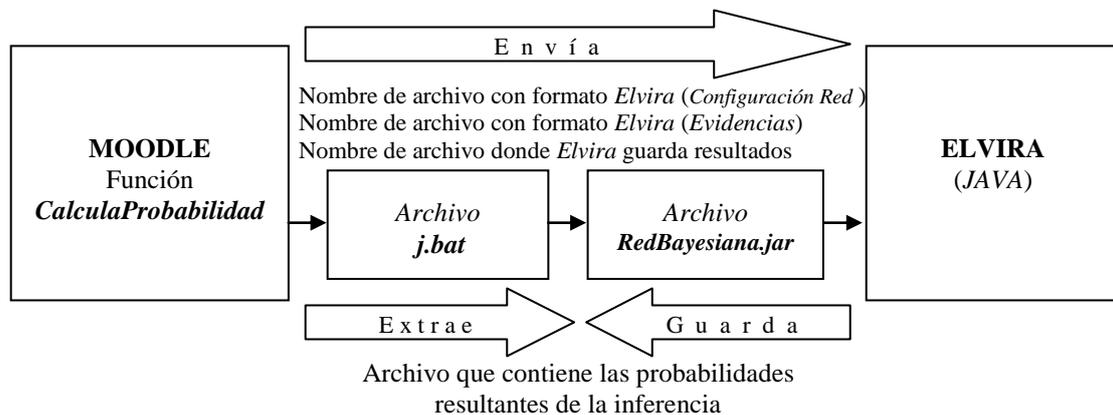


Figura 42. Proceso de comunicación entre Moodle y Elvira

Desde Moodle, a través de la función *CalculaProbabilidad*, se ejecuta un archivo con extensión *bat*, el cual recibe como parámetros el nombre de cada uno de los tres archivos. El orden de los parámetros es: nombre del archivo con la estructura de la red, nombre del archivo con las evidencias y nombre del archivo en donde *Elvira* almacenará los resultados. La ejecución del archivo *bat* se realiza mediante el siguiente código:

```
$Path = "C:\appserv\www\moodle\j.bat /". $NombreArchivoElvira. " /". $NombreArchivoEvidencias. " /". $NombreArchivoResultados;
shell_exec($Path);
```

El archivo *bat* al recibir los parámetros, los almacena en las variables %1, %2 y %3, el contenido de éste archivo se muestra a continuación:

```
"C:\Archivos de programa\Java\jdk1.6.0_21\bin\java.exe" -jar "C:\elvira\RedBayesiana\dist\RedBayesiana.jar" %1 %2 %3
```

Como se puede observar, el archivo *bat* sólo contiene una línea, mediante la cual llama al archivo *RedBayesiana.jar*, y le envía como parámetro mediante las variables %1, %2 y %3, el nombre de cada uno de los tres archivos.

El programa *RedBayesiana.jar* fue desarrollado en java para interactuar con *Elvira*. Mediante este programa se realiza el llamado a las clases de *Elvira* para que realice la inferencia sobre la red bayesiana. En la Tabla 55 se describen las librerías y clases de *Elvira* utilizadas. El código del programa *RedBayesiana.jar* se muestra a continuación:

```
package redbayesiana;

import elvira.*;
import elvira.inference.elimination.VariableElimination;
import elvira.parser.ParseException;
import java.io.IOException;

public class Main {

    public static void main(String[] args) throws ParseException, IOException {

        Bnet b = new Bnet(args[0]);

        Evidence e = new Evidence(args[1], b.getNodeList());

        VariableElimination ve = new VariableElimination((Bnet)b, e);
        ve.obtainInterest();

        ve.propagate(args[2]);

    }
}
```

La función *main* recibe como parámetro el nombre de los tres archivos en el arreglo *args* de tipo *String*. Los parámetros se almacenan dentro de este arreglo de la siguiente forma:

POSICIÓN	PARÁMETRO
0	Estructura Red
1	Evidencias
2	Resultados

Tabla 55. Librerías y clases de *Elvira* utilizadas para la inferencia

CLASE	DESCRIPCIÓN
<i>elvira.*</i>	Librería que contiene las clases <i>Bnet</i> y <i>Evidence</i> , utilizadas para realizar inferencia en una red bayesiana.
<i>elvira.inference.elimination.VariableElimination</i>	Librería que contiene la clase que realiza la propagación mediante el método exacto de eliminación de variables en redes bayesianas y diagramas de influencia.
<i>elvira.parser.ParserException</i>	Librería que contiene la clase que permite manejar excepciones.
<i>Bnet</i>	Clase encargada de almacenar una red Bayesiana.
<i>Evidence</i>	Clase que representa un conjunto de evidencias (observaciones) en una red Bayesiana.
<i>propagate</i>	Clase que lleva acabo la propagación guardando los resultados en el archivo que se le especifique.

El primer paso es crear un objeto de la clase *Bnet*, el cual recibe como parámetro en su instanciación, el nombre del archivo de texto que contiene la estructura de la red bayesiana, enviado como primer parámetro a *Elvira*.

Posteriormente, se crea un objeto de la clase *Evidence*, el cual recibe como parámetro de instanciación, el nombre del archivo de texto que contiene las evidencias de la red bayesiana, enviado como segundo parámetro a *Elvira*.

Después de haber creado los objetos *Bnet* y *Evidence*, se procede a crear un objeto de la clase *VariableElimination*, el cual es el método a utilizar para realizar la inferencia y que recibe como parámetros de instanciación los objetos *Bnet* y *Evidence*.

Por último, se realiza una propagación mediante la clase *Propagate*, almacenando los resultados en el tercer archivo enviado como parámetro a *Elvira*, es decir, el archivo donde *Elvira* va aguardar los resultados de la inferencia. Este es el archivo que será procesado por Moodle, para extraer las probabilidades resultantes de la inferencia.

6.2.3 Recuperación de los resultados generados por *Elvira*

Una vez que *Elvira* realizó la inferencia sobre la red bayesiana escribe en el archivo de texto especificado, las probabilidades resultantes de la inferencia, bajo un cierto formato. *Elvira* sólo escribirá en este archivo, las probabilidades resultantes de los nodos que *no* hayan sido enviados como evidencia. Un ejemplo del formato en el que *Elvira* guarda los resultados en el archivo de texto *txt* se muestra en el Anexo K.

La información almacenada en la tabla *mdl_probabilidades_competencias*, es utilizada para **mostrar al alumno** un diagnóstico acerca del porcentaje que posee de cada competencia asociada al tema que acaba de presentar.

Antes de mostrarle los resultados al alumno, se verifica que la probabilidad *SI*, de cada una de las competencias asociadas al tema del examen actual, se encuentre por encima de un umbral previamente definido por el administrador. El umbral es definido en la variable global *\$UMBRAL*, declarada en el archivo *config.php* nativo de Moodle que se encuentra ubicado en `\moodle`, el valor asignado a esta variable es de 0.30.

En el caso donde las probabilidades *SI*, de cada una de las competencias asociadas al tema del examen actual, cumplan con la condición de ser mayor que el umbral, solamente se le muestran al alumno sus porcentajes diagnosticados en cada competencia y la retroalimentación que Moodle normalmente realiza acerca del examen, y la función *CalculaProbabilidad* retornará un FALSE, por lo cual el proceso del **Agente Selector** no se ejecutará.

Por el contrario, si al menos una de las competencias relacionadas al tema del examen actual, *no* alcanzó el umbral, la función *CalculaProbabilidad* retornará un TRUE, por lo cual el proceso del **Agente Selector** será ejecutado y se realizarán algunas recomendaciones al alumno.

Las recomendaciones consisten en sugerir al alumno que vuelva a revisar temas anteriores, en los cuales, al haber presentado el examen correspondiente, hayan obtenido una calificación menor a 70. Estos temas serán sólo aquellos que se encuentren asociados a las competencias relacionadas con el examen que acaba de presentar y de las cuales el alumno *no* haya alcanzado el umbral. El archivo modificado para poder mostrar ésta información al alumno es *review.php* nativo de Moodle y ubicado en `\moodle\mod\quiz\`.

La Figura 43 muestra una imagen de cómo se le presenta al alumno los porcentajes correspondientes a cada competencia asociada con el tema correspondiente al examen actual, así como también la recomendación sobre los temas que debe volver a revisar, esto debido a que al menos en una competencia no alcanzó el umbral especificado. Ésta información se le muestra al alumno dentro de la página en donde Moodle le presenta la calificación y la retroalimentación del examen actual.

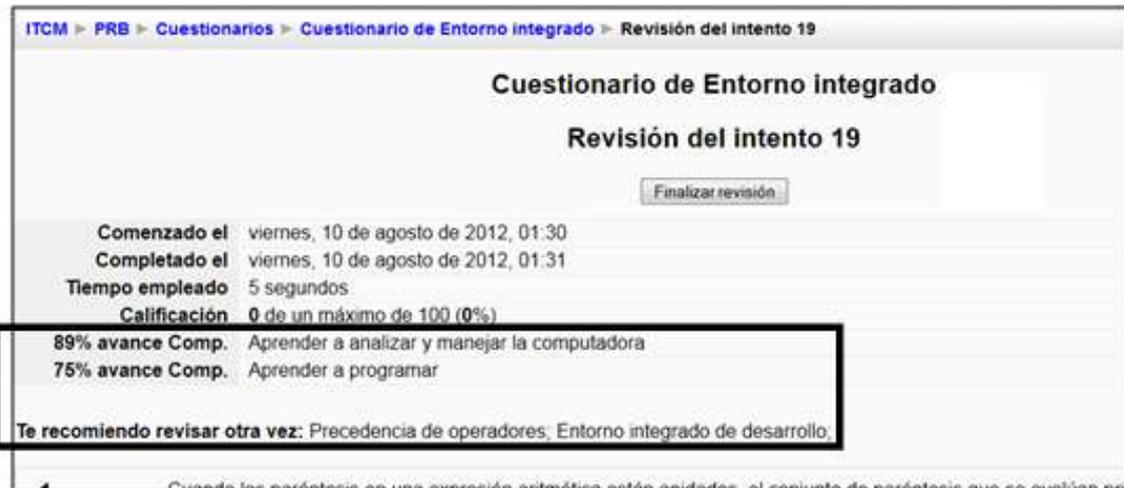


Figura 43. Página que muestra la retroalimentación del examen y los porcentajes de competencia del alumno

El proceso para mostrar la información de la figura anterior, consistió en extraer de la tabla *mdl_probabilidades_competencias*, el identificador de las competencias relacionadas con el examen que acaba de presentar el alumno, así como el porcentaje correspondiente a la probabilidad *SI* de cada competencia. Posteriormente, se recorre cada una de las competencias y se verifica si el porcentaje *SI* de alguna de ellas, es menor que el umbral definido previamente. Si alguna de ellas no alcanzó el umbral, se concatena en la variable *\$CompetenciasIDs* de tipo string el identificador de la competencia que no alcanzó el umbral. Este proceso se lleva a cabo mediante el siguiente código:

```
global $UMBRAL;

$rs_Porcentajes = get_recordset_sql("SELECT mpc.CompetenciaId, mpc.Porc_No, mpc.Porc_Si, mc.Name
                                   FROM mdl_probabilidades_competencias mpc, mdl_competencias mc
                                   WHERE (mpc.CompetenciaId = mc.Id) and
                                           mpc.CourseId = $course->id AND
                                           mpc.ExamenId = $attempt->quiz AND
                                           mpc.IntentosExamen = $attempt->attempt AND
                                           mpc.UsuarioId = $attempt->userid");

$CompetenciasIDs = "";
while ($row = rs_fetch_next_record($rs_Porcentajes)) {

    if ($row->Porc_Si < $UMBRAL) {
        $CompetenciasIDs .= $row->CompetenciaId.", ";
    }
    $rows[] = '<tr><th scope="row" class="cell">'. $row->Porc_Si * 100. '% avance Comp.</th><td class="cell">'. $row->Name. '</td></tr>';
}

rs_close($rs_Porcentajes);
```

Los identificadores de las competencias que *no* alcanzaron el umbral, son almacenados en la variable *\$CompetenciasIDs*, debido a que son utilizados para extraer,

mediante una consulta, los nombres de los temas que ya fueron presentados por el alumno y en los cuales obtuvo una calificación menor a 70. Otra condición que deben cumplir los temas es que deben estar relacionados con la competencia que *no* alcanzó el umbral.

En cada iteración del ciclo *While*, se recorre a cada uno de los temas no acreditados y se concatena su nombre en la variable *\$Recomendaciones*, de tipo *string*. Esto se realizó mediante el siguiente código:

```

$rs_Recomendaciones = get_recordset_sql("SELECT DISTINCT mte.ScoreId, ms.Name, mte.ExamenId, mqq.Grade
FROM mdi_competencias_temas act, mdi_tema_examen mte, mdi_Score ms, mdi_quiz_grades mqq
WHERE (act.CourseId = mte.CourseId and act.ScoreId = mte.ScoreId) and
(mte.ScoreId = ms.Id and act.CourseId = ms.Course) and
(mqq.Quiz = mte.ExamenId and mqq.UserId = {attempt->userid}) and
act.CourseId = {course->id} and act.CompetenciaId in [ {CompetenciasIds } ] and mqq.Grade < 70");

$Recomendaciones = "";
$HuboRecomendacion = false;
while ($rs_Recomendaciones = rs_fetch_next_record($rs_Recomendaciones)) {
    $Recomendaciones .= $rs_Recomendaciones->Name, " ";

    $HuboRecomendacion = true;
}
rs_close($rs_Porcentajes);

```

Al terminar el ciclo *While*, se tiene almacenado en la variable *\$Recomendaciones*, los nombres de los temas *no* acreditados. Estos temas son recomendados al alumno para que los vuelva revisar, ya que forman parte de la competencia que *no* alcanzó el umbral. Para enviar a pantalla el mensaje de recomendación para el alumno se realizó mediante el siguiente código:

```

if ($HuboRecomendacion) {
    echo "<b>Te recomiendo revisar otra vez: </b>{$Recomendaciones <BR><BR>";
}

```

Antes de que se muestre al alumno la retroalimentación sobre los temas que debe revisar nuevamente, se realiza el llamado al *agente selector*. El objetivo del agente selector es elegir una estrategia de enseñanza distinta a la que tiene asignada como actual el alumno, esto con el fin de filtrar a los contenidos del curso, que se encuentren elaborados de acuerdo con la nueva estrategia elegida por el agente selector. El funcionamiento del agente selector se describe en la siguiente sección.

6.3 Implementación del Agente Selector

La ejecución del agente selector está en función del proceso para el diagnóstico de las competencias. Esto es debido a que si un alumno no cumple con el nivel de competencia establecido, el proceso de diagnóstico le indicará al agente selector sobre esta situación, por lo tanto, éste se ejecutará. El propósito del agente selector es reforzar el aprendizaje del alumno, seleccionando una estrategia de enseñanza que vaya de acuerdo al estilo de aprendizaje actual del alumno. Una vez que agente selector eligió y estableció la estrategia de enseñanza actual, el agente interfaz la considera para filtrar (elegir el material que se encuentra elaborado para el estilo y estrategia actual) y mostrar los contenidos al alumno.

En este capítulo se describe el proceso que realiza el agente selector, así como también se describe el proceso para el filtrado de los contenidos a mostrar al alumno.

6.3.1 Proceso de selección del agente selector

A continuación se describe el funcionamiento del agente selector, después de cada explicación se muestra el código utilizado en la implementación.

Cuando el agente selector se ejecuta, el primer paso que realiza es ejecutar una consulta a la base de datos para obtener todas las estrategias de enseñanza que se encuentren relacionadas al estilo de aprendizaje actual del alumno. El estilo actual del alumno se encuentra cargado en memoria en el objeto *\$USER->Estiloactual*. Al consultar las estrategias, éstas son ordenadas de acuerdo a su identificador. Este proceso se lleva a cabo mediante la siguiente consulta.

```
$rs_Estrategia = get_recordset_sql("SELECT Distinct msee.EstrategiaId
FROM mdl_Scorms_ee msee, mdl_Estrategias me
WHERE (msee.EstrategiaId = me.Id) and
      msee.CursoId = $Curso and
      msee.EstiloId = $USER->Estiloactual
ORDER BY msee.EstrategiaId");
```

Habiendo obtenido las estrategias asociadas al estilo actual del alumno, las cuales se ordenaron en base al identificador de la estrategia, se busca entre éstas, el siguiente ID mayor al de la estrategia actual, la cual se encuentra cargada en memoria en el objeto *\$USER->EstrategiaActual*. En el caso de encontrar una estrategia que

cumpla con la condición, ésta será establecida como la estrategia actual. El código con el cual se realizó el proceso descrito se muestra a continuación:

```
while ($row = rs_fetch_next_record($rs_Estrategia)) {
    if ($USER->EstrategiaActual < $row->EstrategiaId){
        $estrategia_id = $row->EstrategiaId;
        $Encontrado = 1;

        break;
    }
}

rs_close($rs_Estrategia);
```

En el caso, donde no se encuentre una estrategia con un ID mayor al que tiene la estrategia actual, esto indicará que ya se utilizaron todas las estrategias asociadas al estilo de aprendizaje actual del alumno. Por lo que el agente selector será capaz de cambiar el estilo de aprendizaje actual del alumno y elegir, de acuerdo a las puntuaciones obtenidas por el alumno en la prueba VARK, al estilo de aprendizaje que tenga la siguiente puntuación menor al estilo actual. Para lo cual es necesario realizar lo siguiente:

- Extraer el identificador de todos los estilos, ordenados de forma ascendente de acuerdo a las puntuaciones de cada estilo y ordenados de forma descendente de acuerdo al identificador de cada estilo. Para obtener estos datos, se accede a la tabla *mdl_user_info_data*, a continuación se muestra un fragmento de esta tabla:

id	fieldid	data
2	3	1
4	5	2
3	4	6
1	2	6

El campo *data* almacena la puntuación obtenida por cada estilo, la cual es el primer criterio de ordenación, de forma ascendente. El campo *id* es el identificador de cada estilo y forma el segundo criterio de ordenación, se ordena de forma descendente y el campo *fieldid* representa el identificador del campo personalizado asociado a cada estilo. Por ejemplo, si el estilo actual fuera el registro con el campo *id* igual a 3, el siguiente estilo a seleccionar sería el del registro con el identificador igual a 4. En el caso donde el estilo actual del

alumno sea el de menor puntaje, indica que ya han sido utilizados todos los estilos de aprendizaje, por lo que se volverá a seleccionar el estilo de mayor puntaje, tal como se realizó cuando el alumno inició sesión por primera vez en Moodle, este procedimiento se asemeja a una ruleta. El código encargado de la selección del nuevo estilo se presenta a continuación:

```
while ($row = rs_fetch_next_record($rs_Estilos)) {
    if ($USER->Estiloactual == $row->id && $Contador == 0){

        while ($row = rs_fetch_next_record($rs_Estilos)) {

            $estilo_id = $row->id;

        }

        break;
    }
    else {
        if ($USER->Estiloactual == $row->id){
            break;
        }

        $estilo_id = $row->id;
        $Contador = $Contador + 1;
    }
} //while

rs_close($rs_Estilos);
```

Posteriormente a la selección del nuevo estilo, se procede a seleccionar el ID de la primera estrategia asociada al nuevo estilo seleccionado, la cual debe cumplir la condición de estar asociada algún material didáctico. Esto se realiza mediante el siguiente código.

```
$estrategia_id = get_field_sql("SELECT Distinct msee.EstrategiaId
FROM mdl_Scorms_ee msee, mdl_Estrategias me
WHERE (msee.EstrategiaId = me.Id) and
msee.CursoId = $Curso and
msee.EstiloId = $estilo_id
ORDER BY msee.EstrategiaId");
```

Después de que el *agente selector* realizó el cambio, ya sea de estilo de aprendizaje y/o estrategia de enseñanza actuales, se procede a actualizar la información correspondiente a estos campos. Esta actualización se realiza en tres lugares distintos: en los campos correspondientes al estilo y la estrategia actual de la tabla *mdl_user_info_data*, en la tabla *mdl_historial_agenteselector* la cual almacena un historial de los cambios de estilo y estrategia que el agente selector va realizando, y por último se modifican en memoria mediante la actualización de los atributos *Estiloactual* y *EstrategiaActual* del objeto *\$USER*. El siguiente fragmento de código actualiza a las

propiedades *EstiloActual* y *EstrategiaActual* del objeto *\$USER*, con los nuevos valores asignados por el agente selector. Esto es necesario debido a que el objeto mantiene en memoria, durante la sesión activa de un alumno, los valores correspondientes al estilo de aprendizaje y la estrategia actual del alumno. Los cuales pueden ser consultados en cualquier momento, sin necesidad de accederlos desde la base de datos.

```
$USER->EstiloActual = $estilo_id;
$USER->EstrategiaActual = $estrategia_id;
```

Para actualizar los valores de la tabla *mdl_user_info_data*, correspondientes a los campos estilo actual y estrategia actual del alumno, se realizó mediante el siguiente código.

```
get_field_sql("Update mdl_user_info_data
              Set data = ".$estilo_id."
              WHERE fieldid = 1 and UserId = ".$USER->id);

get_field_sql("Update mdl_user_info_data
              Set data = ".$estrategia_id."
              WHERE fieldid = 6 and UserId = ".$USER->id);
```

La actualización del historial acerca de los cambios de estilo y estrategia realizados por el agente selector, se realiza en la tabla *mdl_historial_agenteselector*, mediante el siguiente código:

```
execute_sql("Insert Into mdl_historial_AgenteSelector (UserId, uniqueid, attempt, Calificacion, Estilo_Actual, Estrategia_Actual, Nuevo_Estilo,
                                                    Nueva_Estrategia, Fecha_Hora, cursoid, scoornid)
            Values($USER->id, $uniqueid, $attempt, $Calificacion, $tmp_estilo_id, $tmp_estrategia_id,
                  $estilo_id, $estrategia_id, Now(), $Curso, $Score)", false);
```

Con estas tres actualizaciones termina la función del agente selector, ahora, el proceso siguientes es que el agente interfaz filtre los contenidos a mostrar al alumno, considerando que estos están diseñados para el estilo y estrategia actual del alumno. Este proceso se detalla en la siguiente sección.

6.3.2 Proceso para el filtrado de los contenidos

Como ya se ha mencionado, uno de los objetivos de esta tesis, es presentar al alumno, los materiales didácticos de una forma que facilite su aprendizaje. Para esto, cada uno de los temas de un curso, debe ser diseñado repetidamente al menos una vez, para cada uno de los diferentes estilos de aprendizaje del modelo VARK, utilizando al menos una

de las estrategias de enseñanza relacionadas a cada estilo. Esto permite tener la información de un mismo tema, diseñada en diferentes modalidades. De esta forma, se tiene la posibilidad de ir presentando los temas, de acuerdo al estilo y estrategia que faciliten el aprendizaje al alumno. El encargado de seleccionar la estrategia o en caso de ser necesario, elegir otro estilo, es el agente selector, para lo cual considera los resultados obtenidos por el alumno en el cuestionario VARK, como se describió en la sección anterior.

Por otro lado, también se tuvo que trabajar en la parte de la visualización de los temas, ya que se debe mostrar el material didáctico de acuerdo al estilo y estrategia actuales del alumno, previamente definidos por el agente selector. Por tal motivo, se incorporó un *agente interfaz*, el cual, en base al estilo y estrategia definidos por el agente selector, selecciona el material didáctico del tema a mostrar, diseñado para el estilo y estrategia actual del alumno.

El agente interfaz fue incorporado, modificando el proceso nativo de visualización de materiales didácticos de Moodle. La función que Moodle utiliza para extraer de la base de datos la información de los materiales didácticos a mostrar es *scorm_get_scoes* y está almacenada en el archivo *moodle/mod/scorm/locallib.php*. Este archivo que contiene funciones para acceder a la información de los paquetes SCORM. La modificación realizada a este archivo se muestra a continuación.

```
function scorm_get_scoes($id,$organisation=false,($filtroAgente="")) {
    Global $USER;
    $organizationsql = '';
    if (!empty($organisation)) {
        $organizationsql = "AND organization='{$organisation}'";
    }

    if ($scoes = get_records_select('scorm_scoes',"scorm='{$id}'($filtroAgente) $organizationsql order by id ASC")) {
        // drop keys so that it is a simple array as expected
        $scoes = array_values($scoes);

        foreach ($scoes as $sco) {
            if ($scodatas = get_records('scorm_scoes_data','scoid',$sco->id)) {
                foreach ($scodatas as $scodata) {
                    $sco->{$scodata->name} = stripslashes_safe($scodata->value);
                }
            }
        }

        return $scoes;
    } else {
        return false;
    }
}
```

La función *get_records_select*, es utilizada por Moodle para extraer información de la base de datos. Esta función recibe dos parámetros, el primero indica el nombre de la tabla de donde se va a extraer la información y el segundo define las restricciones y ordenamiento a considerar en la extracción de la información.

La modificación realizada a esta función fue agregar una restricción más a la consulta. La nueva restricción está contenida dentro de la variable *\$filtroAgente*. A continuación se presenta un ejemplo de la información extraída por la función *get_records_select*, considerando las restricciones antes y después de la modificación realizada.

Antes de la modificación

```
scorm='41' AND organization='ORG-7A4C324C-3F44-2497-3FFC-6D95D2D74BF8' order by id ASC
```

Después de la modificación

```
scorm='41' and id in (439) AND organization='ORG-7A4C324C-3F44-2497-3FFC-6D95D2D74BF8' order by id ASC
```

La modificación provoca que Moodle extraiga de la tabla *mdl_scorm_scoes*, el material didáctico definido en el campo *launch* (este campo almacena el nombre de la página que contiene el recurso didáctico), del registro con identificador 439 del SCORM 41.

La variable *\$filtroAgente* fue enviada como parámetro a la función *scorm_get_scoes*, desde el archivo *moodle/mod/scorm/datamodels/scorm_12lib.php*, el cual se encarga de seleccionar y visualizar los temas disponibles. El código agregado a este archivo para generar la nueva restricción, la cual es enviada como parámetro, se muestra a continuación.

```
//TESIS
$rs_Scoes = get_recordset_sql("SELECT scoid
                                FROM mdl_scorm_scoes
                                WHERE m.scormid = $scorm->id and
                                       estiloid = $USER->EstiloActual and estrategioid = $USER->EstrategiaActual");

$filtroAgente = " and id in (";
$ScoesId = "";
while ($row = rs_fetch_next_record($rs_Scoes)) {
    $ScoesId = $ScoesId.$row->scoid.", ";
}

rs_close($rs_Scoes);

$ScoesId = trim($ScoesId); //Elimina el espacio final
$ScoesId = substr($ScoesId, 0, strlen($ScoesId) - 1); //Elimina la coma
$filtroAgente = $filtroAgente.$ScoesId." )";
```

Para generar la nueva restricción, se considera que los contenidos didácticos deben estar previamente relacionados con el estilo y la estrategia para el cual fueron diseñados (ver sección 5.2.6.3). Por lo tanto, la consulta extraerá sólo a aquel recurso didáctico que cumpla con la condición de estar diseñado para el estilo y estrategia actual del alumno.

El identificador del material a mostrar se concatena en la variable *\$ScoesId*, para posteriormente asignarse a la variable *\$filtroAgente*, la cual es enviada como parámetro a la función *scorm_get_scoes*, tal como se muestra a continuación:

```
if ($scoes = scorm_get_scoes($scorm->id, $currentorg, $filtroAgente){
```

De esta forma se genera la nueva restricción que permite extraer el material didáctico que sea compatible con el estilo y la estrategia actual del alumno, el cual será mostrado al mismo mediante la interfaz de Moodle.

Este proceso se realiza por cada paquete SCORM, ya que como se ha mencionado, cada paquete representa un tema del curso.

Capítulo 7

EXPERIMENTACIÓN Y RESULTADOS

El objetivo de este trabajo fue incorporar en la arquitectura básica de un STI, el concepto de estilos de aprendizaje y el concepto de educación basada en competencias. Para lo cual, se creó en el módulo tutor de la plataforma Moodle, un agente selector el cual se encarga de elegir la estrategia de enseñanza que favorezca al estilo de aprendizaje del alumno. Así como también, se incorporó en la plataforma, un proceso para diagnosticar la probabilidad existente de que un alumno obtenga cierta calificación al presentar un examen, a causa de que tiene desarrollada la o las competencias asociadas al tema evaluado.

La experimentación consistió en observar y analizar el desempeño obtenido por un grupo de alumnos, al interactuar con la versión modificada en esta tesis de la plataforma Moodle. Esto con el fin de: a) sugerir las características para organizar la muestra a examinar, b) indicar qué factores se deben considerar al momento de evaluar el desempeño del alumno y c) realizar recomendaciones para la elaboración del material didáctico.

En este capítulo se describen, las características de la muestra examinada y del material didáctico utilizado, así como el proceso realizado durante la experimentación y el análisis de los resultados obtenidos.

7.1 Características de la muestra examinada y del material didáctico utilizado.

Para llevar acabo la experimentación se trabajó en el Centro de Cómputo con una muestra de alumnos que interactuaron con la plataforma Moodle. Los alumnos cursaban el primer semestre de la carrera de Ingeniería en Sistemas Computacionales en el ITCM. El material didáctico utilizado en ésta experimentación fue el tema *Arreglos*, el cual fue utilizado también con el grupo de alumnos del cual se obtuvieron las probabilidades para la red bayesiana, ver sección 6.1.2 del capítulo 6. Cabe mencionar que el grupo

utilizado en esta experimentación y el grupo del cual se obtuvieron las probabilidades de la red, no es el mismo, no obstante en su momento ambos grupos cursaban la materia de Fundamentos de Programación. Las características de los alumnos con los que se realizó esta experimentación se describen en la Tabla 56. En la Tabla 57, se muestran los elementos configurados en Moodle, para llevar a cabo la experimentación.

Tabla 56. Características del grupo muestra utilizado en la experimentación

SEXO	EDAD PROMEDIO	CANTIDAD ALUMNOS
Femenino	18	2
Masculino	19	8
Total		10

Tabla 57. Elementos utilizados en la experimentación

CURSO	COMPETENCIA	TEMA (PAQUETE SCORM)	ESTRATEGIAS UTILIZADAS	CUESTIONARIO
Fundamentos de Programación- Personalizado	Construir programas que utilicen arreglos unidimensionales y multidimensionales para solucionar problemas.	Arreglos	Textos subrayados- Visual Vídeo con sonido- Auditivo Lectura- Lectura/Escritura Ejemplos- Kinestesico	Examen con 10 preguntas
Fundamentos de Programación-No Personalizado		Arreglos	Lectura- Lectura/Escritura	

Como parte de la experimentación se evaluaron 2 cursos de Moodle: 1) el curso *Fundamentos de Programación-Personalizado* el cual muestra al alumno el tema de forma personalizada, considerando el estilo de aprendizaje y la estrategia actual del alumno, y 2) el curso *Fundamentos de Programación-No Personalizado* que muestra el mismo recurso a todos los alumnos, es decir *sin* personalizar. Para ambos cursos se aplicó un mismo examen: 10 preguntas de tipo opción múltiple. Cada pregunta tenía cuatro opciones y sólo se podía elegir una. A cada pregunta se le dio un puntaje diferente dependiendo de su importancia, dando un total de 100 puntos, como calificación máxima. Cabe mencionar que en ambos cursos también se consideró el tema *Matrices*, para configurar la red bayesiana, tal como se describió en la sección 6.1.2 del capítulo 6, solo que este tema no fue evaluado.

7.2 Proceso realizado en la experimentación

A continuación se describen los pasos realizados en la experimentación:

1. Cada alumno realizó su registro en la plataforma Moodle.
2. Cada alumno contestó el cuestionario con el cual se detectó su estilo de aprendizaje.
3. De los diez alumnos del grupo examinado, se le pidió a cinco de ellos que se registraran en el curso *Fundamentos de Programación-Personalizado* y a los cinco alumnos restantes, que se registraran en el curso *Fundamentos de Programación-No Personalizado*. El registro de los alumnos en cada curso fue realizado al azar.
4. A cada alumno registrado en el curso *Fundamentos de Programación- Personalizado* se le mostró el tema elaborado en una estrategia de enseñanza que va de acuerdo a su estilo de aprendizaje. A los alumnos registrados en el curso *Fundamentos de Programación-No Personalizado* se les mostró a todos por igual el tema elaborado en una misma estrategia de enseñanza, es decir no se consideró su estilo de aprendizaje.
5. En ambos cursos, después que los alumnos revisaron el tema, respondieron al cuestionario correspondiente, teniendo como máximo 5 intentos para contestar el examen.
6. En el curso *Fundamentos de Programación-Personalizado* se le mostraba al alumno el tema de acuerdo a su estilo de aprendizaje. Cada que un alumno contestaba el cuestionario, se realizaba el proceso para conocer si éste poseía la competencia relacionada al tema, ejecutando al agente selector en caso de ser necesario. Con respecto al curso *Fundamentos de Programación-No Personalizado* siempre se mostró el tema en el mismo formato, es decir no se consideró su estilo de aprendizaje. El formato en el cual estaba diseñado el tema estaba dirigido hacia el estilo de aprendizaje de *Lectura/Escritura*.

7.3 Resultados obtenidos

En esta sección se describen los resultados obtenidos en la experimentación realizada en este tema de investigación. En la gráfica de la Figura 44 se muestra el promedio general de las calificaciones obtenidas por los alumnos de cada curso. Este promedio general fue el resultado de sumar todas las calificaciones obtenidas por los alumnos, en cada uno de los intentos realizados. El número de intentos realizados depende de las veces que tuvieron que realizar la actividad para aprobar el examen y esta limitado a un máximo de 5.

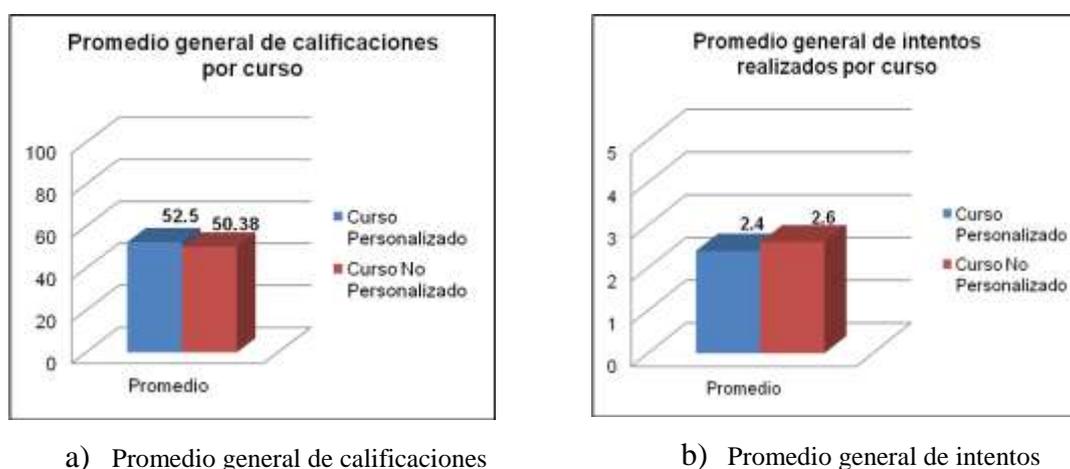


Figura 44. Promedio general de calificaciones de los alumnos y de intentos realizados para cada tipo de curso

En el inciso *a* de la gráfica anterior se muestran los promedios generales de calificaciones obtenidas por los alumnos de cada curso, en cada uno de los intentos realizados, hasta lograr la aprobación, es decir una calificación al menos de 70. En el inciso *b* se muestra el promedio de intentos realizados por los alumnos de cada curso, para obtener una calificación aprobatoria.

Como se puede observar en la gráfica anterior, los promedios de las calificaciones obtenidas por los alumnos de cada curso resultaron muy similares. Sin embargo, se esperaba que los alumnos del curso *Personalizado* obtuvieran un mejor desempeño, tanto en calificaciones como en intentos realizados, dado que se les presentó la información de acuerdo a su estilo de aprendizaje.

Debido a lo anterior, se revisó cada uno de los intentos realizados, con el fin de analizar el comportamiento de los alumnos de ambos cursos. En la Figura 45, se

presenta una gráfica de los promedios generales obtenidos, en cada intento para ambos cursos.

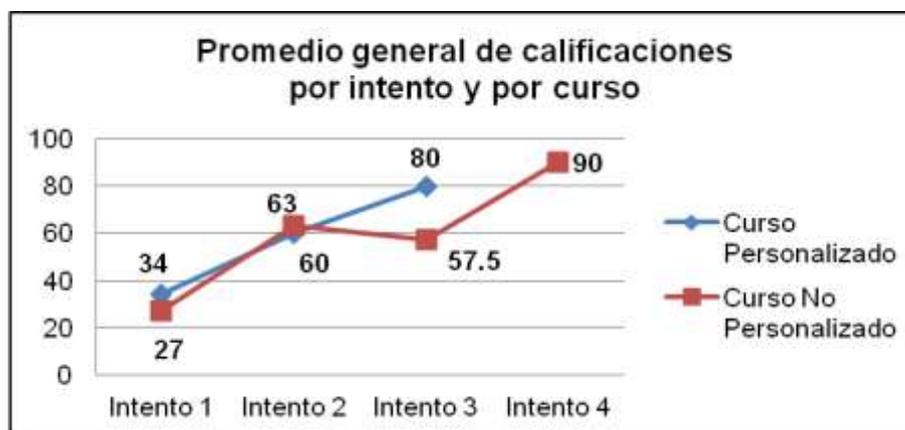


Figura 45. Promedio general de calificaciones obtenido en cada intento por curso

La cantidad de alumnos que realizaron el primer y segundo intento en ambos cursos fue de cinco. La cantidad de alumnos que realizaron el tercer intento en ambos cursos fue de dos. La cantidad de alumnos que realizaron el cuarto intento sólo fue un alumno del curso *No Personalizado*, ya que en el tercer intento reprobó y necesitó realizar un intento más.

Como se puede observar en la gráfica anterior, en el primer y segundo intento, los alumnos de ambos cursos obtuvieron un promedio semejante. En el tercer intento la diferencia sí fue notoria, debido a que un alumno del curso *No Personalizado*, obtuvo una calificación no aprobatoria y los alumnos del otro curso que presentaron el tercer intento, todos obtuvieron una calificación aprobatoria. En el cuarto intento no existe comparación, ya que sólo un alumno del curso *No Personalizado* lo realizó. Se observó que la cantidad de alumnos que realizaron los intentos, coincidió en ambos cursos hasta el tercer intento. Esto confirma que los dos grupos de alumnos tuvieron un desempeño similar.

Por tal motivo, se analizaron las frecuencias de los estilos de aprendizaje detectados a cada uno de los alumnos de cada curso, para conocer si estos influyeron en el desempeño de los alumnos. En la Figura 46 y 47 se presentan las gráficas correspondientes a los estilos de aprendizajes.



Figura 46. Distribución de estilos de aprendizaje detectados en el curso *Personalizado*

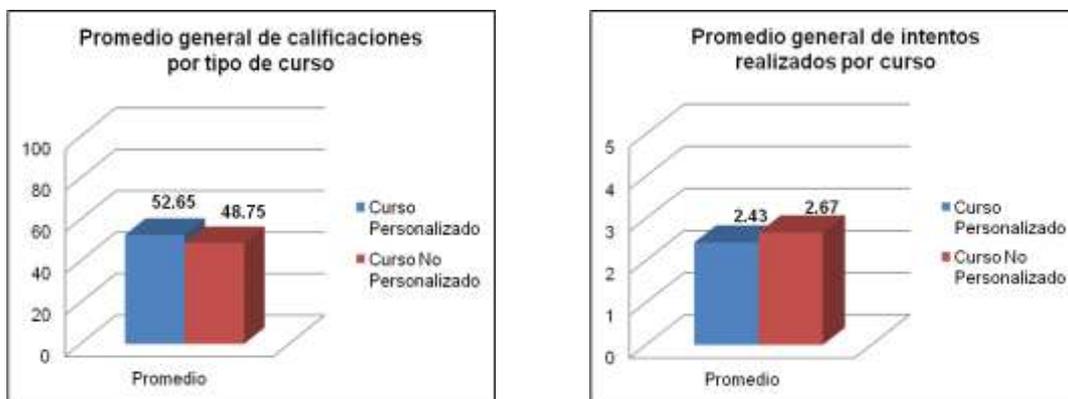


Figura 47. Distribución de estilos de aprendizaje detectados en el curso *No Personalizado*

Como se puede observar, en ambos cursos el estilo de aprendizaje predominante en los estudiantes fue el de *Lectura/Escritura*. Al conocer esta información se dedujo lo siguiente: una causa que pudo haber influido de forma positiva en las calificaciones de los alumnos del curso *No Personalizado*, fue que sin ser el objetivo, a los alumnos con un estilo de *Lectura/Escritura* también se les mostró la información de forma personalizada. Esto debido a que el material didáctico con el que interactuaron los alumnos de este curso, estaba diseñado para el estilo de *Lectura/Escritura*. Por lo tanto, como dos de los alumnos de este curso presentaron ese estilo de aprendizaje, esto pudo haber influido para que el desempeño de los alumnos de este curso resultara similar al del otro curso.

Debido a esto, se calculó nuevamente el promedio general de calificaciones de todos los intentos realizados, y el promedio de intentos, por cada curso, reubicando del curso *No Personalizado* al curso *Personalizado*, a los dos alumnos con el estilo de

aprendizaje de *Lectura/Escritura*, obteniendo los resultados mostrados en la gráfica de la Figura 48.



a) Promedio general de calificaciones

b) Promedio general de intentos

Figura 48. Promedio de calificaciones y de intentos por curso, reubicando del curso *No Personalizado* al curso *Personalizado*, a los dos alumnos con estilo de *Lectura/Escritura*

Como se puede observar, no obstante al cambio realizado de los dos alumnos de un curso a otro, el promedio general de calificaciones y de intentos de ambos cursos, tuvo una variación menor en el promedio de calificaciones del curso *Personalizado* y un decremento de 2 puntos en el promedio de calificaciones del curso *No Personalizado*. A continuación se presentan en la Figura 49, los nuevos promedios de cada uno de los intentos realizados por los alumnos, considerado el cambio de alumnos.

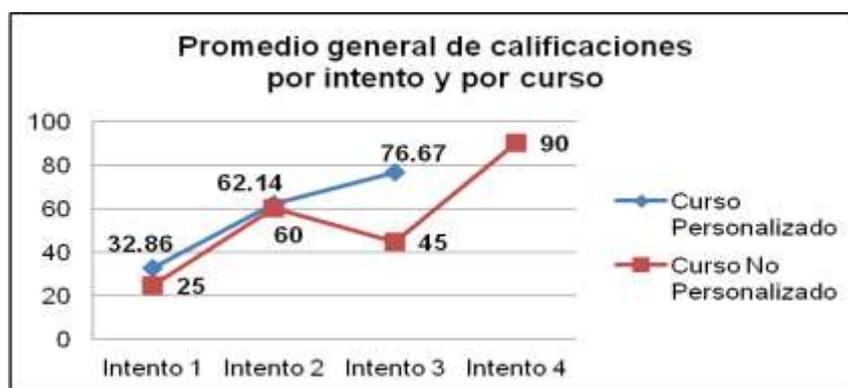


Figura 49. Promedio general de calificaciones obtenido en cada intento por curso

Una vez conformada la nueva composición (Curso *Personalizado* quedó integrado por 7 alumnos y el otro curso por 3 alumnos) y para tratar de deducir porque

no existe una gran diferencia entre los promedios de las calificaciones, se analizó otro factor que pudo influir para que los alumnos del curso *No Personalizado* obtuvieran un desempeño similar al del otro curso.

El otro factor que pudo haber influido de manera positiva al curso *No Personalizado*, fue que existían alumnos que presentaban mayor habilidad para entender los temas de la materia (contaban con un background), estos fueron señalados por el profesor a cargo de la asignatura. De acuerdo a lo señalado, el número de alumnos con las habilidades más desarrolladas eran seis, de los cuales, cuatro se ubicaron en el curso *Personalizado* y dos en el curso *No Personalizado*. Las Tabla 58 y 59 muestran a los alumnos integrantes de los cursos agrupados según su estilo de aprendizaje e indicando si presentan un conocimiento previo de la materia. Donde se observa que el 66% de los alumnos del curso *No Personalizado* contaban con un background y el *Personalizado* solo tenía un 57%. Esto pudo haber contribuido positivamente a los resultados obtenidos en este curso, tanto en calificaciones como en el número de intentos realizados.

Tabla 58. Alumnos del curso *Personalizado*, categorizados por estilo de aprendizaje y background

ALUMNOS DEL CURSO PERSONALIZADO		
ESTILO DE APRENDIZAJE	NÚMERO DE ALUMNOS	
	CON BACKGROUND	SIN BACKGROUND
Visual	-	1
Auditivo	-	-
Lectura/Escritura	4	1
Kinestesico	-	1

Tabla 59. Alumnos del curso *No Personalizado*, categorizados por estilo de aprendizaje y background

ALUMNOS DEL CURSO NO PERSONALIZADO		
ESTILO DE APRENDIZAJE	NÚMERO DE ALUMNOS	
	CON BACKGROUND	SIN BACKGROUND
Visual	1	-
Auditivo	-	1
Lectura/Escritura	-	-
Kinestesico	1	-

En la Figura 50 se muestra una gráfica, con los promedios generales de calificación obtenidos por cada uno de los estilos de aprendizaje, El cálculo de los promedios considera todas las calificaciones obtenidas por los alumnos en cada uno de

los intentos realizados. En la gráfica se indican con una línea sólida los estilos de aprendizaje en donde se ubicaron los alumnos con un background.

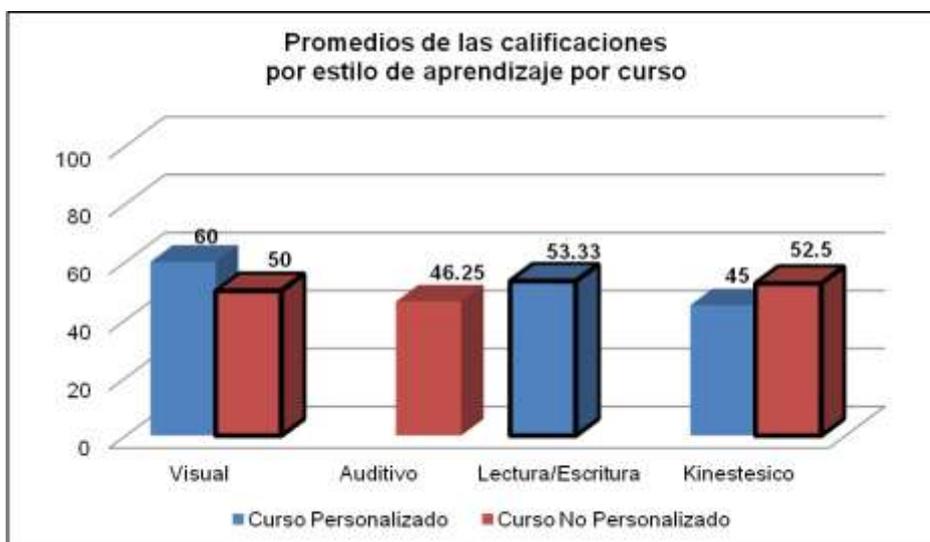


Figura 50. Promedio general de calificación obtenido por cada uno de los estilos de aprendizaje

En esta gráfica se puede apreciar que los promedios de los alumnos con un estilo de aprendizaje *Visual*, no difieren mucho. Esto pudo deberse a que el alumno del curso *No Personalizado* presentaba mayor habilidad para entender el tema, y a consecuencia de eso, obtuvo una calificación cercana a la calificación del alumno del curso *Personalizado*, a pesar de que no se le mostró el material de acuerdo a su estilo de aprendizaje. Los alumnos con un estilo *Auditivo* y de *Lectura/Escritura* no fue posible compararlos, debido a que no se presentaron alumnos con esos mismos estilos de aprendizaje en el curso contrario. Con respecto a los alumnos con un estilo de aprendizaje *Kinestésico*, obtuvo un mejor promedio el alumno del curso *No Personalizado*, posiblemente debido a que contaba con un background de la materia.

En la gráfica de la Figura 51, se muestran los promedios de calificaciones Aprobatorias de cada estilo de aprendizaje, obtenidos en cada curso. Para calcular estos promedios, se utilizó solamente la última calificación de los alumnos.

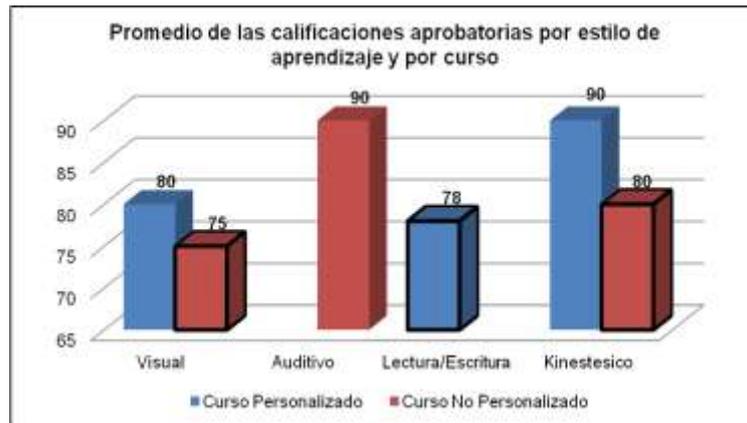


Figura 51. Promedios generales presentados por cada uno de los estilos de aprendizaje

Analizando la gráfica se observa, que los alumnos que presentan un background y un estilo de aprendizaje *Visual* y *Kinestésico*, obtuvieron un promedio menor, en comparación con los alumnos que les fue presentado el material de forma personalizada. Los estilos restantes no pudieron ser comparados, por las razones explicadas anteriormente.

Por otro lado, también se analizó la cantidad de alumnos a los cuales se les diagnosticó que poseían la competencia relacionada con el tema, esto en cada uno de los intentos, ver Figura 52.

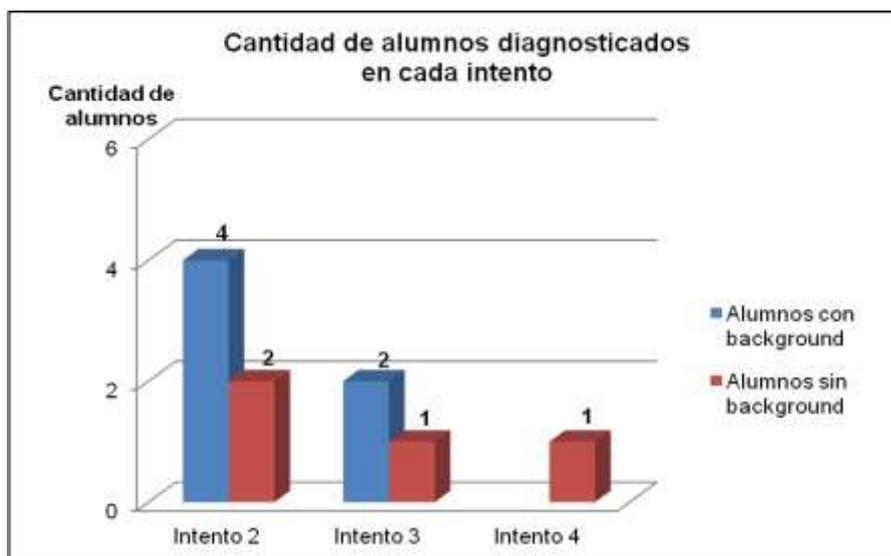


Figura 52. Cantidad de alumnos diagnosticados con competencia, en cada intento realizado

En la gráfica anterior, se puede observar que a los alumnos con un conocimiento previo de la materia, les fue diagnosticado antes que a los otros alumnos, que sí poseían

la competencia. Por lo tanto, se puede pensar que el funcionamiento del diagnóstico de las competencias se asemeja a la realidad.

7.4 Discusión de resultados

De acuerdo a los resultados obtenidos y a las observaciones realizadas durante el proceso de experimentación, en esta sección se describen algunas sugerencias o recomendaciones para llevar a cabo una experimentación más adecuada.

En particular, se observaron las siguientes tres áreas, en las cuales se pueden realizar mejoras en cuanto a: 1) las características para organizar la muestra a examinar, 2) los factores para evaluar el desempeño del alumno y 3) las consideraciones para la elaboración del material didáctico. A continuación se detalla cada una de ellas.

Características para organizar la muestra a examinar

- Considerar un mayor número de integrantes.
- Confirmar que estén presentes los cuatro estilos de aprendizaje del modelo VARK.
- Confirmar que la frecuencia de cada estilo de aprendizaje sea el mismo.
- Distribuir de manera uniforme a los alumnos que presenten un background de la materia.
- Trabajar con alumnos que sea del mismo semestre y carrera.
- Distribuir uniformemente el número de hombres y mujeres.

Factores para evaluar el desempeño del alumno

- Considerar el número de veces que revisan el material.
- Examinar el tiempo que tardan en revisar el material.
- Examinar el tiempo que tardan en contestar el examen de un tema.
- Proponer actividades que aporten al desarrollo de las competencias y que sean evaluables dentro de la plataforma Moodle.

Recomendaciones para la elaboración del material didáctico

- Diseñar un mismo tema, utilizando al menos cuatro distintas estrategias de enseñanza para un mismo estilo de aprendizaje.

- Considerar en el diseño del material didáctico algunos de estos elementos: ejemplos, ejercicios interactivos, imágenes, sonido, video, diagramas de flujo y mapas conceptuales, para hacer el tema más entendible.
- Presentar la secuencia de la información de forma coherente.
- Diseñar de forma neutral o utilizando distintas estrategias de enseñanza, el material didáctico que será mostrado a un grupo de alumnos al cual no se le presentará la información de forma personalizada. Esto con la intención de no favorecer a un estilo de aprendizaje en particular.
- Diseñar más a detalle, los materiales didácticos para el estilo de aprendizaje *Auditivo* y *Kinestesico*, ya que son los estilos que obtuvieron en la experimentación realizada, el desempeño más bajo.

Capítulo 8

CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se presentan las conclusiones de este proyecto de tesis, así como también se puntualizan algunas ideas que permiten extender esta línea de investigación.

8.1 Conclusiones

En este trabajo de investigación se propuso la creación de un agente inteligente en el módulo Tutor de un STI, el cual incorpora el concepto de estilo de aprendizaje para elegir una estrategia de enseñanza-aprendizaje, que será utilizada para mostrar los contenidos didácticos al alumno. También se propuso la creación de un proceso para diagnosticar si un alumno posee las competencias asociadas a los temas que le serán enseñados.

Para lograr estos dos objetivos fue necesario añadir y modificar algunos elementos de la arquitectura básica de un STI. Los principales elementos añadidos al módulo Tutor fueron: un módulo llamado *diagnóstico de las competencias* y un módulo llamado *agente selector*. Así como también, se modificó en el módulo Interfaz, el proceso encargado de mostrar el contenido a los alumnos, añadiendo a éste un proceso llamado *agente interfaz*.

El *diagnóstico de las competencias* es el encargado de llamar al *agente selector* cuando un alumno no obtiene un valor de probabilidad que indique que tiene desarrollada la competencia. Su función es determinar la probabilidad existente de que un alumno haya obtenido cierta calificación en un examen de algún tema, a causa de que tiene desarrollada la competencia requerida.

El *módulo agente selector* le indica al *agente interfaz* cual estrategia de enseñanza que debe considerar para filtrar los contenidos que le serán mostrados al alumno. Por lo tanto, es el encargado de elegir una estrategia de enseñanza que sea afín al estilo de aprendizaje actual del alumno.

El *agente interfaz* se encarga de filtrar los contenidos a mostrar al alumno, siempre y cuando los contenidos cumplan con la estrategia de enseñanza establecida por el *agente selector*.

La implementación de los módulos propuestos se llevo a cabo en el *Sistema Gestor de Aprendizaje Moodle*. La implementación fue probada utilizando a un grupo de alumnos para que interactuaran con la plataforma y así obtener resultados para su posterior análisis. La experimentación del proyecto consistió en, *a)* presentarle a un grupo de alumnos, mediante la plataforma Moodle, un tema el cual fue diseñado en cuatro distintas estrategias de enseñanza, cada una destinada hacía uno de los cuatro estilos de aprendizaje del modelo VARK. Esto con el fin de mostrarle al alumno el contenido de forma personalizada, es decir considerando su estilo de aprendizaje, *b)* presentarle a otro grupo de alumnos el mismo tema, pero sin considerar su estilo de aprendizaje, es decir, el contenido se les mostró a todos los alumnos en el mismo formato.

Los resultados obtenidos de la interacción entre los alumnos y la plataforma Moodle, fueron analizados. Concluyendo que los elementos utilizados en la experimentación, no fueron los idóneos para *asegurar* que se puede diagnosticar que un alumno posee una competencia dada. Como no se observó una diferencia significativa en la superioridad de las calificaciones logradas por los alumnos que recibieron el material personalizado, no se puede *aseverar* que mostrarle al alumno la información de acuerdo a su estilo de aprendizaje, conlleva que obtenga un mejor desempeño que otro alumno, al cual no se le enseña conforme a su estilo de aprendizaje. Dado estas situaciones, se identificaron factores que pueden mejorar el escenario de la experimentación, esperando obtener resultados más confiables.

8.2 Aportaciones

La principal aportación de esta tesis de investigación fue, modificar el módulo Tutor de la arquitectura básica de un STI, incorporando un agente inteligente selector de estrategias de enseñanza y un proceso para diagnosticar las competencias de un alumno.

Otras aportaciones realizadas en esta investigación fueron las siguientes:

- La implementación e incorporación en Moodle, del cuestionario VARK, con el cual se detecta el estilo de aprendizaje del alumno.
- La implementación de un proceso para diagnosticar si un alumno tiene desarrollada(s) cierta(s) competencia(s).
- La implementación de un agente selector encargado de elegir una estrategia de enseñanza que favorezca al estilo de aprendizaje del alumno.
- La modificación del proceso encargado de mostrar al alumno los materiales, incorporando un agente interfaz.
- La identificación de los elementos necesarios para llevar a cabo una experimentación más adecuada.

8.3 Trabajos Futuros

De acuerdo a los resultados obtenidos al implementar los módulos propuestos en esta investigación, se proponen los siguientes trabajos futuros:

- Mejorar el proceso de selección de la estrategia de enseñanza-aprendizaje, basándose en las calificaciones históricas del alumno.
- Analizar qué actividades de Moodle pueden ser utilizadas para desarrollar en el alumno las competencias, con el propósito de incluirlas como parte de la estructura de la red bayesiana.
- Expandir la estructura de la red bayesiana a más de dos niveles, modificando las interfaces de vinculación y de captura de probabilidades, para configurar las actividades descritas en el punto anterior. Esto con el fin de predecir si el alumno logrará adquirir las competencias de la materia.
- Integrar a la red bayesiana un método de aprendizaje de parámetros, para determinar los valores de las probabilidades para cada nodo, a partir de los datos.
- Diseñar los materiales didácticos para futuras evaluaciones, en base a las recomendaciones realizadas en esta tesis, producto de los resultados obtenidos en la experimentación.

ANEXOS

Anexo A. Creación y configuración de cursos en Moodle

El procedimiento descrito en esta sección, es para ilustrar la manera en la cual se crearon y configuraron en Moodle, los cursos utilizados en este proyecto de tesis. Para ejemplificar el procedimiento se crea un curso llamado *Fundamentos de Programación*, el cual consta de 5 unidades.

A continuación se presentan los pasos que conforman este proceso, acompañados de una imagen para reforzar la explicación.

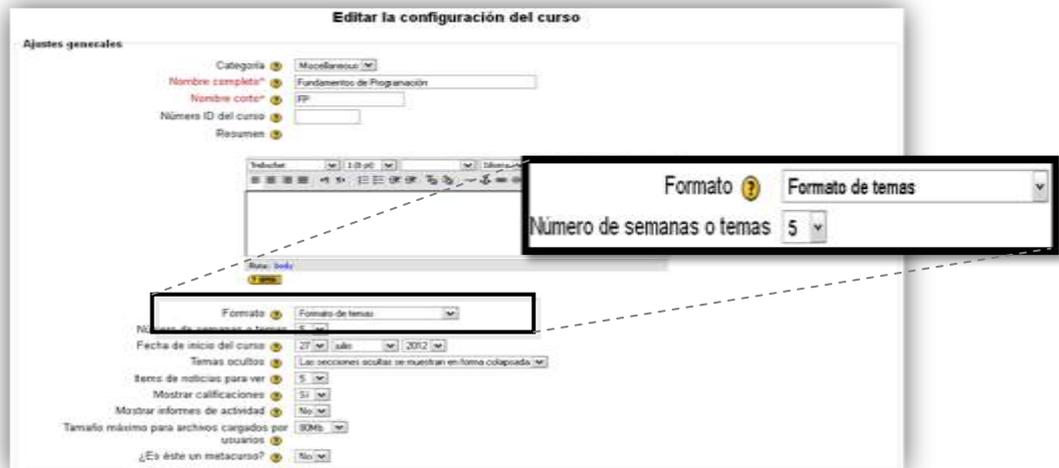
1. Entrar a Moodle como administrador.



2. Ubicar en la parte izquierda de la página principal, el menú *Administración del sitio*, abrir la carpeta *Cursos* y elegir *Agregar/editar cursos*. Enseguida se mostrará un formulario, elegir el botón *Agregar un nuevo curso*, aparecerá una página que permitirá editar la configuración del curso.



3. En la página *Editar la configuración del curso* se muestran varios campos. El valor para los campos *Nombre completo* y *Nombre corto* debe ser *Fundamentos de Programación* y *FP*, respectivamente. El campo *Formato* debe tener el valor *Formato de temas* y el campo *Número de semanas o temas* el valor *5*, el cual corresponde a las unidades del curso. Todos los demás campos dejarlos con los valores asignados por default.



4. Para finalizar la creación del curso, se oprime el botón *Guardar cambios* ubicado en la parte inferior de la misma página. El curso creado se visualiza de la siguiente forma.



5. Una vez creado el curso, se procede a dar nombre a las 5 unidades, para esto se debe oprimir el botón *Activar Edición* ubicado en la parte superior de la pantalla.



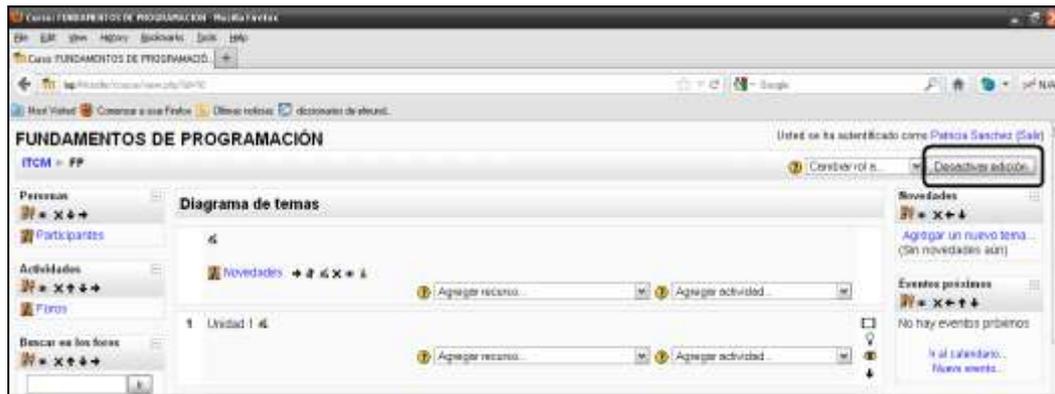
6. Dar clic en el ícono *Editar informe* en la sección correspondiente a la unidad 1.



7. Aparecerá una página para escribir el nombre de la unidad, en este caso *Unidad 1*. Posteriormente se debe dar clic al botón *Guardar cambios*.



8. Repetir los pasos 6 y 7 para nombrar a cada unidad. Cuando se haya terminado de nombrar la última unidad, oprimir el botón *Desactivar Edición* ubicado en la parte superior de la pantalla.



9. Finalmente la estructura del curso se mostrará como en la siguiente imagen.

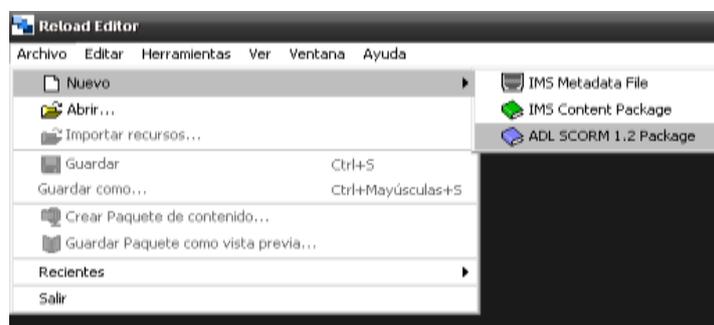
Diagrama de temas	
Novedades	
1	Unidad 1 <input type="checkbox"/>
2	Unidad 2 <input type="checkbox"/>
3	Unidad 3 <input type="checkbox"/>
4	Unidad 4 <input type="checkbox"/>
5	Unidad 5 <input type="checkbox"/>

Anexo B. Creación de paquetes SCORM y su empaquetamiento utilizando el software Reload Editor

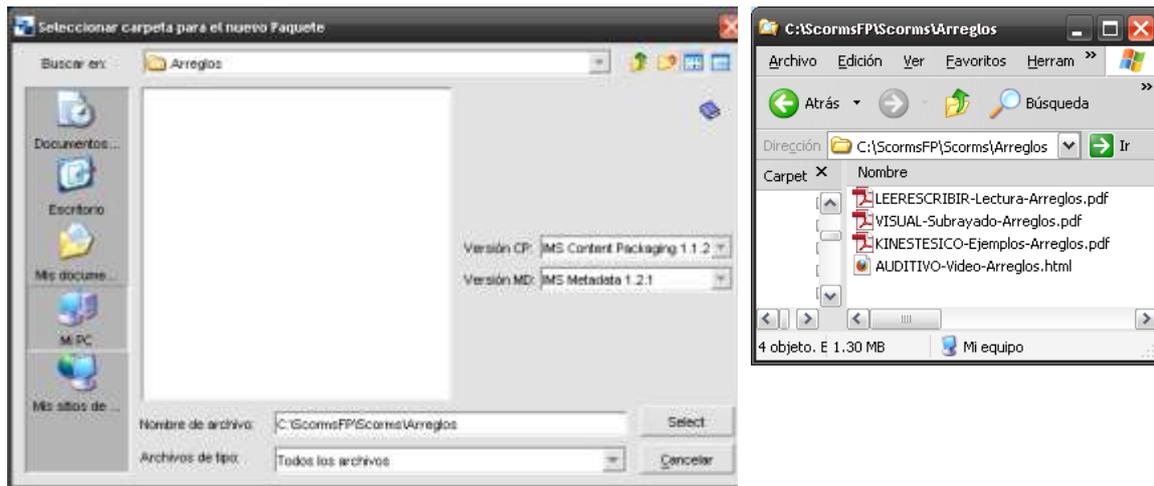
El procedimiento que se describe en esta sección, explica como se crearon los paquetes SCORM utilizados en este trabajo de tesis, utilizando el software *Reload Editor* (RE). Para ejemplificar el procedimiento se considera lo siguiente: que los contenidos que se empaquetan son cuatro archivos que tratan acerca del mismo tema, que cada archivo está diseñado utilizando una estrategia de enseñanza para cada uno de los cuatro estilos de aprendizaje del modelo VARK, y que la carpeta donde se encuentran almacenados dichos archivos se llama *Arreglos*.

A continuación se presentan los pasos que conforman este proceso, acompañados de una imagen para reforzar la explicación.

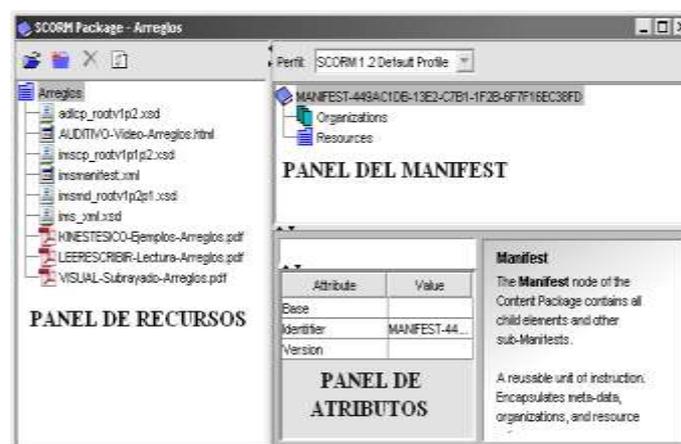
1. **Crear un paquete SCORM con *Reload Editor* (RE).** Abrir el programa *RE* y crear un paquete, eligiendo del menú *Archivo* la opción *Nuevo* y *ADL SCORM 1.2 Package*.



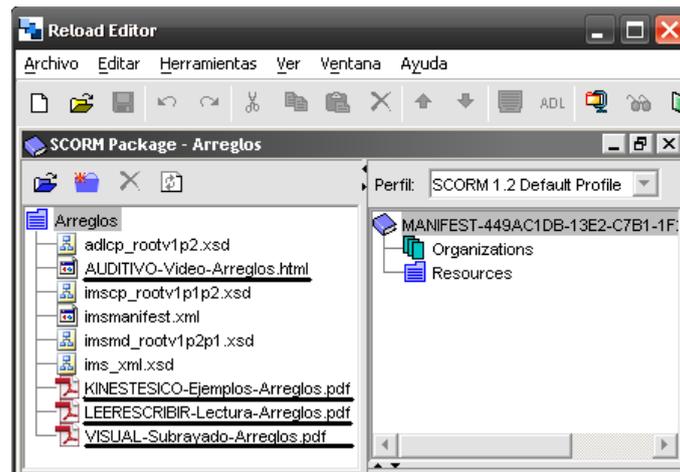
2. **Elegir la carpeta para crear el paquete.** Al crear un nuevo paquete se muestra una pantalla (imagen izquierda) solicitando seleccionar una carpeta para crear ahí el nuevo paquete. La carpeta a seleccionar, debe ser aquella donde se encuentran guardados los materiales didácticos (imagen derecha), ya que de esta manera esos recursos serán importados automáticamente al paquete que se está creando. Las opciones de *Versión CP* y *MD* que aparecen en la misma pantalla permanecerán con los valores sugeridos.



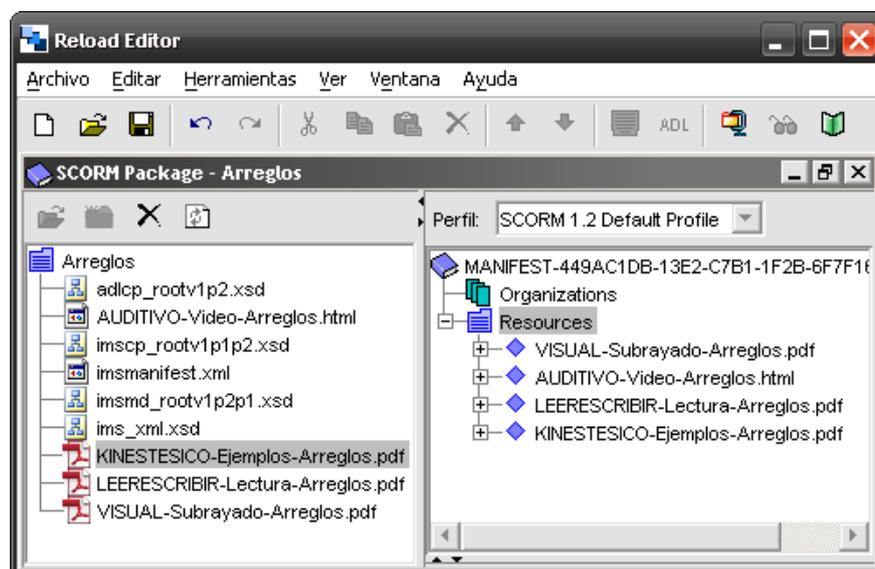
3. **Conocer el espacio de trabajo de RE.** Es necesario conocer el espacio de trabajo de *RE* ya que será referenciado a lo largo del procedimiento. El espacio consiste en tres paneles: el **Panel de Recursos** (izquierda), el **Panel del Manifest** (derecha) y el **Panel de Atributos** (abajo).



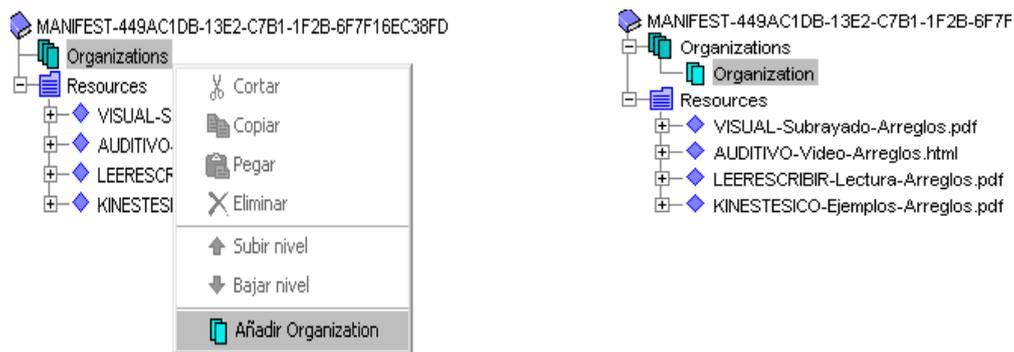
4. **Importar los archivos al paquete creado.** En el **Panel de Recursos** se aprecia que los archivos del tema *Arreglos* fueron importados al paquete creado. También se observan algunos otros archivos los cuales son creados por *RE*.



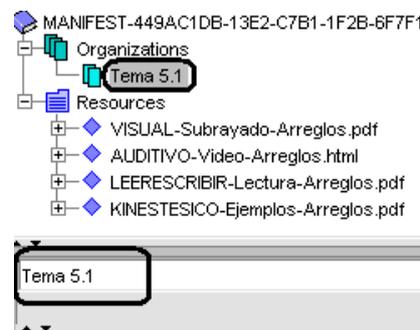
5. **Copiar los contenidos a la carpeta *Resources*.** Arrastrar cada uno de los archivos del tema *Arreglos* hacía la carpeta llamada *Resources* que se encuentra en el **Panel del Manifest**. La organización que deben seguir estos archivos, va de acuerdo al orden que sigue el modelo VARK, Visual, Auditivo, LecturaEscritura y Kinestesico. Esto es posible debido a que el nombre de cada archivo del tema *Arreglos* indica para cual estilo de aprendizaje fue diseñado.



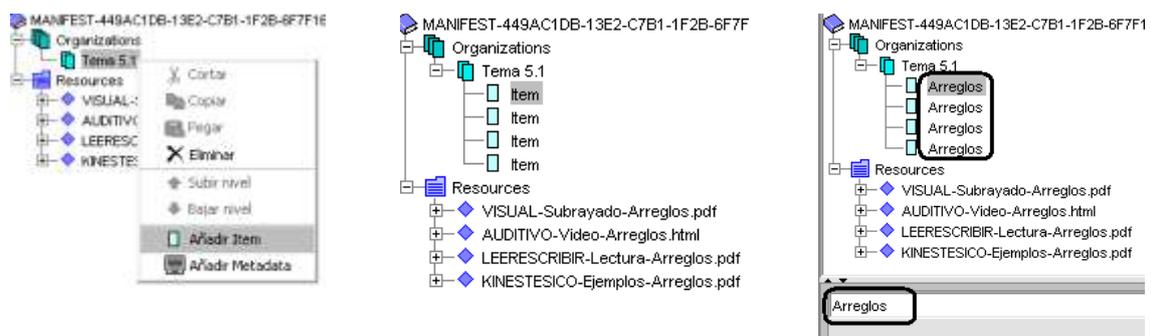
6. **Crear un elemento *Organization*.** Localizar el elemento *Organizations* en el **Panel del Manifest**, dar clic derecho sobre él para que aparezca un menú donde se debe elegir la opción *Añadir Organization* (imagen izquierda), la cual añadirá un elemento nuevo llamado *Organization* (imagen derecha).



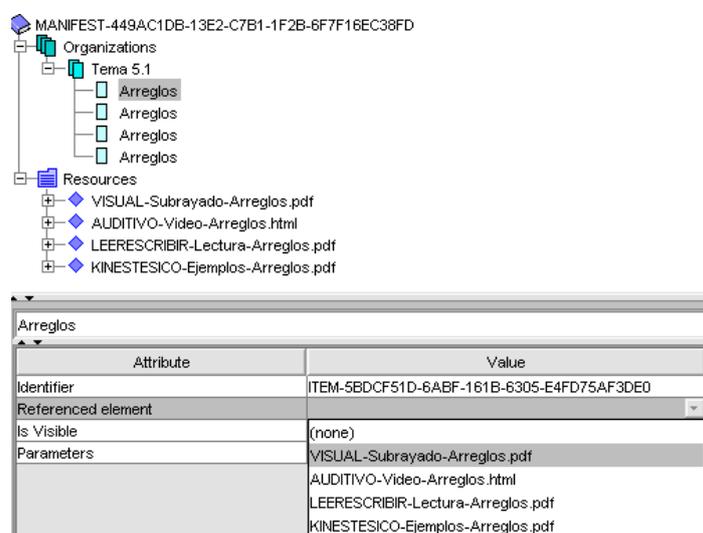
7. **Cambiar el nombre al elemento *Organization*.** Dar clic sobre el elemento *Organization* y localizar en el **Panel de Atributos**, un cuadro de texto que contiene la palabra *Organization*. Se elimina dicho texto y se escribe el nuevo nombre del elemento, en este caso *Tema 5.1*.



8. **Añadir ítems.** Añadir tantos ítems como archivos a empaquetar, en este caso son cuatro los archivos que se van a empaquetar, por lo tanto se añaden cuatro ítems. Un ítem se agrega dando clic derecho sobre el elemento *Tema 5.1* para que despliegue un menú en el cual se elige la opción *Añadir Ítem*. De esta forma se añaden los cuatro ítems, los cuales deben tener el mismo nombre, en este caso *Arreglos*. Para cambiar el nombre a los ítems se sigue el mismo procedimiento que en el paso 7.



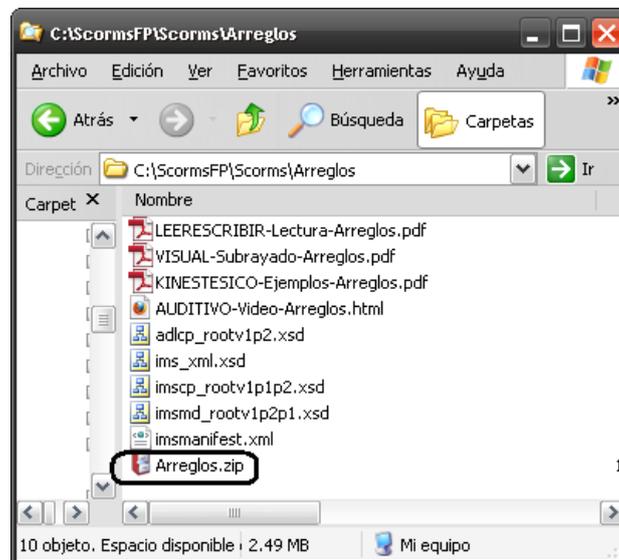
9. **Relacionar ítems con recursos.** Seleccionar el primer ítem *Arreglos*, ubicar en el **Panel de Atributos** el atributo *Referenced Element*. En ese mismo renglón, pero de la columna **Value**, dar un clic el cual hará que se despliegue una lista con todos los nombres de los archivos (recursos) que se importaron al paquete, como se muestra en la siguiente imagen. Se relaciona a cada ítem con sólo un recurso. Por lo tanto, para el primer ítem *Arreglos* se selecciona el recurso diseñado para el estilo de aprendizaje *Visual*, el segundo ítem *Arreglos* se relaciona con el recurso diseñado para *Auditivo* y así sucesivamente obedeciendo el orden de los estilos de aprendizaje del modelo VARK. Los demás atributos permanecen con los valores por default.



10. **Guardar el paquete en formato zip.** Para importar este paquete de contenidos a Moodle, es necesario empaquetarlo en un archivo con formato zip. Para lo cual se debe localizar en la barra de herramientas el ícono de los archivos con formato zip  (imagen derecha). Dar clic al ícono, probablemente se muestre un mensaje informando que el paquete ha sido modificado que si se guardan los cambios, elegir *Sí*. A continuación se muestra una ventana solicitando la ubicación donde se guardará el archivo zip (imagen izquierda), se debe elegir la carpeta en la cual se ha estado trabajando, asignando al archivo, el nombre de *Arreglos.zip*.



11. **Verificar la existencia del archivo zip.** Por último se verifica que el archivo zip se haya creado en el directorio seleccionado, con lo cual se concluye el empaquetado de contenidos en formato SCORM.



Anexo C. Importar paquetes SCORM a Moodle

En esta sección, se ilustra el procedimiento realizado para importar a Moodle, los paquetes SCORM utilizados en este trabajo de tesis. Para ejemplificar el procedimiento se considera lo siguiente: el nombre del curso es *Fundamentos de Programación*, el paquete SCORM se llama *Arreglos.zip* el cual se importa a la *Unidad 1* del curso.

A continuación se presentan los pasos que conforman este proceso, acompañados de una imagen para reforzar la explicación.

1. Entrar a la plataforma como administrador.



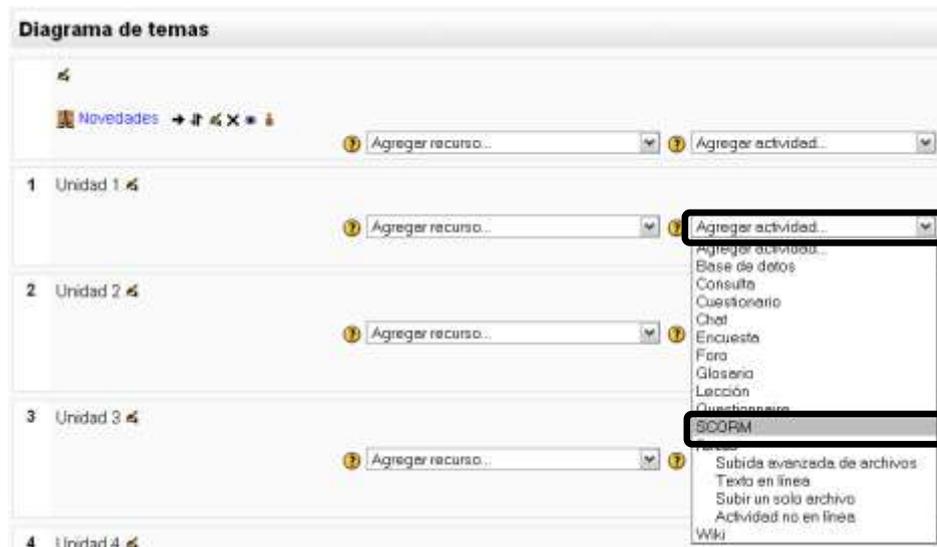
2. Elegir el curso al que se desea importar el paquete SCORM, en este caso *Fundamentos de Programación*.



3. Una vez dentro del curso dar clic en el botón *Activar edición*.



4. Dar clic al *combobox* **Agregar actividad** ubicado en el lado derecho de la *Unidad 1*, éste despliega un menú del cual se debe elegir la opción **SCORM**.



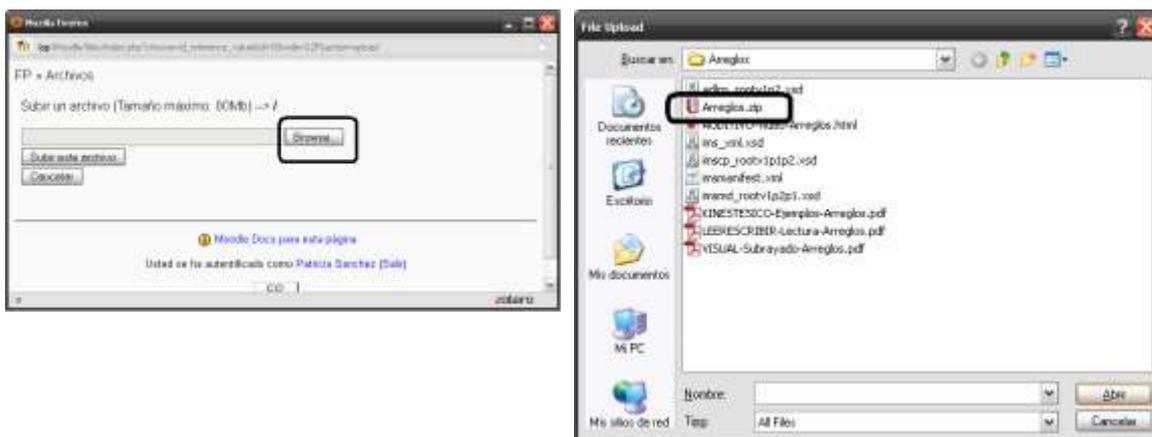
5. Al haber elegido la actividad SCORM, se muestra una página para configurar el paquete. La página tiene tres campos obligatorios que deben llenarse: **Nombre**, **Resumen** y **Paquete**. En el campo **Nombre** se especifica el nombre del paquete, el cual será con el que se identifica en el curso, en este caso *Arreglos*. En el campo **Resumen** se escribe una síntesis del tema. En el campo **Paquete** dar clic en el botón que se encuentra a la derecha llamado **Elija o suba un archivo**.



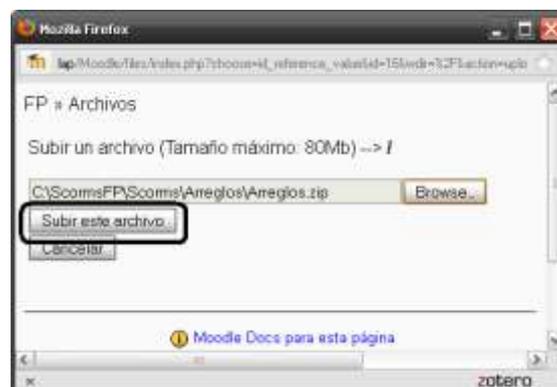
6. El botón *Elija o suba un archivo* abre otra página, donde se dará clic al botón *Subir un archivo*.



7. Al dar clic al botón *Subir un archivo*, se muestra una página, oprimir el botón *Browse* el cual sirve para localizar el archivo *zip* que se desea importar a Moodle, en este caso se utilizó el archivo *Arreglos.zip* creado en el Anexo B.



8. Después de ubicar y seleccionar el archivo *Arreglos.zip*, dar clic al botón *Subir este archivo*.



9. Una vez subido el archivo, se muestra una página en la cual se selecciona el *checkbox* que se encuentra en el lado derecho del nombre del archivo *Arreglos.zip*, después seleccionar la opción **Elegir** ubicada en el lado izquierdo del archivo. Con esto se finaliza la importación del paquete SCORM a Moodle, e inmediatamente se retorna automáticamente a la página inicial donde se configura el paquete.



10. Los demás campos de la página donde se configura el paquete, deben permanecer con los valores por default. Por último se oprime el botón **Guardar cambios y regresar al curso** ubicado en la parte inferior de la página, el cual retorna automáticamente a la página principal del curso. Estando en la página principal del curso se oprime el botón **Desactivar edición**.



11. El curso con el paquete SCORM *Arreglos.zip* integrado en la *Unidad 1* del curso, se muestra como en la siguiente imagen.

Diagrama de temas		
	 Novedades	
1	Unidad 1  Arreglos	<input type="checkbox"/>
2	Unidad 2	<input type="checkbox"/>
3	Unidad 3	<input type="checkbox"/>
4	Unidad 4	<input type="checkbox"/>
5	Unidad 5	<input type="checkbox"/>

Anexo D. Creación de exámenes utilizando Moodle

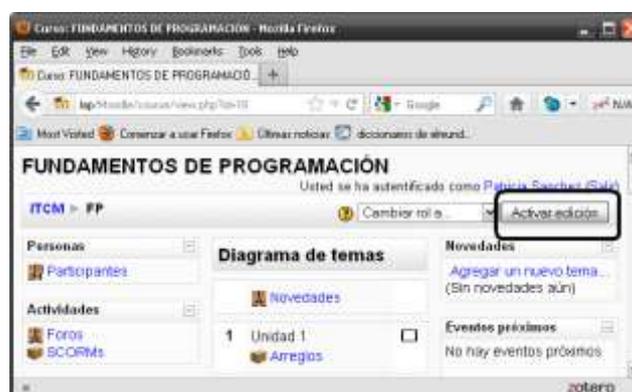
En esta sección se detalla el procedimiento realizado para crear los exámenes utilizados en este trabajo de tesis. Para ilustrar el procedimiento se considera lo siguiente: el examen se crea en la *Unidad 1* del curso *Fundamentos de Programación* y el examen es acerca del tema *Arreglos*. El examen consta de *5 preguntas* y para cada pregunta existen *4 posibles respuestas*, es decir el examen es de tipo *opción múltiple*. Cada pregunta del examen tiene un valor de *20 puntos*, sumando un total de *100 puntos*, siendo la calificación máxima del examen.

A continuación se presentan los pasos que conforman este proceso, acompañados de una imagen para reforzar la explicación.

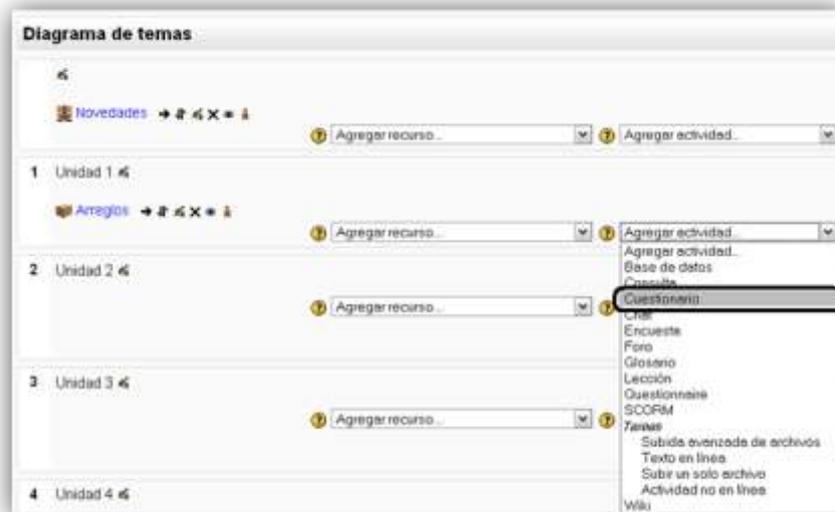
1. Entrar a la plataforma como administrador y elegir el curso.



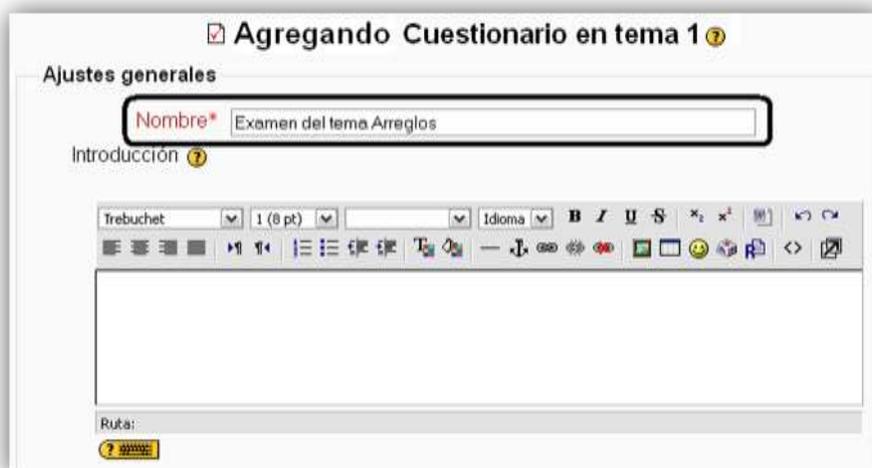
2. Activar edición.



3. En la sección de la *Unidad 1* compuesta por el tema *Arreglos*, seleccionar *Agregar actividad* de tipo *Cuestionario*.



4. Inmediatamente se presenta una página con varias secciones para configurar el cuestionario. En la sección **Ajustes Generales**, el campo **Nombre** es obligatorio, por lo que se nombró al cuestionario “*Examen del tema Arreglos*”. El campo **Introducción** dejarlo vacío.



5. En la sección **Mostrar**, el campo **Número máximo de preguntas por página** dejarlo con el valor por default. El valor para los campos **Barajar preguntas** y **Barajar dentro de las preguntas** debe ser **Sí**.

Mostrar

Número máximo de preguntas por página

Barajar preguntas

Barajar dentro de las preguntas

6. En la sección **Intentos**, el valor del campo **Intentos permitidos** debe ser **Sin límite**, y el valor para los campos **Cada intento se basa en el anterior** y **Modo adaptativo** debe ser **No**.

Intentos

Intentos permitidos

Cada intento se basa en el anterior

Modo adaptativo

7. En la sección **Calificaciones**, el valor del campo **Método de calificación** debe ser **Último intento**, el valor del campo **Aplicar penalizaciones** debe ser **No** y el valor del campo **Número de decimales en calificaciones** debe ser **0**.

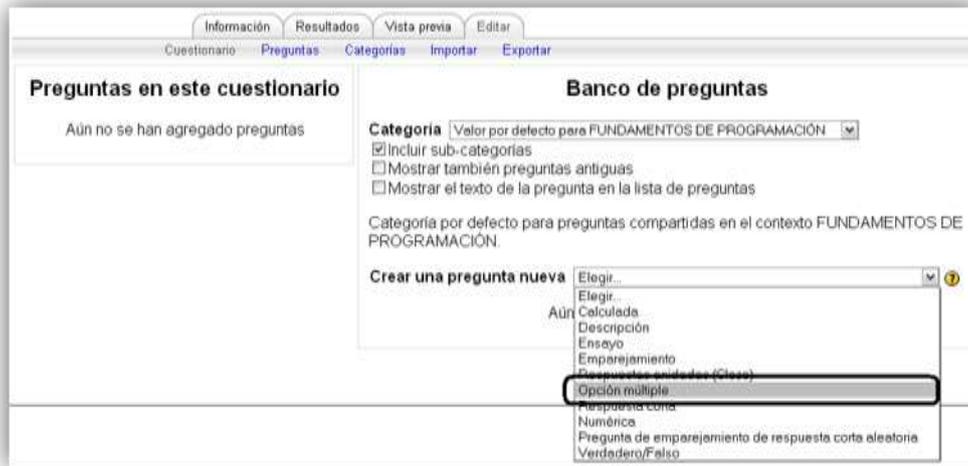
Calificaciones

Método de calificación

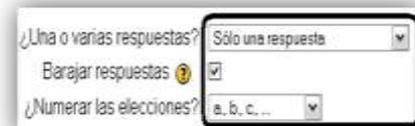
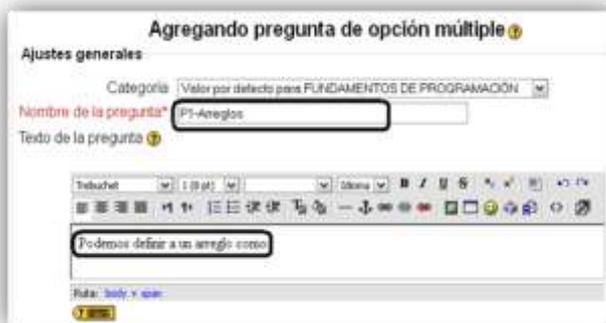
Aplicar penalizaciones

Número de decimales en calificaciones

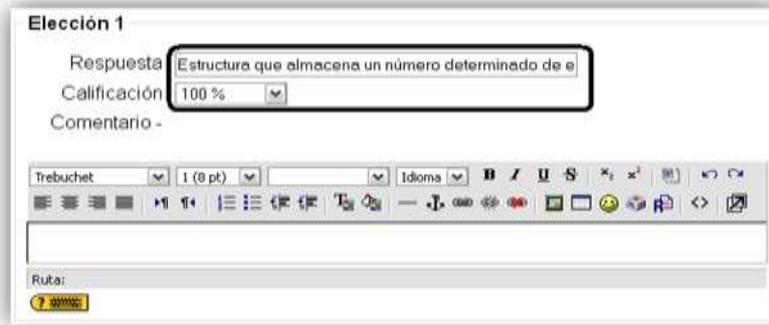
8. Existen otras secciones que no son fundamentales para este proyecto de tesis, por lo tanto se quedarán con los valores asignados por default. Para finalizar la configuración y creación del cuestionario se debe dar clic al botón **Guardar cambios y mostrar** ubicado en la parte inferior de la página, inmediatamente se presenta otra página para crear las preguntas del cuestionario.
9. En la página para editar preguntas, el valor para el campo **Crear una pregunta nueva** debe ser **Opción múltiple**, los demás campos se dejan con los valores asignados por default.



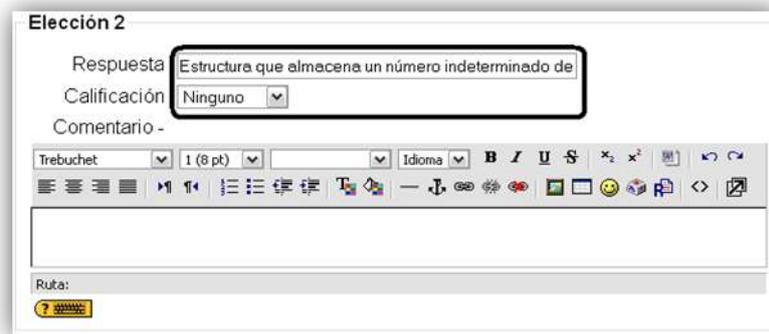
10. Inmediatamente se presenta otra página con varias secciones, la cual permite editar la nueva pregunta y sus posibles respuestas. En la sección **Ajustes generales** el valor para el campo **Nombre de la pregunta** es *P1-Arreglos*. El valor para el campo **Texto de la pregunta** es la pregunta en sí que se desea realizar, el valor para el campo **¿Una o varias respuestas?** debe ser *Sólo una respuesta*, el valor para el campo **Barajar respuestas** debe estar elegido y el campo **¿Numerar las elecciones?** debe tener el valor *a., b., c...*. Los campos restantes de dicha sección deben permanecer con los valores por default.



11. En la sección **Elección 1**, se escribirá la primera opción de respuesta, la cual debe ser la correcta. En el campo **Respuesta** debe ir el texto correspondiente a la respuesta correcta. El campo **Calificación** debe tener un valor de **100%**, debido a que esta respuesta es la única correcta y representa al 100% del valor de la pregunta. El campo **Comentario** se debe dejar en blanco.



12. En las secciones *Elección 2* hasta la *4*, se deben escribir las otras tres opciones de respuesta que deben ser incorrectas. Por lo tanto el valor para el campo *Calificación* estas tres respuestas debe ser *Ninguno*. Las otras secciones se dejan con los valores por default.



13. Cuando se hayan editado las cuatro posibles respuestas, se oprime el botón *Guardar cambios* que se encuentra al final de la página. Se presenta nuevamente la página para crear preguntas, en la cual se muestra la pregunta creada. Se vuelve a repetir el proceso a partir del paso 9 al 13 para crear las cuatro preguntas restantes. Una vez que se crearon las cinco preguntas, se habilita el *checkbox* que se encuentra a la izquierda del nombre de cada una de las preguntas, posteriormente se da clic al botón *Añadir a cuestionario*.



14. Las preguntas son copiadas a la sección *Preguntas en este cuestionario*. Cada pregunta tiene en su campo *Calificación* un valor de default, éste debe ser borrado y asignar el valor **20**, que es el puntaje para cada pregunta. El valor por default del campo *Calificación máxima*, debe ser borrado y asignar el valor **100**, el cual representa la suma de los puntos de cada pregunta. Por último oprimir el botón *Guardar cambios*.



15. Una vez estructurado el examen, se puede tener una vista previa del mismo. Se debe elegir la pestaña *Vista previa* ubicada en la parte superior de la página y el examen será visualizado.



Vista previa del cuestionario

[Comenzar de nuevo](#)

1 Cuál de las siguientes expresiones ejemplifica la creación e inicialización predeterminada de un arreglo:

Puntos: 20

Seleccione una respuesta.

- a. `int[] vector = {3, 4, 6, 1, 8};`
- b. `int[] numeros = new int[10];`
- c. `char[] cadena;`
- d. `cadena = new char[5];`

2 Cuál de las siguientes expresiones ejemplifica **exclusivamente** la declaración de un arreglo:

Puntos: 20

Seleccione una respuesta.

- a. `numeros = new int[10];`
- b. `int[] vector = {3, 4, 6, 1, 8};`
- c. `int[] numeros;`
- d. `int[] numeros = new int[10];`

Anexo E: Configuración de campos de perfil del usuario

En esta sección se describe el procedimiento realizado para añadir los campos *Visual*, *Auditivo*, *LeerEscribir*, *Kinestesico*, *Estilo_actual* y *EstrategiaActual*. Estos campos se utilizan en este proyecto de tesis, para guardar las puntuaciones obtenidas por cada alumno al contestar el cuestionario VARK, así como también almacena el estilo de aprendizaje y la estrategia de enseñanza actual utilizadas por el agente selector.

A continuación se presentan los pasos que conforman este proceso, acompañados de una imagen para reforzar la explicación.

1. Entrar a Moodle como administrador, ubicar el menú *Administración del sitio*, abrir la carpeta *Usuarios – Cuentas* y elegir *Campos de perfil del usuario*.



2. Se mostrará una página donde se elegirá el valor *Entrada de Texto*, para el campo *Crear un nuevo campo de perfil*. Enseguida se abre un formulario para configurar el nuevo campo agregado.



3. En la sección *Ajustes comunes*, se deben llenar obligatoriamente los campos *Nombre corto* y *Nombre*, con los datos del campo que se quiere crear. El valor para el campo *¿Está este campo bloqueado?* debe ser *Sí*, el valor para el campo *¿Quién puede ver este campo?* debe ser *No visible*. Los demás campos de esta sección y de la sección *Ajustes específicos*, quedarán con los valores default. Esta configuración es para agregar los campos correspondientes a los estilos de aprendizaje.

Creando un nuevo "Entrada de texto" campo de perfil

Ajustes comunes

Nombre corto (único)*

Nombre*

Descripción del campo

Rich text editor (Trabuchal) with toolbar and content area.

Reglas:

¿Es este campo necesario? No Sí

¿Está este campo bloqueado? No Sí

¿Deberían ser únicos los datos? No Sí

¿Mostrar página al inscribirse? No Sí

¿Quién puede ver este campo? No visible Visible

Categoría Otros campos

Ajustes específicos

Valor por defecto

Mostrar tamaño

Longitud máxima

¿Es éste un campo de contraseña?

Nota: La configuración para el campo *Edad*, se muestra a continuación.

¿Es este campo necesario?

¿Está este campo bloqueado?

¿Deberían ser únicos los datos?

¿Mostrar página al inscribirse?

¿Quién puede ver este campo?

Categoría

Anexo F: Script para añadir las tablas a la base de datos de Moodle

```

-- MySQL Administrator dump 1.4
--
-----
-- Server version      5.0.51b-community-nt-log

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

--
-- Definition of table `historial_agenteselector`
--
DROP TABLE IF EXISTS `mdl_historial_agenteselector`;
CREATE TABLE `mdl_historial_agenteselector` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `Estilo_Actual` int(10) unsigned NOT NULL,
  `Estrategia_Actual` int(10) unsigned NOT NULL,
  `Nuevo_Estilo` int(10) unsigned NOT NULL,
  `Nueva_Estrategia` int(10) unsigned NOT NULL,
  `Fecha_Hora` datetime NOT NULL,
  `userid` int(10) unsigned NOT NULL,
  `calificacion` float NOT NULL,
  `uniqueid` int(10) unsigned NOT NULL,
  `attempt` int(10) unsigned NOT NULL,
  `cursoid` int(10) unsigned NOT NULL,
  `scormid` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=327 DEFAULT CHARSET=utf8;

--
-- Definition of table `historial_vark`
--
DROP TABLE IF EXISTS `mdl_historial_vark`;
CREATE TABLE `mdl_historial_vark` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `userid` int(10) unsigned NOT NULL,
  `respuestas` varchar(200) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=81 DEFAULT CHARSET=utf8 ROW_FORMAT=DYNAMIC;

--
-- Definition of table `mdl_competencias`
--
DROP TABLE IF EXISTS `mdl_competencias`;
CREATE TABLE `mdl_competencias` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `cursoid` int(10) unsigned NOT NULL,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_competencias_temas`
--
DROP TABLE IF EXISTS `mdl_competencias_temas`;
CREATE TABLE `mdl_competencias_temas` (

```

```

    `id` int(10) unsigned NOT NULL auto_increment,
    `cursoid` int(10) unsigned NOT NULL,
    `competenciaid` int(10) unsigned NOT NULL,
    `scormid` int(10) unsigned NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=40 DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_estilos`
--
DROP TABLE IF EXISTS `mdl_estilos`;
CREATE TABLE `mdl_estilos` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;

--
-- Dumping data for table `mdl_estilos`
--
/*!40000 ALTER TABLE `mdl_estilos` DISABLE KEYS */;
INSERT INTO `mdl_estilos` (`id`,`name`) VALUES
  (1,'Visual'),
  (2,'Auditivo'),
  (3,'LeerEscribir'),
  (4,'Kinestesico');
/*!40000 ALTER TABLE `mdl_estilos` ENABLE KEYS */;

--
-- Definition of table `mdl_estrategias`
--
DROP TABLE IF EXISTS `mdl_estrategias`;
CREATE TABLE `mdl_estrategias` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `estiloid` int(10) unsigned NOT NULL,
  `name` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_probabilidades_competencias`
--
DROP TABLE IF EXISTS `mdl_probabilidades_competencias`;
CREATE TABLE `mdl_probabilidades_competencias` (
  `Id` int(10) unsigned NOT NULL auto_increment,
  `CursoId` int(10) unsigned NOT NULL,
  `CompetenciaId` int(10) unsigned NOT NULL,
  `Porc_No` decimal(8,2) NOT NULL,
  `Porc_Si` decimal(8,2) NOT NULL,
  `ExamenId` int(10) unsigned NOT NULL,
  `IntentoExamen` int(10) unsigned NOT NULL,
  `UsuarioId` int(10) unsigned NOT NULL,
  PRIMARY KEY (`Id`)
) ENGINE=InnoDB AUTO_INCREMENT=109 DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_rb_competencias_porc`
--
DROP TABLE IF EXISTS `mdl_rb_competencias_porc`;
CREATE TABLE `mdl_rb_competencias_porc` (
  `cursoid` int(10) unsigned NOT NULL,
  `competenciaid` int(10) unsigned NOT NULL,
  `porc_si` decimal(8,2) NOT NULL,
  `porc_no` decimal(8,2) NOT NULL,
  PRIMARY KEY (`cursoid`,`competenciaid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_rb_temas_competencias_comb`
--
DROP TABLE IF EXISTS `mdl_rb_temas_competencias_comb`;
CREATE TABLE `mdl_rb_temas_competencias_comb` (
  `cursoid` int(10) unsigned NOT NULL,

```

```

`partida` int(10) unsigned NOT NULL,
`scormid` int(10) unsigned NOT NULL,
`competenciaid` int(10) unsigned NOT NULL,
`valor_binario` int(10) unsigned NOT NULL,
`SubPartida` int(10) unsigned NOT NULL,
PRIMARY KEY USING BTREE (`partida`,`cursoid`,`SubPartida`,`scormid`,`competenciaid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_rb_temas_competencias_porc`
--
DROP TABLE IF EXISTS `mdl_rb_temas_competencias_porc`;
CREATE TABLE `mdl_rb_temas_competencias_porc` (
  `cursoid` int(10) unsigned NOT NULL,
  `scormid` int(10) unsigned NOT NULL,
  `Porc_Rango1` decimal(8,2) NOT NULL,
  `Porc_Rango2` decimal(8,2) NOT NULL,
  `Porc_Rango3` decimal(8,2) NOT NULL,
  `Porc_Rango4` decimal(8,2) NOT NULL,
  `Partida` int(10) unsigned NOT NULL,
  PRIMARY KEY USING BTREE (`cursoid`,`scormid`,`Partida`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_scorms_ee`
--
DROP TABLE IF EXISTS `mdl_scorms_ee`;
CREATE TABLE `mdl_scorms_ee` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `scormid` int(10) unsigned NOT NULL,
  `scoid` int(10) unsigned NOT NULL,
  `estiloid` int(10) unsigned NOT NULL,
  `estrategiaid` int(10) unsigned NOT NULL,
  `cursoid` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=232 DEFAULT CHARSET=utf8;

--
-- Definition of table `mdl_tema_examen`
--
DROP TABLE IF EXISTS `mdl_tema_examen`;
CREATE TABLE `mdl_tema_examen` (
  `id` int(10) unsigned NOT NULL auto_increment,
  `cursoid` int(10) unsigned NOT NULL,
  `seccionid` int(10) unsigned NOT NULL COMMENT 'Valor obtenido de la tabla:
mdl_course_sections',
  `scormid` int(10) unsigned NOT NULL,
  `examenid` int(10) unsigned NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=27 DEFAULT CHARSET=utf8 ROW_FORMAT=DYNAMIC
COMMENT='Enlaza cada tema con su respectivo examen';

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

```

Anexo G: Cuestionario VARK

1. Estás ayudando a una persona que desea ir al aeropuerto, al centro de la ciudad o a la estación del ferrocarril. Tu:
 - irías con ella.
 - le dirías cómo llegar.
 - le darías las indicaciones por escrito (sin un mapa).
 - le darías un mapa.

2. No estás seguro si una palabra se escribe como: trascendente o tracendente. Tu:
 - verías las palabras en tu mente y elegirías la que mejor luce.
 - pensarías en cómo suena cada palabra y elegirías una.
 - las buscarías en un diccionario.
 - escribirías ambas palabras y elegirías una.

3. Estás planeando unas vacaciones para un grupo de personas y desearías la retroalimentación de ellos sobre el plan. Tu:
 - describirías algunos de los atractivos del viaje.
 - utilizarías un mapa o un sitio web para mostrar los lugares.
 - les darías una copia del itinerario impreso.
 - les llamarías por teléfono, les escribirías o les enviarías un e-mail.

4. Vas a cocinar algún platillo especial para tu familia. Tu:
 - cocinarías algo que conoces sin la necesidad de instrucciones.
 - pedirías sugerencias a tus amigos.
 - hojearías un libro de cocina para tomar ideas de las fotografías.
 - utilizarías un libro de cocina donde sabes que hay una buena receta.

5. Un grupo de turistas desea aprender sobre los parques o las reservas de vida salvaje en tu área. Tu:
 - les darías una plática acerca de parques o reservas de vida salvaje.
 - les mostrarías figuras de Internet, fotografías o libros con imágenes.
 - los llevarías a un parque o reserva y darías una caminata con ellos.
 - les darías libros o folletos sobre parques o reservas de vida salvaje.

6. Estás a punto de comprar una cámara digital o un teléfono móvil. ¿Además del precio, qué más influye en tu decisión?
 - lo utilizas o lo pruebas.
 - la lectura de los detalles acerca de las características del aparato.
 - el diseño del aparato es moderno y parece bueno.
 - los comentarios del vendedor acerca de las características del aparato.

7. Recuerda la vez cuando aprendiste cómo hacer algo nuevo. Evita elegir una destreza física, como montar bicicleta. ¿Cómo aprendiste mejor?:
 - viendo una demostración.
 - escuchando la explicación de alguien y haciendo preguntas.

-
-
- siguiendo pistas visuales en diagramas y gráficas.
 - siguiendo instrucciones escritas en un manual o libro de texto.
8. Tienes un problema con tu rodilla. Preferirías que el doctor:
- te diera una dirección web o algo para leer sobre el asunto.
 - utilizara el modelo plástico de una rodilla para mostrarte qué está mal.
 - te describiera qué está mal.
 - te mostrara con un diagrama qué es lo que está mal.
9. Deseas aprender un nuevo programa, habilidad o juego de computadora. Tu debes:
- leer las instrucciones escritas que vienen con el programa.
 - platicar con personas que conocen el programa.
 - utilizar los controles o el teclado.
 - seguir los diagramas del libro que vienen con el programa.
10. Te gustan los sitios web que tienen:
- cosas que se pueden picar, mover o probar.
 - un diseño interesante y características visuales
 - descripciones escritas interesantes, características y explicaciones
 - canales de audio para oír música, programas o entrevistas.
11. Además del precio, ¿qué influiría más en tu decisión de comprar un nuevo libro (que no sea de ficción)?
- la apariencia te resulta atractiva.
 - una lectura rápida de algunas partes del libro.
 - un amigo te habla del libro y te lo recomienda.
 - tiene historias, experiencias y ejemplos de la vida real.
12. Estás utilizando un libro, CD o sitio web para aprender cómo tomar fotografías con tu nueva cámara digital. Te gustaría tener:
- la oportunidad de hacer preguntas y que te hablen sobre la cámara y sus características.
 - instrucciones escritas con claridad, con características y puntos sobre qué hacer.
 - diagramas que muestren la cámara y qué hace cada una de sus partes.
 - muchos ejemplos de fotografías buenas y malas y cómo mejorarlas.
13. Prefieres a un profesor o un expositor que utiliza:
- demostraciones, modelos o sesiones prácticas.
 - preguntas y respuestas, charlas, grupos de discusión u oradores invitados.
 - folletos, libros o lecturas.
 - diagramas, esquemas o gráficas.
14. Has terminado una competencia o una prueba y quisieras una retroalimentación. Quisieras tener la retroalimentación:
- utilizando ejemplos de lo que has hecho.
 - utilizando una descripción escrita de tus resultados.
 - escuchando a alguien haciendo una revisión detallada de tu desempeño.
 - utilizando gráficas que muestren lo que has conseguido.

15. Vas a elegir tus alimentos en un restaurante o café. Tu:

- elegirías algo que ya has probado en ese lugar.
- escucharías al mesero o pedirías recomendaciones a tus amigos.
- elegirías a partir de las descripciones del menú.
- observarías lo que otros están comiendo o las fotografías de cada platillo.

16. Tienes que hacer un discurso importante para una conferencia o una ocasión especial. Tu:

- elaborarías diagramas o conseguirías gráficos que te ayuden a explicar las ideas.
- escribirías algunas palabras clave y practicarías tu discurso repetidamente.
- escribirías tu discurso y te lo aprenderías leyéndolo varias veces.
- conseguirías muchos ejemplos e historias para hacer la charla real y práctica.

TABLA DE RESPUESTAS

Pregunta	Respuesta a	Respuesta b	Respuesta c	Respuesta d
1	K	A	R	V
2	V	A	R	K
3	K	V	R	A
4	K	A	V	R
5	A	V	K	R
6	K	R	V	A
7	K	A	V	R
8	R	K	A	V
9	R	A	K	V
10	K	V	R	A
11	V	R	A	K
12	A	R	V	K
13	K	A	R	V
14	K	R	A	V
15	K	A	R	V
16	V	A	R	K

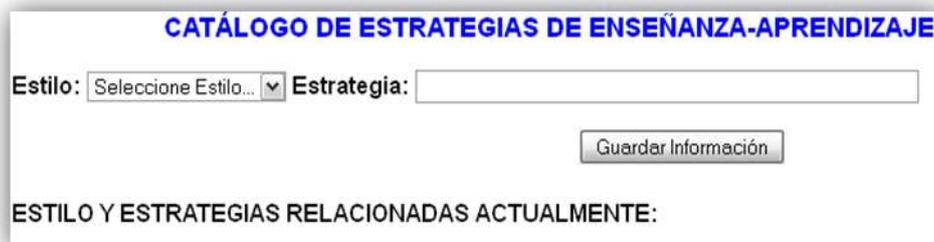
Anexo H: Interfaces para los catálogos de estilos, estrategias y competencias

Catálogo de Estilos. Este catálogo sólo muestra los estilos de aprendizaje del Modelo VARK.



Id	Estilo
1	Visual
2	Auditivo
3	LeerEscribir
4	Kinestetico

Catálogo de Estrategias de enseñanza. Permite editar nuevas estrategias de enseñanza.



CATÁLOGO DE ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE

Estilo: Estrategia:

ESTILO Y ESTRATEGIAS RELACIONADAS ACTUALMENTE:

Catálogo de Competencias. Permite editar nuevas competencias específicas, relacionándolas con el curso a enseñar.



CATÁLOGO DE COMPETENCIAS ESPECÍFICAS

Curso:

Competencia:

COMPETENCIAS ESPECÍFICAS REGISTRADAS ACTUALMENTE:

Anexo I: Interfaces para realizar la vinculación entre temas, estilos de aprendizaje, estrategias de enseñanza y competencias.

Vincular tema – estilo – estrategia. Permite editar la relación existente entre un tema y el estilo de aprendizaje y estrategia de enseñanza para el cual fue diseñado, esto en un curso específico.

VINCULAR CADA TEMA CON EL ESTILO Y LA ESTRATEGIA PARA EL CUAL FUE DISEÑADO

Curso:

Tema: Recurso: Estilo: Estrategia:

TEMAS RELACIONADOS ACTUALMENTE A UN ESTILO Y A UNA ESTRATEGIA DE ENSEÑANZA:

Vincular temas con su respectivo examen. Permite editar la relación de un tema, de un curso y unidad específicos, con su examen correspondiente.

VINCULAR CADA TEMA CON SU RESPECTIVO EXAMEN

Curso:

Unidad / Sección:

Tema: Examen:

TEMAS Y EXÁMENES RELACIONADOS ACTUALMENTE:

Vincular temas con la competencia correspondiente. Permite editar la relación existente entre un tema y la o las competencias con las cuales se encuentra relacionado. Esto para un curso específico.

VINCULAR CADA TEMA CON LA COMPETENCIA ESPECÍFICA CORRESPONDIENTE

Curso:

Tema: Competencia:

TEMAS Y COMPETENCIAS RELACIONADOS ACTUALMENTE:

Anexo J: Interfaces para capturar las probabilidades

Configurar las probabilidades de las competencias específicas. Permite editar las probabilidades correspondientes a los nodos Competencia. Solamente se debe capturar la probabilidad correspondiente al estado **Si** de la competencia.

CONFIGURAR LAS PROBABILIDADES DE LAS COMPETENCIAS ESPECÍFICAS

Esta configuración tiene como objetivo **Elegir un curso** y **asignar las probabilidades** a cada una de las **Competencias** que componen al curso. Cada competencia tiene dos estados: Si y No, de los cuales sólo se debe capturar el valor para el estado Si, ya que el valor para el estado No, será calculado automáticamente.

Curso: 11

Competencia	Probabilidad (Si)
26 - Dominar los conceptos básicos de la programación	0.00

PROBABILIDADES CAPTURADAS ACTUALMENTE:

Competencia	Prob. (Si)	Prob. (No)	Acción
26	0.60	0.40	Editar , Eliminar

Configurar las probabilidades de los temas. Permite capturar las probabilidades correspondientes a los nodos Tema.

CONFIGURAR LAS PROBABILIDADES DE LOS TEMAS

Esta configuración tiene como objetivo **Elegir un curso** y el **tema** al cual se **asignarán las probabilidades** que lo componen. Cada tema tiene cuatro estados, los cuales representan los rangos en los que se puede ubicar la calificación obtenida por un alumno al contestar el examen correspondiente al tema en cuestión. Los estados y el rango que comprende un Tema son: **Reprobado(0-69)**, **Suficiente(70-80)**, **Bueno(81-90)**, **Excelente(91-100)**. La tabla de probabilidad para cada tema se obtiene de realizar todas las combinaciones posibles entre los cuatro estados del tema con los dos estados de cada Competencia a la cual se encuentre relacionado el tema. Los estados de la Competencia están representados como sigue: el estado Si por un 1, y el estado No por un 0.

Curso: 16

Scorms:

Competencias Asociadas al Tema:
C23 = Aprender a analizar y manejar la computadora

C23	Reprobado	Suficiente	Bueno	Excelente
0	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00

PROBABILIDADES CAPTURADAS ACTUALMENTE:

C23	Reprobado	Suficiente	Bueno	Excelente	Acción
0	0.10	0.20	0.30	0.40	Editar , Eliminar
1	0.05	0.25	0.35	0.35	Editar , Eliminar

Anexo K: Formatos Elvira

Formato del archivo de texto *elv*

```
// Bayesian Network
// Elvira format

bnet "Moodle" {

// Network Properties
kindofgraph = "directed";
visualprecision = "0.00";
version = 1.0;
default node states = (presente , ausente);

// Variables
node C25(finite-states) {
kind-of-node = chance;
type-of-variable = finite-states;
pos_x =459;
pos_y =86;
relevance = 7.0;
purpose = "";
num-states = 2;
states = ("no" "si");
}

node T62(finite-states) {
kind-of-node = chance;
type-of-variable = finite-states;
pos_x =459;
pos_y =86;
relevance = 7.0;
purpose = "";
num-states = 4;
states = ("r0" "r1" "r2" "r3");
}

// Links of the associated graph:
link C25 T62;

//Network Relationships:
relation C25 {
comment = "new";
kind-of-relation = potential;
deterministic=false;
values= table (0.64 0.36);
}

relation T62 C25 {
comment = "new";
kind-of-relation = potential;
deterministic=false;
values= table (0.14 0.00 0.58 0.50 0.14 0.25 0.14 0.25 );
}
}
```

Formato del archivo de texto *evi*

```
// Evidence case
//   Elvira format

evidence NoName {

// Evidence Properties

title = "Untitled";
version = 1.0;

T62 = 1,
}
```

Formato del archivo de texto *txt*

```
node C11
values= table (
           [0] = 0.10638297872340427,
           [1] = 0.8936170212765958,
           ):
node C12
values= table (
           [0] = 0.4,
           [1] = 0.6,
           ):
node T42
values= table (
           [0] = 0.15322144017325393,
           [1] = 0.4376827287493232,
           [2] = 0.2567677314564158,
           [3] = 0.15232809962100705,
           ):
node T43
values= table (
           [0] = 0.344,
           [1] = 0.182,
           [2] = 0.278,
           [3] = 0.196,
           ):
```

Anexo L: Publicaciones

M. L. Morales R., J. A. Ramírez S., A. Hernández R., **J. P. Sánchez S.**, J. A. Martínez F., “Agente Selector de Estrategias de Enseñanza-Aprendizaje para la Educación Basada en Competencias”, 17th International Congress on Computer Science Research (CIICC'11), pp. 53-63, Oct. 2011.

M. L. Morales R., J. A. Ramírez S., A. Hernández R., **J. P. Sánchez S.**, J. A. Martínez F., “Architecture for an Intelligent Tutoring System that considers learning styles”, Fourth Workshop on Intelligent Learning Environments WILE 2011, Nov. 2011.

M. L. Morales R., J. A. Ramírez S., A. Hernández R., **J. P. Sánchez S.**, J. A. Martínez F., “Implementación del estilo de aprendizaje VARK en Moodle”, V Encuentro de Investigadores, Dic. 2011.

REFERENCIAS BIBLIOGRÁFICAS

Andrade, R. (2005). Un acercamiento al enfoque por competencias profesionales. Universidad de Guanajuato / Universidad Marista de Querétaro, México.

Badaracco, M., & Martínez, L. (2010). Arquitectura para un Sistema Tutor Inteligente basado en Competencias (STI-C). Evaluación empírica simulada, CACIC 2010 - XVI Congreso Argentino de Ciencias de la Computación.

Beinlich, I. A., Suermondt, H. J., Chavez, R. M., & Cooper, G. F. (1988). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *In proceedings of the 2nd European Conference on Artificial Intelligence in Medicine* (p. 10).

Ben-Gal, I. (2007). Bayesian Networks. Encyclopedia of Statistics in Quality and Reliability. John Wiley & Sons, Ltd. Retrieved from <http://onlinelibrary.wiley.com/doi/10.1002/9780470061572.eqr089/abstract>

Bickmore, T. (1994). *Real-Time Sensor Data Validation* (No. NASA Contractor Report 195295). National Aeronautics and Space Administration.

Bloom, B. S. (1984). The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-one Tutoring, *Educational Researcher*. Vol. 13, No. 6.

Butz C.J., Hua S., Maguire R.B. (2006). A Web-based Bayesian Intelligent Tutoring System for Computer Programming.

Cárdenas, J. L. (2006). Sistemas de Gestión del Aprendizaje, Sga (Lms). Facultad de Educación, Universidad Autónoma de Yucatán, México.

Castro, S. E. (2002). Estándares en los sistemas de Gestión de Aprendizaje. Dirección de Tecnología Educativa IPN.

Cataldi, Z., & Lage, F. (2009). Sistemas Tutores Inteligentes orientados a la enseñanza para la comprensión. *EDUTECH Revista electrónica de Tecnología Educativa*, No. 28.

Cataldi, Z., & Lage, F. (2010). Modelado del Estudiante en Sistemas Tutores Inteligentes. *TE&ET Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, No. 5 Artículo 4.

Caviedes, P. D., Medina, G. V., & García, P. O. (2009). Diseño de un sistema tutor inteligente basado en estilos cognitivos. Realizado en la Maestría de Ciencias de la Información y las Comunicaciones de la Universidad Distrital Francisco José de Caldas en Bogotá – Colombia.

Cejas, M. (2004). La educación basada en competencias: una metodología que se impone en la Educación Superior y que busca estrechar la brecha existente entre el sector educativo y productivo. Presented at the Comunicación presentada al 3r Congreso Internacional Docencia Universitaria e Innovación, Universidad de Carabobo, Venezuela.

- Conati, C. (2009). *Intelligent Tutoring Systems: New Challenges and Directions*. Department of Computer Science University of British Columbia.
- Díaz, B. A. (1999). *Estrategias docentes para un aprendizaje significativo*. México.
- Dougiamas, M. (2011). Moodle.org: open-source community-based tools for learning. Retrieved August 20, 2011, from <http://moodle.org/>
- Druzdel, M. J., & Van Der Gaag, L. C. (2000). Building probabilistic networks: “Where do the numbers come from?” *IEEE Transactions on knowledge and data engineering*, vol. 12, no. 4, 481–486.
- Ezawa, K., & Schuermann, T. (1995). Fraud/Uncollectible Debt Detection Using a Bayesian Network Based Learning System: A Rare Binary Outcome with Mixed Data Structures. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA.
- Felgaer, P. (2005). *Optimización de Redes Bayesianas Basada en Técnicas de Aprendizaje por Inducción* (Tesis de Grado en Ingeniería Informática). Facultad de Ingeniería, Universidad de Buenos Aires, Argentina.
- Fleming, N. (2011). VARK -- A Guide to Learning Styles. Retrieved August 20, 2011, from <http://www.vark-learn.com/english/index.asp>
- García, P. C. (2011). elearning y scorm. Retrieved May 20, 2011, from <http://elearning-scorm.com/>
- Gomes, A., Miranda, R., Vale, Z., & Faria, L. (2010). A WEB-BASED INTELLIGENT TUTORING SYSTEM ON TEACHING AND LEARNING ELECTRICAL PROJECT. *ICERI2010 Proceedings*, 5288–5294.
- Gómez, N. C. (2004). *Manual de Estilos de Aprendizaje*. Secretaría de Educación Pública, Dirección General del Bachillerato.
- González, S. C. (2004). Sistemas inteligentes en la educación: una revisión de las líneas de investigación y aplicaciones actuales. *RELIEVE Revista electrónica de investigación y evaluación educativa*, vol. 10, No. 001, pp. 3–22.
- Guerra, H. A. (2010). *Programación de Agentes y Sistemas Multiagentes*. Departamento de Inteligencia Artificial, Universidad Veracruzana, México.
- Hernández, Y., Rodríguez, G., & Arroyo, F. G. (2010). Integrating learning styles and affective behavior into an intelligent environment for learning.
- Itmazi, J. (2005). *Sistema flexible de gestión del e-learning para soportar el aprendizaje en las universidades tradicionales y abiertas* (Tesis Doctoral). Departamento de Lenguajes y Sistemas Informáticos, Universidad de Granada, España.
- Kim, J. H., & Pearl, J. (1983). *A Computational Model for Combined Causal and Diagnostic Reasoning in Inference Systems* (Technical Report No. R-25) (pp. pp 190–193). Karlsruhe, West Germany: In *Proceedings of the Eighth International Joint Conference on AI*.

- Larraín, U., & González, M. L. (2005). Formación universitaria por competencias. Universidad Industrial de Santander, Colombia.
- López, P. J., & García, G. J. (2008). Sistemas de Tutorización Inteligente Basados en Redes Bayesianas. *Revista Electrónica de Metodología Aplicada, Almería, España, vol. 13, no. 1*, pp 13–25.
- Melis, E., & Siekmann, J. (2004). ActiveMath: An Intelligent Tutoring System for Mathematics. German Research Institute for Artificial Intelligence.
- Millán, V. D. (2000). *Sistema bayesiano para modelado del alumno* (Tesis doctoral). Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, España.
- Pearl, J. (1982). Reverend Bayes on Inference Engines: a Distributed Hierarchical Approach. *Proceedings, AAAI National Conference on AI* (pp. pp 133–136). Pittsburgh, PA.
- Pearl, J. (1999). *Bayesian Networks* (Technical Report No. R-246) (p. 4). In MIT Encyclopedia of the Cognitive Sciences, Cambridge, MA: UCLA Cognitive Systems Laboratory.
- Pearl, J., & Russell, S. (2000). *Bayesian networks* (No. R-277). Computer Science Department: University of California, Berkeley, CA.
- Peter, S. E., Bacon, E., & Dastbaz, M. (2009, April 6). Learning styles, personalisation and adaptable e-learning. Retrieved June 29, 2012, from <http://gala.gre.ac.uk/1888/>
- Ramírez, L., & Medina, M. G. (2011). Educación Basada en Competencias y el Proyecto Tuning en Europa y Latinoamérica. Observatorio Ciudadano de la Educación, Portal de Recursos Educativos Abiertos. Retrieved from <http://www.temoa.info/es/node/49021>
- Rodríguez, D., & Dolado, J. (2007). Redes Bayesianas en la Ingeniería del Software.
- Russell, S. J., & Norvig, P. (2004). *Inteligencia Artificial Un enfoque moderno* (2da Edición.). Prentice Hall.
- Salgueiro, F. (2005). *Sistemas Inteligentes para el Modelado del Tutor* (Tesis de grado en Ingeniería Informática). Facultad de Ingeniería, Universidad de Buenos Aires, Argentina.
- Stephenson, T. A. (2000). An introduction to Bayesian network theory and usage. IDIAP Research Report 00- 03.
- Vernet, D., Nicolas, R., Golobardes, E., Fornells, A., & Garcia-Piquer, A. (2010). Work in progress-Intelligent tutoring system framework for the acquisition of knowledge and competences. *Frontiers in Education Conference (FIE), 2010 IEEE* (pp. T4G–1 –T4G–2). doi:10.1109/FIE.2010.5673610