



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

MODALIDAD PRESENCIAL

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

**AGENTE DE MODELADO DE USUARIO PARA EL ENTORNO
VIRTUAL DE APRENDIZAJE DE LA UTPL**

Proyecto de fin de carrera previo a la obtención del título
de ingeniero en sistemas informáticos y computación

Autor:

Juan Carlos Sarango Briceño

Directora:

Ing. Priscila Maricela Valdiviezo Diaz

Loja - Ecuador

2010

Ing. Priscila Valdiviezo

DIRECTORA DE TESIS

Certifica:

Que el Sr. Juan Carlos Sarango Briceño, autor de la tesis AGENTE DE MODELADO DE USUARIO PARA EL ENTORNO VIRTUAL DE APRENDIZAJE DE LA UTPL, ha cumplido con los requisitos estipulados en el Reglamento General de la Universidad Técnica Particular de Loja, la misma que ha sido coordinada y revisada durante todo el proceso de desarrollo, desde su inicio hasta la culminación, por lo cual autorizo su presentación.

Loja, septiembre del 2010

Ing. Priscila Valdiviezo

DIRECTORA DE TESIS

Ing. Janneth Chicaiza

COORDIRECTORA DE TESIS

Cesión de Derechos

Yo, **Juan Carlos Sarango Briceño**, declaro ser autor del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja, que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través o con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

Juan Carlos Sarango Briceño

Autoría

Las ideas, opiniones, conclusiones, recomendaciones y más contenidos expuestos en el presente informe de tesis son de absoluta responsabilidad del autor.

Juan Carlos Sarango Briceño

Dedicatoria

Con mucho cariño quiero dedicar la presente tesis:

A mi Dios, ser de infinito poder que me infunde cada día la fuerza suficiente para vencer todos los obstáculos que me presenta la vida.

A mis queridos padres, Anibal y Cecilia, que siempre me han sabido dar su apoyo y consejo, en los diferentes anhelos y metas que me he propuesto en todo este tiempo.

A mis hermanas, Johanna y María, que han sido tolerantes para comprender al hermano tan radical que les preparo el destino.

A Eylin, que con su amor, me ha ayudado a orientar todos los aspectos de mi vida y ha sabido compartir su cariño y dulzura durante el tiempo que llevamos juntos.

A mi sobrina, Arianita, desde que naciste has llenado de alegría mi hogar y mi corazón.

A mis amigos, Juan Carlos, Xavier, Julio, Ernesto, Ivan, Saida, Jefferson, Andrés, Johanna, Alexandra, Stalin, Nancy, Paul, y un gran grupo de personas que no nombro, que han sabido ser más que la gente con la que he compartido la universidad, los considero hermanos que siempre me apoyan y en quienes puedo confiar y contar incondicionalmente.

A mis profesores que siempre trataron de sacar de mi, todo el potencial que como ingeniero puedo brindar a la sociedad,

gracias por todo...

Juan Carlos

Agradecimiento

Durante el transcurso de mi vida universitaria y el desarrollo de esta tesis, muchas personas, me han sabido brindar su apoyo y confianza y mediante esta líneas expreso mi mas sincero agradecimiento.

En el desarrollo de esta tesis a sido crucial el apoyo de mi directora Priscila, que tuvo la confianza de poner un tema extenso y difícil el mis manos, y mediante su objetividad y constancia ha sabido obtener de mi todo el esfuerzo necesario para sacarlo adelante.

Agradezco también al grupo de compañeros, Juan Pablo, Angel, Jose Miguel y Byron, con los cuales compartimos durante cuatro meses el desarrollo e integración de nuestros proyectos de tesis, y a la vez, extendo el agradecimiento a sus directores, los ingenieros, Fausto, Greyson y Martha los cuales nos brindaron consejos y sugerencias que nos ayudaron a cristalizar nuestros proyectos.

Quiero agradecer de manera especial, a la Ingeniera Inés Jara, que desde que tuve el gusto de conocerla, me ha sabido brindar palabras de aliento y apoyo que me han impulsado a seguir adelante en el trascurso de mi vida universitaria.

En la escuela de ciencias de la computación agradezco, a todos los profesores y de manera especial a los ingenieros, Armando Cabrera, Patricio Abad, Fausto Loja, Janneth Chicaiza, Jorge Lopez, Nelson Piedra y Hector Gomez, de los cuales aparte de recibir sus enseñanzas, han sido mi modelo a seguir en los diferentes campos en que incursiona nuestra carrera.

Finalmente quiero agradecer a las personas del departamento de Gestión de Servicios de la UGTI de nuestra universidad, Silvana, Johanna, Maritza, Janneth,

Enmita, Anita, Bibiana, Frank, Xavier, Juan Pablo y Franco, con los cuales estoy empezando a compartir mis primeras experiencias laborales, gracias por recibirme y dejarme ser parte de su familia.

Índice de Contenidos

Certificación.....	i
Cesión de Derechos.....	ii
Autoría.....	iii
Dedicatoria.....	iv
Agradecimiento.....	v
Índice de Contenidos.....	vii
Índice de Tablas.....	xiv
Índice de Figuras.....	xvi
Parte I: ESTADO DEL ARTE DEL MODELADO DE USUARIO y AGENTES.....	1
Capítulo 1 Modelo de usuario.....	2
1.1 Introducción.....	3
1.2 Modelo de usuario.....	3
1.3 Tipos de modelos de usuario.....	5
1.4 Representación del modelo.....	6
1.5 Proceso para la construcción de un modelo de usuario.....	7
1.5.1 Modelado de usuario.....	9
1.5.2 Pasos del modelado de usuario.....	9
1.5.3 Elementos mínimos de un modelo de usuario.....	10
1.6 Requisitos del modelo de usuario.....	11
1.7 Datos para el modelo de usuario.....	12
1.7.1 Datos de usuario.....	12
1.7.2 Datos de uso del sistema.....	13
1.7.3 Datos de entorno.....	17
1.7.4 Métodos de adquisición de datos.....	19

1.8 Estilos de aprendizaje y modelado de usuario.....	21
1.8.1 Estilos de aprendizaje.....	22
1.8.2 Clasificación de los estilos de aprendizaje.....	23
1.9 Modelado de usuario con aprendizaje automático.....	26
1.9.1 Tareas de aprendizaje.....	27
1.9.2 Mejora del rendimiento de tareas de aprendizaje.....	27
1.9.3 Algoritmos de aprendizaje automático.....	30
Capítulo 2 Agentes inteligentes.....	35
2.1 Introducción.....	36
2.2 Agentes inteligentes.....	36
2.2.1 Características de los agentes inteligentes.....	37
2.2.2 Clasificación de los agentes inteligentes.....	38
2.3 Estructura de los agentes inteligentes.....	40
2.4 Agente inteligente de modelado de usuario.....	42
2.5 Metodologías de desarrollo de agentes.....	43
2.5.1 Vowell Engineering.....	44
2.5.2 Mas-CommonKADS.....	45
2.5.3 BDI.....	47
2.5.4 MaSE.....	48
2.5.5 ZEUS.....	49
2.5.6 GAIA.....	50
2.5.7 INGENIAS.....	51
Parte II: DISEÑO Y COSTRUCCIÓN DEL AGENTE DE MODELADO DE USUARIO.....	54
Capítulo 3 Descripción del proceso y Selección de herramientas y técnicas	55
3.1 Introducción.....	56
3.2 Descripción del proceso de construcción del modelo de usuario.....	56

3.2.1	<i>Actividades de construcción del modelo de usuario</i>	56
3.3	Selección de herramientas y técnicas.....	59
3.3.1	<i>Selección del tipo de modelo de usuario</i>	60
3.3.2	<i>Selección de estilos de aprendizaje</i>	60
3.3.3	<i>Selección de datos para el modelo de usuario</i>	65
3.3.4	<i>Selección de la metodología de desarrollo del agente</i>	69
3.3.5	<i>Selección de algoritmos de aprendizaje automático</i>	71
	Capítulo 4 Minería de datos para el modelo de usuario	73
4.1	Introducción.....	74
4.2	Características de la población estudiantil de la UTPL Modalidad a Distancia.....	74
4.2.1	<i>Dispersión geográfica</i>	75
4.2.2	<i>Edad y estado</i>	76
4.2.3	<i>Motivación</i>	76
4.2.4	<i>Intereses profesionales</i>	76
4.2.5	<i>Falta de tiempo</i>	77
4.2.6	<i>Aislamiento y deserción</i>	77
4.3	Extracción de conocimiento del entorno virtual de aprendizaje UTPL.....	79
4.3.1	<i>Proceso de extracción de conocimiento a seguir</i>	79
4.3.2	<i>Integración y recopilación</i>	82
4.3.3	<i>Selección, limpieza y transformación</i>	85
4.3.4	<i>Minería de datos</i>	88
4.3.5	<i>Evaluación e interpretación</i>	92
	Capítulo 5 Diseño del modelo de usuario para el EVA UTPL	100
5.1	Modelo de usuario de los estudiantes de la UTPL.....	101
5.1.1	<i>Descripción global del modelo</i>	101
5.1.2	<i>Descripción de los elementos del modelo</i>	102
5.1.3	<i>Mecanismos de identificación de atributos del modelo de usuario</i>	104
5.2	Diseño de tareas de aprendizaje para el modelo de usuario.....	106

5.2.1 <i>Definición de tareas de aprendizaje</i>	106
5.3 Modelo de datos para el modelado de usuario.....	107
5.4 Descripción de tablas del modelo de usuario.....	109
5.4.1 <i>Entidades del modelo de usuario</i>	109
5.4.2 <i>Entidades de apoyo al modelo de usuario</i>	114
5.4.3 <i>Entidades de apoyo al sistema</i>	117
5.5 Inicialización del modelo de usuario.....	118
Capítulo 6 Construcción del agente de modelado de usuario para el EVA UTPL.....	121
6.1 Introducción.....	122
6.1.1 <i>Modelado de sistemas multiagente con INGENIAS</i>	122
6.2 Conceptualización.....	124
6.3 Análisis – Inicio.....	125
6.4 Análisis – Elaboración.....	126
6.4.1 <i>Casos de uso ampliados</i>	129
6.4.2 <i>Modelos de interacción</i>	130
6.4.3 <i>Descomposición de objetivos</i>	132
6.4.4 <i>Objetivos y tareas asociadas</i>	133
6.4.5 <i>Tareas y entidades asociadas</i>	134
6.4.6 <i>Modelos de agente</i>	138
6.4.7 <i>Nuevos elementos identificados en el entorno del sistema</i>	140
6.4.8 <i>Organización refinada del sistema</i>	141
6.5 Diseño – Elaboración.....	143
6.5.1 <i>Detalle de flujos de trabajo del sistema</i>	146
6.5.2 <i>Refinamiento de interacciones</i>	147
6.5.3 <i>Protocolos para el desarrollo de las interacciones</i>	148
6.5.4 <i>Responsabilidades en las tareas por parte de los roles</i>	149
6.5.5 <i>Dependencias entre objetivos</i>	150

6.5.6	<i>Percepción de los agentes</i>	151
6.5.7	<i>Arquitectura del Sistema</i>	151
6.6	Implementación	152
6.6.1	<i>Aplicaciones y enlaces de código fuente</i>	152
6.6.2	<i>Tareas y enlaces de código fuente</i>	153
6.6.3	<i>Diseño e implementación del módulo de acceso a datos</i>	154
6.6.4	<i>Diseño e implementación del módulo de obtención de estilos de aprendizaje del entorno virtual de aprendizaje de la UTP</i>	156
6.6.5	<i>Diseño e implementación del módulo de obtención de atributos por medio de reglas de inferencia</i>	156
6.6.6	<i>Diseño e implementación del módulo de obtención de atributos por medio de aprendizaje automático (WEKA-Util)</i>	157
6.6.7	<i>Diseño e implementación del módulo de obtención de cambios en la interacción</i>	159
6.6.8	<i>Diseño e implementación del módulo de actualización de estadísticas del entorno virtual de aprendizaje de la UTP</i>	161
6.7	Pruebas	162
6.7.1	<i>Plan de pruebas desarrolladas al agente de modelado de usuario</i>	162
6.7.2	<i>Ejecución de las pruebas al agente de modelado de usuario</i>	163
6.7.3	<i>Evaluación de las pruebas desarrolladas al agente de modelado de usuario</i>	165
	Discusión	169
	<i>Resultados de la selección de datos para el modelo de usuario</i>	170
	<i>Resultados de la ejecución de la minería de datos al entorno virtual de aprendizaje</i>	171
	<i>Resultados del diseño del modelo de usuario</i>	173
	<i>Resultados de la Implementación del esquema de datos del modelo de usuario</i>	174
	<i>Resultados de la definición de estereotipos</i>	175
	Conclusiones y Recomendaciones	177
	Conclusiones.....	178
	Recomendaciones.....	180
	ANEXOS	182
	Anexo A Test de estilos de aprendizaje	183

A.1 Tests de estilos de aprendizaje.....	184
A.2 Test de Felder y Silverman.....	184
Anexo B Elementos de navegación y tracking obtenibles y no obtenibles del EVA UTPL	188
B.1 Trazas de navegación y elementos de tracking obtenibles del EVA UTPL.....	189
<i>B.1.1 Número de accesos al curso.....</i>	<i>190</i>
<i>B.1.2 Número de aportes al curso.....</i>	<i>190</i>
<i>B.1.3 Número de preguntas al profesor.....</i>	<i>191</i>
<i>B.1.4 Número de paginas por curso visitadas.....</i>	<i>191</i>
<i>B.1.5 Número de respuestas en los foros.....</i>	<i>192</i>
<i>B.1.6 Tiempo promedio de interacción.....</i>	<i>192</i>
<i>B.1.7 Número de tareas realizadas.....</i>	<i>193</i>
<i>B.1.8 Número de tareas propuestas.....</i>	<i>193</i>
B.2 Información de tracking que no se puede obtener del EVA UTPL.....	194
<i>B.2.1 Patrones de navegación.....</i>	<i>194</i>
<i>B.2.2 Preferencias de selección.....</i>	<i>194</i>
<i>B.2.3 Monitorear el scrolling y el enlarging.....</i>	<i>195</i>
<i>B.2.4 Monitorear selección de elementos de ayuda.....</i>	<i>195</i>
<i>B.2.5 Etiquetado y rating de objetos de aprendizaje.....</i>	<i>195</i>
<i>B.2.6 Monitorear acciones con objetos de aprendizaje.....</i>	<i>196</i>
Anexo C Base de datos de modelos de usuario.....	197
C.1 Inicialización de la base de datos de modelos de usuario.....	198
Anexo D Documentación de las fases de desarrollo.....	201
D.1 Resultados de la fase de Análisis-Inicio.....	202
<i>D.1.1 Casos de uso asociados al problema.....</i>	<i>202</i>
<i>D.1.2 Modelo de entorno.....</i>	<i>203</i>
<i>D.1.3 Modelo de organización sistema.....</i>	<i>204</i>
Anexo E Archivos del modulo de obtención de los estilos de aprendizaje	

del EVA.....	206
E.1 Archivos del modulo.....	207
E.2 Modificación a la base de datos.....	209
Referencias Bibliográficas.....	212

Índice de Tablas

Tabla 3.1: Comparativa de técnicas de identificación de estilos de aprendizaje.....	62
Tabla 3.2: Resultados obtenidos en el test de estilos de aprendizaje de Felder – Silverman.....	64
Tabla 3.3: Resultados obtenidos en el test de estilos de aprendizaje de Honey - Alonso.....	65
Tabla 3.4: Atributos de entrada que definen el entorno del usuario del EVA UTPL.....	66
Tabla 3.5: Atributos de entrada que definen los datos de usuario del EVA UTPL.....	67
Tabla 3.6: Atributos de entrada que definen los datos de interacción del usuario del EVA UTPL.....	68
Tabla 3.7: Comparativa de metodologías de desarrollo de SMA.....	69
Tabla 4.1: Síntesis comparativa de las características de los estudiantes de la UTPL (Rubio, 2002).....	78
Tabla 4.2: Carreras de la modalidad a distancia con mayor número de interacciones.....	81
Tabla 4.3: Tablas para la integración y recopilación de datos.....	83
Tabla 4.4: Tablas para la limpieza y transformación de datos.....	86
Tabla 4.5: Discretación de datos para el atributo nivel de interés en el curso.....	87
Tabla 4.6: Algoritmos de prueba y de generación de modelos.....	90
Tabla 4.7: Resumen de resultados para cada algoritmo ejecutado.....	91
Tabla 4.8: Comparativa de la precisión de algoritmos aplicados a atributos del modelo de usuario.....	93
Tabla 4.9: Selección de algoritmos aprendizaje automático para atributos del modelos de usuario.....	99
Tabla 5.1: Campos de la entidad mdl_course.....	109
Tabla 5.2: Campos de la entidad mdl_user.....	110
Tabla 5.3: Campos de la entidad mu_atributos_modelo_usuario.....	111
Tabla 5.4: Campos de la entidad mu_valores_atributos.....	111
Tabla 5.5: Campos de la entidad mu_modelos_usuario.....	112
Tabla 5.6: Campos de la entidad mu_tipos_modelos_usuario.....	112
Tabla 5.7: Campos de la entidad mu_datos_entorno_usuarios.....	113
Tabla 5.8: Campos de la entidad mu_acciones_sobre_objetos.....	113
Tabla 5.9: Campos de la entidad mu_asignaciones_estereotipos.....	114
Tabla 5.10: Campos de la entidad mu_valores_estereotipos.....	114
Tabla 5.11: Campos de la entidad mu_estilos_aprendizaje_resultados.....	115
Tabla 5.12: Campos de la entidad mu_datos_interaccion_curso.....	115

Tabla 5.13: Campos de la entidad mu_datos_interaccion.....	116
Tabla 5.14: Campos de la entidad mu_acciones.....	117
Tabla 5.15: Campos de la entidad mu_tipos_objeto.....	117
Tabla 5.16: Campos de la entidad mu_estados.....	118
Tabla 5.17: Estereotipos definidos para cada estilo de aprendizaje.....	119
Tabla 5.18: Valores de atributos según los estereotipos.....	119
Tabla 6.1: Actividades para la construcción del modelo del entorno del agente.....	126
Tabla 6.2: Actividades para la construcción del modelo de organización del sistema.....	126
Tabla 6.3: Actividades para la construcción de modelos de interacción del sistema.....	127
Tabla 6.4: Actividades para la construcción del modelo de agente.....	128
Tabla 6.5: Actividades para la construcción de modelos de tareas y objetivos.....	128
Tabla 6.6: Actividades para la optimización del modelo del entorno.....	129
Tabla 6.7: Actividades para la optimización del modelo de organización del Sistema	129
Tabla 6.8: Actividades para la refinación de flujos en tareas.....	143
Tabla 6.9: Actividades para el desarrollo del detalle de los flujos del SMA.....	144
Tabla 6.10: Actividades para la especificación del control de los agentes.....	144
Tabla 6.11: Actividades para el refinamiento del entorno y percepción la de Agentes.....	145
Tabla 6.12: Actividades para el refinamiento de la satisfacción de objetivos y tareas.....	145

Índice de Figuras

Figura 1.1: Proceso de trabajo de una representación híbrida de modelos de usuario. Adaptación de la imagen realizada por Elena Gaudioso (2002).....	7
Figura 1.2: Elementos mínimos de un modelado de usuario expresados en forma de ontología.....	10
Figura 2.1: Taxonomía de agentes de Franklin y Gresser (1996).....	39
Figura 2.2: Clasificación de agentes de software de Nwana (1996).....	40
Figura 2.3: Modelo general para agentes que aprenden (Russell y Norving 2004).....	41
Figura 2.4: Como trabajan los agentes de interfaz. Imagen adaptada de Maes (1994) citada por Nwana (1996).....	43
Figura 3.1: Actividades para la diseño e implementación del agente de modelado de usuario.....	59
Figura 4.1: Distribución de la población estudiantil de la UTPL.....	75
Figura 4.2: Fases del proceso de descubrimiento de conocimiento en bases de datos, KDD. (Hernández, Ramírez y Ferri, 2004)	80
Figura 4.3: Introducción de conocimiento previo para el atributo nivel de interés en el curso.....	87
Figura 4.4: Consulta para la vista minable del atributo nivel de interés en el curso.....	88
Figura 4.5: Matrices de confusión para el atributo nivel de interés en el curso.....	96
Figura 4.6: Matrices de confusión para el atributo nivel de interés en recursos.....	96
Figura 4.7: Matrices de confusión para el atributo nivel de interacción.....	97
Figura 5.1: Modelo de datos para la representación de los modelos de usuario.....	108
Figura 5.2: Ejemplo de modelo de usuario generado en las entidades de la base de datos	120
Figura 6.1: Puntos de vista de INGENIAS para el modelado de SMA.....	123
Figura 6.2: Ampliación al diagrama de casos de uso.....	130
Figura 6.3: Modelo de interacción y especificación UML para la interacción activar agente.....	131
Figura 6.4: Modelo de interacción y especificación UML para la interacción comprobar interacción.	131
Figura 6.5: Descomposición de objetivos de sistema.....	132
Figura 6.6: Tareas asociadas al cumplimiento de los objetivos identificados.....	133
Figura 6.7: Entidades consumidas y producidas por la tarea activar agente MU.....	135
Figura 6.8: Entidades consumidas\producidas por la tarea solicitar estilos aprendizaje.....	135
Figura 6.9: Entidades consumidas\ producidas por la tarea prototipar modelo usuario.....	136

Figura 6.10: Entidades consumidas\producidas por la tarea detectar cambios interacción usuario.....	137
Figura 6.11: Entidades consumidas y producidas por la tarea modificar atributos modelo usuario.....	137
Figura 6.12: Modelo del agente de modelado de usuario.....	139
Figura 6.13: Modelo del agente monitor del EVA.....	140
Figura 6.14: Refinación de modelo del entorno con los nuevo elementos identificados.....	141
Figura 6.15: Organización del sistema multiagente de modelado de usuario del EVA UTPL.....	142
Figura 6.16: Detalle de la tarea procesar modelado.....	146
Figura 6.17: Detalle del flujo de trabajo monitorear interacción.....	147
Figura 6.18: Refinamiento de la interacción pasar datos al agente de modelado de usuario.....	147
Figura 6.19: Refinamiento de la interacción comprobar interacción.....	148
Figura 6.20: Protocolo utilizado por la interacción activar agente.....	148
Figura 6.21: Protocolo utilizado por la interacción comprobar interacción.....	149
Figura 6.22: Responsabilidades de la ejecución de tareas para el agente de modelado de usuario.....	149
Figura 6.23: Responsabilidades de la ejecución de tareas para el flujo monitorear interacción.....	150
Figura 6.24: Dependencias entre objetivos.....	150
Figura 6.25: Percepción de los agentes.....	151
Figura 6.26: Arquitectura del agente de modelado de usuario y su interacción con el EVA UTPL.....	152
Figura 6.27: Aplicaciones y el código fuente asociado a su implementación.....	153
Figura 6.28: Tareas y componentes de código asociados a su implementación.....	154
Figura 6.29: Diagrama de clases de módulo de acceso a datos.....	155
Figura 6.30: Diagrama de clases del módulo de obtención de atributos por medio de reglas de inferencia.....	157
Figura 6.31: Diagrama de clases del módulo de obtención de atributos por medio de aprendizaje automático (Weka-Util).....	158
Figura 6.32: Interfaz del módulo de obtención de atributos por medio de aprendizaje automático (WEKA-Util).....	159
Figura 6.33: Diagrama de clases del módulo de obtención de cambios en la interacción.....	160
Figura 6.34: Diagrama de clases del módulo de actualización de estadísticas del EVA.....	161
Figura A.1: Test de estilos de aprendizaje implementado en la plataforma.....	187
Figura B.1: Campos que componen la entidad mdl_log.....	189

Figura B.2: Consulta para la obtención del número de accesos al curso de un estudiante.....	190
Figura B.3: Consulta para la obtención del número de aportes al curso de un estudiante.....	191
Figura B.4: Consulta para la obtención del número de preguntas al profesor en un curso de un estudiante.....	191
Figura B.5: Consulta para la obtención del número de paginas por curso visitadas por el estudiante..	192
Figura B.6: Consulta para la obtención del número de respuestas en los foros de un estudiante en un curso.....	192
Figura B.7: Consulta para la obtención del número de tareas realizadas por un estudiante en un curso	193
Figura B.8: Consulta para la obtención del número de tareas propuestas por un tutor en un curso.....	193
Figura C.1: Datos iniciales de la entidad valores de estereotipos.....	198
Figura C.2: Datos iniciales de la entidad asignaciones de estereotipos.....	199
Figura C.3: Datos iniciales de la entidad tipos de modelos de usuario.....	199
Figura C.4: Datos iniciales de la entidad acciones.....	199
Figura C.5: Datos iniciales de la entidad tipos de objeto.....	200
Figura C.6: Datos iniciales de la entidad atributos del modelo de usuario.....	200
Figura D.1: Diagrama de casos de uso iniciales para el agente de modelado de usuario.....	202
Figura D.2: Modelo de entorno para el agente de modelado de usuario.....	204
Figura D.3: Modelo de organización del sistema de agentes.....	204
Figura E.1: Modelo de datos de modulo de obtención de estilos de aprendizaje.....	210

Parte I:
ESTADO DEL
ARTE DEL
MODELADO
DE USUARIO y
AGENTES

Capítulo 1

Modelo de usuario

1.1 Introducción

En primera parte de esta tesis se desarrolla un estado del arte del modelado de usuario y la construcción de agentes de software, describiendo el entorno de trabajo mediante el cual se va a desarrollar el presente proyecto de tesis.

En el capítulo presente se realiza un estudio que sustenta el modelo de usuario y el proceso de modelado de usuario, así como los elementos principales del modelo, las características a considerar, los tipos de modelos existentes, la representación que se debe dar al modelo de usuario, un proceso para la construcción de modelos de usuario y la descripción de un conjunto de datos los cuales se tomarán como parte del modelo.

Además se incorpora el estudio de los estilos de aprendizaje como parte de los modelos de usuario, planteando las clasificaciones que han tenido mayor impacto en entornos educativos.

También se discuten las propuestas existentes en la inclusión de aprendizaje automático en la obtención de atributos para el modelo de usuario, incluyendo el concepto de tareas de aprendizaje como base para la incorporación de este tipo de atributos en el modelo, así como procesos para mejorar los resultados obtenidos en su aplicación.

Mediante este estudio se busca desarrollar la selección de los mejores elementos, características, y técnicas de construcción del modelo de usuario de los estudiantes de la UTPL.

1.2 Modelo de usuario

El *Modelo de Usuario* es un elemento importante y relevante en los sistemas que requieren adaptar el aprendizaje a las necesidades y preferencias del usuario. Este contiene información del usuario como: datos personales, preferencias e intereses, y datos del entorno.

Se define al modelo de usuario como:

“Una representación explícita de las propiedades de un usuario específico. Se

utiliza para razonar acerca de las necesidades, preferencias o comportamiento futuro del usuario” (Gaudioso, 2002).

Esta definición engloba los elementos principales del modelo de usuario así como las causas de su origen, ya que la inflexibilidad de los sistemas tradicionales sumada con los altos requerimientos que exige la construcción de sistemas de apoyo a la enseñanza a distancia condicionan el proceso para encontrar nuevas alternativas de trabajo con los usuarios.

Los modelos de usuario pueden tomar diversas representaciones dependiendo de los objetivos para los están pensados. Así, de acuerdo a Webb, Pazzani, y Billsus, (2001) podemos encontrar modelos cuyo objetivo es describir:

- ◆ Las características generales de los usuarios (casi todos los sistemas guardan información acerca de esto).
- ◆ Los objetivos del usuario.
- ◆ Los patrones de comportamiento y preferencias de los usuarios. Esta información es la base de cualquier sistema de recomendación.
- ◆ Las destrezas de los usuarios, etc.

Los sistemas adaptativos, dependiendo del tipo de información que albergan, pueden ser utilizados en diferentes entornos de aplicación. Se los ha utilizado con gran énfasis en las áreas de:

- ◆ Comercio Electrónico
- ◆ Sistemas Educativos
- ◆ Sistemas Turísticos, etc.

Existe una amplia gama de aplicaciones posibles para el modelado de usuarios, pero en ámbitos comerciales todavía no son muy utilizados debido a la dificultad de integración con los sistemas existentes. Se ve su aplicación en sistemas nuevos pensados y construidos para que funcionen mediante modelos de usuario.

1.3 Tipos de modelos de usuario

Elena Gaudioso (2002), presenta una clasificación de los diferentes tipos de modelos de usuario y afirma que los criterios para clasificar los modelos de usuario varían de acuerdo a lo siguiente:

- El tipo de usuarios que se modelan:
 - a) Usuarios estándares (estereotipos) o,
 - b) Modelos más individualizados.
- Fuentes de información para realizar el modelado:
 - a) Modelos construidos explícitamente por el usuario (estos modelos se construyen a partir de los datos que proporciona el usuario mediante formularios de entrada) o,
 - b) Abstraídos por el sistema basándose en el comportamiento del usuario (datos de interacción fundamentalmente).
- La caducidad de la información que se guarda en el modelo:
 - a) Modelos a corto plazo con información muy específica y,
 - b) Modelos a largo plazo con información más general.
- Actualización del modelo:
 - a) Modelos estáticos,
 - b) Modelos dinámicos y,
 - c) Modelos híbridos.

Cada clasificación incorpora diferencias a la hora de tratar asuntos como la actualización de la información, que dependerá de la estructura del mismo, ya que por ejemplo, los modelos a corto plazo, o de comportamiento de usuario, o individualizados, necesitan una actualización dinámica de los mismos. Algunos de los tipos de modelos, al mantener información que se trabaja a corto plazo, se los conoce como modelos de tareas, los cuales varían de acuerdo a la tarea que el usuario se encuentre realizando en esos momentos.

El modelo más simple a considerar es el conocido como *modelo estático* con un

usuario estándar. Este es fácilmente acoplable a sistemas porque, de hecho, es una representación explícita de modelos implícitos. Para la actualización de usuarios individualmente la forma en que se debe realizar este proceso es dinámicamente, además de la incorporación de métodos explícitos que dictaminen como debe actuar el sistema dependiendo del estado en que se encuentra el modelo.

1.4 Representación del modelo

Existen tres formas de representar un modelo de usuario, mencionados anteriormente: mediante los mecanismos explícitos, implícitos y las propuestas híbridas. La aplicación de cada una de ellas dependerá del dominio de aplicación de cada caso en particular, y del alcance que se quiera obtener del sistema que se está diseñando.

Representación Explícita.- Implementa conocimiento aprendido anteriormente para representar así modelos de usuario explícitos. Se basan en reglas de adquisición de conocimiento incorporadas inicialmente.

Representación Implícita.- Utiliza principalmente aprendizaje automático para la generación del modelo implícito de usuario, para lo cual se basa en la recolección de los datos en cada una de las interacciones del usuario con el sistema, tratando de solventar los problemas que presentan las representaciones explícitas.

Como ya se dijo anteriormente utilizan un algoritmo de aprendizaje, que es el mecanismo por medio del cual esta representación procesa los datos de entrada. Y según Gaudioso (2002) para ello se sigue el siguiente proceso:

- ◆ Para una tarea de aprendizaje se procesa un conjunto de datos de entrenamiento y se guarda el resultado en un formato específico.
- ◆ Luego, se utilizan los valores aprendidos en la adaptación de la tarea relacionada con los datos de entrenamiento.

Representación Híbrida.- Combina las dos técnicas anteriormente descritas y de esta manera supera las carencias que supone el uso de cada técnica de manera individual. La desventaja principal es la dificultad en su implementación ya que se

necesita encontrar un nivel de armonía entre la representación implícita y la representación explícita.

Esta representación híbrida necesita una clasificación de los datos que se van a utilizar dentro del modelo de usuario ya que los datos que se incluyen en la representación Explícita no son los mismos que se incluyen en la representación implícita. Además las acciones para el procesamiento de cada tipología de datos varía en cada caso. En la figura 1.1 se muestra este proceso para cada caso.

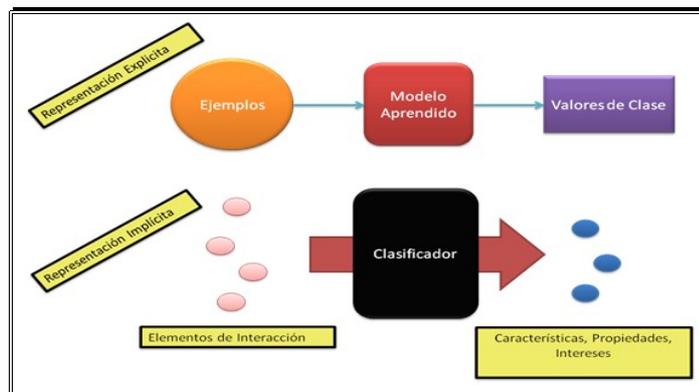


Figura 1.1: Proceso de trabajo de una representación híbrida de modelos de usuario. Adaptación de la imagen realizada por Elena Gaudioso (2002).

En la figura 1.1 se representa el proceso en que las dos partes de la representación explícita e implícita trabajan para lograr el modelo híbrido que se pretende construir, en la combinación propuesta, la representación explícita es aquella que maneja un modelo aprendido, que se obtiene mediante el uso de reglas de inferencia y trata datos que no necesitan de un procesamiento sofisticado para la deducción de sus valores. Mientras que la representación implícita del modelo ya necesita que la forma en que se procesa los elementos de interacción con el sistema se realice mediante clasificadores que interpreten estos elementos y deduzcan de ellos las propiedades, características e Intereses de la población estudiantil.

1.5 Proceso para la construcción de un modelo de usuario

Una vez que se conocen las definiciones sobre modelos de usuario y su clasificación, se propone en esta sección un proceso en el cual nos permita construir un modelo de usuario que cumpla con las expectativas propuestas en los estudios

anteriores.

Se ha avanzado mucho en la construcción de modelos eficientes a lo largo de todos los intentos de desarrollo que se han concretado por diferentes entidades comerciales, universitarias y privadas, pero es difícil sostener que hasta la fecha estos modelos hayan satisfecho todas las necesidades planteadas, siempre hay limitantes que están sujetas al ámbito de aplicación del modelo y sobre todo por su caracterización de considerarlos explícitos. Pero la mayoría de autores coinciden en dos técnicas destacadas para la elaboración de los mismos:

- ◆ Predefinir el modelo de usuario en base al conocimiento de expertos. Y representándolo en el modelo mediante reglas que guían la inferencia para la adaptación. Para este modelo por lo general se utilizan los llamados estereotipos que según Gaudioso (2002) se definen como “una colección de atributos que pueden ser comunes en las personas, y su importancia radica en que permita al sistema hacer un mayor número de inferencias o recomendaciones basándose en un pequeño número de observaciones”. Los estereotipos se definen en función de:
 - d) El nivel de pericia en el uso del sistema.
 - e) El nivel de conocimiento de partida.
 - f) El tipo de aprendizaje preferido por el alumno.
 - g) Tipo de interacción.
- ◆ Diseñar el sistema de manera que sea éste el que aprenda autónomamente obteniendo información de las interacciones de los usuarios y tratándolas mediante el aprendizaje automático.

Estas dos técnicas de construcción de modelos de usuario pueden ser combinadas de cierta forma para que cuando se predefina el estereotipo de un estudiante este pueda ser modificado en función del comportamiento del usuario en el sistema mediante alguna técnica, como por ejemplo el aprendizaje automático Gaudioso (2002).

1.5.1 Modelado de usuario

El proceso de modelado de usuario se define como: “El modelado del usuario, consiste en la definición de clases o perfiles de usuarios en base a atributos comunes. Los atributos sobre los que se hará la clasificación dependen de la información que se tenga de la audiencia, pero normalmente se tratarán de atributos tales como necesidades de información, condiciones de acceso, experiencia y conocimientos.” (Hassan y otros, 2004).

Esta definición describe la categorización que se debe realizar de los perfiles de usuarios, en la cual se trataran atributos comunes, estos atributos en el caso particular de la UTPL pueden ser, estilos de aprendizaje, preferencias, intereses y acciones sobre objetos del entorno virtual de aprendizaje.

1.5.2 Pasos del modelado de usuario

Se pueden encontrar algunas propuestas a la hora de construir modelos de usuario, al analizar algunos casos de aplicación se deducen los siguientes pasos a seguir, los cuales nos darán como resultado los elementos necesarios para la consecución del modelo esperado. Elena Gaudioso (2002) señala los siguientes:

- ◆ **Recogida de datos:** El modelado del usuario es una tarea que implica una recogida intensiva de datos y su posterior proceso. El objetivo es identificar los datos que consideramos que es necesario incluir en un modelo de usuario en un sistema de aprendizaje adaptativo.
- ◆ **Diseño del modelo de usuario:** El objetivo de este paso es identificar cuáles son los principales elementos necesarios para que el modelo de usuario satisfaga los requerimientos de datos de los módulos y aplicaciones de él dependan.
- ◆ **Construcción del modelo:** Al realizar este paso lo que se pretende es identificar un mecanismo de representación e inferencia que nos permita la construcción de modelos de usuario, abiertos y flexibles.
- ◆ **Validar el modelo construido:** Es necesario realizar dos tipos de evaluaciones. Por un lado, es necesario validar las técnicas utilizadas en la construcción y

actualización del modelo. Por otro, es necesario realizar una evaluación empírica de los modelos construidos, mediante la realización de una serie de experiencias con alumnos reales.

Esta secuencia de pasos asegura que el modelo obtenido cumpla con las expectativas, y que brinde la información más relevante sobre los estudiantes, además de permitir la incorporación de nuevos elementos en el modelo que ayuden a su especialización y mejor desenvolvimiento.

1.5.3 Elementos mínimos de un modelo de usuario

Los trabajos realizados en el campo de la ingeniería del conocimiento pretenden obtener ontologías explícitas que permitan desarrollar componentes reutilizables entre aplicaciones. Lo que realizan estas ontologías es capturar el conocimiento de un dominio en un modo genérico y proporcionan un mecanismo de comprensión que puede ser utilizada mediante aplicaciones y grupos de usuarios. El resultado de su utilización define un lenguaje común y niveles de formalidad que dan significado a los términos básicos y a sus relaciones (Chandrasekaran, Josephson, y Benjamins, 1999).

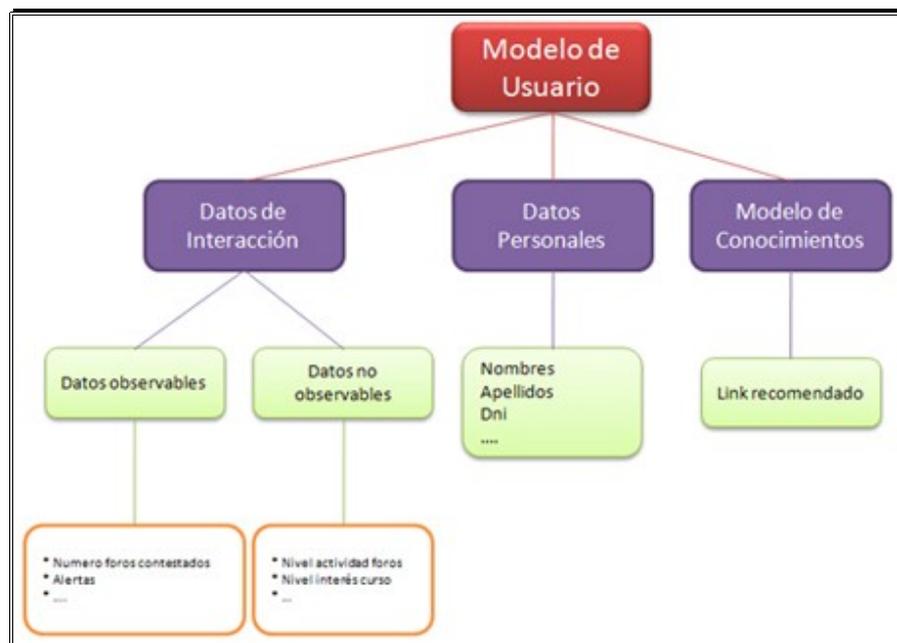


Figura 1.2: Elementos mínimos de un modelado de usuario expresados en forma de ontología

Pero pese a los arduos esfuerzos por estandarizar el proceso de representación de los elementos de un modelo de usuario, la aparición de una ontología lo suficientemente general que pueda ser utilizada por cualquier sistema en un dominio determinado, está aún distante, pero lo que se puede obtener como resultado del estudio de estos planteamientos es la identificación de los elementos mínimos comunes que debe incluir cualquier modelo de usuario. Los elementos que se incluyen se describen en la Figura 1.2.

En esta figura se reflejan los elementos que se consideran indispensables para que un modelo de usuario pueda ser considerado como tal. Se toman en cuenta datos de interacción en los cuales se diferencian los datos observables y los no observables, y como se puede ver en los ejemplos, los datos observables se pueden generar directamente aplicando reglas y los datos no observables se podrían obtener mediante aprendizaje automático, también se incluyen datos personales. Para el modelo de conocimiento se involucran los recursos con los cuales cuenta el sistema para la interacción con los usuarios.

1.6 Requisitos del modelo de usuario

Se necesita que el modelo que se va a diseñar cumpla con un conjunto determinado de requerimientos los cuales contribuirán al mejor desempeño de dicho modelo. Estos requisitos se tendrán que satisfacer independientemente del tipo de representación e inferencia que se utilice:

- ◆ *El modelo debe ser explorable.*- Ya que los usuarios necesitan mantener control sobre los datos y decisiones que se han tomado sobre él, porque no les gusta que los clasifiquen sin conocer las razones que influyeron en ello, debe darse acceso a que los mismos puedan manipular o cambiar ciertas partes del modelo, por ejemplo, puede decidir que parte del modelo es pública o qué parte es privada.

No solo por cuestiones de privacidad es que se considera valiosa la información que se obtiene del usuario sino que el propio usuario puede conocerse mejor y aportar su criterio en datos que le competen a su persona (Kobsa, 2002).

- ◆ *El modelo debe ser fácilmente escalable.*- En entornos web, tan cambiantes como los de hoy en día, éstos están sujetos a que fácilmente amplíen sus

servicios y funcionalidades. Por lo tanto, es muy deseable que no implique mucho esfuerzo adecuar o extender el modelo ya existente para que involucre a los nuevos servicios que se incrementen.

- ♦ *El modelo debe ser autónomo.*- Lo que se busca con el cumplimiento de este requerimiento es que el modelo que se va a construir pueda ser aplicado en una plataforma diferente pero que a su vez pueda compartir datos y sirva como una plataforma de colaboración de Internet.

1.7 Datos para el modelo de usuario

La selección apropiada de los datos a incluir en un modelo de usuario es crucial ya que de la calidad e integridad de los mismos depende en gran medida que los procesos que dependan del modelo cumplan efectivamente la labor para la cual han sido construidos. En esta sección se describen un conjunto de datos los cuales se consideran de relevancia para la definición de los datos de entrada que deben tomarse en cuenta para la definición de modelos de usuario.

En un sistema en el cual se desea saber las características y las preferencias de cada usuario, son necesarias y primordiales las tareas de recolección de información, algunos de estos datos pueden ser tomados directamente del sistema, mientras que otros requerirán de pasos adicionales y de interacción con otros sistemas para obtenerlos. En esta sección se detallan los diferentes tipos de datos que los sistemas adaptativos deben tener en cuenta, Según Kobsa, Koenemann, y Pohl (1999) estos datos deben contener a tres tipos importantes, una de ellas es la información del usuario, información acerca del uso del sistema, e información acerca del entorno del usuario, (software, hardware e información demográfica). A continuación se describe a cada uno de ellos:

1.7.1 Datos de usuario

Se conoce como información del usuario a las características personales de usuario, mientras que los datos de usuario son aquellos que están relacionados con el comportamiento del usuario en el sistema, en esta clasificación se puede divisar que existen diferencias a la hora de obtener los datos, ya que como se dijo anteriormente

habrá información que se obtendrá directamente del sistema pero otra necesita que se sea generada del uso del sistema, en la mayoría de casos esta información necesita ser procesada para poderla tomar en cuenta, de esto se puede decir que en general la complejidad de esta adquisición de datos será alta.

a) Datos demográficos

Los datos demográficos del usuario son datos específicos acerca del usuario e incluyen datos como los siguientes:

- ◆ Datos personales (username, password, nombres y apellidos, email, dirección y teléfono)
- ◆ Datos geográficos (código de área, ciudad, provincia, país, centro regional)
- ◆ Características del usuario (edad, sexo, educación, estado civil)
- ◆ Estatus social (datos acerca del estilo de vida, como si trabaja o no)
- ◆ Datos académicos (información acerca del rendimiento del estudiante en periodos anteriores).

1.7.2 Datos de uso del sistema

Puede ser adquirida directamente o necesitara del análisis mediante observación. En los sistemas de hipermedia que son basados en HTML solo son capaces de guardar las paginas y archivos que han sido pedidos al servidor (incluye una traza de navegación y los de los links del sitio). Otros sistemas que tienen mejor control acerca de la interacción como, por ejemplo Applets de Java, pueden guardar información hasta el nivel de clics de ratón y movimientos en la pantalla. También existen sistemas que guardan logs¹ de la interacción del usuario con el mismo, así como el monitoreo de la interacción de los usuarios en un nivel de widgets².

Otro enfoque depende del uso de una interfaz que se centra en el monitoreo de las

1 *Log*. Un log es un registro oficial de eventos durante un periodo de tiempo en particular que generalmente se almacenan en archivos de texto o modelos relacionales diseñados para el caso.

2 *Widget*. Un widget es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine o como módulos o complementos de aplicaciones más grandes.

acciones que el usuario desarrolla en el sistema o bien las interacciones con el mismo (abrir o cerrar páginas o contestar e ingresar a un foro).

Para el desarrollo de este modelo, casi toda la información de navegación e interacciones con el sistema por parte de los usuarios es relevante. Ya que, por ejemplo los datos de uso de los foros permitirá detectar aspectos como el nivel de actividad y nivel de uso de los mismos. Por otra parte, una traza de navegación bien nutrida (páginas accedidas, mensajes del foro leídos, etc.), permite detectar objetivos del usuario, patrones de navegación o intereses. Esta información va a permitir construir perfiles de interacción de los usuarios, formar grupos de trabajo compuestos por usuarios con perfiles e intereses semejantes, etc.

En este caso, los datos de uso se basarán fundamentalmente, en la forma en que los usuarios utilizan los servicios que se ofrecen sistema (foros, chats, etc.). Dichos servicios estarán disponibles en cualquier curso que lo requiera y, por tanto, los modelos construidos se podrán aplicar a todos los cursos que cumplan unas determinadas características. En los párrafos siguientes se describen los diferentes tipos de información los cuales se clasifican en datos de uso observables y los que nos ayudaran a generar las preferencias, hábitos y niveles de experiencia.

a) Uso observable

En sistemas adaptativos dedicados al aprendizaje podemos encontrar diferentes formas en que los usuarios pueden interactuar con el sistema, y estas pueden tener un mayor o menor grado de relevancia a la hora de intervenir en la selección de datos de entrada. En las siguientes líneas se describen los tipos de interacciones que sirven para ser tomadas como observaciones en el sistema.

- ◆ *Acciones Selectivas.*- Según Kobsa, Koenemann y Pohl (1999) se obtienen de la interacción con un sistema basado en la Web en la que se dé clic en un link, al realizar esto, el usuario hace una elección si para su parecer los elementos contenidos en ese link son competitivos. Este tipo de acciones pueden ser tomados como indicadores, del nivel de interés en cierto contenido, así como el nivel de desconocimiento del sistema, y además obtener preferencias de objetos de los usuarios. Entre los indicadores que se puede obtener mediante acciones

selectivas tenemos:

- ◆ *Interés* (de los usuarios a agrupaciones de ítems presentados).
- ◆ *Desconocimiento* (de términos técnicos de parte de los usuarios)
- ◆ *Preferencias* (de usuarios a determinados objetos presentados)
- ◆ *Ratings*³.- Se basa en la calificación de objetos (documentos, noticias, artículos, productos, etc.). Esta calificación indica que relevante o interesante es ese objeto para el usuario, los usuarios pueden aplicar esta calificación mediante, escalas explícitas o mediante escalas discretas. En la tienda en línea Amazon.com se ofrece a los usuarios la oportunidad de calificar sus productos y mediante un algoritmo basado en filtrado se genera los ratings (Schwab, Kobsa, y Koychev, 2000).
- ◆ *Otras acciones de Aprobación o de Desaprobación*.- Kobsa, Koenemann y Pohl (1999) sostienen que las acciones de desaprobación ofrecen fuertes indicadores de la selección de objetos. Entre las acciones confirmatorias, podemos anotar: guardar un documento, imprimirlo, las acciones de incluir marcadores.

Considerando todo esto se puede decir que, existen muchas alternativas a la hora de obtener indicadores de interés mediante acciones de aprobación y de desaprobación, pero en gran medida se facilitara su uso en la identificación de tendencias de interés teniendo a nuestro haber un conjunto de datos, que muchas veces son difíciles de obtener debido a que para contar con ellos, se necesita que el sistema cuente con los mecanismos apropiados de extracción. En algunos casos se necesita que el sistema cuente con mecanismos tan sofisticados como agentes de interfaz ya que la mayoría de sistemas en funcionamiento no han sido pensados como para recolectar información de las acciones que el usuario realiza con el sistema.

b) Regularidades de uso

En algunos casos el observar las interacciones no se relaciona directamente con las adaptaciones. Es necesario un procesamiento adicional de datos para adquirir información acerca de las preferencias de los usuarios, hábitos y niveles de

³ *Rating*. El rating es un indicador de calificación o clasificación.

aprendizaje, para un manejo profesional de los datos en un sistema adaptativo se puede basar en el comportamiento del usuario en el sistema. En esta sección se describirán ejemplos típicos del uso de la información adquirida de observar la información de uso del sistema. Para esto incluiremos la distribución de tipos de datos realizada por Kobsa, Koenemann y Pohl (1999).

- ◆ *Frecuencia de uso.*- Quizás el modo más obvio para procesar la información de uso es el categorizar los eventos y contar sus frecuencias. Debevc, Meyer, Donlagic, y Svecko (1996), describen un mecanismo que controla la configuración de una barra de herramientas adaptativa para Microsoft Word. Los iconos son añadidos o removidos en la barra, basándose en la frecuencia de uso de los mismos o de los comandos relacionados, Linton y Schaefer (2000) compararon el uso individual a largo plazo de los comandos de Word con un grupo de datos de uso. Basados en esta comparación, los comandos que no son usados por usuarios individuales pero frecuentemente empleados por el grupo pueden ser puestos fuera del contexto de uso individual.
- ◆ *Correlaciones entre Acción y Situación.*- En los noventa aparecieron las nociones de agentes de interfaz y los asistentes personales, en gran medida por las investigaciones de Mitchell, Caruana, Freitag, McDermott, y Zabowski (1994). Un hecho común en los primeros trabajos en este ámbito es que todos ellos utilizaban las correlaciones entre acción y situación como base para el comportamiento adaptativo. Both, Mitchell y Maes (1995) describen un asistente de correo electrónico que genera recomendaciones de cómo el usuario podría tratar con los mensajes entrantes. Esas recomendaciones eran basadas en estadísticas que expresan correlaciones generalizadas entre los correos anteriores (situaciones) y como el usuario los procesó (acciones). Maes (1996) también describe un agente de contactos que trabaja en un modo similar, pero procesa peticiones de contacto en vez de correos electrónicos.
- ◆ *Secuencias de Acciones.*- Las secuencias de acciones son mayoritariamente analizadas para:
 - ◆ Recomendar la generación de macros para secuencias de acciones
 - ◆ Predecir acciones futuras en base a acciones pasadas, y

- ◆ Recomendar acciones basadas en secuencias de acciones frecuentes de otros usuarios.

Varios métodos son usados para determinar regularidades y similitudes en secuencias de acciones.

1.7.3 Datos de entorno

Se conocen como datos de entorno a todos aquellos que son tomados, con respecto a la plataforma hardware y software en el que el sistema es accedido por los usuarios. En el lado del cliente en un sistema basado en la web, el rango de hardware y software que es usado es extremadamente amplio. Y aun más, este espectro aun está aumentando con la aparición de dispositivos que soportan el acceso web, pero con capacidades limitadas. (Dispositivos como netbooks, PDAs, y teléfonos celulares). Estos cambios a la hora de construir o de adecuar las aplicaciones basadas en la web, proveen un número mayoritario de posibles entornos. El uso de la web puede ser influenciado por el software y el hardware de un usuario individual, y por las características de los usuarios en el entorno local. Se van a considerar, cada una de esas limitantes en las siguientes secciones (Kobsa, Koenemann, y Pohl 1999).

a) Entorno de software

Cada pieza de software para la web no necesariamente soportan todas las características actuales, las versiones más antiguas de los navegadores tienen capacidades muy limitadas. Los nuevos tipos de dispositivos móviles, con capacidades web, se incluyen a esta diversidad para implantar nuevos paradigmas y escenarios en los cuales se debe obtener cierto tipo de información para realizar adaptaciones adecuadas. Los datos que se deben considerar para delimitar el entorno de software en que se encuentra el sistema, son los siguientes:

- ◆ *Versión de Software y Navegador*
- ◆ *Disponibilidad de plugins*

- ◆ *Java y JavaScript⁴ habilitados*

Como ya se conoce, Internet es una red cambiante que cada vez incorpora nuevas tecnologías, y por esto obliga a que los sistemas que conviven en ella obedezcan a estos cambios así que la disponibilidad de plugins, descrita anteriormente, toma un grado mayor de relevancia tanto como Java o JavaScript ya que muchas de las aplicaciones en línea los utilizan para el funcionamiento de sus servicios. Un caso muy conocido es Youtube que sin el uso del plugin Adobe Flash Player, mencionado anteriormente, no podría realizar el trabajo para el que fue desarrollado, entonces se hace primordial que los mecanismos que detectan estos complementos en los navegadores sean capaces de detectar la mayor parte de los mismos y así describir mejor el entorno de software que rodea a un usuario de un sistema en línea.

b) Entorno de hardware

Hay una gran diferencia entre el hardware que usan los desarrolladores y diseñadores, y el hardware que tienen a su haber los visitantes de un sistema en la web. Para las aplicaciones de intranet, precisar la información acerca de las restricciones de hardware que puede haber, es tarea difícil. En el caso de Internet, solo se puede obtener suposiciones y estimados de esa información. Entonces para dicha tarea Kobsa, Koenemann, y Pohl (1999) proponen también algunos elementos a considerar en la descripción de un entorno de hardware para un modelo de usuario que incluyen los siguientes:

- ◆ *Ancho de Banda* (influye en el tiempo de descarga de contenidos)
- ◆ *Velocidad de procesamiento* (en la interpretación de los archivos descargados)
- ◆ *Dispositivos de presentación* (contenidos adecuado para el dispositivo con el que se accede al sistema)
- ◆ *Dispositivos de Entrada* (flexibilidad para la incorporación de los nuevos dispositivos de entrada)

⁴ *JavaScript*. Lenguaje de programación para WWW desarrollado por Netscape. Al igual que VBScript, pertenece a la familia Java pero se diferencia de este último en que los programas están incorporados en el fichero HTML (Fernández, 2008).

c) **Localidad**

La información de uso local puede ser utilizada para filtrar contenidos para la adaptación, por ejemplo, formatos de presentación y contenidos multimedia, y para hacer recomendaciones basados en un conocimiento geográfico y observaciones pasadas para obtener regularidades de los usuarios (Fink, 2001). Esta información acerca del uso local puede incluir los siguientes tipos:

- ◆ *La localización Actual de los Usuarios.*- La granularidad requerida en la información de localización varía directamente dependiendo del rango de cada país, desde delimitaciones políticas de los estados (decir que un usuario pertenece, a un continente, a un país o una ciudad) hasta mediciones en metros o hasta centímetros en el caso de aplicaciones que determinan la localización.
- ◆ *Características de uso local.*- Información que puede ser tomada en una cuenta que incluye el nivel de ruido y brillo del entorno, así como información acerca de los lugares y objetos en el medio ambiente inmediato.

1.7.4 **Métodos de adquisición de datos**

En el apartado anterior se describieron los tipos de datos de usuario, uso del sistema, y del entorno de usuario, que son potencialmente usables como entradas de datos en sistemas de hipermedia educativos. Ahora en este apartado se describen los métodos que pueden ser usados para obtener esos datos, Además, dentro del proceso de modelado, las técnicas que se describen a continuación permiten expresar en forma de modelo la información adquirida y que esta pueda ser utilizada posteriormente.

a) **Información proporcionada por parte del usuario**

Una estrategia obvia para obtener información acerca del usuario es dejar que el mismo sea quien la brinde (Kobsa, Koenemann, y Pohl, 1999). Para algunos datos de usuario (datos demográficos especialmente) el ingreso de los mismos por parte del usuario es la forma más sencilla de obtenerlos. Cuando se desea aplicar a sistemas ya en funcionamiento, como es el caso de la UTPL, esta información se puede obtener directamente porque ya se encuentra dentro del mismo.

Esta obtención de datos de usuario se puede realizar por medio de preguntas individuales propuestas por el sistema, o en tests completos, típicamente en fases iniciales del uso del sistema. Estos tests pueden ser de varios tipos dependiendo del contexto en el que van a ser usados, pudiendo abarcar temas como: Identificación de estilos de aprendizaje, conocimientos iniciales del sistema y del entorno o conocimiento de una asignatura (en caso de sistemas de hipermedia educativos). Algunos sistemas presentan preguntas controladas, pruebas y ejercicios y así pueden evaluar más objetivamente a cada usuario.

Esta forma de realizar adquisición de datos de entrada mantiene una gran paradoja, que consiste en que los usuarios se rehúsan a desperdiciar tiempo alimentando al sistema con información. En algunos casos se ha visto que los usuarios pueden llegar al punto de no visitar un determinado sitio por tener que contestar una entrevista inicial. Para ello, en algunos sistemas se ha relegado esta tarea a una fase posterior, para que el usuario se familiarice con los beneficios que puede obtener del mismo.

Los usuarios pueden tener la oportunidad de brindar información acerca de sus características personales en una forma más personalizada, por ejemplo, como parte de cuadros de diálogo de preferencias o en momentos arbitrarios de la interacción con el sistema. El estándar P3P World Wide Web (WWW) da la facilidad a los usuarios para que le faculten a los sitios con políticas de privacidad confiables a preguntarles con cierta frecuencia sobre datos personales y así no tener que ingresarlos en todos los sitios en que sean requeridos.

b) Reglas de adquisición

Para Elena Gaudio (2002) los métodos pasivos de adquisición de datos (en los que el usuario no brinda la información sino que el sistema la infiere automáticamente), también llamados implícitos, prometen ser un poco menos molestos para los usuarios que una adquisición activa. Por definición, los métodos pasivos de adquisición de datos no necesitan iniciar ningún tipo de interacción con los usuarios.

La forma más usada para la generación de suposiciones acerca del usuario, son las reglas de adquisición de datos, que se entienden como reglas de inferencia que se

ejecutan cuando está disponible nueva información acerca del usuario. En la mayoría de los casos las reglas de adquisición, se relacionan con acciones observadas del usuario o una interpretación menos fuerte del comportamiento del usuario en el sistema. Las reglas de adquisición van a ser específicas para cada dominio de aplicación o pueden ser independientes de la aplicación.

Las reglas de adquisición son completamente específicas para la aplicación, son quizá las más populares para la adquisición de modelos de usuario, porque estas son sencillas de implementar. Sin embargo, no son muy flexibles para utilizar, y sus propiedades generalmente no son descritas formalmente. En algunos casos, las reglas específicas de la aplicación o sus procesos son aplicados para derivar mejores conclusiones de las interpretaciones de las reacciones de los usuarios para implementar o testear ítems (Akoulchina y Ganascia, 1997).

1.8 Estilos de aprendizaje y modelado de usuario

Los estilos de aprendizaje dentro de los sistemas de hipermedia adaptativos han generado un gran campo de investigación en los últimos tiempos. En estudios como el de Fontalvo y otros (2007), se presentan varias intervenciones de la teoría de estilos de aprendizaje en el desarrollo de sistemas de hipermedia adaptativos y en el desarrollo de ambientes de aprendizaje virtual, siendo de gran acogida y tomándolos como base para la generación de las personalizaciones, esto se debe principalmente a las delimitaciones en la forma de generar el contenido, adaptándolo a las necesidades de cada forma de aprender del usuario.

Dentro de modelos de usuario, los estilos de aprendizaje, en algunos sistemas tienen un rol primordial, por ejemplo, en el desarrollo del Sistema de Tutoría Multiagente MAS-PLANG, de Peña, Marzo, de la Rosa y Fabregat (2007) se modela a cada estudiante de acuerdo a su estilo de aprendizaje a través del modelo de Felder y Silverman y luego se refina por medio del análisis de las interacciones del estudiante en el entorno de aprendizaje mediante aprendizaje automático.

En esta sección haremos una introducción a la teoría de estilos de aprendizaje, enfatizando los modelos más importantes que se están aplicando en investigaciones acerca de su aplicación en sistemas de soporte al aprendizaje y de hipermedia

adaptativos.

1.8.1 Estilos de aprendizaje

Desde la aparición de las nuevas tecnologías de la información muchos aspectos del quehacer humano han tenido que adaptarse a ellas, uno de estos aspectos es la educación que ha tenido que acoplarse a los cambios profundos que va desarrollando el hombre, fruto de esta unión han aparecido los llamados cursos virtuales, así como tecnologías que apoyan el trabajo de los mismos, una de esas tecnologías son los LMS que implementan herramientas que ayudan a la gestión de cursos en línea. Pero luego de un periodo de tiempo han aparecido nuevos requerimientos educativos y se hace necesario conocer más de los estudiantes que interactúan en estos cursos en línea para poder aportar en el proceso de su aprendizaje, es aquí donde se ha creído conveniente incluir el conocimiento de estilos de aprendizaje que ayuden a que estos sistemas presenten contenidos acorde a las necesidades individuales de cada estudiante.

Existen muchos factores que influyen considerablemente en la forma de aprender de las personas, además que las demandas de la sociedad actual, exigen que el aprendizaje sea continuo, para que los individuos se conviertan en sujetos autónomos, que tengan consciencia de sus procesos mentales y puedan resolver problemas, analizarlos, planificar soluciones, y evaluar su actuación propia (García y Pascual, 1994).

Estudios acerca de la concepción del aprendizaje de los seres humanos han demostrado que estos aprenden de diferentes formas (Alonso, 2001). Entonces, es pertinente que los estudiantes entiendan que existen varias factores que inciden en su aprendizaje y los maestros deben comprender su importancia ya que así podrán incorporar nuevas herramientas y técnicas a la hora de impartir sus conocimientos. Ante esto, el campo de la psicología se ha manifestado afirmando que las personas tenemos diferentes *estilos de aprendizaje*, que inciden en gran medida en la forma en que las personas aprenden.

Una de las definiciones más divulgadas internacionalmente en la actualidad sobre lo que se entendería como estilos de aprendizaje, según Alonso y otros (1999), es la

de Keefe (1988), quien propone asumir los estilos de aprendizaje en términos de “Aquellos rasgos cognitivos, afectivos y fisiológicos, que sirven como indicadores relativamente estables de cómo los dicentes perciben, interaccionan y responden a sus ambientes de aprendizaje”.

Sin dudas, y como afirma Curry (1983), uno de los obstáculos más importantes para el desarrollo y aplicación de la teoría de los estilos de aprendizaje en la práctica educativa, es la confusión que provoca la diversidad de definiciones que rodean al término, a lo cual se suma también la heterogeneidad de clasificaciones que abundan entre los diferentes autores. Además, que las clasificaciones fueron diseñadas para satisfacer a los diferentes puntos de vista que sus autores conciben sobre la forma que estos influyen en los procesos de aprendizaje de los estudiantes.

1.8.2 Clasificación de los estilos de aprendizaje

Según Cabrera y Fariñas (2005) existen muchas clasificaciones que se fundamentan principalmente en dos preceptos fundamentales: Las formas de percibir la información y las formas en que se la procesa. En esta sección se considera un conjunto de clasificaciones de estilos de aprendizaje, incluidos por la relevancia que han tenido en el ámbito de la educación sus investigaciones y su inclinación hacia la aplicación y evaluación de estrategias de enseñanza basados en los estilos de aprendizaje y mediados por el uso de software educativo.

El modelo propuesto Rita y Keneth Dunn (1978, 1982), fue uno de los primeros intentos por clasificar a los estilos de aprendizaje y se basa en la distinción de lo que sus autores llamaron modalidades perceptuales, las cuales delimitan las acciones con las que responderán los estudiantes a las tareas de aprendizaje. Esta clasificación incluye los siguientes estilos de Aprendizaje *estilo visual* o *estilo auditivo* y *estilo táctil* o *estilo kinestésico*.

El modelo teórico de gran relevancia propuesto por el psicólogo norteamericano David Kolb (1976, 1984), clasifica a los estudiantes en grupos como *convergentes* o *divergentes*, *asimiladores* o *acomodadores*, en dependencia de como perciben y cómo procesan la información. Para el autor los estudiantes *divergentes* captan la información por medio de experiencias concretas y la procesan reflexivamente, los

convergentes se caracterizan por percibir la información de forma abstracta, mediante la forma teórica de formulación conceptual y la asimila mediante una experimentación activa. Los *asimiladores* o *analíticos*, tienden perciben también la información de forma abstracta, pero la procesan reflexivamente. Por ultimo, los *acomodadores* captan la información a partir de experiencias concretas y la procesan activamente.

Se presenta la clasificación realizada por Ronal Schmeck (1982, 1988), el cual va de la mano directamente con estudios sobre estrategias de aprendizaje. Define tres estilos de aprendizaje, cada uno de los cuales supone el uso de una estrategia de aprendizaje particular por parte del estudiante: *Estilo de profundidad*: propio de aquel alumno que usa la estrategia de conceptualización, lo cual quiere decir que cuando estudia abstrae, analiza, relaciona y organiza las abstracciones (estrategia facilitadora de un aprendizaje de alto nivel); *Estilo de elaboración*: el cual implica la utilización por parte del estudiante e una estrategia personalizada. Para este estudiante el contenido de estudio ha de estar relacionado directamente con él mismo, con sus experiencias, con lo que ha pasado o piensa que va a pasar (estrategia facilitadora de un aprendizaje de nivel medio); y *Estilo superficial*: el cual implica el uso de una estrategia centrada en la memorización; el alumno solo recuerda el contenido que repasó al estudiar (estrategia facilitadora de un aprendizaje de bajo nivel).

Felder y Silverman (1988) originalmente consideraron que el aprendizaje de las personas debe ser evaluado en cinco dimensiones: percepción, entrada, organización, procesamiento y comprensión; pero en una corrección del mismo hacen una supresión de la dimensión de organización (inductivo y deductivo) dejando cuatro dimensiones las cuales se describen a continuación:

- ♦ **Aprendizaje visual y verbal (Dimensión entrada).**- Las personas visuales recuerdan mejor lo que ven: figuras, diagramas, cuadros, películas, demostraciones. Las personas auditivas recuerdan más lo que escuchan y mucho más lo que ellas dicen. Ellas aprenden a partir de la discusión y prefieren las explicaciones verbales a las demostraciones visuales.
- ♦ **Aprendizaje activo y reflexivo (dimensión procesamiento).**- En esta dimensión se intenta enmarcar el proceso mental complejo mediante el cual la

información percibida es convertida en conocimiento: la experimentación activa implica hacer algo en el mundo externo con la información (discutirla, explicarla o chequearla de alguna manera), y observación reflexiva implica examinar y manipular la información mediante meditación y conciencia. Según esto, una persona activa es aquella que se siente más cómoda con la experimentación activa que con la observación reflexiva, al revés de una persona reflexiva. Las personas activas no aprenden mucho en situaciones en las que ellas deben estar pasivas, tales como lecturas; trabajan bien en grupos y tienden a ser experimentalistas. Las personas reflexivas no aprenden bien en situaciones que no les proporcionan la oportunidad de pensar sobre la información que se les presenta; trabajan mejor solas o a lo sumo con una persona más.

- ♦ **Aprendizaje secuencial y global (dimensión comprensión).**- El aprendizaje secuencial es el aplicado en la mayoría de la educación formal, e implica la presentación de material en un orden progresivo y lógico, y de acuerdo con el avance del aprendizaje regido por el tiempo. Cuando una parte del material ha sido cubierto los estudiantes son evaluados y recién pueden pasar al siguiente nivel. El aprendizaje global es aquel que no se rige por el tiempo, por el contrario se pueden pasar días o semanas ocupados en resolver un simple problema o demostrando una comprensión rudimentaria hasta que de pronto se le “encienden las luces” y logran una rápida comprensión del todo.
- ♦ **Aprendizaje sensitivo e intuitivo (dimensión percepción).**- Las personas que presentan una tendencia hacia la dimensión sensitiva demuestran ser sujetos a los que les gusta trabajar con hechos, datos concretos y experimentación. En el caso de las personas catalogadas dentro la dimensión intuitiva, estos prefieren tener acceso a una información basada en teorías y principios, tratando siempre de innovar en la forma como perciben un nuevo conocimiento, llegando sin ninguna dificultad a crear nuevos conceptos, evitando ser metódicos y poniendo a prueba toda su creatividad.

Finalmente se describe la clasificación propuesta por Honey y Mumford (1990), Estos autores consideran que el proceso de aprender implica un recorrido cíclico por

cuatro etapas sucesivas: tener una experiencia, reflexionar sobre ella, extraer conclusiones y planificar los pasos siguientes a aplicar. El estudiante recorre las cuatro etapas y en el proceso presenta preferencias distintas, en cada una de ellas lo que a la final expone su estilo de aprendizaje. Además proponen cuatro estilos de aprendizaje, que surgen directamente de la relación que existe de ellos con las etapas de este ciclo, estos estilos son: aprendizaje activo, aprendizaje reflexivo, aprendizaje teórico y aprendizaje pragmático.

Según Cabrera y Fariñas (2005) estas clasificaciones presentan un alto valor heurístico, pero a su vez incluyen un alto nivel cognitivista que limita el tratamiento que los estilos de aprendizaje dentro de la personalidad integral del estudiante.

Desde mi punto de vista también hay que tomar en cuenta el hecho fundamental de que los estilos de aprendizaje contemplan un carácter flexible que ayuda a que las nuevas concepciones en educación complementen su crecimiento. Además tienen que surgir nuevas propuestas y técnicas para una aplicación más efectiva de sus principios.

1.9 Modelado de usuario con aprendizaje automático

Antes de adentrarnos en el estudio de las técnicas a considerar para la introducción de aprendizaje automático en la definición de modelos de usuario, se cree conveniente explicar la razón por la cual se ha considerado necesario la inclusión de dicha teoría.

El fundamento principal según Gaudioso (2002) es que los sistemas adaptativos, y por ende, los sistemas de apoyo a la educación necesitan que se analice el comportamiento de sus usuarios, para que las observaciones realizadas permitan predecir acciones futuras del usuario en el sistema. En este campo muchos autores han visto y estudiado la factibilidad de aplicación de ciertas técnicas de aprendizaje automático, en este campo, la implementación de aprendizaje supervisado es de gran incidencia, pero conlleva un gran problema y es la necesidad de grandes cantidades de ejemplos de entrenamiento para poder realizar su labor (Webb, Pazzani y Billsus, 2001). Y es ahí en donde algunos sistemas han implementado aprendizaje no supervisado, donde se trata de ordenar jerárquicamente según las regularidades de los pares atributo-valor sin usar un atributo de clase como se realiza en el aprendizaje

supervisado. En los dominios de aplicación en donde se ha aplicado el aprendizaje automático como medio para la construcción y formalización de modelos de usuario los resultados han sido satisfactorios, pero según Gaudioso (2002), si lo que se pretende es aprender modelos de usuario más complejos:

*“Es necesario realizar varias **tareas de aprendizaje** o bien combinar distintas técnicas de aprendizaje para una misma tarea de manera que se pueda aumentar su eficacia”.*

Existen algunas implementaciones de modelos de usuario a partir de técnicas de aprendizaje automático, y se dice que son heterogéneos ya que utilizan varias técnicas de aprendizaje así como tipologías diferentes de datos. Además de implementar variadas técnicas de aprendizaje un sistema de esta magnitud debe poder seleccionar que técnica de aprendizaje automático se ajusta de mejor manera a un conjunto de datos, o si necesita que para una mejor clasificación de las instancias se realice una combinación de clasificadores, ayudando al refinamiento de las tareas de aprendizaje.

1.9.1 Tareas de aprendizaje

Las tareas de aprendizaje no son más que tareas de clasificación, las cuales necesitan de un conjunto de datos de entrenamiento etiquetados. Esta forma de trabajo presenta un gran problema, que se fundamenta en la necesidad de una gran cantidad de datos de entrenamiento, como solución nace la propuesta de diseñar tareas de aprendizaje no supervisado que agilicen el trabajo de las mismas. Otro inconveniente es que en los sistemas de apoyo a la educación, la definición de todas las tareas de aprendizaje es difícil en la etapa de modelado, por lo que se sugiere es delimitar un mecanismo mediante el cual se puedan ir incorporando nuevas tareas de aprendizaje al modelo de usuario, en una representación híbrida.

En el siguiente apartado mostramos algunas estrategias de combinación de clasificaciones para mejorar el trabajo de algunas tareas de aprendizaje dentro del modelado de usuario.

1.9.2 Mejora del rendimiento de tareas de aprendizaje

Como vimos en las secciones anteriores las tareas de aprendizaje juegan un papel preponderante en la definición de atributos en los modelos de usuario, pero, en una representación híbrida de modelos de usuario se ve la necesidad de que algunas de las tareas de aprendizaje implementen métodos más elaborados de definición de sus atributos, debido esto esencialmente a la heterogeneidad de los datos que se obtienen dentro de plataformas Virtuales. En esta sección se muestran los fundamentos por los cuales las variaciones en los datos de entrada pueden modificar el rendimiento de algunas tareas de aprendizaje.

Fundamentación

Los algoritmos de aprendizaje automático implementan estrategias de búsqueda, que no son más que un conjunto de métodos y heurísticas que se utilizan para explorar el espacio de búsqueda definido por un conjunto de representaciones posibles. Se dice entonces que no existen algoritmos mejores que otros en diferentes dominios. Según Mitchell (1990) cada algoritmo implementa varias representaciones implícitas y explícitas que hacen que se elijan unas hipótesis frente a otras. Entonces, se puede decir que cada algoritmo funciona de mejor manera si los dominios se ajustan mejor a los métodos de representación de cada algoritmo. Por lo tanto se encuentran dos tipos de errores: unos debidos a la representación del algoritmo y otros debidos a la dependencia entre el modelo aprendido y el conjunto de datos de entrenamiento (también llamada *varianza*).

Por las razones descritas en el párrafo anterior se sostiene que pequeñas variaciones en los datos de entrada, pueden producir resultados muy diferentes dentro de los clasificadores, pero no son sensibles a los cambios en los atributos. Estas implicaciones influyen mucho ya que cuando se conoce, antes de la clasificación, los datos y el método de representación es posible encontrar un algoritmo que trabaje correctamente con los mismos. Pero en el escenario contrario donde no conocemos estos factores la selección de un algoritmo se vuelve complicada, para esto se introduce una combinación de clasificadores que trate de minimizar el impacto de dentro de la selección de los algoritmos.

Combinación de clasificadores

La combinación de clasificadores también conocida como *modelos múltiples* se utiliza para maximizar de cierta forma el resultado obtenido por cada algoritmo en la clasificación de nuevos ejemplos (Dietterich, 1997).

Se necesita que los algoritmos que son parte de esta representación cumplan con una característica esencial, la cual es: que los algoritmos tienen que ser a la vez precisos y diversos. Un algoritmo es preciso si tiene un error menor al que se obtendría eligiendo una clase arbitrariamente. La diversidad en los clasificadores de evidencia si los errores que estos presentan en la clasificación de nuevas instancias son diferentes.

Tipos de combinaciones de clasificadores

Según Gaudioso (2002) existen varios tipos de combinaciones de clasificadores, pero generalmente obedecen a dos criterios de clasificación: según el método que utilizan para combinar predicciones, o bien, según los métodos que se utilicen para generar los clasificadores.

En función del método de combinación de las predicciones de los clasificadores que intervienen en la combinación se menciona:

- ◆ *Métodos con o sin votación.*- Los métodos con votación utilizan un contador de votos (con o sin peso) de la predicción que cada algoritmo ofrece, y en los métodos de no votación cada algoritmo obtiene una estimación de la probabilidad de la clase en vez de una simple predicción.
- ◆ *Métodos estáticos y métodos dinámicos.*- Los métodos estáticos consultan todos los clasificadores que intervienen en la combinación, por otra parte los métodos dinámicos determinan para cada clasificador cual es su área de competencia y se consulta solo aquellos en los que el área de competencia de los mismos incluya a dicha entrada. Destaca entonces el método MAI⁵ (Ortega, 1995).

En función del método de construcción de los clasificadores, las combinaciones de

⁵ MAI. del inglés model applicability induction

clasifican de la siguiente forma:

- ◆ *Combinación de clasificadores homogéneos.*- Combinan clasificadores generados por un único algoritmo de aprendizaje. Se basa en que todos los algoritmos manipulan los mismos datos de ejemplo para generar las hipótesis. Entonces, el algoritmo se ejecuta varias veces, utilizando en cada una de ellas un conjunto diferente de ejemplos de entrenamiento. Se destacan dos métodos de este tipo: *bagging* y *boosting*;
- ◆ *Combinación de clasificadores heterogéneos.*- en este método se combinan clasificadores que se han generado a partir de diferentes algoritmos de aprendizaje, destacándose el *método de pila*.

Como se ve hay algunas alternativas a la hora de realizar combinaciones de clasificadores para maximizar el poder de trabajo de los mismos lo que nos brinda un gran espectro para luego escoger el método que más se ajuste al trabajo al que va a ser aplicado.

1.9.3 Algoritmos de aprendizaje automático

En esta sección se desarrolla una descripción de los algoritmos de aprendizaje automático que serán utilizados en la experimentación con los datos de interacción de los estudiantes de la UTPL con el entorno virtual de aprendizaje.

a) Árboles de decisión C4.5 (J48)

El algoritmo J48 de WEKA es una implementación propia del algoritmo C4.5, propuesto por Quilan (1993). Es uno de los algoritmos de minería de datos más utilizados. Se trata de un refinamiento del modelo generado con OneR que supone una mejora moderada en las prestaciones.

Genera árboles de decisión para atributos discretos y continuos, utilizando una razón de ganancia para seleccionar el atributo de cada nodo y aplica estrategias de poda para reducir el ruido de los datos de entrenamiento.

El proceso de construcción del árbol comienza por el nodo raíz, el que tiene asociados todos los ejemplos o casos de entrenamiento. Lo primero es seleccionar la

variable o atributo a partir de la cual se va a dividir la muestra de entrenamiento original (nodo raíz), buscando que en los subconjuntos generados haya una mínima variabilidad respecto a la clase. Este proceso es recursivo, es decir, una vez que se haya determinado la variable con la que se obtiene la mayor homogeneidad respecto a la clase en los nodos hijos, se vuelve a realizar el análisis para cada uno de los nodos hijos. Aunque en el límite este proceso se detendría cuando todos los nodos hojas contuvieran casos de una misma clase, no siempre se desea llegar a este extremo, para lo cual se implementan métodos de pre-poda y post-poda de los árboles.

El algoritmo J48 amplía las funcionalidades del C4.5, tales como permitir la realización del proceso de post-poda del árbol mediante un método basado en la reducción del error (`reducedErrorPruning`) o que las divisiones sobre las variables discretas sean siempre binarias (`binarySplits`) 4.5. Algunas propiedades concretas de la implementación son las siguientes:

- ◆ Admite atributos simbólicos y numéricos, aunque la clase debe ser simbólica.
- ◆ Se permiten ejemplos con valores desconocidos
- ◆ El criterio de división está basado en la entropía y la ganancia de información

b) Árboles de decisión REPTree

Su criterio principal para la construcción del árbol de decisión es utilizar la ganancia en la información recolectada y realizando con esto una poda de error reducido. Para la ordenación se realiza una única pasada que lista ordenadamente los valores numéricos. Los valores que no se encuentran presentes se obtienen dividiendo las instancias correspondientes en varios segmentos (Witten y otros, 2000).

c) Redes neuronales con perceptrón multicapa (MultilayerPerceptron)

Los perceptrones multicapa aparecieron de la dificultad de los perceptrones simples en clasificar ciertos tipos de atributos en los cuales se necesitaba que se combinen los anteriores para poder inferirlos. Generalmente este tipo de atributos aparecen en aproximaciones de entrada-salida que no son lineales, para lo cual se implementan redes con diferente número de capas y con funciones de activación no

lineales, además de una matriz de pesos asociada. En investigaciones realizadas por varias personas se demostró que redes con capas ocultas y con funciones de activación no lineales continuamente diferenciables, pueden aproximar cualquier mapeo suave, si se producen suficientes neuronas en las capas ocultas. Generalmente se utiliza el teorema de Stone-Weirstrass (Seung, 2002).

d) Clasificador NaiveBayes

Al estar el algoritmo basado en variables, este sostiene que las variables que representan a las instancias son estadísticamente independientes. Esta simplificación del problema arroja resultados con una buena aproximación. Se calcula la probabilidad de que una instancia pertenezca a la clase, además de calcular la probabilidad condicional de que un atributo tome un valor si la instancia pertenece a una determinada clase, luego al terminar de recoger estos datos la probabilidad de pertenencia a la clase se puede calcular utilizando la fórmula de Bayes y asumiendo independencia entre las variables. La clasificación de la instancia dada será la que maximice esta probabilidad (Duda y Hart, 1973).

Se genera un modelo gráfico probabilístico de la hipótesis de independencia asumida por este algoritmo, en el que el nodo raíz se toma como la clase y los demás atributos se consideran nodos hoja que son precedidos por su padre la variable de clase (Hernández, Ramírez y Ferri, 2004).

e) Redes Bayesianas (BayesNet)

Las Redes Bayesianas son un formalismo que tiene un gran potencial como modelo de representación del conocimiento con incertidumbre. Nació de la aportación de diferentes campos de investigación: teoría de toma de decisiones, estadística e inteligencia artificial.

Representan el conocimiento mediante un grafo dirigido acíclico, que mantiene relaciones de independencia/dependencia entre las variables que componen el modelo, abarcan desde la independencia completa de las variables hasta una dependencia funcional de las mismas. Su calidad como modelo gráfico hace que sea atractiva su utilización en minería de datos.

Además de modelar cuantitativamente el conocimiento expresan de forma numérica la solidez con la que se relacionan las variables, esto generalmente se realiza implementando distribuciones de probabilidad para medir la fortaleza de las relaciones entre las variables de clase (Hernández, Ramírez y Ferri, 2004).

Para la construcción de un clasificador basado en redes bayesianas se utilizan diferentes métodos, uno de los más usados es el denominado *TAN*⁶, que construye una red un poco más compleja que el algoritmo Naive Bayes, la idea principal de esto es conservar la simplicidad computacional del Naive Bayes pero tratando de mejorar la tasa de acierto en la clasificación, construyendo una red con dependencias entre las variables en forma de árbol (Friedman, Geiger y Goldszmidt, 1997).

Luego de representar el conocimiento se hace necesaria una forma en la que se pueda formalizar los datos en esta representación, una forma de hacerlo es mediante la maximización de la métrica de adaptación de los datos al esquema propuesto. Para desarrollar este proceso se han implementado métodos de búsqueda para resolver el problema de aprendizaje estructural. Uno de los más conocidos es el denominado *Algoritmo K2* (Cooper y Herskovits, 1992), que fue el primero en la optimización de métricas bayesianas, utiliza un esquema voraz en la búsqueda de soluciones candidatas cada vez mejores. Parte de que las variables están ordenadas de cierta forma en que los supuestos padres de las mismas están antes que ellas mismas, lo que repercute en que el algoritmo este restringido por esta característica pero en compensación el algoritmo solo tiene que buscar un conjunto de padres de las variables predecesoras en el orden propuesto.

f) Combinación de clasificadores por el método Bagging

La combinación de clasificaciones surge para mejorar las predicciones de modelos individuales combinando los resultados de cada uno de ellos en vez de elegir una hipótesis unitaria. El método que generalmente se utiliza para la combinación de las predicciones individuales de cada modelo es la votación, en la que luego de la predicción de cada modelo se comparara los resultados y la clase más votada por los modelos es la que se le asigna a la instancia.

⁶ *TAN*. (Tree Aumented Naive Bayes)

Dentro de los métodos de combinación de modelos sobresale el denominado *Bagging*, que proviene del mecanismo denominado *bootstrap aggregation*, el cual genera subconjuntos de entrenamiento formados por m ejemplos seleccionados aleatoriamente y con reemplazamiento, con los cuales se genera los modelos aplicando un algoritmo de generación de clasificadores, al ejecutar el modelo resultante se toman los resultados de cada clasificador y se evalúa el resultado final mediante votación mayoritaria (Hernández, Ramírez y Ferri, 2004).

Capítulo 2

Agentes inteligentes

2.1 Introducción

En este segundo capítulo se hace un recorrido, sobre la tecnología de agentes, en el que se definen las características que presenta este paradigma de desarrollo de sistemas, estas características son referidas por los estudios a los agentes dentro de la Inteligencia Artificial. Además se han clasificado los grupos más importantes, en los que se puede enmarcar una tipología definida de agentes.

Se describe la estructura básica que debe contener un agente inteligente enfatizando sus elementos principales.

Se define lo que se entiende como agente de modelado de usuario, clasificándolo dentro de una tipología de agentes inteligentes,. Además de idear un bosquejo de su arquitectura básica.

Por último se incorpora el estudio de un conjunto de metodologías de desarrollo de sistemas multiagente, definiendo sus objetivos, como realizan el proceso de desarrollo del sistema y que plataforma ofrecen para el desarrollo del sistema.

Mediante los puntos tratados en este capítulo se busca definir un entorno apropiado de estudio y creación del agente de modelado de usuario ya que se incorporan los elementos más cruciales del paradigma de desarrollo de sistemas basados en agentes.

2.2 Agentes inteligentes

En el estudio de Botti (2000), se presenta un ejemplo muy claro de una entidad que se considera como un agente inteligente, esta entidad se encarga de filtrar información proveniente de la Internet de diferentes medios, adecuándola a las acciones típicas de un usuario, y aprendidas de sus comportamientos analizados por el agente. Es importante, reconocer algunos elementos que se encuentran en casi todos ellos, elementos como: el *análisis de entradas* que luego de un procesamiento de las mismas producen *acciones* que realiza el agente. En el libro de Russell y Norving (2004) se encuentra una definición muy acertada de lo que son los agentes inteligentes que menciona: “Un agente es cualquier cosa capaz de percibir su medioambiente con

la ayuda de sensores y actuar en ese medio utilizando actuadores¹". Esta definición nombra los componentes del agente que realizan las observaciones y efectúan las acciones que realiza el agente con su entorno.

Esta definición tiene un problema y es que debido a su carácter general, sumado a la repentina popularidad del tema de los agentes, muchos sistemas que no son agentes, pretenden ser tomados como tal, así que es necesario remarcar que el uso de la tecnología de agentes es sugerido cuando las técnicas comunes de resolución de problemas en entornos informáticos han fracasado.

2.2.1 Características de los agentes inteligentes

En general se ha tratado de definir un conjunto de características que tienen los agentes (en mayor o menor grado), que le sirven para la resolución de problemas en diferentes dominios de aplicación. Autores como Franklin y Gresser (1996) y Nwana (1996) definieron las siguientes:

- ◆ *Autonomía.*- Un agente inteligente tiene autonomía si es capaz de actuar por su propia cuenta basándose en su experiencia
- ◆ *Racionalidad.*- El agente siempre realiza *lo correcto*² a partir de las percepciones que recibe del entorno.
- ◆ *Reactividad.*- La reactividad hace que el agente actúe en función de los sucesos que se producen en su entorno.
- ◆ *Proactividad.*- La proactividad hace que el agente controle los objetivos a cumplir a pesar de los cambios en su entorno.
- ◆ *Sociabilidad.*- Esta característica permite al agente comunicarse con otros agentes, sistemas o usuarios, para compartir información
- ◆ *Cooperación.*- Para que el agente trabaje con otros en el desarrollo de tareas de mayor complejidad (base para el trabajo de los sistemas multiagente)

¹ *Actuadores.* Se usa este término para indicar el elemento que reacciona a un estímulo realizando una acción.

² *"Lo Correcto" en términos de agentes.* Realizar lo correcto depende de la medida de rendimiento, el entorno y los sensores y actuadores del agente. Para lo cual Russell y Norvig (2004) sostienen "En cada posible secuencia de percepciones, un agente racional deberá emprender aquella acción que supuestamente maximice su medida de rendimiento, basándose en las evidencias aportadas por la secuencia de percepciones y en el conocimiento que el agente mantiene almacenado".

- ◆ *Movilidad*.- Característica que puede tener un agente para trasladarse de ambiente a otro para reunirse con uno u otros agentes, o para recopilar y analizar nuevos datos y entornos.
- ◆ *Adaptabilidad*.- Mediante el aprendizaje que el agente desarrolla a lo largo de su trabajo, este es capaz de cambiar su comportamiento basándose en este aprendizaje.

2.2.2 Clasificación de los agentes inteligentes

Franklin y Gresser (1996) presentan en su taxonomía sobre agentes autónomos un conjunto de clasificaciones de las cuales podemos mencionar los siguientes:

- ◆ *Agentes Móviles y de Aprendizaje*.- que nace de la adición de la característica de aprendizaje a los agentes móviles por ende se los puede incluir dentro de ellos.
- ◆ *Agentes de Software*.- que pueden ser clasificados de acuerdo a las tareas que realizan, entonces pueden existir por ejemplo agentes para el filtrado de correo electrónico o agentes de recolección de información. También pueden ser clasificados de acuerdo a su arquitectura de control.
- ◆ *Grado de Sensibilidad*.- los agentes pueden también clasificarse por el grado de sensibilidad de sus sensores, por el rango de efectividad de sus acciones o por la cantidad de estados internos que posee.
- ◆ *Taxonomía de Brustoloni (1991)*.- Este autor incluyó tres formas de clasificación.
 - ◆ *Agentes de Regulación*.- Reacciona a cada sensación de entrada según como esta llega al agente, y siempre sabe cómo reaccionar a cada entrada.
 - ◆ *Agentes de Planeación*.- Utilizan planes para resolver problemas (*agentes de resolución de problemas*), o usan el paradigma de la condición-acción (*agentes basados en la condición-acción*), o también usan operaciones de descubrimiento, o utilizan algoritmos aleatorios (*agentes aleatorios*).
 - ◆ *Agentes Adaptativos*.- Estos agentes no solo planean, sino que aprenden.

Entre estos se puede enmarcar a los *agentes adaptativos de resolución de problemas*.

Además de las clasificaciones descritas anteriormente en la figura 2.1 se presenta la taxonomía generada por Franklin y Gresser (1996), como curiosidad se incluye dentro de este estudio a los virus como agentes ya que estos, cumplen con algunas características como son la autonomía, reactividad, movilidad, etc., que los han hecho meritorios a ser considerados como tales. Como se ve esta taxonomía ubica a todos los agentes que se puedan construir y enmarca sobre todo a los agentes de software y desglosa sus derivaciones.

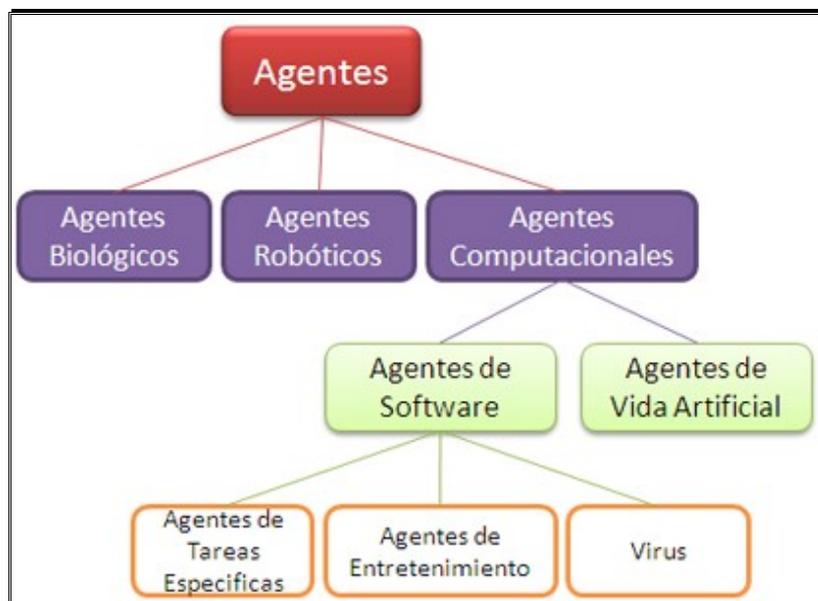


Figura 2.1: Taxonomía de agentes de Franklin y Gresser (1996)

Finalmente incluimos la visión de la clasificación de agentes generada por Nwana (1996) que considera algunas dimensiones para generar las tipologías de agentes que nacen de la combinación de estas dimensiones:

- ◆ Por la movilidad que presentan
- ◆ Reactivos o deliberativos
- ◆ Por sus atributos (ideales o primarios) que pueda mostrar
- ◆ Por sus roles

En resumen este autor presenta un conjunto de tipologías de agentes de software

que incluyen todas las dimensiones mencionadas, esta clasificación se presenta en la figura 2.2.

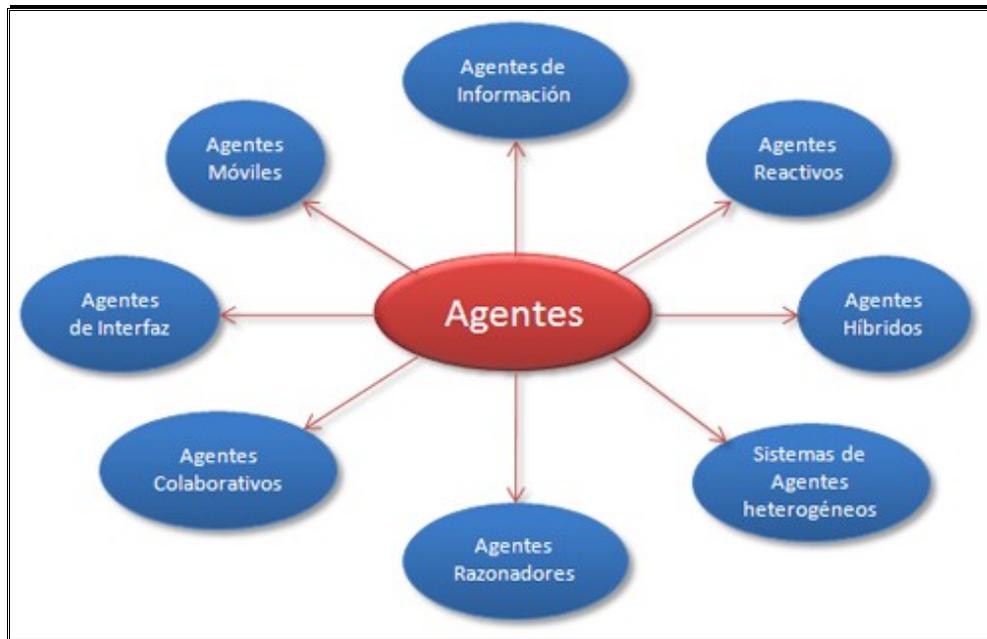


Figura 2.2: Clasificación de agentes de software de Nwana (1996)

2.3 Estructura de los agentes inteligentes

Lo que se busca en esta sección es, describir la estructura básica de los agentes inteligentes así como los componentes que estos necesitan para cumplir con su trabajo adecuadamente.

Según Russell y Norving (2004) Un agente contiene dos elementos estructurales principales, uno es el *programa del agente*, que es sujeto de estudio principal de la Inteligencia Artificial, y es este programa, quien implementa la función del agente y es a su vez quien interpreta las percepciones y genera las acciones que desencadena el agente. El otro componente de la estructura básica de un agente es el medio físico en que trabaja este programa, es él quien incorpora los sensores físicos y actuadores (si los hubiese), a este medio se lo conoce como *Arquitectura*. Como consideración importante se menciona que el programa del agente, así como la arquitectura en la que se sustenta dicho software deben ser complementarios, ya que de no concordar no se podrían cumplir las acciones que desarrolla el agente.

Dentro de las diferentes estructuras de agentes que presentan estos autores se destaca la tipología de *Agentes que aprenden*, o más bien conocidos como Agentes Inteligentes. La idea principal que incorpora esta tipología de agentes es que estos continúen aprendiendo después de enseñarles. Esto incorpora nuevas ventajas, como que el agente actúe en medios inicialmente desconocidos con mayor efectividad que si solo contara con un conocimiento inicial.

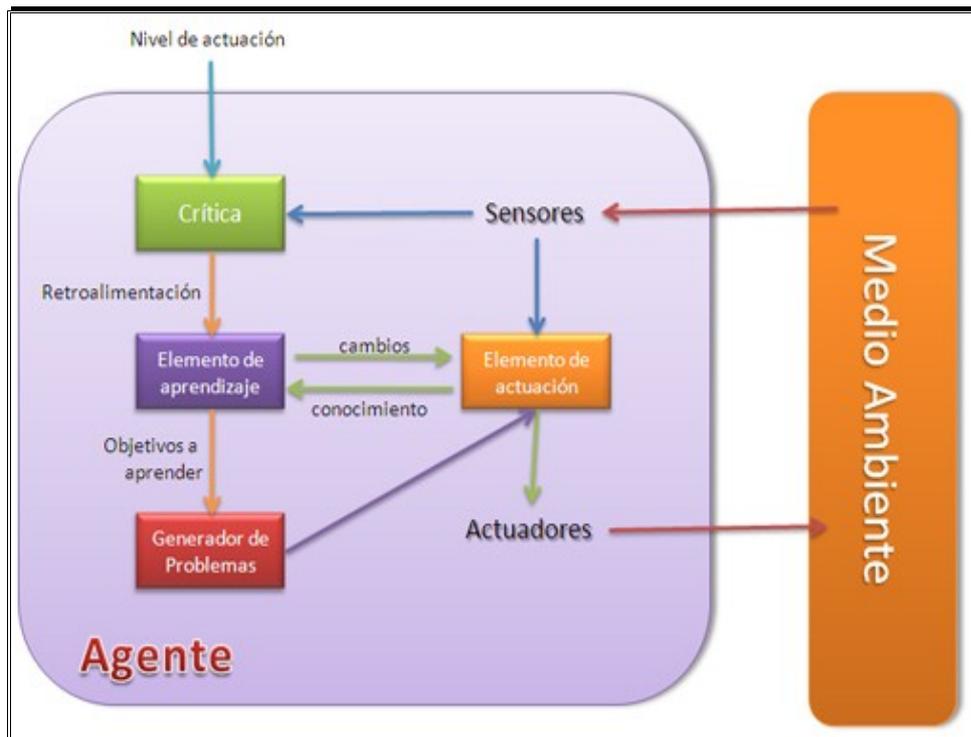


Figura 2.3: Modelo general para agentes que aprenden (Russell y Norving 2004).

En la figura 2.3 se presenta un modelo de trabajo para este tipo de agentes, destacan en él, *el elemento de aprendizaje* y *el elemento de actuación*, estos se encargan de lo siguiente: el primero, de las mejoras, es decir, de aprender y modificar el comportamiento del agente, estudiando el entorno del agente. Y el segundo, de elegir las acciones que el agente devuelve al entorno. Se consideran destacados estos componentes en esta tipología de agentes, porque son los que le dan la naturaleza de inteligente. Los demás elementos, hablando de la *crítica* y el *generador de problemas*, son definidos: el primero, como aquel que mide la actuación del agente en cada una de las acciones tomadas, es decir, por medio de este mecanismo el agente sabe que las acciones que está ofreciendo al entorno son satisfactorias; el generador de problemas,

en cambio se encarga de la generación de casos en los cuales el agente pueda obtener nuevos conocimientos, ya que este puede llegar a realizar solo las acciones que considera correctas siempre, y mediante la incorporación de este mecanismo el agente es inducido a experimentar con acciones que, a largo plazo, pueden llegar a ser correctas y le brinden nuevas alternativas a sus respuestas.

2.4 Agente inteligente de modelado de usuario

En la sección anterior se ha descrito el origen de los agentes inteligentes su utilidad, las características que estos podrían incorporar así como los puntos de vista para la clasificación de los mismos, esto ha ayudado a delimitar el entorno en que los agentes inteligentes enfocan su objetivo de trabajo. En el Capítulo 1 se hizo un análisis exhaustivo sobre el modelado de usuarios, sus implicaciones, elementos, características y alcance. En esta sección se enmarca el entorno de trabajo de los agentes de modelado de usuario, así como sus principales funciones, lo que creara un marco teórico para la posterior definición del agente de modelado de usuario para el EVA de la UTPL.

Primero, se debe delimitar ¿Qué es un agente de modelado de Usuario? Desde mi punto de vista un agente de modelado de usuario es aquel que después de un proceso de prototipado del usuario, forma un modelo de usuario, que luego de la observación de la interacción de dicho usuario con el sistema refina dicho modelo. Esta definición aclara la función principal de este agente e intenta definir en cierto modo los atributos que este puede incorporar.

Segundo, se debe incorporar al agente de modelado de usuario en una tipología, para poder definir en base a ello sus características esenciales. Sabemos por definición que el agente de modelado de usuario pertenece a la clasificación de agentes de software o softbots. Dentro de la taxonomía de Nwana (1996) la tipología de agentes de software que más concordancia presenta con los atributos que debe implementar este agente es la tipología de *agentes de interfaz* la cual sustenta que el agente de este tipo debe monitorear la interacción de los usuarios con un sistema y de esta interacción aprende el comportamiento del usuario, este tipo de agentes puede, luego de esto, imitar los comportamientos del usuario o anticiparse a las acciones del mismo

en el sistema.

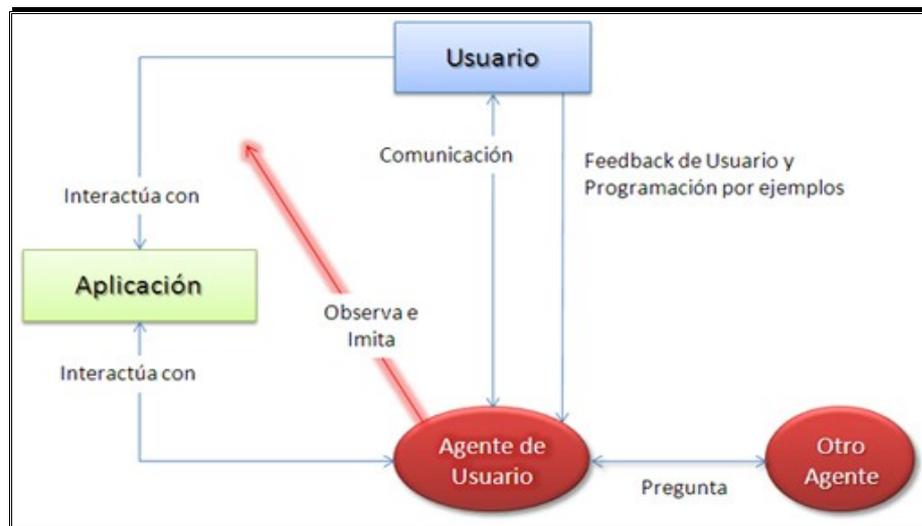


Figura 2.4: Como trabajan los agentes de interfaz. Imagen adaptada de Maes (1994) citada por Nwana (1996)

En la figura 2.4 se muestra el funcionamiento típico de los agentes de interfaz, o agentes de usuario como se los denomina en este esquema, se identifica entonces los sensores y actuadores que afectan a todos los actores dentro de esta interacción. Debe notarse que el agente observa la interacción del usuario con el sistema y de esta observación el agente se alimenta de conocimiento que le permite imitar el comportamiento del usuario con el sistema, esto le permite poder realizar sus experimentaciones y de esta forma prever las acciones a realizar por el usuario o formalizar los creencias que tiene de los mismos. Para realizar esta interacción con el sistema y el usuario, se muestra en la gráfica las relaciones que esta tipología de agentes generalmente implementa con estos actores. Además se contempla la comunicación con otros agentes para obtener datos que no son observables del sistema o de los mismos usuarios.

2.5 Metodologías de desarrollo de agentes

La ingeniería de software a lo largo de su existencia como rama de las ciencias informáticas, está sujeta a evoluciones constantes, debido principalmente al apareamiento de nuevas tecnológicas de tratamiento de información, esta innovación siempre tiene que ir de la mano de procesos de que estandaricen su construcción para

que los productos (sistemas) resultantes gocen de la credibilidad necesaria y la modificación o actualización de dichos productos sea satisfactoria y no presente dificultades.

Con el desarrollo de la tecnología de agentes inteligentes se incorpora nuevos requerimientos para la construcción de los denominados *Sistemas Multiagente* (SMA), cuyo diseño principalmente conlleva un cambio de concepción de la ingeniería de software ya que en vez de modelar componentes que ejecutan métodos, el diseñador de sistemas tiene que pensar en los objetivos que estos componentes tienen que alcanzar y en las tareas para que se puedan cumplir estos objetivos (Bratman, 1987).

Según Gómez (2003) las *Metodologías Basadas en Agentes* nacen para aplacar las dificultades que tienen las metodologías tradicionales en la construcción de SMA. Aunque se ha avanzado mucho en la integración de este tipo de metodologías con las ya existentes, existen vacíos considerables, debido esencialmente a la falta de herramientas de soporte que dirijan el progreso en el trabajo de este tipo de sistemas.

En el estudio de Gómez (2003) se incluye la comparación de siete metodologías orientadas a agentes las cuales obedeciendo a tres criterios de evaluación, se tomaron en cuenta para describir su forma de trabajo y elementos que las constituyen. Los criterios que se evaluarán para cada metodología serán los siguientes

- ◆ Utilización de vistas para especificar el sistema.
- ◆ Incorporación de un proceso de desarrollo
- ◆ Integración de técnicas de ingeniería y teoría de agentes.

A continuación se presenta una descripción de las metodológicas de desarrollo de SMA revisadas en este proyecto:

2.5.1 Vowell Engineering

Esta metodología centra su trabajo en la concentración de aspectos identificados por cuatro vocales: A de agentes, E de entorno, I de interacciones, O de organización. Para cada uno de los aspectos propuestos es necesaria la utilización de métodos y

técnicas específicos (Demazeau, 1995).

Propósito

El propósito de esta metodología es la construcción de librerías de componentes que den soluciones de diseño a cada uno de estos aspectos, posteriormente en otra etapa el diseñador seleccionara un modelo de agente, un modelo del entorno, un modelo de interacciones, y un modelo de la organización a instanciar.

Proceso de desarrollo

El proceso de desarrollo consiste ordenar en cierto orden los elementos que propone esta metodología para así dar mayor o menor importancia a cada uno de ellos. El orden se decide en función del tipo de sistema que queramos tener. Por ejemplo, si se empieza por la vocal O, se tendrá un sistema en el que las relaciones sociales son lo más importante. Si se empieza por la A, se tendrá un sistema en el que probablemente la organización surja como resultado de la interacción de agentes aislados.

Plataforma de desarrollo

En estudios recientes se propone el desarrollo de Agentes usando esta metodología implementada en la plataforma *Volcano* (Ricordel, 2001), la cual utiliza como método de trabajo el ensamblaje de componentes que pertenecen a una categoría concreta de las cuatro consideradas por la metodología. Para esta metodología existe un entorno de desarrollo, pero no es de acceso público. Sin embargo el trabajo desarrollado para esta metodología es importante ya que abarca algunos dominios de aplicación.

2.5.2 Mas-CommonKADS

Surge como extensión de la metodología CommonKADS utilizada para el desarrollo de sistemas basados en conocimiento. Maneja la idea central de metodologías orientadas a objetos, en la construcción de SMA. Implementa el concepto del modelo de experiencia y extiende la metodología CommonKADS para tener la posibilidad de que dos componentes del sistema interactúen entre sí (Tansley y Hayball, 1993).

Propósito

Integrar las el desarrollo de SMA con el ciclo de vida del software, para ello propone siete modelos que cubren los diferentes aspectos del desarrollo de SMA, y estos son: agente (AM), tareas (TM), experiencia (EP), coordinación (CoM), comunicación (CM), organización (OM) y diseño (DM).

Proceso de desarrollo

Esta metodología es la primera en incorporar un proceso de ingeniería completo para la implementación de SMA al estilo de Pressman (2006), definiendo detalladamente el proceso de creación del sistema y sus componentes y teniendo en cuenta las dependencias entre los modelos.

La aplicación de esta metodología consiste en construir cada uno de los distintos modelos. Las fases que incorpora esta metodología son las siguientes:

- ◆ *Conceptualización.*- Se obtiene una descripción preliminar del problema y se describen casos de uso que puedan ayudar a obtener requerimientos de información, y a probar el sistema.
- ◆ *Análisis.*- Se determinan los requerimientos del sistema iniciando por establecer el problema.
- ◆ *Diseño.*- Se determina el modelo del diseño pasando de los requerimientos del análisis. También se definen la red multiagente global, y cada agente individual.
- ◆ *Codificación y Pruebas de cada Agente.*- En donde se construyen los diseños y se generan los planes de pruebas y se las ejecuta.
- ◆ *Integración.*- Se prueba el sistema completo.
- ◆ *Operación y Mantenimiento.*- En el cual se somete al trabajo para el que fue creado el sistema.

Plataforma de desarrollo

Existen herramientas para el desarrollo con MAS-CommonKADS pero se limitan a ser frameworks y arquitecturas para el desarrollo de agentes, pero no existen herramientas de apoyo al diseño y el análisis de SMA dentro de esta metodología, lo

que dificulta alcanzar el nivel de detalle que esta exige en la documentación que acompaña al desarrollo, que se basa esencialmente en lenguaje natural lo que resulta costoso de aplicación ya que toda la documentación se genera manualmente. Sin embargo esta metodología goza de gran aceptación debido a su grado de similitud con las metodologías tradicionales de desarrollo de software y por su rigurosidad a la hora de definir los componentes del sistema.

2.5.3 BDI

Conocida así por sus siglas en ingles *Beliefs - Desires - Intentions* (Creencias - Deseos – Intenciones), donde las *Creencias* se refieren esencialmente al conocimiento del agente de su entorno, los *Deseos* se derivan de las creencias y corresponden a hechos que se cumplirán en estados futuros, las *Intenciones* se ilustran como los objetivos que busca el agente en el momento presente (Kinny, Georgeff y Rao, 1997).

Propósito

Especificar el SMA a partir de un conjunto de modelos que operan en dos niveles e abstracción: *Interno* y *Externo*. Un SMA desde un punto de vista interno, se construye mediante *Modelos Internos* que permiten incorporar una estructura sobre el estado de la información y motivación de los agentes y las estructuras de control que definen su comportamiento. Desde un punto de vista externo un SMA se modela como una jerarquía de herencia de clases de agentes, de donde los agentes individuales, son instancias. Estas clases se caracterizan por su propósito, por sus responsabilidades, los servicios que ejecutan, la información del entorno que necesitan, y las interacciones externas.

Proceso de desarrollo

La integración con el proceso de vida del software es reducido. Alternativamente se proponen una serie de pasos a seguir para construir los modelos, estos pasos se repiten para generar progresivamente los modelos así como para probarlos y refinarlos. Además que la aplicación de esta metodología conlleva ciertas restricciones en lo que se refiere a la arquitectura de soporte de los modelos.

Plataforma de desarrollo

Es uno de los puntos débiles de esta metodología ya que no existen herramientas que soporten el diseño y el análisis aplicando BDI. Así que los modelos que se crean en esta metodología se deberán de generar manualmente con las implicaciones que esto conlleva.

2.5.4 MaSE

La visión de MaSE sobre los SMA es que los agentes son una especialización de los objetos, esta especialización se basa en que los agentes conversan unos con otros para coordinar acciones en conjunto para alcanzar metas en forma individual y para el sistema al que pertenecen (DeLoach 2001). Su nombre proviene de las siglas en inglés (Multi-agent Systems Software Engineering).

Propósito

El propósito de MaSE es generar una serie de modelos para describir los tipos de agentes en un sistema y sus interfaces con otros agentes, independientemente de la arquitectura interna de cada uno de ellos. Para ello sus autores se han enfocado en la creación de técnicas, y herramientas para la construcción de sistemas basados en agentes.

Proceso de desarrollo

El proceso de desarrollo en MaSE sigue un conjunto de pasos los cuales se controlan casi en su totalidad dentro de la herramienta que lo acompaña denominada *AgentTool*. El proceso de desarrollo en esta metodología consta de dos fases principales: El análisis y el diseño. En la fase de análisis se incluyen tres pasos: obtención de objetivos, aplicación de casos de uso y la redefinición de roles. En la fase de diseño se incluyen cuatro pasos: creación de las clases de agentes, construcción de las conversaciones, ensamblaje de las clases de agentes y el diseño del sistema.

Plataforma de desarrollo

La plataforma de desarrollo es un punto fuerte de MaSE, ya que posee, su propia herramienta de soporte que se denomina *AgentTool*, que permite la generación de código desde la especificación del sistema, ósea del conjunto de diagramas que son el resultado final de la aplicación de esta metodología. Además se incluye dentro de la herramienta utilidades para el diseño de los protocolos que utilizan los agentes.

2.5.5 ZEUS

Zeus al igual que MaSE ofrece una metodología y una herramienta de soporte en el proceso de creación de SMA. Zeus ha sido punto de referencia de como deberían ser las herramientas de soporte para la construcción de SMA. Ya que incorpora en una sola aplicación ejecutable lo nuevos paradigmas de la construcción de agentes, además este conjunto de herramientas incluye funciones para la inicialización y el monitoreo del sistema.

Propósito

El propósito de la metodología Zeus es el permitir a los ingenieros de software, el rápido desarrollo de nuevas aplicaciones multiagente mediante la abstracción, dentro de un conjunto de herramientas, de los principios comunes y los componentes de un proceso de desarrollo de SMA, la idea central que se maneja en este desarrollo es crear un conjunto de herramientas modificables de propósito general, enmarcadas en la colaboración de los agentes, y que ingenieros con competencias básicas en la tecnología de agentes puedan crear SMA funcionales. Entre los elementos que se tienen que definir dentro de la metodología Zeus encontramos: Planificación de los agentes, definición de ontologías, asignación de responsabilidades y relaciones sociales entre agentes (Nwana, Ndumu, Lee y Collis, 1999).

Proceso de desarrollo

En el proceso de desarrollo con esta metodología genera cuatro resultados visibles:

- ◆ *Análisis del Dominio.*- Para el análisis del dominio esencialmente lo que se trata de obtener es un modelo de roles, compuesto por diagramas de clases

UML para identificar roles, así como de diagramas UML de colaboración, para identificar los mensajes con los que se comunican los agentes y por ultimo fichas para describir los mencionados roles.

- ◆ *Diseño del Agente.-* En este entregable lo que se busca es identificar las necesidades de los agentes para realizar su labor. Revisando tecnologías y disciplinas relacionadas en el diseño de agentes.
- ◆ *Implementación de los Agentes.-* Utilizando la herramienta de Zeus en la implementación de los diseños. Como resultado se obtiene un modelo ejecutable del sistema.
- ◆ *Soporte en Tiempo de Ejecución.-* Además de implementar el sistema la herramienta de soporte al diseño e implementación de SMA de Zeus. Incorpora funciones para el control de la ejecución del sistema, de la comunicación y estadísticas de todo el proceso.

Plataforma de desarrollo

Como se menciona anteriormente la plataforma de desarrollo es uno de los puntos fuertes de Zeus ya que esta soporta la creación del sistema en muchas de las fases de la metodología. La opciones que presenta dicha herramienta son diversas, pero se extraña un proceso de desarrollo más específico y no solo la creación de roles, ya que la metodología plantea más paradigmas de implementación.

2.5.6 GAIA

Al aparecimiento del nuevo paradigma del desarrollo de sistemas basados en agentes aparecen nuevas formas de soportar el proceso de desarrollo de los mismos, Gaia es una metodología pensada para ofrecer técnicas de desarrollo de software específicas para SMA. Ya que las técnicas de desarrollo orientadas a objetos resultan inadecuadas para el análisis de este tipo de sistemas (Wooldridge, Jennings y Kinny, 2000).

Propósito

El propósito esencial de Gaia es la obtención de sistemas capaces de maximizar

una medida de calidad global, para ello se guía al analista del proyecto en la transición desde unos requisitos iniciales a un diseño detallado, que puede ser implementado directamente. El análisis debe ser una tarea que lleve a la comprensión global del sistema y de su estructura, sin rozar los puntos de implementación, y esto en Gaia se consigue tomando como referencia una organización de roles los cuales se relacionan de cierto modo con otros e incorporan cuatro aspectos importantes: responsabilidades del rol, los recursos que este puede utilizar, las tareas asociadas al rol y las interacciones que este incluye.

Proceso de desarrollo

El proceso que propone Gaia para el desarrollo de SMA inicialmente es de alto nivel, e incluye tres fases fundamentales: el análisis, el diseño de la arquitectura, y el diseño en detalle. En este análisis (diseño de la arquitectura) se aplican dos modelos: *el modelo de roles*, donde se identifican los roles principales en el sistema y sus propiedades definitorias, y *el modelo de interacciones*, que define las referencias ligadas a un modelo la implementado. Luego se pasa a una etapa de diseño en alto nivel (diseño a detalle), en donde se generan tres modelos: *el modelo de agentes*, donde se enmarcan los tipos existentes de agentes, cuantas instancias de cada tipo se distinguen y la función que estos desempeñan en el sistema, *el modelo de servicios*, que identifica las funciones del agente asociadas a cada rol, y *el modelo de conocidos*, que define las comunicaciones que realizan los agentes. El resto del proceso se desarrolla de forma similar a las técnicas clásicas de desarrollos orientados a objetos.

Plataforma de desarrollo

Gaia es una de las metodologías que no ofrece herramientas de soporte al proceso de diseño e implementación del sistema, por lo que es necesario que para su aplicación los ingenieros de software estén consientes de las carencias de esta metodología.

2.5.7 INGENIAS

Una metodología, relativamente nueva, que nace de los principios y el trabajo realizado en la metodología MESSAGE, es INGENIAS, donde se evolucionan y

actualizan las ideas propuestas anteriormente profundizando el proceso de desarrollo e incorporando herramientas que soporten dicho proceso (Gómez y Fuentes, 2002).

Propósito

El propósito de Ingenias es definir una metodología para el desarrollo de SMA, integrando los resultados de las investigaciones en el campo del paradigma de desarrollo de agentes, con un proceso de desarrollo de software bien definido como es el caso de Rational Unified Process (RUP). Está basada en la definición de meta-modelos³ que describen los elementos que forman un SMA de diferentes puntos de vista, lo que permite poder definir una especificación del lenguaje para SMA. Los puntos de vista que toma esta metodología son cinco: agentes, interacciones, organización, entorno y objetivos-tareas (Botía, González, Gómez y Pavón, 2008).

Proceso de desarrollo

Los meta-modelos, se construyen mediante un lenguaje de meta-modelado, GOPRR (Graph, Object, Property, Relationship and Role), integrando las practicas conocidas de ingeniería de software, mediante un conjunto de actividades que guían las fases de análisis y diseño. Estos meta-modelos generan un conjunto de diagramas, o modelos, de la misma forma que los que define UML, especializados para definir SMA. Las actividades que incorpora esta metodología ayudan al desarrollo de los modelos y las relaciones entre los mismos. Como vemos Ingenias presenta un proceso que incorpora muchos de los elementos de metodologías tradicionales de desarrollo de sistemas, adaptados al paradigma de desarrollo de SMA. Es importante mencionar que esta metodología soporta, mediante el uso de herramientas especializadas, todo el proceso de diseño y análisis del SMA.

Plataforma de desarrollo

INGENIAS incorpora herramientas que soportan el proceso de análisis y diseño de SMA, este conjunto de herramientas se conoce como *INGENIAS Development Kit* (IDK) (Gómez y Pavón, 2004) Que incorpora funciones gráficas para la creación de los meta-modelos y de los modelos explícitos, además incluye funciones para la

³ *Meta-Modelo*. Descripción de alto nivel de qué elementos componen a un modelo (Gómez, 2003)

verificación y validación, así como motores para de transformación e implementación de código. Estas funciones se incorporan a la herramienta por medio de módulos.

Parte II: DISEÑO
Y
CONSTRUCCIÓN
DEL AGENTE DE
MODELADO DE
USUARIO

Capítulo 3

*Descripción del proceso y
Selección de herramientas y
técnicas*

3.1 Introducción

En este capítulo se busca describir, en primera instancia, el proceso a seguir para el diseño del modelo de usuario a aplicar a los estudiantes de la UTPL, tomando en cuenta los avances que se han logrado en la aplicación de esta tecnología en sistemas de apoyo al aprendizaje.

En segunda instancia, en este capítulo se desarrolla una selección de las herramientas y técnicas que se van a utilizar a lo largo del desarrollo de esta tesis, tratando en la medida de lo posible de incluir las técnicas y herramientas de mayor incidencia dentro de su campo de acción y que además contengan las mejores prestaciones, para que su utilización genere los resultados previstos y consecuentemente se puedan inferir conclusiones precisas de dichos resultados.

Se prevé la utilización de una metodología de desarrollo de agentes, en este capítulo se contempla la selección de esa metodología para su posterior seguimiento.

3.2 Descripción del proceso de construcción del modelo de usuario

La obtención del modelo de usuario considerando los datos de los cursos que siguen los estudiantes de la UTPL es el eje central de este proyecto, de esto se desprende la importancia en que el proceso de su construcción sea el más adecuado, por ello, en esta sección se describe este proceso incorporando las actividades con mayor relevancia y que aseguren que el modelo que se va a proponer sea el que mejor se acople a la población estudiantil de la UTPL.

3.2.1 Actividades de construcción del modelo de usuario

Para diseñar el modelo de usuario de los estudiantes de la UTPL se contemplan un conjunto de actividades fundamentadas en el proceso de construcción de modelos de usuario, descrito en la sección 1.5 del Capítulo 1 de esta tesis, en ella se contemplaron dos formas de construir modelos de usuario, la primera mediante la generación de estereotipos (asunciones sobre los usuarios) y la segunda mediante aprendizaje automático, también se demostró que se puede utilizar una combinación

de las dos técnicas para generar el modelo y es la elección que se tomara en cuenta para el desarrollo de proceso de modelado del agente. Además se considera para este proceso el conjunto de requisitos descritos en la sección 1.6 y la inclusión de la teoría de estilos de aprendizaje descrita en la sección 1.8 del Capítulo 1, ya que los requerimientos en gran medida definen las propiedades que brinda el modelo y la teoría de estilos de aprendizaje, que se incorporara al modelo de usuario, para poder definir cómo se los obtendrá dentro del modelo.

En los pasos para la construcción de modelos de usuario se incluyeron cuatro fases, para cada una las cuales definimos las actividades para llegar a su ejecución estas se describen a continuación:

- **Recolección de datos**

- a) *Selección de atributos que conforman el modelo.*- Esta actividad consiste en elegir de un conjunto de datos disponibles dentro del EVA UTPL aquellos que sean necesarios para la generación de los atributos que va a contener el modelo de usuario además se debe contemplar los mecanismos de extracción de los mismos.
- b) *Minería de datos del EVA UTPL.*- Luego de escoger los atributos que se va a considerar dentro del modelo y los datos que los van a inferir, se procede con la selección de una muestra de estudiantes, lo que permitirá reconocer tendencias dentro de los datos y esencialmente inferir conocimiento de esos datos y nos ayuden a corroborar las definiciones de los atributos y sus valores.

- **Diseño del modelo**

- a) *Diseño de la estructura del modelo de usuario.*- En esta actividad formalizamos definitivamente los componentes del modelo.
- b) *Diseño de mecanismos de representación del modelo.*- Con esta actividad se busca definir cómo se van a representar los modelos, y en qué medios se van a almacenar, tomando en cuenta el proceso de construcción y actualización de los modelos.
- c) *Diseño de tareas de aprendizaje.*- Las tareas de aprendizaje son los medios por los cuales se van a generar los atributos del modelo de usuario así que

esta actividad es muy importante ya que en ella se diseñan los mecanismos implícitos y explícitos que infieren los atributos del modelo de usuario. Además se define como las técnicas de aprendizaje supervisado colaboran entre sí para construir dichos atributos

- **Construcción del modelo**

- a) *Definición de estereotipos.*- La actividad de definición de estereotipos se deriva directamente de las actividades de recolección de información y de diseño del agente, en las cuales se adquiere el conocimiento suficiente para definir un conjunto de elementos genéricos que puedan servir para encajar inicialmente a un estudiante en uno de los estereotipos que se han de definir en esta actividad.
- b) *Implementación de mecanismos de representación.*- Ya diseñados los mecanismos de representación se contempla la implementación de los mismos mediante un lenguaje de programación. Brindando medios que faciliten la construcción y actualización de los modelos.
- c) *Implementación de tareas de aprendizaje.*- Otra actividad que conlleva la implementación de software es la que construye las tareas de aprendizaje en donde se contempla la incorporación de la combinación de clasificadores (en cuanto a los atributos que se obtienen por aprendizaje automático).

- **Validación del modelo**

- a) *Evaluación de técnicas de construcción y actualización del modelo.*- En esta actividad se busca evaluar que tan efectivas son las técnicas elegidas para la construcción y actualización del modelo de usuario. Para lo cual se contempla la evaluación del error de cada una de ellas mediante herramientas de software que se seleccionaran posteriormente en este capítulo.
- b) *Evaluación de los modelos de prueba construidos.*- En vista de que para la evaluación de las técnicas de construcción y actualización se tienen que definir muestras de estudiantes para probar las dichas técnicas que inferirán modelos que también deben ser evaluados para comprobar si el

proceso de modelado propuesto define modelos utilitarios para el sistema.

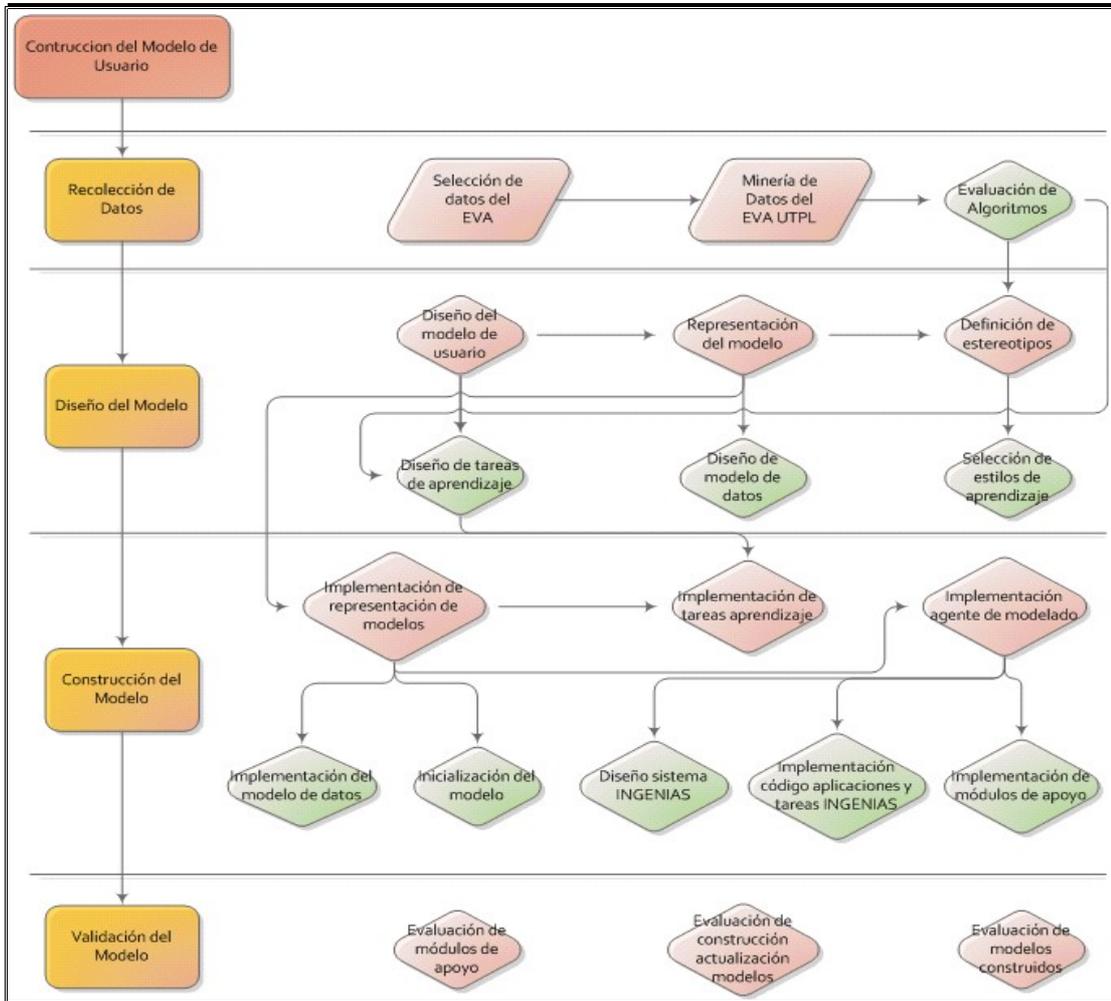


Figura 3.1: Actividades para la diseño e implementación del agente de modelado de usuario

En la figura 3.1 se gráfica el flujo de las actividades, como se ve las actividades propuestas contemplan todo el proceso de construcción del modelo de usuario, el detalle específico de cada una de estas actividades se describe paso a paso en su ejecución, estas actividades son fundamentales ya que algunas dependen directamente de otras así que es prácticamente indispensable que se realicen cada una de ellas y en el proceso de su ejecución pueden aparecer nuevas actividades que contemplar lo que conllevara ejecutar procesos de respaldo que soporten a estas nuevas actividades.

3.3 Selección de herramientas y técnicas

Luego de definir abiertamente el proceso a seguir para la construcción del modelo

de estudiantes de la UTPL se busca en esta sección identificar las técnicas y herramientas que acompañaran durante todo este proceso de construcción del modelo de usuario así como las técnicas y herramientas que apoyan el proceso de desarrollo del agente de modelado de usuario.

Esta selección debe fundamentarse en estudios científicos que abalen su valía dentro del desarrollo de las actividades propuestas. Además se presentan comparativas de herramientas y técnicas que sirven para escoger las más opcionadas para formar parte del trabajo de esta tesis.

3.3.1 Selección del tipo de modelo de usuario

En el análisis presentado en la tesis de grado, “Educación Adaptativa en la Web: Estado del Arte”, por Valdiviezo, Jara y Pesantez (2009), se determinó que la clasificación que más se ajusta a las necesidades de la UTPL para su idea de un entorno adaptativo de aprendizaje es aquella que se basa en la actualización del modelo y de los modelos propuestos por esta clasificación. Además se concluyó que el modelo que más se ajusta a las pretensiones de la UTPL fue el modelo híbrido, del cual se desarrollo una descripción en la sección 1.3 del Capítulo 1.

3.3.2 Selección de estilos de aprendizaje

La identificación de los estilos de aprendizaje de los estudiantes es una parte importante dentro de los atributos a considerar para el modelo de usuario, existen varias clasificaciones que reflejan el trabajo de muchos autores y a su vez existen un conjunto de técnicas para su obtención en las siguientes sub-secciones se desarrolla una evaluación y selección de las mismas.

a) Técnicas de identificación de estilos de aprendizaje (análisis comparativo)

En el Capítulo 1 sección 1.8 , se desarrolló una introducción a los estilos de aprendizaje en el ámbito educativo, y se presentó los intentos de involucrar los métodos de su obtención en los diferentes campos que la educación ha incursionado. Para ello en su inicios se ha contemplado la obtención de los mismos mediante test

escritos¹, ya que estos son la alternativa más factible que los investigadores encontraron, en su momento, para identificar en las personas los estilos de aprendizaje. Para cada propuesta de clasificación que los especialistas desarrollaron se creó un tipo de test específico, por lo general estos test constan de un conjunto de preguntas que se encuentran divididas en grupos para cada estilo de aprendizaje.

El método de calificación se basa en que cada pregunta que existe en los grupos creados en el test, se crean opciones para comprobar la pertenencia al estilo de aprendizaje (como las preguntas del test de Honey-Alonso) o también se han creado opciones en las que la pertinencia puede ser, en cada caso, a una dicótoma específica (como las preguntas de test de Felder y Silverman), luego de contestadas todas las preguntas y la tabulación de las mismas y el veredicto final de la pertenencia de la persona evaluada a un estilo de aprendizaje obedece a el mayor número de aciertos para el grupo de preguntas que definen un estilo específico.

En la actualidad se han iniciado investigaciones para incluir estilos de aprendizaje en el desarrollo de cursos en línea, en estudios como el de Baldiris, Santos, Boticario y Fabregat (2007) o de Arias, Ovalle y Cadavid (2008) se utilizó la teoría de estilos de aprendizaje y se ha automatizado el proceso de obtención de los mismos en los estudiantes, codificando el test desarrollado por Felder y Silverman y luego interpretando los resultados obtenidos por los estudiantes para que los contenidos de los cursos se adapten a el estilo de aprendizaje de cada uno de los educandos. En otra investigación desarrollada por Durán y Costaguta (2007) se tomó el test de Felder y Silverman a los estudiantes y luego se almacenaron los resultados en una base de datos para luego someterlos a un proceso de KDD² para interpretar los datos obtenidos en un solo conjunto. Finalmente en la investigación desarrollada por Rubio y otros (2008) se implementó un módulo que realiza el proceso de toma de test de estilos de aprendizaje en la plataforma Moodle, este módulo implementa por defecto el test de estilos de aprendizaje desarrollado por Honey y Alonso, pero tiene la posibilidad de expandir su funcionamiento mediante la incorporación de nuevos test mediante la carga de archivos XML, además incorpora algunas herramientas de

¹ *Test escritos*. Test que generalmente se imprimen y se toman directamente a los aplicados de los mismos.

² *KDD*. Knowledge Discovery in Databases, o Proceso de descubrimiento de conocimiento en bases de datos relacionales.

administración para la posterior interpretación de los resultados obtenidos en los test.

Como vemos hay pocas alternativas a la hora en encontrar técnicas de obtención de estilos de aprendizaje en sistemas orientados a la educación, así que de los existentes se presenta a continuación, en la tabla 3.1, una comparativa de estas técnicas que servirá, para elegir la más apropiada para implementarla en el Agente de Modelado de Usuario.

Tabla 3.1: Comparativa de técnicas de identificación de estilos de aprendizaje

	Test escritos	Test en línea	Módulos de test
Facilidad de aplicación	Fácil	Media. Se necesita de conocimientos informáticos	Media. Se necesita de conocimientos informáticos
Facilidad de tabulación	Retardada. Se debe contar una a una las respuestas al test	Inmediata. Se realiza automáticamente	Inmediata. Se realiza automáticamente
Entrega de resultados	Retardada. Se realiza luego de la interpretación de las respuestas al test	Inmediata. La calificación es instantánea lo que significa resultados al instante	Inmediata. La calificación es instantánea lo que significa resultados al instante
Integración con plataformas de aprendizaje	Retardada. Hay que introducir los resultados en la base de datos de la plataforma	Retardada. Los resultados obtenidos se deben pasar a la base de datos de la plataforma	Inmediata. Ya que el modulo es parte de la plataforma.
Facilidad de obtención de estadísticas de estilos de aprendizaje	Muy Difícil. Ya que luego de la obtención de los resultados del test hay que generar estadísticas de un grupo de estudiantes manualmente	Difícil. Ya que se necesita implementar cada estadística que se quiere mostrar	Fácil. Ya que algunos módulos desarrollados para este fin traen consigo herramientas de generación de estadísticas

Como se ve se ha evaluado varios puntos que tienen que tener a favor los las técnicas de obtención de estilos de aprendizaje. En este análisis los puntos que se han

tratado han sido propuestos para que identifiquen la opción más adecuada para la obtención de estilos de aprendizaje de los estudiantes de la UTPL.

En la comparación que se muestra en la tabla 3.1 se observa claramente que la alternativa que puede ofrecer mejores resultados a la hora de trabajar en la obtención de estilos de aprendizaje de estudiantes en sistemas orientados a la educación es aquella que se basa en la construcción de módulos dentro de dichos sistemas, de estos módulos se necesita que tengan la suficiente flexibilidad para soportar a todos los estudiantes de la plataforma así como poder extenderlo con la aplicación de varios test y además brinde facilidades para la interpretación de los resultados obtenidos por conjuntos de estudiantes así como estadísticas de tendencias de tipologías de estilos de aprendizaje. Entonces dicho esto se propone que la técnica que se selecciona para la obtención de estilos de aprendizaje de los estudiantes de la UTPL en su Entorno Virtual de Aprendizaje es aquella que implementa módulos en dicha plataforma. Para ello en los siguientes capítulos se describirá el proceso para su implementación.

b) Evaluación y selección de clasificaciones de estilos de aprendizaje

Uno de los grandes dilemas que presenta la obtención de estilos de aprendizaje en sistemas orientados a la educación es elegir cuál de las clasificaciones propuestas, es la más efectiva y que ayudara a que los resultados obtenidos por los estudiantes sirvan de mejor forma a la adaptación de contenidos, material y recursos de aprendizaje. Para Peña, Marzo, de la Rosa y Fabregat. (2002). el modelo más apropiado para capturar los estilos de aprendizaje, es el modelo de estilos de aprendizaje de Felder y Silverman por el alto grado de aplicación que este ha tenido en sistemas de soporte a la educación en los últimos tiempos y por los buenos resultados que se han obtenido en la aplicación de dicho modelo en grupos controlados de estudiantes de carreras técnicas. Otra propuesta que tiene amplia difusión es la de Honey-Alonso que ha sido probado en grupos de estudiantes de diferentes áreas de aprendizaje y con amplios resultados las investigaciones desarrolladas por autores como son muestra de su eficacia a la hora de aplicar dicha clasificación.

Para esta investigación se aplicaron los test de Honey-Alonso y de Felder-Silverman a un grupo de treinta estudiantes de la UTPL. Del test de estilos de

aprendizaje, lo que se busca obtener, es la tipología a la que pertenece el estudiante, y los resultados que ha obtenido en las diferentes tipologías existentes en la clasificación. Estos datos se utilizan dentro del modelo para poder desarrollar el prototipo de los modelos de usuario. Y fuera del modelo dentro de los sistemas adaptativos para generar recomendaciones acopladas a la tipología de estilos de aprendizaje. Además es posible observar la predisposición de los estudiantes ya que los test elegidos constan de un número elevado de preguntas a contestar (44 el de Felder-Silverman y 80 el de Honey-Alonso). Los test aplicados se los puede encontrar en el Anexo A. Lo se puede describir de este proceso lo siguiente:

- ◆ Los Estudiantes tuvieron un cierto reparo al contestar el test de Honey-Alonso debido a que este test tiene una magnitud mayor en lo que se refiere al tamaño del mismo ya que consta de ochenta preguntas, prueba de este reparo es que algunos de los cuestionarios no fueron completados en su totalidad. Esto puede ser un problema ya que la idea es que se desea automatizar esta tarea en la plataforma del EVA y si los estudiante no son motivados a completar el test los resultados que se obtengan no van a ser óptimos para la generación del estilo de aprendizaje de cada uno de ellos.

Tabla 3.2: Resultados obtenidos en el test de estilos de aprendizaje de Felder – Silverman

Test de Felder y Silverman	
Estilos de aprendizaje	Número de estudiantes
Activo	3
Reflexivo	1
Sensitivo	2
Intuitivo	1
Visual	4
Verbal	1
Secuencial	1
Global	2
TOTAL	1

- ◆ El test de Felder y Silverman tuvo buena aceptación debido al diseño del mismo ya que la naturaleza dicótoma de las preguntas y las opciones de cada una de ellas motiva a los estudiantes a completar los mismos además que la reducción de las preguntas de casi la mitad (44 preguntas comparada con las 80 de Honey-

Alonso) hace que los estudiantes no se casen de llenar dicho cuestionario.

- ◆ En cuanto a los resultados obtenidos por los estudiantes que contestaron el cuestionario de Felder y Silverman vemos, en la tabla 3.2, que el estilo que predomina dentro de esta clasificación es el visual, ya que expresa un número mayor de estudiantes pertenecientes a esta tipología.
- ◆ Los resultados obtenidos por los estudiantes que contestaron el cuestionario de Honey - Alonso vemos, en la tabla 3.3, que el estilo que predomina dentro de esta clasificación es el teórico, ya que expresa un número mayor de estudiantes pertenecientes a esta tipología.

Tabla 3.3: Resultados obtenidos en el test de estilos de aprendizaje de Honey - Alonso

Test de Honey y Alonso	
Estilos de aprendizaje	Número de estudiantes
Activo	3
Reflexivo	4
Teórico	5
Pragmático	3
TOTAL	15

Luego de la experiencia obtenida también es necesario exponer, que desde mi punto de vista la clasificación expuesta por Felder y Silverman brinda más ventajas para la utilización de sus estilos de aprendizaje en la adaptación de contenidos en sistemas orientados a la educación, ya que considera dicótomas y dimensiones como puntos de vista para la clasificación de los mismos. Además utiliza un número reducido de preguntas más amistosas a los estudiantes y que brinda más facilidades a la hora de su implementación y posterior interpretación. Dichas estas aseveraciones se propone tomar esta clasificación como parte del modelo de usuario de los estudiantes de la UTPL.

3.3.3 Selección de datos para el modelo de usuario

En el Capítulo 1 se definieron los pasos para el desarrollo de un modelado de usuario y sus implicaciones más importantes, así como en la sección 3.3.1 de este Capítulo se realizó la selección de la tipología de modelo de usuario a utilizar en la propuesta que se describe en este documento. Es momento entonces de concentrar

esos conceptos y empezar a desarrollar el proceso de modelado que defina los elementos que van a ser parte de modelo de usuario de la UTPL. Para ello se inicia con la fase de recolección de información del sistema y de otros medios que formara parte de los datos de entrada del modelo de usuario del entorno virtual de aprendizaje (EVA) de la UTPL. Entonces en este apartado se presenta como propuesta un conjunto de datos que servirán de base para la obtención del modelo de usuario. Se han tomado los datos que se consideran importantes para la identificación de propiedades, características, e intereses de estudiantes en una plataforma de aprendizaje virtual. Estos atributos se tomaran como datos de entrada del Modelo de Usuario del Entorno Virtual de aprendizaje de la UTPL.

a) Datos de entorno

Los datos que se que se ha elegido como importantes para la definición del entorno del usuario tratan de determinar que hardware y software posee el usuario así como la localidad de donde se accede al sistema, para que, dependiendo de ello estos datos permitan determinar un perfil de acceso al sistema en donde se describan los medios comunes con los que cada usuario accede al sistema.

Tabla 3.4: Atributos de entrada que definen el entorno del usuario del EVA UTPL

ATRIBUTO	DESCRIPCIÓN	TABLA DE MOODLE	
		Atributos	Tabla
Plugins: Java, JavaScript, flash, Acrobat.	Atributo en el que se obtiene si en el equipo que se accede al sistema existen estos plugins que ayudan a la experiencia del usuario con el sistema.	No se aplica porque se obtienen del navegador con el que se accede al sistema	
Navegador	Describe el software que se utiliza para navegar en el sistema	No se aplica porque se obtienen del navegador con el que se accede al sistema	
Sistema operativo	Describe el Sistema Operativo con el que el usuario accede al sistema	No se aplica porque se obtienen del navegador con el que se accede al sistema	
Ultima dirección IP	La dirección de red del último equipo con el cual se accedió al sistema	lastip	mdl_user
Centro	Indica el centro universitario al que pertenece el estudiante (usuario)	centro	mdl_enrol_utpl

En la tabla 3.4 de nombran y describen los datos que se deben obtener del entorno de los usuarios para el modelo de Usuario del Entorno Virtual del Aprendizaje de la

UTPL.

Se debe mencionar que se han dejado de lado algunos argumentos importantes como, el ancho de banda de acceso al sistema, así como la identificación los dispositivos se acceso al sistema debido a la dificultad de obtención de los mismos, pero como alternativa la arquitectura que se propondrá será flexible a la incorporación de nuevos datos para formalizar mas el modelo y el dominio que de quiere diseñar.

b) Datos usuario

Tabla 3.5: Atributos de entrada que definen los datos de usuario del EVA UTPL

ATRIBUTO	DESCRIPCIÓN	TABLA DE MOODLE	
		Atributos	Tabla
Nombre de usuario	Campo para el acceso al sistema	username	mdl_user
Apellidos y nombres	Nombres y apellidos de los usuarios del sistema	firstname, lastname	mdl_user
Cedula	Campo que incluye el numero de cedula de los usuarios	idnumber,	mdl_user
Id usuario	Id del usuario en el sistema	id	mdl_user
Genero	Indica el sexo al que pertenece el usuario del sistema	sexo	mdl_user_utpl
Id curso	Id del curso que el usuario está cursando en Moodle	id	mdl_course
Nombre del curso	Nombre del curso al que pertenece el alumno en Moodle	fullname	mdl_course
Id periodo	Id del periodo que está cursando el estudiante en el sistema	pdoid	mdl_periodo_utpl
Periodo	Nombre del periodo académico que se encuentra cursando el estudiante en el sistema	name	mdl_periodo_utpl
Fecha inicio	Fecha en la que empieza el periodo académico que cursa un usuario	inicio	mdl_periodo_utpl
Fecha fin	Fecha en la que finaliza el periodo académico que cursa un usuario	fin	mdl_periodo_utpl
Modalidad	Modalidad de estudios a la que pertenece el estudiante en el sistema	modalidad	mdl_enrol_utpl

Los datos de usuario se que se han escogido para el modelo de usuario son aquellos esenciales que su objetivo primordial es identificar aspectos concretos del usuario así como los cursos que está tomando. A continuación en la tabla 3.5, mostramos estos datos, describiendo la información que contienen y las tablas de donde se tomaran

dichos atributos, se ha tratado que el conjunto de estos datos identifiquen directamente al usuario al que pertenecen.

c) Datos de interacción

Tabla 3.6: Atributos de entrada que definen los datos de interacción del usuario del EVA UTPL

ATRIBUTO	DESCRIPCIÓN	TABLA DE MOODLE	
		Atributos	Tabla
Último Acceso	Atributo que contiene la fecha y hora del último acceso que el usuario realiza al sistema.	lastaccess	mdl_user
Acciones sobre objetos	Atributo compuesto que describe que acción se realizó sobre un objeto	userid, course, module, action	mdl_log
Número de accesos al curso	Atributo que describe el número de veces que el usuario accede a la vista de su curso	userid, course, action(view), module(course)	mdl_log
Número de aportaciones al curso	Este atributo compuesto engloba el número de aportaciones que el usuario realiza al curso en las distintas herramientas que el sistema presta para ese propósito	userid, course, module, action	mdl_log
Número de preguntas al tutor	Atributo que contiene la cantidad de cuestionamientos que el estudiante realiza a su tutor.	userid, course, module (message), action(history)	mdl_log
Número de páginas por curso visitadas	Atributo que describe el número de páginas a las que le el usuario ha accedido al sistema, navegando entre las diferentes herramientas.	userid, course, action(view)	mdl_log
Número respuestas en los foros	Indicador del número de veces que un usuario accede y aporta a un conjunto de foros que estén disponibles en una materia que este cursando.	userid, course, module (forum), action (add discussion)	mdl_log
Tiempo promedio de interacción en el curso	Atributo que contiene el tiempo promedio que un usuario accede al sistema en cada curso que tiene asignado.	userid, course, time, module, action	mdl_log
Número de tareas subidas	Número de archivos de tarea que el usuario sube al sistema en calidad de tareas referentes al curso que se encuentra cursando dicho usuario.	userid, course, module (upload), action (upload)	mdl_log
Número de tareas propuestas	El número total de tareas que los tutores de cada curso han propuesto para el usuario en cuestión.	userid, course, module	mdl_log

Como ya se menciona en el capítulo anterior los datos de interacción del usuario con el sistema son los que definirán el comportamiento del usuario en el sistema, así

como sus intereses y preferencias, entonces se necesita que los datos que se tomen en cuenta para el modelado de esta parte del modelo de usuario sean los más apropiados y relevantes, pero en gran medida también condiciona esta extracción, la disponibilidad de esta información en el sistema. Entonces, los datos que se incluyen en la tabla 3.6 se han definido pensando en estas premisas y a criterio del autor son suficientes para determinar adecuadamente los indicadores de preferencia e interés que se desea obtener de los mismos. Además de esto se incluye una descripción de los mismos así como se menciona las tablas del sistema de donde se los va a obtener.

3.3.4 Selección de la metodología de desarrollo del agente

En la sección 2.5 del Capítulo 2, se desarrollo una introducción a las metodologías de desarrollo de agentes inteligentes, describiendo puntos importantes de cada una de ellas, tratando siempre de sentar las bases para una comparación de las mismas. En esta sección se presenta una comparativa de estas metodologías para desarrollar una selección de la que presente las mejores prestaciones en la implementan del agente de modelado de usuario del entorno virtual de aprendizaje de la UTPL.

Tabla 3.7: Comparativa de metodologías de desarrollo de SMA

Metodología	Vistas de especificación del sistema	Integración con el proceso de desarrollo	Integración con la teoría de agentes	Herramientas de soporte al proceso de desarrollo
Vowell Engineering	Vocales que representan los componentes del sistema	Solo vocales como resumen de elementos a considerar y tecnologías aplicadas	Combinación de aspectos agrupados en vocales	No consta con herramientas de soporte al desarrollo
Mas-CommonKADS	Desarrollo en torno modelo de experiencia Considera agentes de usuario y el sistema	Integración con el ciclo de vida del software. Siete Modelos (agente, tareas, experiencia, coordinación, comunicación, organización y diseño)	Identificación de elementos y relaciones entre estos dentro del sistema.	No consta con herramientas de soporte al desarrollo
BDI	Los agentes utilizan modelo del mundo por medio de creencias, deseos, e Intenciones.	Proceso iterativo incremental con retro-alimentaciones Pasos específicos para la generación de los modelos	Dependencias entre los modelos Modelos que operan en dos niveles de abstracción	No consta con herramientas de soporte al desarrollo
MaSE	Parte de la Orientación a Objetos Agentes	Conjunto de pasos que se llevan a cabo en la herramienta de soporte El proceso consta de:	Utilización de maquinas de estados y representación simple de los elementos	<i>AgentTool</i> . Sigue el proceso del desarrollo del sistema con

Metodología	Vistas de especificación del sistema	Integración con el proceso de desarrollo	Integración con la teoría de agentes	Herramientas de soporte al proceso de desarrollo
	especialización de objetos	captura de objetivos, diagramas de roles, y tareas asociadas a los roles	característicos de la tecnología de agentes	generación de código dependiente del diseño. No considera la dependencia entre los modelos
ZEUS	Abstracción del sistema dentro de un conjunto de herramientas basadas en la colaboración de los agentes	Se producen como resultado: Análisis del dominio, Diseño del agente, Implementación de los agentes y soporte en tiempo de ejecución.	Combina los distintos resultados de la investigación de agentes (planificación, ontologías, responsabilidades y relaciones)	<i>ZEUS Agent Building Tool</i> . Generación de programas para el arranque del sistema diseñado e implementado. Carencia de un proceso detallado
GAIA	Maximizar una medida de calidad global por parte del sistema	Proceso detallado desde requisitos iniciales hasta la fase de diseño, fácil de implementar. Propone: modelo de roles y de interacciones. Se omite dependencias entre modelos	Uso de formulas lógicas que son más efectivas pero muy complicadas de implementar	No consta con herramientas de soporte al desarrollo
INGENIAS	Incorpora filosofía de MESSAGE evolucionando ideas e incorporando herramientas de soporte	Proceso similar al de RUP. Se definen meta-modelos para describir el sistema, que a su vez generan modelos similares a los de UML. Se contemplan entidades y las relaciones entre ellas y la relación entre los modelos generados. Se definen actividades que construyen los modelos que se organizan siguiendo paradigmas de ingeniería de software	Se integra los paradigmas de desarrollo de agentes dentro de los meta-modelos en el diseño de las interacciones entre los componentes del sistema y la definición de los mismos.	<i>INGENIAS Development Kit (IDK)</i> . Proceso de generación de código más flexible. Plantillas para la generación de documentación de los agente.

En la tabla 3.7 se presenta una comparativa de metodologías en la cual se describe un conjunto homogéneo de las características más significativas de cada una de ellas. Se puede observar que cada metodología propone un punto de vista diferente de la especificación del sistema, y diferentes grados de especialización en la teoría de agentes, así como en la integración con el proceso de desarrollo de software. Esta variedad formas de tratar los sistemas multiagente (SMA) hace que cada metodología se aplique de acuerdo a la experiencia de los desarrolladores que las van a utilizar. Estas metodologías se especializan en áreas concretas. Así que para la selección, se tiene que tomar en cuenta esta premisa, también hay que considerar la experiencia de

los usuarios en el desarrollo de sistemas, para que la metodología escogida sea familiar al usuario que la aplica.

De lo expuesto anteriormente, se cree que la mejor opción a la hora de la implementación del agente de modelado de usuario del entorno virtual de aprendizaje de la UTPL, es *INGENIAS*. Existen varias razones que hacen que esta metodología se considere como primera opción, una de ellas es que su punto de vista sobre la idea de los SMA es fundamentada en una metodología altamente probada como lo es MESSAGE, además incorpora todos los resultados actuales en la investigación de la tecnología de agentes. Otra razón es, que en el proceso de desarrollo del sistema es una de las más robustas, ya que es la que más se asemeja al Proceso Unificado de Desarrollo de Software, incluyendo nuevas ideas aplicables a la teoría de agentes (los meta-modelos), e incorpora eficientemente los paradigmas de ingeniería de software. Una razón importante para su selección es que es una de las pocas metodologías que describe más formalmente la interacción entre sus modelos, fundamental para el desarrollo de SMA. Y finalmente, una razón de peso para su elección es que *INGENIAS* presenta la mejor herramienta de soporte al proceso de diseño y construcción de SMA, contemplando desde la documentación ordenada de los agentes, hasta generación flexible de código ejecutable del sistema.

3.3.5 Selección de algoritmos de aprendizaje automático

Dentro de la inferencia de atributos mediante aprendizaje automático, para el modelado de usuario, es necesario que los clasificadores elegidos sean seleccionados luego de un proceso de prueba con los datos que van a ser sujeto de su estudio. Se realiza esta tarea de esta forma debido a que las instancias a clasificar pueden tener características muy diferentes hasta para atributos de un mismo dominio. Entonces se sigue un proceso de descubrimiento de Información en bases de datos, en el los datos seleccionados generen modelos con varios algoritmos de clasificación de instancias los cuales se comparen y puedan elegirse los más apropiados.

En el Capítulo 4 se describe a detalle el proceso de KDD desarrollado a la base de datos del Entorno Virtual de Aprendizaje (EVA) de la UTPL el cual, mediante varias experiencias permitieron general modelos de las instancias en cursos pasados para tres

atributos contenidos dentro del modelo de usuario de los estudiantes de la UTPL. Para cada atributo se generaron nueve modelos con distintos algoritmos que se escogieron de acuerdo a varios criterios:

- ◆ Su grado de aplicación en trabajos similares
- ◆ Su grado de concordancia con los datos disponibles para la experimentación
- ◆ Y su incidencia dentro del campo de extracción de conocimiento en la minería de datos y el aprendizaje automático.

La cantidad de modelos generados y probados así como el número de instancias disponibles para la experimentación, desde el punto de vista del autor, aseguran la variedad y eficacia de los resultados obtenidos. Los algoritmos escogidos son:

- ◆ Árboles de decisión C4.5 (J48)
- ◆ Árboles de decisión REPtree
- ◆ Redes neuronales con perceptrón multicapa (MultilayerPerceptron)
- ◆ Clasificador NaiveBayes
- ◆ Redes Bayesianas (BayesNet)
- ◆ Combinación de clasificadores por el método Bagging

De los algoritmos escogidos se presenta una descripción en la sección 1.9.3 del Capítulo 1.

Capítulo 4

*Minería de datos para el
modelo de usuario*

4.1 Introducción

Dentro del estudio que concierne a este proyecto de tesis, una parte fundamental del mismo es el desarrollo de un proceso de minería de datos, el cual permita encontrar los indicadores adecuados de los atributos considerables dentro del modelo de usuario de los estudiantes de la UTPL. Para ello partimos del hecho que los sistemas adaptativos de apoyo al proceso de aprendizaje han tenido un crecimiento exponencial en los últimos años (Zytkon y Klosgen, 2001), lo que ha motivado el desarrollo de proyectos exitosos, aplicando esencialmente técnicas de minería de datos a este tipo de sistemas, para el descubrimiento de conocimiento que sirva para guiar a los estudiantes en su aprendizaje y en la maximización del mismo.

En este capítulo se presenta un proceso de extracción de conocimiento (KDD) que fue aplicado al Entorno Virtual de Aprendizaje (EVA) de la UTPL, para la selección de los atributos presentes dentro del modelo de usuario de los estudiantes de la UTPL, en las secciones contenidas en este capítulo, primero se incluye un estudio de la población estudiantil, para finalmente proseguir con el desarrollo del proceso de extracción de conocimiento.

4.2 Características de la población estudiantil de la UTPL Modalidad a Distancia

Antes de describir detenidamente en el proceso de minería de datos, se considera relevante describir a la población estudiantil de la Universidad Técnica Particular de Loja, que es sujeto de estudio de este proyecto. Se realiza esta descripción para conocer sus características, para que el proceso de minería que se tomo como referencia guarde relación con los sujetos de estudio de proyecto, los estudiantes.

La población estudiantil de la UTPL, que cuenta con un sistema de estudios a distancia donde los estudiantes tienen características heterogéneas, es decir, que pueden variar drásticamente de un sujeto a otro, para lo cual, se hace necesario que el modelo que se proponga satisfaga las necesidades y requerimientos de estos estudiantes y que albergue en él datos que sirvan objetivamente a las tareas para las cuales se están diseñando.

En esta parte se toma como referencia lo mencionado por María José Rubio (2002) sobre la población estudiantil de la UTPL estudio que refleja en gran medida las características y preferencias a la hora de aprender de los estudiantes y que servirá de mucho en la creación y diseño del modelo de datos para el EVA. Además se incluyen estadísticas nuevas, con datos actualizados que reflejen el real estado de la población estudiantil utepelina.

Actualmente la población estudiantil a distancia ha tenido cambios y el perfil que tenía a sus comienzos va tomando matices diferentes con el tiempo, esta población estudiantil reúne un conjunto de características que se describen a continuación:

4.2.1 Dispersión geográfica

Por lo general está alejado de los centros de educación superior, disperso en toda la geografía nacional, o bien vive en una gran ciudad en donde necesita de mucho tiempo para desplazarse a la universidad presencial. En la siguiente gráfica se refleja la distribución geográfica de los estudiantes de la UTPL.

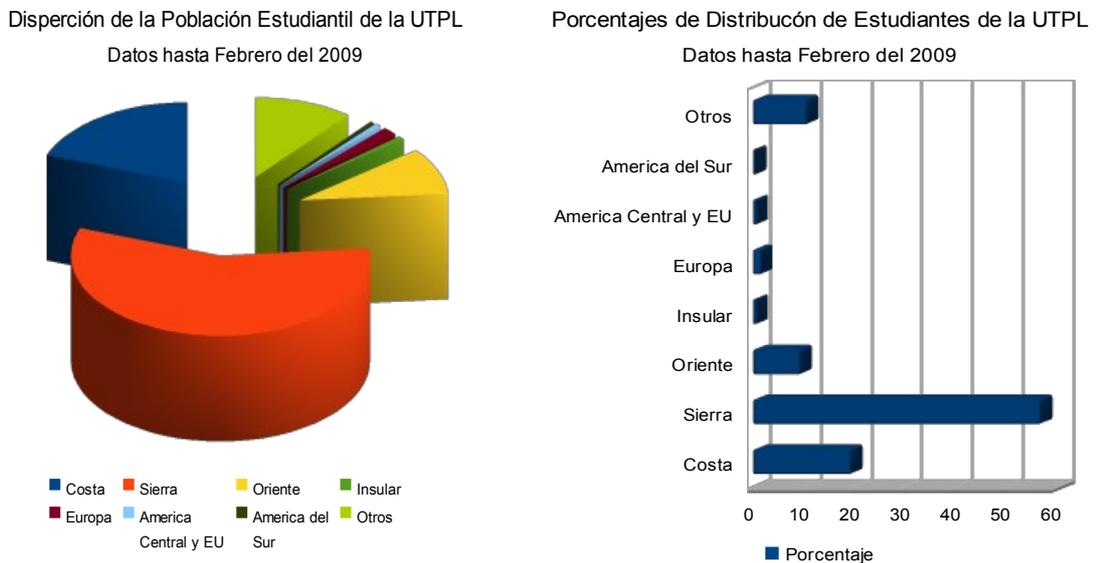


Figura 4.1: Distribución de la población estudiantil de la UTPL

En la figura 4.1 se puede apreciar que la mayor concentración de estudiantes se encuentran en algunas ciudades la sierra ecuatoriana, con un porcentaje de 56.9 % de la población estudiantil, así también destacan la costa y el oriente ecuatorianos con los

puestos siguientes. Es oportuno mencionar también que se encuentra un número importante de estudiantes de la universidad en el extranjero que en conjunto suman el 2.7 % de la población estudiantil, es decir 1393 estudiantes.

4.2.2 Edad y estado

Si analizamos las estadísticas de las universidades que imparten estudios a distancia vemos como la mayoría de sus alumnos se sitúan entre los 25 y 40 años de edad. Es decir una edad adulta pero relativamente joven, con aspiraciones profundas de cambio en sus vidas y en gran parte con una estabilidad familiar, suelen ser casados y con varios hijos.

4.2.3 Motivación

Es una de las características comunes de los estudiantes a la distancia que podemos apreciar muy bien en los comienzos de la carrera. Esta ha sido elegida libremente, buscando cumplir aspiraciones que se han tenido desde hace tiempo, se espera concluir su carrera buscando un cambio en la vida o en la formación como satisfacción personal.

4.2.4 Intereses profesionales

Un motivo que lleva a los estudiantes de nivel superior y en particular a los estudiantes de la UTPL a obtener su título de nivel superior es el poder ocupar cargos más altos en el trabajo que viene desempeñando, ósea la obtención de un título de educación superior le permite promocionar su trabajo.

La titulación superior también abre las puertas para que los estudiantes puedan desempeñarse en actividades laborales por las que tienen afición y para los que ya se encuentran inmersos dentro de un ámbito laboral pueden optar con esto, por un cambio en sus actividades de trabajo por aquellas que más les puedan satisfacer.

Algunas personas deciden emprender estudios de nivel superior debido a que poseen negocios o son parte de sociedades familiares y pretenden obtener una titulación para asumir adecuadamente esos retos profesionales, y si ya son titulados buscan obtener una profesión más acorde con el trabajo que realizan.

4.2.5 Falta de tiempo

Una característica importante a considerar de la mayoría de estudiantes de la UTPL, especialmente los de la modalidad de estudios a distancia, es que tienen muy poco tiempo para dedicarlo a actividades académicas debido a sus varias responsabilidades adicionales como son el hogar, el trabajo y en algunos casos el campo académico puede llegar a ser considerado secundario.

4.2.6 Aislamiento y deserción

Por aislamiento se debe tomar no solo el hecho de que algunos estudiantes se encuentran lejos de los centros de aprendizaje, sino también el hecho de que un estudiante de la modalidad a distancia tiene que estudiar solo, ya que no conoce a la mayoría de los compañeros y se encuentra distante de su tutor. Además padece de dificultades de comprensión de los contenidos que se tratan en la asignatura, debido esencialmente a falta de tiempo, a la falta de apoyo académico, y a la carencia de una figura permanente de enseñanza. Se debe mencionar que también existen algunos factores que son causa de la deserción de los estudiantes, uno de ellos es la dificultad de las asignaturas, además un estudiante siente aislamiento por la lejanía del centro universitario (en el caso de los estudiantes de la modalidad abierta) y de sus tutores, también se incrementa este fenómeno debido a problemas familiares así como problemas de índole laboral.

Rubio (2002) incluye en su estudio de características de los estudiantes de la UTPL un *síntesis comparativa de características de los estudiantes* que se detalla en la tabla siguiente:

En la tabla 4.1 se muestra una comparativa de las características de los estudiantes tradicionales (modalidad clásica) y de los estudiantes a distancia (modalidad a distancia). En ésta, se reflejan características propias de cada tipología de estudiante y que son de gran impacto en la definición de un modelo de datos de los estudiantes de la UTPL.

Tabla 4.1: Síntesis comparativa de las características de los estudiantes de la UTPL (Rubio, 2002)

Rol / Situación del Estudiante Tradicional	Rol / Situación del Estudiante a Distancia
• Dedicación a tiempo completo	• Dedicación a tiempo parcial
• Homogéneos en edad	• Heterogéneos en edad
• Persona joven sin experiencia	• Persona adulta con experiencia de vida y laboral
• Sin responsabilidades fuera de sí mismo	• Con responsabilidades adicionales a las propias
• Económicamente Dependiente	• Económicamente Independiente
• Cercano a la institución de aprendizaje	• Distante físicamente de la institución
• Difiere o suprime otros roles	• No le es posible diferir o suprimir otros roles por su precedencia
• Homogéneos en conocimientos	• Heterogéneos en conocimientos
• Homogéneos en cualificación	• Heterogéneos en cualificación
• Familiarizado con los procesos académicos por la larga y continua experiencia.	• Poco familiarizado con los procesos académicos por las interrupciones
• El principal ambiente educativo es el campus universitario	• Varios Ambientes: trabajo, hogar, otros
• Aprender es la principal ocupación del estudiante	• Aprender es una ocupación secundaria del estudiante
• Aprender es requerido por ley	• Aprender no es requerido por ley
	• El estudio es voluntario por las ventajas laborales o calidad de vida.
• Sigue un currículo obligatorio	• Sigue un currículo en parte autodeterminado
• Aprendizaje dependiente	• Aprendizaje Independiente
• Poco motivado para el estudio	• Muy motivado para el estudio

Como conclusión se puede argumentar que se han encontrado cambios en la

población estudiantil de la UTPL en los últimos años. Estos cambios se han producido por el crecimiento de la misma, además de la creación de nuevos centros y carreras que han generado que las tendencias se remarquen mucho más. Estos datos nos ayudan a comprender los cambios que la educación está teniendo en la Universidad Técnica Particular de Loja. Se comprueba también que la educación a distancia está incidiendo directamente en la forma en que la población ecuatoriana se educa a nivel superior, ya que el ritmo en que la población de la modalidad a distancia está creciendo, da muestra fehaciente de que está convirtiéndose en una seria alternativa de educación para los estudiantes con otras ocupaciones aparte del estudio y para las que por circunstancias ajenas a su consentimiento tuvieron que relegar su preparación profesional a un nivel secundario.

4.3 Extracción de conocimiento del entorno virtual de aprendizaje UTPL

Un proceso de minería de datos se basa en la formulación de modelos que se representan de formas diversas y cada una de esas representaciones determina la técnica que debe de usarse para inferir los datos que se desea conocer.

Esencialmente se identifican, según Hernández, Ramírez y Ferri (2004), dos tipos de modelos: *predictivos* y *descriptivos* de los cuales se plantea la utilización de un modelo predictivo debido a que esta tipología de modelo se ajusta de mejor forma a las necesidades que presenta la inferencia de los atributos pertenecientes al modelo de usuario de los estudiantes de la UTPL. Además sostienen que las tareas que implementan modelos predictivos son la *clasificación* y la *regresión*, de estas se utilizo a la clasificación para la predicción los atributos, debido a su amplia utilización en investigaciones relacionadas a la inferencia de atributos de modelos de usuario en sistemas de hipermedia adaptativos aplicados a la Web.

4.3.1 Proceso de extracción de conocimiento a seguir

El proceso de extracción de conocimiento planteado en este trabajo es de amplia aceptación entre los investigadores afines y engloba en gran medida el trabajo a realizar referente a la construcción de conocimiento a partir de un modelo predictivo y

aplicando tareas de clasificación, para la predicción de atributos de un modelo del estudiante en un entorno de aprendizaje orientado a la Web.

El modelo propuesto consta de cuatro fases que se detallan en la Figura 4.2 definiendo las tareas que se realizaron para la generación de conocimiento de la base de datos existentes.

En esta sección también se presentan las experiencias que se obtuvieron al realizar el proceso de minería de datos a la base de datos del Entorno Virtual de Aprendizaje (EVA) de la UTPL.

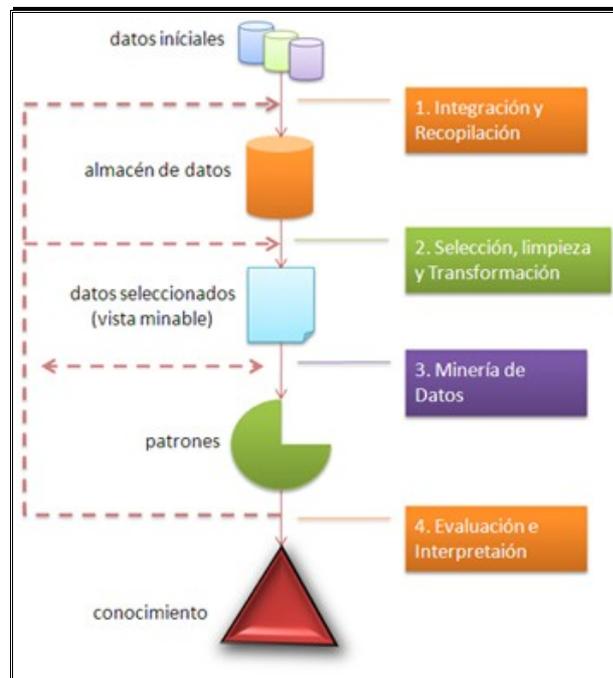


Figura 4.2: Fases del proceso de descubrimiento de conocimiento en bases de datos, KDD. (Hernández, Ramírez y Ferri, 2004)

La base de datos utilizada para este estudio fue tomada el 7 de Enero del 2009 del servidor de pruebas UTPL <http://evadev.utpl.edu.ec> que replica la información del servidor de producción, contiene información de todos los cursos dictados por la UTPL, en sus dos modalidades de estudio (Clásica y a Distancia), desde el periodo académico Abril/2004 – Agosto/2004, cuando entro en funcionamiento, hasta el periodo Octubre/2008 – Diciembre/2008, que fue tomado el último estado de la base de datos.

Para el desarrollo de este estudio se tomo como población los estudiantes de las carreras de la modalidad a distancia que se muestran en la Tabla 4.2, dando una población total de 6845 estudiantes, de los periodos académicos comprendidos entre Abril/2007 – Agosto/2007 hasta Octubre/2008 – Diciembre/2008. Se tomo como población estas carreras ya que son las que registran un alto grado de interacción comparadas con las demás.

Tabla 4.2: Carreras de la modalidad a distancia con mayor número de interacciones

Carrera	Número de interacciones
Informática	321777
Jurisprudencia	217968
Administración de empresas	157604
Psicología	120120
Ingeniería en contabilidad y auditoria	102662
Ciencias de la educación - ingles	74078
Comunicación social	56058
Educación infantil	42165
Administración en banca y finanzas	38756
Gestión ambiental	36979
Economía	31215
Administración turística	26217
Asistencia gerencia y relaciones publicas	25611

Luego de seleccionar la población se procedió mediante la técnica de *Muestreo Aleatorio Estratificado* (Hernández, Ramírez y Ferri, 2004), implementada mediante scripts de lenguaje PHP¹ a seleccionar un conjunto de estudiantes para que sean sujetos de este trabajo de minería.

En el muestreo aleatorio estratificado se utilizan estratos, que no son más que agrupaciones de las instancias a seleccionar, en nuestro caso los estratos se definieron como cada una de las carreras que fueron tomadas en cuenta para la población, luego para la selección de la muestra se toman n atributos de cada estrato, mediante muestreo aleatorio simple, para nuestra muestra se tomaron diez estudiantes por carrera, dando un total de 130. Se tomo como referencia esta técnica de muestreo debido a que, según sus autores, es estadísticamente aconsejable su uso cuando

¹ *PHP*. (Personal Home Page Tools – Herramientas para Páginas Iniciales Personales) Lenguaje de programación tipo script para entornos Web utilizado sobre todo en servidores Linux para personalizar la información que se envía a los usuarios que acceden a un sitio web. Es un programa de software libre con unas funciones muy semejantes a las de ASP y JSP (Fernández, 2008).

existen dentro de la población estratos de los cuales se necesita que existan casos de cada uno de ellos, además los elementos de un mismo estrato son generalmente similares lo que ayuda a mantener la diversidad en la muestra ya que se tiene grupos equilibrados de las instancias.

Para cada una de las fases del proceso de extracción de conocimiento se cumplieron algunas actividades que se describen a continuación:

4.3.2 Integración y recopilación

En esta fase lo que se busca es realizar un ordenamiento apropiado de los campos que se consideran de la base de datos disponible. Se contempla la construcción de una aplicación ETL (*Extraction, Transformation Load*)² para la implementación de un almacén de datos con la muestra obtenida, obteniendo la información mediante consultas de lenguaje SQL³ de manipulación de bases de datos. Además de la incorporación de procesos de integración de datos.

El script de PHP creado para la selección de los datos de los usuarios seleccionados, genera un arreglo en el mismo lenguaje que es incluido dentro de la aplicación ETL creada para la preparación de los datos que se incluyeron dentro de la muestra. Este arreglo contiene los identificadores de usuario de los estudiantes seleccionados para luego de esto poder obtener la información referente a los cursos en los que han participado.

En esta fase del proceso de extracción de conocimiento se contaba ya con la selección de atributos propuestos en la sección 3.3.3 del Capítulo 3, así como la selección de estudiantes, descrita en los párrafos anteriores, los cuales debían ser ordenados de cierta forma para que puedan ser integrados y almacenados en una base de datos intermedia, denominada “*mineria_eva*”, para lo cual se diseñaron tres tablas que albergan los datos de interacción, datos personales de los usuarios y sus cursos, y las acciones sobre los objetos.

2 Existen traducciones diversas en castellano, como ETC (Extracción, Transformación y Carga) o ETT (Extracción, Transformación y Transporte)

3 *SQL*. (Structured Query Language – Lenguaje Estructurado de Consulta) Estándar para la construcción de sentencias de acceso, definición y depuración de bases de datos usado por los sistemas gestores de bases de datos más importantes

Tabla 4.3: Tablas para la integración y recopilación de datos

Tabla: cursos_y_alumnos			
Atributo	Tipo	Constraints	Extra
ID	smallint(10) unsigned	primary Key	auto_increment
ID_ALUMNO	decimal(10,0)		
NOMBRES_ALUMNO	varchar(255)		
DNI_ALUMNO	varchar(20)		
USERNAME	varchar(100)		
GENERO	varchar(20)		
ULTIMO_ACCESO	date		
ULTIMO_LOGIN	date		
LOGIN_ACCTUAL	date		
ULTIMO_IP	varchar(100)		
ID_CURSO	decimal(10,0)		
NOMBRE_CURSO	varchar(255)		
MODALIDAD	varchar(20)		
CENTRO	varchar(255)		
ID_PERIODO	decimal(10,0)		
PERIODO	varchar(255)		
FECHA_INICIO	date		
FECHA_FIN	date		
Tabla: interacciones			
Atributo	Tipo	Constraints	Extra
ID	smallint(10) unsigned	primary Key	auto_increment
ID_ALUMNO	decimal(10,0)		
ID_CURSO	decimal(10,0)		
MODULO	varchar(40)		
ACCION	varchar(100)		
RECURSO	varchar(100)		
PLUGINS	varchar(255)		
NAVEGADOR	varchar(255)		
SISTEMA_OPERATIVO	varchar(255)		
Tabla: acciones_objetos			
Atributo	Tipo	Constraints	Extra
ID	smallint(10) unsigned	primary Key	auto_increment
ID_ALUMNO	decimal(10,0)		
ID_CURSO	decimal(10,0)		
MODULO	varchar(40)		
ACCION	varchar(100)		
RECURSO	varchar(100)		

En la tabla 4.3 se muestran los campos que contiene el modelo de datos, resalta a la vista un grado de similitud en los contenidos que albergaran las tablas

“*interacciones*” y “*acciones_objetos*”, pero, son dos puntos de vista distintos ya que la tabla *interacciones* tiene la información de todas las acciones del usuario en un curso, mientras que la tabla *acciones_objetos* esta limitada a las acciones que realizan los usuarios en un curso, pero solo de los módulos del EVA que son considerados como objetos dentro de los mencionados cursos.

Luego de implementar el modelo de datos para el almacenamiento intermedio en un motor de base de datos, que en este caso fue un servidor Windows XP Profesional con el motor de bases de datos MySQL 5.0.11, se procedió al diseño e implementación de una aplicación ETL a medida, el cual en esta fase del proceso de KDD se encargo de extraer los datos de la base de datos del EVA y de hacer un pre-procesamiento eliminando errores en las instancias y redundancias no deseadas.

La aplicación ETL fue implementada mediante scripts del lenguaje PHP, esta aplicación, inicialmente carga el archivo que contiene el arreglo de los identificadores de usuario de los estudiantes seleccionados en la muestra, y mediante el uso de sentencias en lenguaje de manipulación de datos SQL, obtiene primero los datos personales de los distintos estudiantes, para luego obtener información referente a los cursos en los cuales participaron los mismos, después obtiene los datos de interacción en cada uno de los cursos extraídos en el paso anterior. Luego de esto se incorpora la información referente a las acciones sobre los objetos realizadas por parte de los estudiantes en cada uno de los cursos participados. Dentro de los datos de interacción se realiza una integración con datos externos ya que datos como los plugins existentes, navegador o el sistema operativo de acceso al sistema por parte de los usuarios es incorporado de desde los logs de interacción del servidor del EVA.

La forma en que la aplicación ETL incorpora los datos al nuevo modelo de datos es que genera archivos de texto plano los cuales contienen sentencias “INSERT” de lenguaje SQL que se pueden cargar al modelo de datos construido mediante aplicaciones de carga de scripts SQL.

Terminado el proceso de carga de toda la información mencionada, se cuenta con los datos necesarios para poder realizar la siguiente fase del proceso de KDD sobre la muestra de estudiantes seleccionada. Como datos adicionales de la información

obtenida se presenta los totales de filas disponibles en cada tabla.

- ◆ Tabla *cursos_y_alumnos*: 2400 filas
- ◆ Tabla *interacciones*: 35432 filas
- ◆ Tabla *acciones_objetos*: 29323 filas

Según las cifras de cada tabla se cuenta con un número considerable de instancias que servirán de mucho en el proceso de conocer los patrones y tendencias de los estudiantes dentro del sistema de apoyo al aprendizaje EVA.

4.3.3 Selección, limpieza y transformación

La fase de selección, limpieza y transformación de los datos servirá para la eliminación de inconsistencias en los mismos, así como para la integración de los datos en atributos de valor más significativo, se utiliza la *discretización* como técnica para transformar valores numéricos en valores nominales.

En esta segunda fase del proceso de KDD se siguieron algunos pasos los cuales permitieron generar las vistas minables:

Primero se debía conocer que atributos del modelo de usuario se necesitaba que se infieran mediante aprendizaje automático para poder reconocer los datos eran relevantes para su inferencia, y se concluyó que los atributos a inferir eran los siguientes:

- ◆ Nivel de interés en el curso
- ◆ Nivel de interés en los recursos
- ◆ Nivel de interacción

Luego de esto se construyó otra parte la aplicación ETL, esta se encargaría de utilizar los datos extraídos en la fase anterior para desarrollar la tarea de transformación de los mismos. Para lo cual se diseñó e implementó un modelo de datos anexo al desarrollado en la fase de integración y recopilación el cual consta de tres entidades que se detallan en la tabla 4.4.

Tabla 4.4: Tablas para la limpieza y transformación de datos

Tabla: interaccion			
Atributo	Tipo	Constraint	Extra
ID	smallint(10) unsigned	primary Key	auto_increment
ID_ALUMNO	decimal(10,0)		
NUM_ACCESOS_CURSO	decimal(10,0)		
NUM_APORTES_CURSO	decimal(10,0)		
NUM_PREGUNTAS_PROFESOR	decimal(10,0)		
NUM_PAGS_CURSO_VISITADAS	decimal(10,0)		
NUM_RESPUESTAS_FOROS	decimal(10,0)		
TIEMPO_PROMEDIO_INTERACCION	decimal(10,0)		
NUM_TAREAS_REALIZADAS	decimal(10,0)		
NUM_TAREAS_PROPUUESTAS	decimal(10,0)		

Tabla: atributos			
Atributo	Tipo	Constraint	Extra
ID	smallint(10) unsigned	primary Key	auto_increment
ID_INTERACCION	smallint unsigned	foreign key	
PARAMETRO	varchar(100)		
VALOR	varchar(50)		

Tabla: totales_interacciones			
Atributo	Tipo	Constraint	Extra
ID	smallint(10) unsigned	primary key	auto_increment
ID_ALUMNO	decimal(10,0)		
ULTIMO_ACCESO	date		
NUM_TOTAL_SECCIONES_USUARIO	decimal(10,0)		
NUM_TOTAL_APORTACIONES_USUARIO	decimal(10,0)		
TIEMPO_PROMEDIO_INTERACCION	decimal(10,0)		
NUM_TOTAL_PAGS_VISITADAS	decimal(10,0)		

Las tablas descritas contienen la información de interacción en los curso por parte de los usuarios, representada mediante técnicas que incluyen, *discretación*⁴, *numeralización*⁵, y *creación de nuevas características*⁶, todas ellas descritas en el trabajo desarrollado por Hernández, Ramírez y Ferri (2004).

4 *Discretación*. (Cuantización – binning) Transforma un atributo numérico en un valor nominal ordenado que representa un intervalo o “bin”, es aplicable cuando el error en la medida puede ser grande o existen umbrales significativos (Hernández, Ramírez y Ferri, 2004).

5 *Numeralización*. Proceso inverso a la discretación, es decir, se transforman atributos nominales en atributos numéricos, generalmente el número de variables numéricas que se pueden crear es igual o menor que el número de variables nominales (Hernández, Ramírez y Ferri, 2004).

6 *Creación de características*. (Agregación) Consiste en crear nuevos atributos para mejorar la calidad, visualización o comprensibilidad del conocimiento extraído, sin sustituir los valores iniciales (Hernández, Ramírez y Ferri, 2004).

Para la carga de la información en las tablas de la base de datos se utilizaron sentencia SQL que toman los datos del modelo creado en la fase de integración y recopilación. Luego, la aplicación ETL los transforma en los nuevos atributos los valores seleccionados y crea sentencias del tipo “INSERT” que son almacenadas en un archivo de texto plano. Luego de realizar la carga dentro del modelo diseñado para el caso de los nuevos atributos, se procede con la introducción del conocimiento previo, fundamental en el proceso de generación de vistas minables, mediante una interfaz que presenta cada una de las interacciones en el sistema y permite que se califique cada uno de los atributos que se desean evaluar. En la figura 4.3 se presenta una captura de este proceso para el atributo nivel de interés en el curso.

Id Interaccion	numero accesos curso	numero aportes al curso	numero de preguntas al profesor	numero de paginas curso visitadas	numero de respuestas en los foros	tiempo promedio de interaccion	tareas propuestas menos tareas realizadas	nivel_interes_curso
77	7	0	0	7	0	41	0	bajo
78	57	6	0	59	0	597	2	alto
79	70	8	0	73	0	368	7	bajo
80	7	0	0	7	0	10	0	bajo
81	42	2	0	44	0	345	-1	medio
82	28	0	0	28	0	77	0	bajo
83	8	2	0	8	0	114	-1	bajo

Figura 4.3: Introducción de conocimiento previo para el atributo nivel de interés en el curso

Tabla 4.5: Discretación de datos para el atributo nivel de interés en el curso

Atributo	Intervalos					
	Bajo		Medio		Alto	
	Intervalo	Instancias	Intervalo	Instancias	Intervalo	Instancias
NUM_ACCESOS_CURSO	0-93	2322	93-186	30	186-inf	6
NUM_APORTES_CURSO	0-16	2344	16-31	12	31-inf	2
NUM_PREGUNTAS_PROFESOR	0-1	2338	1-2	14	2-inf	6
NUM_PAGS_CURSO_VISITADAS	0-106	2318	106-212	36	212-inf	4
NUM_RESPUESTAS_FOROS	0-6	2350	6-13	6	13-inf	2
TIEMPO_PROMEDIO_INTERACCION	0-236	2068	236-472	138	472-inf	128
NUM_TAREAS_REALIZADAS- NUM_TAREAS_PROPUUESTAS	0-inf	137	(-22)-(0)	2196	inf-(-22)	3

Para la selección de este proceso se han introducido criterios de discretización realizada previamente con la ayuda de la herramienta WEKA⁷, la cual se conecto a la

⁷ WEKA.(Waikato Environment for Knowledge Analysis) <http://www.cs.waikato.ac.nz/~ml/weka/>

base de datos que contiene las instancias creadas en el paso anterior y se discretaron mediante uno de los filtros disponibles en el panel “*Preprocess*” de la utilidad “*Explorer*” de la mencionada herramienta. El filtro utilizado es el “*Discretize*” no supervisado con un número de 3 bins (intervalos), para el caso de los atributos predictivos del nivel de interés en el curso, los resultados se muestran en la tabla 4.5:

Luego de discretar los valores predictivos de los datos se procede a evaluar el nivel de interacción con la aplicación ETL. El criterio utilizado para definir el nivel de interacción en este caso es ponderar el número mayor de niveles por cada atributo e introducir dicha ponderación con la aplicación ETL. Luego de realizar esta tarea con todos los atributos se diseña un conjunto de vistas minables, mediante sentencias SQL de la base de datos con los ejemplos etiquetados, estas consultas serán las que mediante la herramienta WEKA nos permitirán realizar la tarea de minería de datos. Un ejemplo de estas consultas se muestra en la Figura 4.4.

```
SELECT num_accesos_curso, num_aportes_curso, num_preguntas_profesor,
num_pags_curso_visitadas, num_respuestas_foros,
tiempo_promedio_interaccion, (num_tareas_propuestas-
num_tareas_realizadas), valor
FROM interaccion i, atributos a
WHERE i.id = a.id_interaccion and parametro = 'nivel_interes_curso'
```

Figura 4.4: Consulta para la vista minable del atributo nivel de interés en el curso

4.3.4 Minería de datos

En la minería de datos, propiamente dicha, se toman los datos depurados y se transforman en conocimiento, que en nuestro caso se traduce en la generación de los atributos, necesarios para el modelo de usuario, que luego en la construcción del agente se obtendrán definiendo tareas de aprendizaje. Además en esta fase se evalúan diferentes técnicas y algoritmos de aprendizaje automático para elegir las más apropiadas para los datos seleccionados. Para estas actividades se presenta el carácter iterativo del proceso de KDD ya que se necesita explorar la construcción de diferentes modelos hasta encontrar el que mejor se ajuste a la inferencia de los atributos planteados, cabe mencionar que esto se realiza para cada uno de los atributos que se plantean generarlos mediante aprendizaje supervisado. Este proceso de minería está sustentado por la herramienta, open source, de minería de datos y aprendizaje

automático WEKA.

Lo que se busca con la aplicación de estas técnicas es la generación de modelos que describan patrones y relaciones entre los datos que se usan para realizar predicciones en instancias futuras, y según Hernández, Ramírez y Ferri (2004) hay que tomar varias decisiones antes de la ejecución de este proceso:

- ◆ Elegir el tipo de tarea de minería a utilizar
- ◆ Elegir el tipo de modelo que se empleara
- ◆ Elegir el algoritmo que resuelva la tarea planteada u obtenga el tipo de modelo buscado.

En nuestro caso las decisiones acerca de estos puntos dependen principalmente de los datos de entrada que se necesita procesar y de los estudios referentes a la aplicación de técnicas de aprendizaje automático y minería de datos a modelos de usuario.

En los párrafos iniciales de esta sección se menciona que los atributos del modelo de usuario a probar necesitan ajustarse a un modelo predictivo de las instancias, dentro de este tipo de modelos resaltan dos tipos de tareas, que son la clasificación y la regresión, como opciones de producción de este tipo de modelos, de las investigaciones estudiadas la tarea más utilizada y que mantiene los mejores resultados es la clasificación, así que esta es la elección para este proyecto.

En segunda instancia hay que elegir el tipo de modelo que se empleara, en esta parte la elección está sujeta a la experimentación con árboles de decisión, redes neuronales, redes bayesianas y combinaciones de clasificadores (método bagging).

Finalmente se eligen algoritmos que permitan implementar los modelos seleccionados, para esto WEKA cuenta con un conjunto importante de algoritmos que sirven para la generación de modelos en todos los tipos seleccionados, así que para cada técnica de minería en la tabla 4.7 se muestran los algoritmos que se utilizan para la experimentación.

Tabla 4.6: Algoritmos de prueba y de generación de modelos

Tipos de Modelos	Algoritmos
Árboles de Decisión	<ul style="list-style-type: none"> • C4.5 (J48) • REPTree
Redes Neuronales	<ul style="list-style-type: none"> • MultilayerPerceptron
Redes Bayesianas	<ul style="list-style-type: none"> • NaiveBayes • BayesNet
Combinación de Clasificadores	<ul style="list-style-type: none"> • Bagging (J48) • Bagging(REPTree) • Bagging(BayesNet) • Bagging(NaiveBayes)

Se utiliza esencialmente combinación de clasificadores con árboles de decisión y métodos bayesianos, debido a la amplia bibliografía existente acerca de la aplicación de estas técnicas en diferentes proyectos relacionados con la predicción de atributos de modelos de usuario mediante aprendizaje automático, y la primera por ser una propuesta relativamente nueva que busca maximizar la ejecución de las técnicas aplicadas anteriormente, además se utilizan métodos basados en redes neuronales para experimentar resultados en su aplicación a este tipo de proyectos. Se cree que los datos que presentan de las vista minables diseñadas cumplen con los requerimientos de todos los algoritmos seleccionados para su ejecución.

Ahora que ya está todo listo, se puede proseguir con la experimentación con los algoritmos seleccionados para el caso, tratando en cada caso tomar el mayor número de experiencias y resultados para una mayor veracidad en la interpretación de los mismos.

Después de la experimentación dentro de la herramienta elegida para el caso (WEKA) se generaron un conjunto de archivos en los cuales se guardan la información referente a la ejecución de cada algoritmo con los parámetros recopilados de los atributos del modelo de usuario elegidos para el caso. Estos archivos se organizaron en tres directorios, y contienen información acerca de la ejecución del algoritmo los archivos .txt, un archivo .model que contiene el modelo generado para el algoritmo ejecutado y en los casos en los que la herramienta permite una representación gráfica del modelo (J48, REPTree, BayesNet) se genero un archivo .jpg con una imagen del árbol o red generada por el algoritmo. En la tabla 4.7 se presenta un resumen de resultados obtenidos para cada atributo, designados por medio

de porcentajes, y niveles de error.

Tabla 4.7: Resumen de resultados para cada algoritmo ejecutado

Nivel Interés Curso				
Algoritmo	Tiempo de construcción del modelo (Segundos)	Instancias clasificadas correctamente	Instancias clasificadas incorrectamente	Error absoluto medido
C4.5 (J48)	0.13	2285 - 96.9042 %	73 - 3.0958 %	0.0277
REPTree	0.06	2273 - 96.3953 %	85 - 3.6047 %	0.0341
MultilayerPerceptron	9.7	2249 - 95.3774 %	109 - 4.6226 %	0.0428
NaiveBayes	0.06	2211 - 93.7659 %	147 - 6.2341 %	0.0435
BayesNet	0.05	2280 - 96.6921 %	78 - 3.3079 %	0.0297
Bagging (J48)	0.45	2287 - 96.989 %	71 - 3.011 %	0.0283
Bagging(REPTree)	3.27	2283 - 96.8193 %	75 - 3.1807 %	0.0321
Bagging(BayesNet)	0.38	2286 - 96.9466 %	72 - 3.0534 %	0.0307
Bagging(NaiveBayes)	0.14	2214 - 93.8931 %	144 - 6.1069 %	0.043
Nivel Interés Recursos				
Algoritmo	Tiempo de construcción del modelo (Segundos)	Instancias clasificadas correctamente	Instancias clasificadas incorrectamente	Error absoluto medido
C4.5 (J48)	0.02	2331 - 99.9143 %	2 - 0.0857 %	0.0008
REPTree	0.001	2332 - 99.9571 %	1 - 0.0429 %	0.0007
MultilayerPerceptron	3.61	2313 - 99.1427 %	20 - 0.8573 %	0.0043
NaiveBayes	0.001	2279 - 97.6854 %	54 - 2.3146 %	0.0167
BayesNet	0.001	2332 - 99.9571 %	1 - 0.0429 %	0.0015
Bagging (J48)	0.03	2331 - 99.9143 %	2 - 0.0857 %	0.001
Bagging(REPTree)	0.03	2332 - 99.9571 %	1 - 0.0429 %	0.0008
Bagging(BayesNet)	0.06	2332 - 99.9571 %	1 - 0.0429 %	0.0016
Bagging(NaiveBayes)	0.03	2278 - 97.6425 %	55 - 2.3575 %	0.0168
Nivel Interacción				
Algoritmo	Tiempo de construcción del modelo (Segundos)	Instancias clasificadas correctamente	Instancias clasificadas incorrectamente	Error absoluto medido
C4.5 (J48)	0.06	2291 - 98.1997 %	42 - 1.8003 %	0.0123
REPTree	0.22	2281 - 97.7711 %	52 - 2.2289 %	0.0166
MultilayerPerceptron	12.36	2256 - 96.6995 %	77 - 3.3005 %	0.0206

NaiveBayes	0.06	2218 - 95.0707 %	115 - 4.9293 %	0.027
BayesNet	0.09	2288 - 98.0712 %	45 - 1.9288 %	0.0119
Bagging (J48)	0.09	2295 - 98.3712 %	38 - 1.6288 %	0.0126
Bagging(REPtree)	0.3	2291 - 98.1997 %	42 - 1.8003 %	0.0159
Bagging(BayesNet)	0.59	2293 - 98.2855 %	40 - 1.7145 %	0.0119
Bagging(NaiveBayes)	0.2	2215 - 94.9421 %	118 - 5.0579 %	0.0267

4.3.5 Evaluación e interpretación

En la fase de evaluación e interpretación se busca comprobar que el modelo construido guarda un cierto grado de credibilidad en cuanto a la determinación de las clases de las nuevas instancias.

De una forma ideal se desea que los modelos generados sean, precisos, comprensibles e interesantes, pero dependiendo del dominio de aplicación de los mismos se puede estar dispuesto a sacrificar alguna de estas características para potenciar otras.

Para los modelos generados en este trabajo se eligió una técnica de alta aplicación en la validación de modelos generados por algoritmos de aprendizaje automático. Esta técnica se denomina *validación cruzada con n pliegues*⁸ (Hernández, Ramírez y Ferri, 2004), en esta técnica los datos se dividen aleatoriamente en n grupos, de los cuales uno se reserva para el conjunto de prueba y con los grupos restantes se construye un modelo que se usa para predecir los valores del grupo reservado para la prueba. A este proceso se lo repite n veces (n folds), tomando un grupo diferente para cada prueba. Esto permite obtener índices de error en cada una de las pruebas que se promedian y permiten inferir la precisión del modelo creado.

Los parámetros que se promedian para la clasificación generalmente incluyen la evaluación de la *precisión predictiva*, así como la evaluación de lo que se conoce como *matriz de confusión*. La primera nos indica que tan eficiente es el algoritmo para clasificar las instancias y la segunda muestra los aciertos tenidos en la clasificación de instancias, contrastados con los niveles de error en cada clase

⁸ *Validación Cruzada con n pliegues*. Proviene del inglés *n-fold cross validation*.

probada.

Para el proceso propuesto se utilizó esta técnica para la validación de los modelos, que viene implementada dentro de la herramienta WEKA, con 10 folds como es comúnmente implementada. Y se evaluaron la precisión predictiva y la matriz de confusión. Los resultados obtenidos se analizan a continuación.

a) Evaluando la precisión predictiva

En la tabla 4.8 se muestra una comparativa de los resultados obtenidos, en lo que se refiere a la precisión al clasificar las instancias, por cada algoritmo y para cada atributo evaluado, además se incorpora un ordenamiento de los algoritmos para cada atributo, según los resultados obtenidos.

Tabla 4.8: Comparativa de la precisión de algoritmos aplicados a atributos del modelo de usuario

Algoritmo	Nivel interés curso	Pos	Nivel interés recursos	Pos	Nivel de interacción	Pos
C4.5 (J48)	96.90%	2	99.91%	6	98.20%	3
REPTree	96.40%	7	99.96%	4	97.77%	6
MultilayerPerceptron	95.38%	8	99.14%	7	96.70%	7
NaiveBayes	93.77%	9	97.69%	8	95.07%	8
BayesNet	96.69%	6	99.96%	3	98.07%	5
Bagging (J48)	96.99%	1	99.91%	5	98.37%	1
Bagging(REPTree)	96.82%	5	99.96%	2	98.20%	4
Bagging(BayesNet)	96.95%	3	99.96%	1	98.29%	2
Bagging(NaiveBayes)	93.89%	4	97.64%	9	94.94%	9

Como se puede ver a primera vista los resultados para cada algoritmo cambian en cierto grado debido a al comportamiento diverso del modelo generado con cada algoritmo, pero también resalta que las técnicas con mayor acierto en los tres atributos son aquellas que se implementan mediante arboles de decisión y los métodos bayesianos especialmente BayesNet, con y sin combinación de clasificadores. Además se nota que la precisión predictiva de cada modelo varía dependiendo del número de atributos con los que se genera cada uno, por ejemplo, para el atributo nivel de interés en recursos los resultados son más precisos porque para este atributo se utilizaron tres parámetros para su evaluación mientras que en los otros dos se

utilizaron muchos más, pero comparando los resultados entre los atributos nivel de interés en el curso y nivel de interacción se produce el caso contrario ya que primero necesita más parámetros que el segundo, para tener una mayor precisión en sus predicciones.

Luego de hacer una comparación global de los resultados, se compara cada algoritmo en cada atributo. Para el atributo nivel de interés en el curso, el modelo con el mejor resultado es aquel que fue implementado mediante la combinación de clasificaciones con el método Bagging del algoritmo J48 con un 96.99% de acierto en la clasificación de instancias que demuestra un grado aceptable en la ejecución del mismo ya que de 2358 instancias propuestas 2287 fueron clasificadas correctamente. El siguiente modelo con un grado similar de precisión en la clasificación es aquel generado por el algoritmo J48 que tiene un 96.90% de acierto en la clasificación de instancias con 2285 correctamente clasificadas de las 2358 disponibles, así que como se ve el usar una combinación de clasificadores no representa una ventaja significativa en la clasificación de este atributo, sin contar que la implementación del método combinatorio conlleva un gasto mucho mayor de recursos. En el resto de algoritmos la tendencia es la pérdida del grado de precisión de los algoritmos en un grado no muy amplio.

Para el atributo nivel de interés en recursos el escenario cambia un poco ya que tres algoritmos presentan porcentajes idénticos en la precisión con la que los modelos generados con ellos clasifican las instancias, estos son; BayesNet, BayesNet con combinación de clasificaciones y el REPTree con combinación de clasificaciones, ya que los tres presentan un 99.96% de acierto en las clasificaciones y de 2233 instancias disponibles clasificaron 2232 correctamente, entonces, la forma de calificarlos dependerá de los niveles de error y de la complejidad de su implementación, a simple vista se cree que el más oicionado es el modelo generado por el algoritmo BayesNet ya que comparado con las combinaciones de clasificadores de los otros algoritmos su dificultad de implementación y tiempo de respuesta en la clasificación es mucho menor. El resto de algoritmos, al igual que con el atributo nivel de interés en el curso, sufren pequeñas pérdidas de precisión en la clasificación de instancias.

Finalmente para el atributo nivel de interacción, el modelo con mayor grado de

precisión de clasificación de instancias es aquel generado con el algoritmo J48 mediante combinación de clasificadores con un 98.37% de grado de acierto en la clasificación, que cuantificando estos resultados, significa que de 2333 se han clasificado 2295 instancias correctamente. El modelo que le sigue en porcentaje de precisión es el modelo generado mediante el algoritmo REPTree implementando combinación de clasificadores con un 98.20% de grado de acierto en la clasificación de instancias, lo que significa que de 2333 instancias propuestas 2291 fueron clasificadas correctamente. Resalta para este atributo que la combinación de clasificadores es la alternativa mejor opcionada, se toma como opción ya que el nivel de precisión de la clasificación, comparada con los otros atributos, es bajo y solo tomar en cuenta modelos generados por algoritmos no combinados genera un grado de error considerable. Los resultados obtenidos con los modelos generados por los algoritmos REPTree, NaiveBayes y BayesNet corroboran lo antes mencionado.

b) Evaluando la matriz de confusión

Luego de comparar los resultados obtenidos por los algoritmos en cuanto a su porcentaje de precisión, se presenta una comparación de los resultados obtenidos mediante la matriz de confusión, que se genera por la comprobación de los modelos implementados por cada uno de los algoritmos estudiados. En la matriz de confusión los índices de columnas y filas representan las clases que intervienen en la clasificación, en la diagonal principal se presentan las instancias correctamente clasificadas para cada clases y en las demás celdas se muestran las instancias clasificadas incorrectamente en las clases que se ubicaron. Lo cual muestra una visión más global de la efectividad de un algoritmo de clasificación para un conjunto de datos propuestos.

En la figura 4.5 se muestran las matrices de confusión obtenidas de todos los modelos generados para el atributo nivel de interés en el curso.

Evaluando el porcentaje de precisión de clasificación para este atributo el modelo más opcionado para la clasificación fue el generado con el algoritmo J48, la matriz de confusión que presenta este modelo es la más equilibrada y presenta menos discrepancias en la clasificación ya que los valores en las celdas que acompañan a la

diagonal principal son los más bajos comparadas con las matrices generadas por los modelos de los otros algoritmos. El modelo generado por el método bagging implementando un árboles de decisión J48 presenta resultados similares, pero queda relegado a segundo plano por perder al ser más retardada su ejecución.

<p>==== Algoritmo J48 ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2020</td> <td>17</td> <td>0</td> <td>a = bajo</td> </tr> <tr> <td>19</td> <td>205</td> <td>14</td> <td>b = medio</td> </tr> <tr> <td>1</td> <td>22</td> <td>60</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2020	17	0	a = bajo	19	205	14	b = medio	1	22	60	c = alto	<p>==== Algoritmo REPTree ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2017</td> <td>19</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>23</td> <td>198</td> <td>17</td> <td>b = medio</td> </tr> <tr> <td>1</td> <td>24</td> <td>58</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2017	19	1	a = bajo	23	198	17	b = medio	1	24	58	c = alto	<p>==== Algoritmo MultilayerPerceptron ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>1990</td> <td>46</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>34</td> <td>196</td> <td>8</td> <td>b = medio</td> </tr> <tr> <td>1</td> <td>19</td> <td>63</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	1990	46	1	a = bajo	34	196	8	b = medio	1	19	63	c = alto
a	b	c	<-- clasificado como																																															
2020	17	0	a = bajo																																															
19	205	14	b = medio																																															
1	22	60	c = alto																																															
a	b	c	<-- clasificado como																																															
2017	19	1	a = bajo																																															
23	198	17	b = medio																																															
1	24	58	c = alto																																															
a	b	c	<-- clasificado como																																															
1990	46	1	a = bajo																																															
34	196	8	b = medio																																															
1	19	63	c = alto																																															
<p>==== Algoritmo NaiveBayes ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>1990</td> <td>46</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>34</td> <td>196</td> <td>8</td> <td>b = medio</td> </tr> <tr> <td>1</td> <td>19</td> <td>63</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	1990	46	1	a = bajo	34	196	8	b = medio	1	19	63	c = alto	<p>==== Algoritmo BayesNet ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2012</td> <td>24</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>14</td> <td>206</td> <td>18</td> <td>b = medio</td> </tr> <tr> <td>0</td> <td>21</td> <td>62</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2012	24	1	a = bajo	14	206	18	b = medio	0	21	62	c = alto	<p>==== Algoritmo Bagging(J48) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2019</td> <td>18</td> <td>0</td> <td>a = bajo</td> </tr> <tr> <td>19</td> <td>205</td> <td>14</td> <td>b = medio</td> </tr> <tr> <td>0</td> <td>20</td> <td>63</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2019	18	0	a = bajo	19	205	14	b = medio	0	20	63	c = alto
a	b	c	<-- clasificado como																																															
1990	46	1	a = bajo																																															
34	196	8	b = medio																																															
1	19	63	c = alto																																															
a	b	c	<-- clasificado como																																															
2012	24	1	a = bajo																																															
14	206	18	b = medio																																															
0	21	62	c = alto																																															
a	b	c	<-- clasificado como																																															
2019	18	0	a = bajo																																															
19	205	14	b = medio																																															
0	20	63	c = alto																																															
<p>==== Algoritmo Bagging(REPTree) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2015</td> <td>22</td> <td>0</td> <td>a = bajo</td> </tr> <tr> <td>22</td> <td>203</td> <td>13</td> <td>b = medio</td> </tr> <tr> <td>0</td> <td>18</td> <td>65</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2015	22	0	a = bajo	22	203	13	b = medio	0	18	65	c = alto	<p>==== Algoritmo Bagging(BayesNet) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2014</td> <td>22</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>15</td> <td>208</td> <td>15</td> <td>b = medio</td> </tr> <tr> <td>0</td> <td>19</td> <td>64</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2014	22	1	a = bajo	15	208	15	b = medio	0	19	64	c = alto	<p>==== Algoritmo Bagging(NaiveBayes) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>1955</td> <td>75</td> <td>7</td> <td>a = bajo</td> </tr> <tr> <td>24</td> <td>196</td> <td>18</td> <td>b = medio</td> </tr> <tr> <td>0</td> <td>20</td> <td>63</td> <td>c = alto</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	1955	75	7	a = bajo	24	196	18	b = medio	0	20	63	c = alto
a	b	c	<-- clasificado como																																															
2015	22	0	a = bajo																																															
22	203	13	b = medio																																															
0	18	65	c = alto																																															
a	b	c	<-- clasificado como																																															
2014	22	1	a = bajo																																															
15	208	15	b = medio																																															
0	19	64	c = alto																																															
a	b	c	<-- clasificado como																																															
1955	75	7	a = bajo																																															
24	196	18	b = medio																																															
0	20	63	c = alto																																															

Figura 4.5: Matrices de confusión para el atributo nivel de interés en el curso

Para el atributo nivel de interés en recursos las matrices de confusión construidas de la ejecución de los diferentes algoritmos se exponen en la figura 4.6.

<p>==== Algoritmo J48 ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2237</td> <td>0</td> <td>2</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2237	0	2	a = bajo	0	15	0	b = alto	0	0	79	c = medio	<p>==== Algoritmo REPTree ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2238</td> <td>0</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2238	0	1	a = bajo	0	15	0	b = alto	0	0	79	c = medio	<p>==== Algoritmo MultilayerPerceptron ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2236</td> <td>0</td> <td>3</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>0</td> <td>15</td> <td>b = alto</td> </tr> <tr> <td>2</td> <td>0</td> <td>77</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2236	0	3	a = bajo	0	0	15	b = alto	2	0	77	c = medio
a	b	c	<-- clasificado como																																															
2237	0	2	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															
a	b	c	<-- clasificado como																																															
2238	0	1	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															
a	b	c	<-- clasificado como																																															
2236	0	3	a = bajo																																															
0	0	15	b = alto																																															
2	0	77	c = medio																																															
<p>==== Algoritmo NaiveBayes ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2185</td> <td>0</td> <td>54</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2185	0	54	a = bajo	0	15	0	b = alto	0	0	79	c = medio	<p>==== Algoritmo BayesNet ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2238</td> <td>0</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2238	0	1	a = bajo	0	15	0	b = alto	0	0	79	c = medio	<p>==== Algoritmo Bagging(J48) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2237</td> <td>0</td> <td>2</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2237	0	2	a = bajo	0	15	0	b = alto	0	0	79	c = medio
a	b	c	<-- clasificado como																																															
2185	0	54	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															
a	b	c	<-- clasificado como																																															
2238	0	1	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															
a	b	c	<-- clasificado como																																															
2237	0	2	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															
<p>==== Algoritmo Bagging(REPTree) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2238</td> <td>0</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2238	0	1	a = bajo	0	15	0	b = alto	0	0	79	c = medio	<p>==== Algoritmo Bagging(BayesNet) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2238</td> <td>0</td> <td>1</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2238	0	1	a = bajo	0	15	0	b = alto	0	0	79	c = medio	<p>==== Algoritmo Bagging(NaiveBayes) ====</p> <table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>c</th> <th><-- clasificado como</th> </tr> </thead> <tbody> <tr> <td>2184</td> <td>0</td> <td>55</td> <td>a = bajo</td> </tr> <tr> <td>0</td> <td>15</td> <td>0</td> <td>b = alto</td> </tr> <tr> <td>0</td> <td>0</td> <td>79</td> <td>c = medio</td> </tr> </tbody> </table>	a	b	c	<-- clasificado como	2184	0	55	a = bajo	0	15	0	b = alto	0	0	79	c = medio
a	b	c	<-- clasificado como																																															
2238	0	1	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															
a	b	c	<-- clasificado como																																															
2238	0	1	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															
a	b	c	<-- clasificado como																																															
2184	0	55	a = bajo																																															
0	15	0	b = alto																																															
0	0	79	c = medio																																															

Figura 4.6: Matrices de confusión para el atributo nivel de interés en recursos

Lo que se muestra en estas matrices refleja una homogeneidad en los resultados obtenidos por varios algoritmos, lo que resalta es la que los algoritmos en que

coinciden los resultados (BayesNet, Bagging(BayesNet) y Bagging(REPTree)), el grado de clasificación es muy alto ya que el grado de error se resume en una instancia mal clasificada. Los criterios de selección del modelo más apropiado para clasificar este atributo deben provenir de evaluar el tiempo de retardo en ejecución del modelo y la complejidad de implementación de cada uno de los modelos producidos por cada algoritmo.

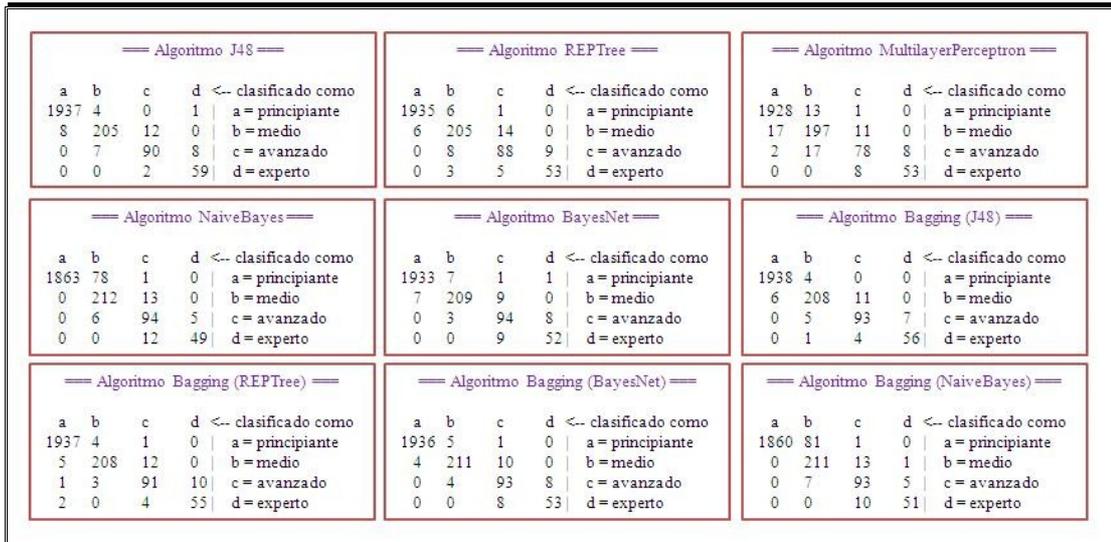


Figura 4.7: Matrices de confusión para el atributo nivel de interacción

Para el atributo nivel de interacción se presentan las matrices de confusión en la figura 4.7.

En este atributo los niveles de acierto en las predicciones son menores debido a la introducción de un número mayor de clases, pero como se muestra los valores no son despreciables. Es más se demuestra una correlación con los otros atributos en el sentido de que los modelos con mejores resultados son aquellos que se implementan mediante algoritmos que se basan en la representación de arboles de decisión y redes bayesianas, la combinación de clasificadores no hace más que potenciar el resultado predictivo que puede implementar un algoritmo.

En este caso, y al igual que en la evaluación de la precisión predictiva, el modelo que presenta el grado menor de error a la hora de clasificar las instancias es el modelo generado por el algoritmo J48 mediante la combinación de clasificadores siguiéndolo muy de cerca el algoritmo REPTree igualmente mejorado mediante combinación de

clasificadores, En este caso se hace necesaria la implementación de esta técnica ya que los resultados con los modelos que no la implementan presentan un grado de error considerable como lo es el caso de modelo generado mediante el algoritmo NaiveBayes que presenta 79 instancias de la clase principiante mal clasificadas como medio o experto lo que lleva a suponer que su implementación para este atributo podría llegar a ser improductiva.

c) Evaluación general

Con los resultados obtenidos se pueden inferir algunas ideas que se han hecho presentes en el transcurso de la experimentación con los atributos y modelos planteados y generados. Una de ellas es que los atributos guardan cierto grado de relación ya que los métodos más efectivos a la hora de clasificar las instancias han sido semejantes, y coincidentalmente son los métodos que han tenido los mejores resultados en proyectos similares con lo son Gaudioso (2002) e Itmazi (2005), lo que varia es la forma de implementarlos ya que para un atributo (nivel de interés en el curso) el método para su inferencia es simple comparado con otro (nivel de interacción), que necesita un método más sofisticado que maximice el grado de predicción que pueda generar el algoritmo base.

Dado lo mencionado anteriormente se hace necesario que para atributos nuevos a incorporarse posteriormente al modelo de usuario y que se necesite que se infieran mediante técnicas de aprendizaje automático, se generen modelos con distintos algoritmos y se prueben similarmente a las evaluaciones desarrolladas para los atributos propuestos en este trabajo, para que se tenga certeza verdadera que los modelos utilizados para inferir un atributo considerado son los más efectivos en esta tarea.

En los párrafos anteriores se ha descrito el proceso llevado cabo para inferir conocimiento acerca que algunos atributos pertenecientes al modelo de usuario de los estudiantes de la UTPL. Se tiene la convicción que incorpora los pasos necesarios y pertinentes para la construcción de modelos validos, novedosos y potencialmente útiles como es el ideal del proceso de KDD (Fayyad, Piatetsky y Smyth 1996).

En resumen el la tabla 4.9 se demuestran las elecciones desarrolladas para cada

atributo, del algoritmo de aprendizaje automático que desarrolla de mejor forma la clasificación de nuevas instancias.

Tabla 4.9: Selección de algoritmos aprendizaje automático para atributos del modelos de usuario

ATRIBUTO	Algoritmo del Modelo
Nivel de interés en el curso	C4.5 (J48)
Nivel de interés en recursos	BayesNet, K2, TAN
Nivel de interacción	Bagging (J48)

Se demuestra entonces que no se puede aplicar una técnica genérica a todos los atributos a inferir dentro del modelo de usuario. Para cada nuevo atributo que se incorpore al modelo de usuario se necesita que la selección de los algoritmos de generación de los modelos pasen por un proceso de extracción de conocimiento que delimite adecuadamente, primero las variables y clases que definen el atributo y luego escoja los modelos más apropiados que se acoplen a los datos obtenidos para que los resultados de las clasificaciones sean los esperados.

Capítulo 5

*Diseño del modelo de usuario
para el EVA UTPL*

5.1 Modelo de usuario de los estudiantes de la UTPL

En los capítulos anteriores se han sentado las bases para la definición del modelo de usuario de los estudiantes de la UTPL. Se ha discutido la estructura y los componentes que este debe tener, así como los orígenes de la información que va generar luego de un proceso de modelado en los atributos del modelo. En las secciones de este capítulo, se describe la propuesta de modelo de usuario para los estudiantes de la UTPL. Para esta propuesta se han tomado en cuenta elementos que son indispensables para que este modelo pueda reflejar información de utilidad para adaptaciones dentro del entorno de aprendizaje virtual de aprendizaje de la UTPL.

5.1.1 Descripción global del modelo

Para el modelo de usuario del entorno virtual de aprendizaje de la UTPL se contempla el diseño de catorce atributos principales divididos en tres categorías, así:

- ◆ **Categoría 1:** *Datos de entorno.*- Para los datos de entorno se han definido dos atributos el uno es la *zona geográfica*, y el otro es *entorno de usuario*.
- ◆ **Categoría 2:** *Datos de usuario.*- En la categoría de datos de usuario se han definido seis atributos: *estilo de aprendizaje*, *nombre de usuario*, *cedula*, *nombres*, *carrera*, *curso* y *modalidad*.
- ◆ **Categoría 3:** *Datos de interacción.*- Aquí, en la categoría de datos de interacción se ha escogido tres atributos principales, *nivel de interés*, *nivel de interacción* y *acciones sobre objetos*.

La agrupación de estos elementos es importante ya que esta define la naturaleza de los mismos, y además facilita la incorporación de nuevos atributos al modelo. Como se ve esta representación obedece a la delimitación de elementos mínimos de un modelo de usuario descrita en la sección 1.5.3 del Capítulo 1. Vale mencionar que la obtención de cada uno de estos atributos estará sujeta a un mecanismo diferente en cada caso, ya que en algunos se utilizara aprendizaje supervisado para la inferencia de los mismos, en otros se realizara mediante reglas de inferencia y en otros se hará una simple obtención directa del sistema.

5.1.2 Descripción de los elementos del modelo

En esta sección describiremos cada elemento que compone el modelo de usuario, y el origen de cada uno de ellos, así como los datos que son necesarios para su inferencia, cabe mencionar que se describió de forma completa estos datos en la sección 1.7 del Capítulo 1 y los orígenes en el EVA en la sección 3.3.3 del Capítulo 3.

a) Descripción de los datos de entorno

Los Datos de Entorno nos ayudaran a definir la localidad del estudiante, para poder ubicarlo en una región o en una población geográfica a la que pertenece dicho estudiante, así como los componentes de hardware y software que el usuario frecuentemente utiliza, para que las adaptaciones que se le presenten estén acorde al tipo de dispositivo con el cual accede al EVA. Dentro de esta categoría de atributos se describen:

- ◆ *Zona geográfica.*- Atributo que describe la localización del estudiante, por medio de este campo se trata de ubicar geográficamente al estudiante. Para este atributo se han definido zonas con las que se van a trabajar que no necesariamente contemplan las geográficamente definidas sino mas bien por la afluencia de estudiantes en dicha zona, esta afluencia de estudiantes fue medida obtenida en la investigación reflejada en la sección 4.3 del Capítulo 4. Las zonas que se han diseñado son las siguientes: *Sierra Norte, Sierra Sur, Costa, Oriente, Insular, América del Sur, América Central y EU, Europa y Otros.* Además al ser un atributo compuesto, consta de parámetros como el país en el que está matriculado el estudiante, la ciudad y el centro al que pertenece dicho estudiante.
- ◆ *Entorno de usuario.*- Atributo que describe el medio mediante el cual el usuario accede al sistema y las implicaciones con las que supone el acceso a los recursos que el sistema brinda a los usuarios. Para este atributo se han contemplado algunas tipologías de medios de acceso que delimitan el entorno con el que el usuario accede al sistema. Y estas son: *tipología PC, tipología dispositivo de acceso móvil.*

b) Descripción de los datos de usuario

Los datos de usuario definen al mismo, es decir, identifican al usuario y a su modelo de usuario en el curso en cuestión. Para estos atributos las tipologías ya se encuentran definidas dentro del EVA y los métodos de su extracción se desarrollan directamente del sistema por lo que no se necesita del trato especializado de dicha información.

c) Descripción de los datos de interacción

Los datos de interacción definen los niveles de interés de los estudiantes en los contenidos de las asignaturas, así como el nivel de intercambio de información entre el usuario y el sistema, otro atributo que se define con un grado alto de importancia es aquel que contempla las acciones que se realizan sobre los objetos, o componentes de la plataforma de soporte a la educación. En esta categoría de atributos definimos los siguientes:

- ◆ *Nivel de interés.*- Es aquel atributo que demuestra el grado de interés que el estudiante presenta en los contenidos de la asignatura que está cursando, estos datos ayudarán, combinados con otros del modelo, a que se puedan hacer sugerencias apropiadas de contenidos a los estudiantes que presentan un nivel inferior de interés, y temas más avanzados a estudiantes en los que el nivel de interés es más alto. Para el atributo en mención se han diseñado tres niveles de interés que engloban las diferentes etapas de crecimiento de este atributo de los estudiantes en temas de una asignatura en concreto, y son: alto, medio y bajo.
- ◆ *Nivel de interacción.*- Este atributo define los niveles de interacción de los estudiantes con el Entorno Virtual de Aprendizaje. Para así saber si los contenidos que se están trabajando con los educandos satisfacen los requerimientos educativos de los mismos. Para este atributo se han definido cuatro niveles de interacción que similarmente a los niveles de interés, engloban las diferentes fases de interacción de los estudiantes con el sistema y las herramientas que en él se ejecutan, y son: *principiante, medio, avanzado y experto*
- ◆ *Acciones sobre objetos.*- Las acciones sobre objetos, entendidas como las

acciones que los estudiantes realizan sobre las herramientas (foros, chats, tareas, evaluaciones y recursos) que los tutores despliegan en cada uno de los cursos, se han contemplado en el modelo de usuario para obtener indicadores de las actividades que los educandos realizan con el sistema. Para este atributo se han incluido acciones que se pueden desarrollar con los mismos, y estas son: *ver*, *ver discusión*, *añadir discusión*, *subir archivo*, *descargar archivo*, *historial*, *añadir post* y *actualizar post*.

5.1.3 Mecanismos de identificación de atributos del modelo de usuario

Así como se han presentado a los componentes del modelo de usuario también se presentan las formas en que estos datos se obtendrán del sistema para lo cual en esta sección se describen un conjunto de técnicas de extracción de datos y los atributos del modelo que serán obtenidos mediante ellas.

a) Extracción directa de los datos del sistema

Con esta técnica se extrae los datos directamente del sistema mediante consultas SQL y se almacenan en el modelo de usuario, el único trato que reciben estos datos es una transformación para su estandarización. Mediante esta técnica de extracción de datos se obtendrán los siguientes:

- ◆ Nombre de usuario
- ◆ Nombre
- ◆ Apellido
- ◆ Ciudad
- ◆ País
- ◆ Ultimo acceso
- ◆ Ultimo login
- ◆ Ultimo ip
- ◆ Contraseña

- ◆ email
- ◆ Dirección
- ◆ Teléfono
- ◆ Cedula
- ◆ Carrera
- ◆ Curso
- ◆ Modalidad

b) Reglas de adquisición de atributos

Mediante esta técnica se utilizan pequeñas reglas que transforman los datos en otros, de forma que sean más entendibles o presenten un grado mayor de utilidad en el trabajo posterior con los mismos. Se piensa obtener mediante esta técnica los siguientes atributos del modelo:

- ◆ Zona geográfica
- ◆ Entorno de usuario
- ◆ Acciones sobre objetos

c) Test electrónicos

Una técnica en la que el principal proveedor de información es el usuario, el que, mediante cuestionarios, es capaz de proveer datos directamente al sistema una descripción completa de esta técnica la encontramos en el inciso a) de la sección 1.7.4 del Capítulo 1. Mediante esta técnica el atributo a obtener es el *estilo de aprendizaje* que posee el estudiante, mediante el denominado test de Felder seleccionado en la sección 3.3.2 del Capítulo 3.

d) Tareas de aprendizaje

Finalmente la técnica que presenta un grado mayor de complejidad comparada con las anteriores, ya que para su desarrollo se necesita utilizar aprendizaje automático para calcular el valor de dichos atributos, es aquella que se basa en el diseño de tareas

de aprendizaje que infieran los valores de los atributos mediante el estudio de un conjunto de interacciones del usuario con el sistema. Estas tareas de aprendizaje fueron definidas a mayor detalle en la sección 1.9.1 del Capítulo 1. Mediante esta técnica los atributos a obtener del modelo de usuario son los siguientes:

- ◆ Nivel de Interés en el Curso
- ◆ Nivel de Interés en Recursos
- ◆ Nivel de Interacción

El diseño de las tareas de aprendizaje propuestas se desarrolla en la sección siguiente de este capítulo entrando en detalle en su representación.

5.2 Diseño de tareas de aprendizaje para el modelo de usuario

En esta sección se presentan las tareas de aprendizaje diseñadas para el modelo de usuario del entorno virtual de aprendizaje de los estudiantes de la UTPL. Así como mostrar una propuesta extensible que permita incluir nuevas tareas de aprendizaje al modelo propuesto, lo que conlleva el enriquecimiento del modelo con nuevos atributos.

Las tareas propuestas en esta sección provienen de los datos de interacción del usuario (sección 3.3.3 c). y el estudio para su ejecución se muestra en la minería de datos desarrollada en el Capítulo 4. Son tareas de clasificación similares a las propuestas por Gaudioso (2002).

5.2.1 Definición de tareas de aprendizaje

Las tareas de aprendizaje consideradas para el modelo de usuario de los estudiantes de la UTPL se definen a continuación:

a) Nivel de interés en el curso

Tarea de aprendizaje en la que se clasifica a los estudiantes en clases de interés midiendo las interacciones, aportaciones y cuestionamientos que los estudiantes han realizado en el sistema. Para lo cual en la sección 4.3.5 del Capítulo 4 se evalúan

técnicas de clasificación que generan modelos de predicción para este atributo. Los niveles diseñados para la clasificación de los estudiantes son: *bajo, medio y alto*.

b) Nivel de interés en recursos

En esta tarea de aprendizaje se clasifica a los estudiantes en niveles de interés en los recursos (foros, chat, tareas, etc.) . Igualmente se evalúan las técnicas que generan los mejores modelos para clasificar este atributo en la sección sección 4.3.5 del Capítulo 4. Los niveles de interés diseñados para la evaluación de este atributo en los datos obtenidos son: *bajo, medio y alto*.

c) Nivel de interacción en la plataforma

En esta tarea se busca medir el nivel en que los estudiantes interactúan con el entorno, midiendo para ello la cantidad de accesos a los cursos, las acciones sobre los recursos propuestos, así como el tiempo promedio de interacción con el sistema. Las técnicas evaluadas para la clasificación de este atributo son presentadas en la sección sección 4.3.5 del Capítulo 4. Los niveles de interacción estimados para evaluación son: *principiante, medio avanzado y experto*.

5.3 Modelo de datos para el modelado de usuario

Para la representación de los modelos de usuario a construir, se diseñó un modelo de datos para una base de datos relacional en la que se guardan los atributos recopilados del sistema, los atributos que provienen de los test escritos, los atributos inferidos mediante reglas y los atributos inferidos mediante las tareas de aprendizaje.

El modelo de datos se presenta en la Figura 5.1, el cual contempla las entidades necesarias para la representación de los atributos contemplados inicialmente y además permite la incorporación de nuevos atributos que podrán ser añadidos en el futuro.

Dentro de esta forma de representación del modelo de usuario se contempla que los valores de cada atributo se representen como una terna *<entidad atributo valor>* que es una forma genérica de representación de conocimiento utilizada en diferentes proyectos relacionados (Gaudioso, 2002).

La forma de calcular los valores de cada atributo dependerá de la entidad a la que pertenecen, así por ejemplo, para un zona geográfica, se obtiene mediante reglas de inferencia para lo cual se incorpora código programado, asociado al atributo, el cual permitirá inferirle de un conjunto de datos de entrada obtenidos de la base de datos del sistema.

Esta forma de representación permite componer al modelo de un conjunto de distintos tipos de atributos los cuales se pueden obtener de formar diversas descritas en esencialmente en la naturaleza de cada atributo.

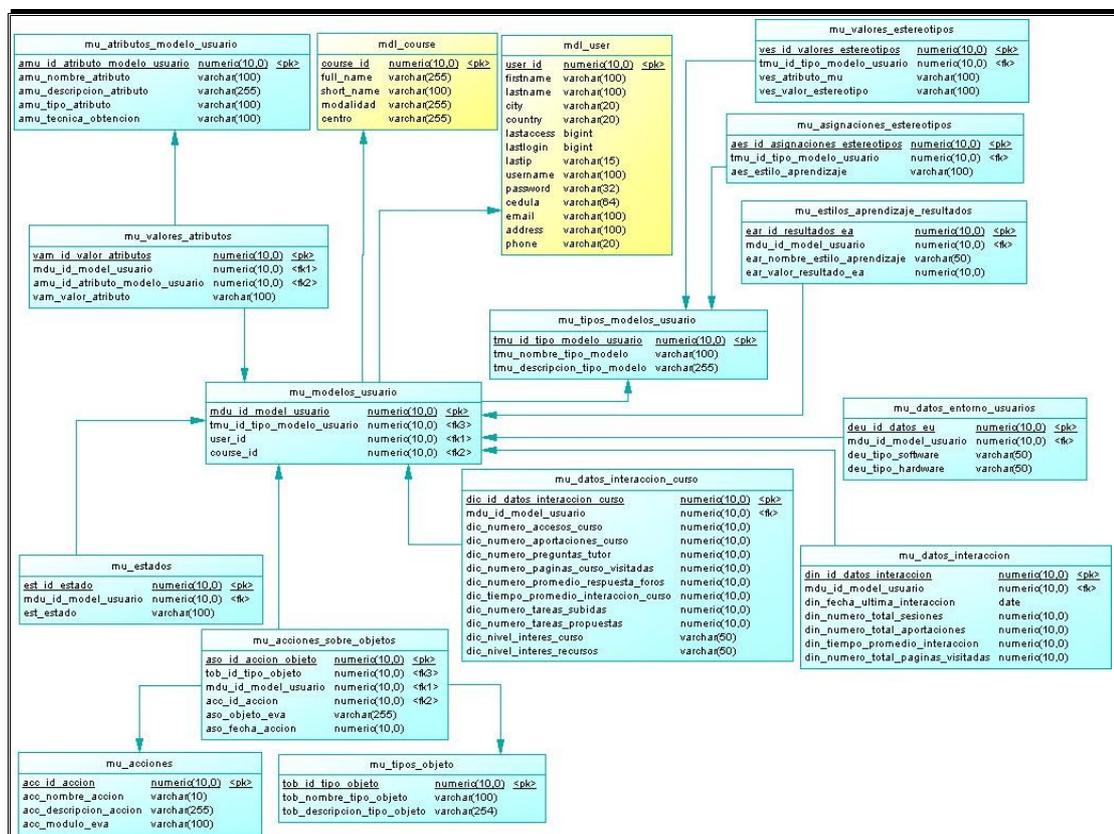


Figura 5.1: Modelo de datos para la representación de los modelos de usuario

Las entidades que conforman la base de datos del modelo de usuario están clasificadas de acuerdo a la utilidad y a la función que desempeñan en el sistema. Ya que existen entidades las que guardan los atributos correspondientes a los modelos de usuario de los estudiantes, entidades que albergan los datos que ayudan a inferencia de los atributos del modelo, y entidades de soporte al sistema de modelado de usuario en las que se almacena información acerca de estados en los que se encuentran los

modelos de usuario

El modelo presentado en la figura 5.1 consta de 16 tablas, y está diseñado para el motor de base de datos MySQL v5.

5.4 Descripción de tablas del modelo de usuario.

En esta sección se describen cada una de las entidades componentes del modelo de datos que alberga a los modelos de usuario de los estudiantes de la UTPL. Además se describen cada uno de los campos que son parte de una entidad.

5.4.1 Entidades del modelo de usuario

Las entidades del modelo de usuario, contienen la información importante de los modelos de usuario de los estudiantes, y son la base para trabajo de sistemas que utilizan modelos de usuario en el desarrollo de su trabajo. Esta información es el resultado del trabajo y la ejecución del sistema multiagente de modelado de usuario. Las entidades que son tomadas en cuenta en esta categoría se describen a continuación:

a) mdl_course

Tabla 5.1: Campos de la entidad mdl_course

Tabla: mdl_course				
Atributo	Tipo	Constraints	Extra	Descripción
<i>course_id</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro. Es idéntico al contenido en la base de datos del EVA
<i>full_name</i>	varchar(255)			Nombre completo del curso
<i>short_name</i>	varchar(100)			Nombre corto del curso
<i>modalidad</i>	varchar(255)			Modalidad (clasica o abierta) en la que está definido este curso.
<i>centro</i>	varchar(255)			El nombre del centro universitario en el que se está dictando este curso

Contiene la información referente al curso del cual se está modelando la información, esta entidad replica algunos de los datos contenidos en la base de datos del entorno virtual de la UTPL (EVA). Conjuntado algunos datos que no se encuentran en esta tabla en su equivalente original. El motivo de desarrollar esto es

para acceder a los atributos de forma más rápida para la inferencia de los modelos de usuario. Y quitar carga de acceso a la base de datos original del EVA. Ya que el proceso de generación y actualización de los modelos de usuario exige una alta demanda de acceso a esta entidad. Los campos contenidos por esta entidad se muestran en la tabla 5.1.

b) mdl_user

Tabla 5.2: Campos de la entidad mdl_user

Tabla: mdl_user				
Atributo	Tipo	Constraints	Extra	Descripción
<i>user_id</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro. Es idéntico al contenido en la base de datos del EVA.
<i>firstname</i>	varchar(100)			Nombres del estudiante.
<i>lastname</i>	varchar(100)			Apellidos del estudiante.
<i>city</i>	varchar(20)			Ciudad en la que se encuentra el estudiante.
<i>country</i>	varchar(20)			Pais en el que se encuentra el estudiante.
<i>lastaccess</i>	bigint(10)			Fecha en formato unix-timestamp ¹ del último acceso al sistema del estudiante
<i>lastlogin</i>	bigint(10)			Fecha en formato unix-timestamp del último login del estudiante.
<i>lastip</i>	varchar(15)			Dirección ip del último punto de red del cual el estudiante inicio sesión en el EVA.
<i>username</i>	varchar(100)			Nombre de usuario del estudiante.
<i>password</i>	varchar(22)			Contraseña del estudiante en el EVA.
<i>cedula</i>	varchar(64)			El numero de cedula de identidad o pasaporte del estudiante.
<i>email</i>	varchar(100)			Correo electrónico del estudiante.
<i>address</i>	varchar(100)			Dirección del estudiante en su localidad.
<i>phone</i>	varchar(20)			Teléfono del estudiante.

Contiene información referente al estudiante del cual se van a crear y actualizar modelos de usuario. Replica y conjunta algunos de los datos contenidos en su equivalente en el entorno virtual de aprendizaje. Al igual que en la entidad

¹ *Unix-Timestamp*. Formato de fecha en el cual se cuentan los segundos contenidos desde el 01 de Enero de 1970 hasta este momento.

mdl_course el motivo de actualizar estos datos en la base de modelos de usuario es el alto acceso que conlleva la creación y actualización de modelos de usuario. Los campos que contiene esta entidad se describen en la tabla 5.2.

c) mu_ atributos_modelo_usuario

Los atributos que debe contener el modelado de usuario se guardan en esta entidad. Así como la técnica que se necesita para la obtención del atributo. Los atributos pertenecen a una tipología. Los campos que componen a esta entidad se describen a continuación:

Tabla 5.3: Campos de la entidad mu_ atributos_modelo_usuario

Tabla: mu_ atributos_modelo_usuario				
Atributo	Tipo	Constraints	Extra	Descripción
<i>amu_id_atributo_modelo_usuario</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>amu_nombre_atributo</i>	varchar(100)			Nombre del atributo del modelo de usuario.
<i>amu_descripcion_atributo</i>	varchar(255)			Una descripción del atributo del modelo de usuario.
<i>amu_tipo_atributo</i>	varchar(100)			Categoría a la que pertenece el atributo el modelo de usuario.
<i>amu_tecnica_obtencion</i>	varchar(100)			Técnica por medio de la cual es obtenido el atributo del modelo de usuario.

d) mu_valores_atributos

Tabla 5.4: Campos de la entidad mu_valores_atributos

Tabla: mu_valores_atributos				
Atributo	Tipo	Constraints	Extra	Descripción
<i>vam_id_valor_atributo</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>mdu_ide_model_usuario</i>	bigint(10)	foreign Key		Número que representa la relación entre esta entidad y la denominada como mu_modelos_usuario.
<i>amu_id_atributo_modelo_usuario</i>	bigint(10)	foreign Key		Número que representa la relación entre esta entidad y la denominada como mu_atributos_modelo_usuario
<i>vam_valor_atributo</i>	varchar(100)			Valor de este atributo para el modelo de usuario especificado.

En esta entidad se almacenan los valores que toma cada atributo para cada modelo de usuario. Los campos de esta representación sufren un procesamiento en cada caso.

Este procesamiento depende de la tipología del atributo. Estos campos se describen en la tabla 5.4:

e) **mu_modelos_usuario**

Entidad en la que se conjuntan los elementos que identifican al modelo de usuario de un estudiante en un curso. Los campos que componen a esta entidad se describen a continuación:

Tabla 5.5: Campos de la entidad mu_modelos_usuario

Tabla: mu_modelos_usuario				
Atributo	Tipo	Constraints	Extra	Descripción
<i>mdu_id_model_usuario</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>tmu_id_tipo_modelo_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como mu_tipos_modelos_usuario.
<i>user_id</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como mdl_user.
<i>course_id</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como mdl_course.

f) **mu_tipos_modelos_usuario**

En esta entidad se guardan las tipologías de modelos de usuario que sirven para realizar el proceso de prototipado de modelos de usuario.

Tabla 5.6: Campos de la entidad mu_tipos_modelos_usuario

Tabla: mu_tipos_modelos_usuario				
Atributo	Tipo	Constraints	Extra	Descripción
<i>tmu_id_tipo_modelo_usuario</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>tmu_nombre_tipo_modelo</i>	varchar(100)			Nombre de la tipología de modelo de usuario utilizada para construir el modelo.
<i>tmu_descripcion_tipo_modelo</i>	varchar(255)			Descripción del tipo de modelo de usuario que se representa el registro.

g) **mu_datos_entorno_usuarios**

Entidad en la que se guardan datos sobre el entorno de usuario, es decir

información acerca de los elementos que modifican el modo de acceso de los estudiantes al EVA.

Tabla 5.7: Campos de la entidad *mu_datos_entorno_usuarios*

Tabla: <i>mu_datos_entorno_usuarios</i>				
Atributo	Tipo	Constraints	Extra	Descripción
<i>deu_id_datos_eu</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro
<i>mdu_id_model_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como <i>mu_modelos_usuario</i> .
<i>deu_tipo_software</i>	varchar(255)			Almacena el tipo de software que está utilizando el estudiante para acceder al EVA.
<i>deu_tipo_hardware</i>	varchar(255)			Almacena el tipo de hardware que está utilizando el estudiante para acceder al EVA.

h) *mu_acciones_sobre_objetos*

En esta entidad se almacenan las acciones que los estudiantes realizan con los recursos que presenta, el entorno virtual de aprendizaje en los diferentes cursos, así como la fecha y hora en que se realizó la acción. Los campos que componen esta entidad se describen a continuación:

Tabla 5.8: Campos de la entidad *mu_acciones_sobre_objetos*

Tabla: <i>mu_acciones_sobre_objetos</i>				
Atributo	Tipo	Constraints	Extra	Descripción
<i>aso_id_accion_objeto</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>tob_id_tipo_objeto</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como <i>mu_tipos_objeto</i> .
<i>mdu_id_model_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como <i>mu_modelos_usuario</i> .
<i>acc_id_accion</i>	varchar(255)			Número que representa la relación entre esta entidad y la denominada como <i>mu_acciones</i> .
<i>aso_objeto_eva</i>	varchar(255)			Link al objeto del EVA al cual hace referencia el registro de la acción.
<i>aso_fecha_accion</i>	bigint(10)			Fecha en formato Unix-Timestamp en la cual se realizó la acción sobre el objeto registrado.

5.4.2 Entidades de apoyo al modelo de usuario

a) mu_asignaciones_estereotipos

Entidad en la que se almacena los estilos de aprendizaje enlazado a los estereotipos del modelo de usuario según a las asignaciones discutidas en la Tabla 5.18. Los campos que componen a esta entidad se describen a continuación:

Tabla 5.9: Campos de la entidad mu_asignaciones_estereotipos

Tabla: mu_asignaciones_estereotipos				
Atributo	Tipo	Constraints	Extra	Descripción
<i>aes_id_asignaciones_estereotipos</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>tmu_id_tipo_modelo_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como mu_tipos_modelos_usuario.
<i>aes_estilo_aprendizaje</i>	varchar(100)			Tipología de estilos de aprendizaje a la que la asignación se desarrollara.

b) mu_valores_estereotipos

Los valores que toman los atributos del modelo de usuario en cada estereotipo se guardan en esta entidad. Para esto se han definido campos para contener la información necesaria. Estos campos se definen a continuación:

Tabla 5.10: Campos de la entidad mu_valores_estereotipos

Tabla: mu_valores_estereotipos				
Atributo	Tipo	Constraints	Extra	Descripción
<i>ves_id_valores_estereotipos</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>tmu_id_tipo_modelo_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como mu_tipos_modelos_usuario.
<i>ves_atributo_mu</i>	varchar(100)			Atributo del modelo de usuario del cual se mantiene un estereotipo.
<i>ves_valor_estereotipo</i>	varchar(100)			Valor que se dará al atributo del modelo de usuario del cual se generara el prototipo.

c) mu_estilos_aprendizaje_resultados

En esta entidad se almacenan los resultados que los estudiantes han obtenido en el

test de estilos de aprendizaje en cada uno de los definidos por la tipología elegida. Los campos contenidos por esta entidad se definen a continuación:

Tabla 5.11: Campos de la entidad mu_estilos_aprendizaje_resultados

Tabla: mu_estilos_aprendizaje_resultados				
Atributo	Tipo	Constraints	Extra	Descripción
<i>ear_id_resultados_ea</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>mdu_id_model_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como mu_modelos_usuario.
<i>ear_nombre_estilo_aprendizaje</i>	varchar(50)			Nombre de la tipología de estilos de aprendizaje del cual se guarda un valor.
<i>ear_valor_resultado_ea</i>	decimal(10,0)			Valor obtenido por el estudiante en el test de estilos de aprendizaje para la tipología actual.

d) mu_datos_interaccion_curso

En esta entidad se almacenan los datos de interacción del estudiante en cada curso que son necesarios para la inferencia de los niveles de interés de los estudiantes. Los campos definidos para esta entidad se definen a continuación:

Tabla 5.12: Campos de la entidad mu_datos_interaccion_curso

Tabla: mu_datos_interaccion_curso				
Atributo	Tipo	Constraints	Extra	Descripción
<i>dic_id_datos_interaccion_curso</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>mdu_id_model_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como mu_modelos_usuario.
<i>dic_numero_accesos_curso</i>	decimal(10,0)			Número de veces que el estudiante ha accedido al curso.
<i>dic_numero_aportaciones_curso</i>	decimal(10,0)			Número de veces que el estudiante realizó un aporte al curso.
<i>dic_numero_preguntas_tutor</i>	decimal(10,0)			Número de veces que el estudiante realizó una pregunta a su profesor.
<i>dic_numero_paginas_curso_visitadas</i>	decimal(10,0)			Número de páginas que el estudiante ha accedido en el curso.
<i>dic_numero_promedio_respuesta_foros</i>	decimal(10,0)			Promedio del número de veces que el estudiante a realizado una aportación en los foros propuestos en el curso

<i>dic_tiempo_promedio_interaccion_curso</i>	decimal(10,0)	Promedio del tiempo que el estudiante realiza acciones dentro del EVA.
<i>dic_numero_tareas_subidas</i>	decimal(10,0)	Número de veces que el estudiante a subido trabajos o tareas dentro del EVA.
<i>dic_numero_tareas_propuestas</i>	decimal(10,0)	Número de tareas que el profesor propone al estudiante dentro del EVA.
<i>dic_nivel_interes_curso</i>	varchar(50)	Atributo en el que se almacena el nivel de interés del estudiante en el curso.
<i>dic_nivel_interes_recurso</i>	varchar(50)	Atributo en el que se almacena del nivel de interés en los recurso que se presentan dentro del curso.

e) **mu_datos_interaccion**

En esta entidad se almacenan los resultados de la interacción de los estudiantes en todos los cursos activos en los que participa. Los campos contemplados para esta entidad se describen a continuación:

Tabla 5.13: Campos de la entidad *mu_datos_interaccion*

Tabla: mu_datos_interaccion				
Atributo	Tipo	Constraints	Extra	Descripción
<i>din_id_datos_interaccion</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>mdu_id_model_usuario</i>	bigint(10)	foreign key		Número que representa la relación entre esta entidad y la denominada como <i>mu_modelos_usuario</i> .
<i>din_fecha_ultima_interaccion</i>	date			Fecha de la última interacción del estudiante con el entorno virtual de aprendizaje.
<i>din_numero_total_sesiones</i>	decimal(10,0)			Número total de sesiones que el estudiante ha realizado en el EVA.
<i>din_numero_total_aportaciones</i>	decimal(10,0)			Número total de aportaciones que el estudiante ha realizado en el EVA.
<i>din_tiempo_promedio_interaccion</i>	decimal(10,0)			Promedio de todas las interacciones del estudiante en el EVA.
<i>din_numero_total_paginas_visitadas</i>	decimal(10,0)			Número total de páginas que el estudiante ha visitado dentro del EVA.

f) **mu_acciones**

En esta entidad se guardan un conjunto de acciones que el estudiante puede desarrollar en los distintos módulos que tiene el entorno virtual de aprendizaje, estas

acciones están relacionadas con los módulos a las que pertenecen y pueden existir acciones similares en distintos módulos. Los campos que soportan esta representación se describen a continuación:

Tabla 5.14: Campos de la entidad mu_acciones

Tabla: mu_acciones				
Atributo	Tipo	Constraints	Extra	Descripción
<i>acc_id_accion</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>acc_nombre_accion</i>	varchar(100)			Nombre de la acción que puede desarrollar el estudiante dentro del EVA.
<i>acc_descripcion_accion</i>	varchar(255)			Descripción de la acción que un estudiante puede desarrollar dentro del entorno virtual de aprendizaje.
<i>acc_modulo_eva</i>	varchar(100)			Nombre de modulo del entorno virtual de aprendizaje al que la acción que realiza el estudiante afecta.

g) mu_tipos_objeto

Los tipos de objetos que se definen en el sistema son categorías en las que se enmarcan los recursos disponibles dentro del entorno virtual de aprendizaje y se almacenan en esta entidad, para la cual los campos que soportan su trabajo se describen a continuación:

Tabla 5.15: Campos de la entidad mu_tipos_objeto

Tabla: mu_tipos_objeto				
Atributo	Tipo	Constraints	Extra	Descripción
<i>tob_id_tipo_objeto</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>tob_nombre_tipo_objeto</i>	varchar(100)			Nombre del tipo de objeto del entorno virtual de aprendizaje.
<i>tob_descripcion_tipo_objeto</i>	varchar(255)			Descripción del tipo de objeto del EVA.

5.4.3 Entidades de apoyo al sistema

a) mu_estados

En esta entidad se almacenan los estados en que se encuentra el modelado de usuario esto sirve para que el sistema pueda continuar con el modelado tomando

como referencia el ultimo estado en que este quedo después del terminado el trabajo del agente de modelado de usuario. Los estados que puede alcanzar un modelo de usuario dentro del sistema son: *inicial*, *prototipado*, *desactualizado* y *actualizado*. Para soportar esta representación se han definido los campos que se mencionan a continuación:

Tabla 5.16: Campos de la entidad mu_estados

Tabla: mu_estados				
Atributo	Tipo	Constraints	Extra	Descripción
<i>est_id_estado</i>	bigint(10) unsigned	primary key	auto_increment	Valor que identifica al registro.
<i>mdu_id_model_usuario</i>	bigint(10)			Número que representa la relación entre esta entidad y la denominada como mu_modelos_usuario.
<i>est_estado</i>	varchar(100)			Estado en el que se encuentra el modelado de usuario.

5.5 Inicialización del modelo de usuario

En el modelo de usuario propuesto se implementa la inicialización del mismo mediante la creación de estereotipos los cuales permitan hacer, desde un principio, asunciones con respecto al estudiante. Estos estereotipos se actualizaran luego con observaciones hechas al comportamiento de los estudiantes en el sistema.

Una de las formas más usadas para la generación de estos estereotipos es utilizar algunos de los atributos mejor obtenibles para la clasificación de un determinado usuario. En nuestro caso, el atributo inicial del cual se puede tener un alto grado de certeza en su valor, es el referente al estilo de aprendizaje del estudiante ya que este atributo se obtiene mediante un test diseñado para el caso. Y ya que luego de este proceso se consta con un atributo definitorio, que será utilizado para definir un estereotipo en cada caso. Según Gaudioso (2002), uno de los criterios para clasificar las formas de obtener estereotipos es utilizar el *tipo de aprendizaje preferido por el alumno*. Así que considerar este atributo se convierte en opción válida y recomendable para la obtención de los mencionados estereotipos.

Entonces para los estilos de aprendizaje definidos en la distribución de Felder y Silverman en la tabla 5.17 se presentan los estereotipos definidos para cada caso.

Tabla 5.17: Estereotipos definidos para cada estilo de aprendizaje

Estilo de aprendizaje	Estereotipo
Activo	medio
Reflexivo	avanzado
Sensitivo	principiante
Intuitivo	avanzado
Visual	avanzado
Verbal	principiante
Secuencial	avanzado
Global	medio

Los estereotipos para el modelo de usuario de los estudiantes de la UTPL se definieron en función de las características de cada estilo de aprendizaje y dependen directamente de las formas de aprender de cada uno de ellos. Ya que por ejemplo, estudiantes con un estilo de aprendizaje visual al tener un alto grado de aceptación por el contenido multimedia, suelen usualmente ser altamente participativos en plataformas basadas en la Web (Durán y Costaguta, 2007). Entonces de este hecho se selecciono que este tipo de estudiantes puede tener un estereotipo avanzado del modelo de usuario.

Tabla 5.18: Valores de atributos según los estereotipos

Estereotipo	Valores
principiante	<ul style="list-style-type: none"> nivel_interes_curso = bajo nivel_interes_recursos = bajo nivel_interaccion = principiante
medio	<ul style="list-style-type: none"> nivel_interes_curso = medio nivel_interes_recursos = medio nivel_interaccion = medio
avanzado	<ul style="list-style-type: none"> nivel_interes_curso = alto nivel_interes_recursos = alto nivel_interaccion = avanzado

Para los estereotipos diseñados los atributos que necesitan valores iniciales se se muestran en la tabla 5.18. Los valores guardan estrecha relación con la definición de cada estereotipo.

Luego de obtener una visión de cómo se comienza con el proceso de modelado de usuario en la figura 5.2 se ejemplifica el modelado de usuario de un estudiante tomando en cuenta todos los datos que son necesarios para calcular los atributos del

modelo de usuario, así como los estados por los que pasa el modelado de usuario.

También se reflejan los datos que permiten construir prototipos del modelo de usuario a partir de los estilos de aprendizaje de los estudiantes, y se lleva un registro de las acciones sobre los recursos disponibles en el curso, además se contemplan las relaciones entre las entidades que componen el modelo de datos.

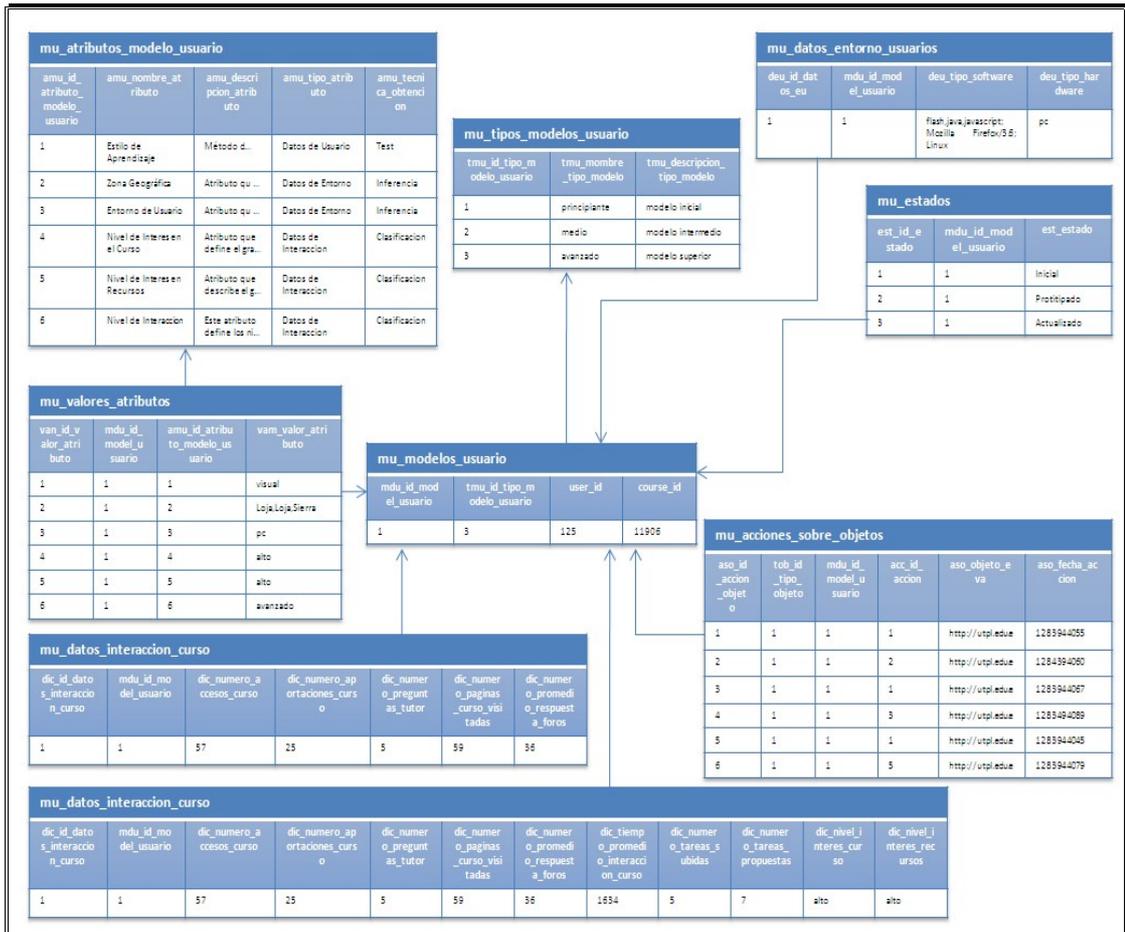


Figura 5.2: Ejemplo de modelo de usuario generado en las entidades de la base de datos

Capítulo 6

*Construcción del agente de
modelado de usuario para el
EVA UTPL*

6.1 Introducción

En este capítulo se presenta un proceso de desarrollo del agente de modelado de usuario de para el entorno virtual de aprendizaje (EVA) de la UTPL. Este proceso se sustenta en la metodología de desarrollo de sistemas multiagente (SMA) denominada INGENIAS y su herramienta de soporte al proceso de desarrollo, el Ingenias Development Kit (IDK). Para el análisis, diseño, implementación y pruebas del agente.

6.1.1 Modelado de sistemas multiagente con INGENIAS

El modelado de SMA en INGENIAS se basa en la generación de meta-modelos y se sustenta bajo tres principios fundamentales, los agentes como paradigma de modelado, implementación sobre diferentes plataformas, y la evolución de la tecnología de agentes.

El desarrollo en INGENIAS se respalda en la utilización del IDK que permite entre otras cosas, editar los modelos, generación automática de código, verificación y validación de los modelos construidos.

Los meta-modelos de INGENIAS se describen mediante cinco puntos de vista que definen la naturaleza de cada uno de los meta-modelos que esta metodología propone:

- ◆ *El Meta-Modelo de Organización.*- define grupos de trabajo, los flujos con que estos trabajan, así como relaciones de poder, y los objetivos que estas organizaciones persiguen, también construye las tareas relevantes para la organización y define restricciones de comportamiento de los agentes como subordinación.
- ◆ *El Meta-Modelo de Agente.*- describe agentes particulares, centrándose en la funcionalidad y en el diseño de su control, es decir que modela las responsabilidades del agente, su comportamiento, los objetivos que persigue controlando todo esto mediante estados mentales que evolucionan en el tiempo fundamentando el aprendizaje que deben poseer estas entidades.
- ◆ *El Meta-Modelo de Objetivos y Tareas.*- recoge las motivaciones del SMA, así

como definir las acciones identificadas en el modelo de organización, interacciones o de agentes y como estas afectan a sus responsables, además asocia los objetivos con las tareas que los cumplen, y descompone los objetivos globales en objetivos más específicos asignables a los agentes.

- ◆ *El Meta-Modelo de Interacción.*- Modela las interacciones existentes entre los agentes y que roles estos toman dentro del SMA, contribuye en la especificación del comportamiento de los componentes de sistema, se identifican los elementos que intervienen en la comunicación mediante roles y sus objetivos.
- ◆ *El Meta-Modelo del Entorno.*- representa el mundo donde se desenvuelve el SMA Trata de modelar las entidades y recursos con los que interactúa el SMA. Además identifica aplicaciones del entorno, asociando operaciones sobre esas aplicaciones, y determina las percepciones que debe incluir los agentes.

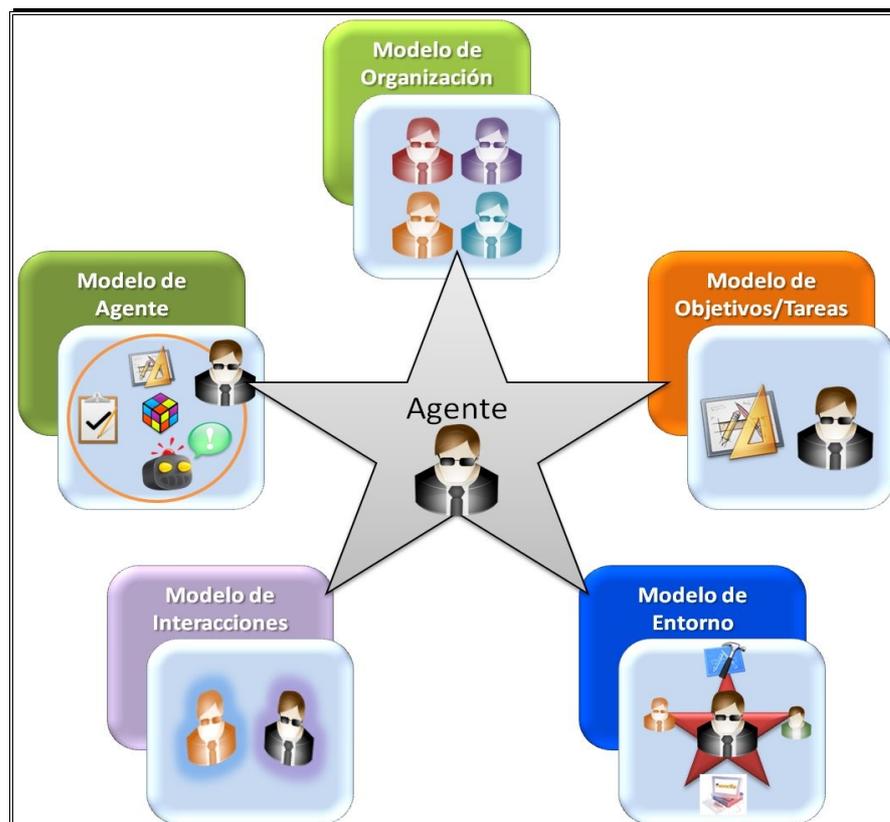


Figura 6.1: Puntos de vista de INGENIAS para el modelado de SMA

La concepción de meta-modelos en el análisis y diseño de SMA incorpora todos los paradigmas actuales del desarrollo de sistemas basados en agentes, poniendo

énfasis en puntos como la comunicación entre los componentes del sistema y su organización. En la Figura 6.1 se muestran los meta-modelos que comprenden este tipo de sistemas.

El desarrollo de SMA con INGENIAS conlleva el seguimiento de flujos de trabajo los cuales se centran en el análisis y diseño para los cuales se incorporan fases en las que se realizan entregables, las fases que se incorporan son: inicio, elaboración y implementación. En cada una de ellas la evolución de los entregables va dando madurez al producto que se va construyendo en cada ciclo.

Para el desarrollo de estas fases y los flujos de trabajo inherentes a cada una de ellas se han diseñado, por parte de los autores de INGENIAS, un conjunto de actividades, expresadas como diagrama de actividades.

Para la implementación del SMA INGENIAS, la generación de código se desarrolla paralelamente con el desarrollo de la especificación mediante las herramientas entregadas a los analistas y diseñadores, lo cual sirve como validación y verificación de los diseños propuestos.

Después de describir brevemente los componentes y el proceso de análisis diseño e implementación de SMA mediante INGENIAS en las secciones siguientes de este capítulo se describen los resultados obtenidos de la aplicación de esta metodología en el proceso de diseño e implementación del agente de modelado de usuario del entorno virtual de aprendizaje (EVA) de la UTPL.

6.2 Conceptualización

Como se ha definido anteriormente el producto que se desea obtener con el desarrollo de este proyecto de tesis es un sistema en el que un agente inteligente desarrolle el proceso de modelado de usuario de los estudiantes de al UTPL mediante los datos de interacción generados por los accesos que el mencionado estudiante realiza al EVA.

El agente tiene que cumplir dos funciones fundamentales, la primera es la *construcción del modelo de usuario* inicial, tomando el test de estilos de aprendizaje,

dentro del EVA para luego de obtenido el resultado del test, mediante el prototipo diseñado para cada estilo, etiquetar los valores iniciales del modelo. En segunda instancia el agente debe *actualizar los modelos de usuario* obtenidos inicialmente, recibiendo como entradas los datos de interacción de los estudiantes dentro del EVA, de notar cambios significativos en la interacción de los estudiantes con el sistema, el agente debe actualizar los valores de los atributos del modelo que necesitan ser actualizados para mantener modelos de usuario actualizados.

Para la construcción y actualización de los modelos se necesita que el agente pueda monitorear periódicamente el estado de los datos que definen los atributos del modelo para poder decidir cuándo actualizarlos, además el agente debe saber que al estar el estudiante matriculado en diferentes cursos hay datos del modelo que se pueden actualizar directamente de los modelos de usuario creados en otros cursos.

El agente, además de su acceso constante al EVA también tiene que tener la posibilidad de acceder al modelo de datos diseñado para la albergar los modelos de usuario para su construcción y actualización.

Para la obtención de los datos de interacción, del entorno de usuario, y los datos de identificación de los usuarios, el agente debe poseer acceso a estructuras de código que le permitan obtener esta información. Además tiene que poder acceder a módulos que le permitan transformar esta información en atributos del modelo de usuario, tanto para los atributos que se obtienen, directamente, mediante reglas de inferencia y mediante tareas de aprendizaje. Para cada tipología de atributos se prevé la existencia de módulos que permitan su construcción.

6.3 Análisis – Inicio

En esta etapa del proceso de desarrollo se producen los primeros entregables referentes a la definición inicial del producto, identificando los casos de uso principales que intervienen en el sistema, además se identifican los componentes principales del sistema, lo que se traduce en diseñar los *modelo de entorno* y el *modelo de organización* del sistema.

Los casos de uso se diseñan mediante UML. La identificación de los componentes

mediante el modelo de organización y el modelo del entorno. El modelo de organización da una visión global de como se organizan los elementos, mientras que el modelo del entorno delimita con que elementos del EVA o con que módulos necesita interactuar el agente.

En la tabla 6.1 se muestran las actividades a desarrollar para obtener el modelo del entorno del agente, se describen las actividades a desarrollar, los resultados a obtener y los recursos que hacen referencia a la representación de cada actividad.

Tabla 6.1: Actividades para la construcción del modelo del entorno del agente

Actividad	Tipo de resultado	Referencia
Identificar aplicaciones del entorno	<ul style="list-style-type: none"> Un conjunto de aplicaciones 	
Asociar operaciones sobre las aplicaciones	<ul style="list-style-type: none"> Operaciones sobre las aplicaciones 	Figura D.2 pagina 204
Determinar la percepción del agente	<ul style="list-style-type: none"> Asociaciones entre agente y aplicaciones. 	

En la tabla 6.2 se muestran las actividades a desarrollar para obtener el modelo de la organización del sistema, se describen las actividades a desarrollar, los resultados a obtener y los recursos que hacen referencia a la representación de cada actividad.

Tabla 6.2: Actividades para la construcción del modelo de organización del sistema

Actividad	Tipo de resultado	Referencia
Identificar grupos	<ul style="list-style-type: none"> Un conjunto de grupos 	
Generar miembros	<ul style="list-style-type: none"> Elementos que componen los grupos 	Figura D.3 pagina 204
Identificar objetivos	<ul style="list-style-type: none"> Objetivos asociados a las organizaciones 	

En las secciones D.1.1 , D.1.2 , D.1.3 del Anexo D se muestran los resultados de esta fase del desarrollo del SMA las referencias a cada uno de los modelos se muestran en las tablas donde se describe la construcción de cada modelo.

6.4 Análisis – Elaboración

En esta fase se analizan los casos de uso para obtener una arquitectura del sistema, la estructura de la organización y los flujos de trabajo fundamentales. Además los

casos de uso definidos en la fase anterior se refinan para identificar como se realizan a cabo estas operaciones.

Lo que se busca es generar *modelos de interacción* con los cuales se muestren las formas en que los agentes contribuyen en el desarrollo de un objetivo. Se debe diseñar modelos de interacción para cada una de las definidas, en la especificación de los casos de uso.

Además en esta fase se identifican los objetivos y tareas que acompañan a las interacciones, estas tareas cumplen la función de satisfacer los objetivos que persiguen las interacciones propuestas.

De las tareas definidas surgen nuevas necesidades de consumo de recursos para poder obtener los resultados que desarrollan estas tareas para lo cual se representa estas necesidades mediante modelos de entorno, modelando las aplicaciones que las tareas necesitan para poder cumplir su cometido.

Finalmente en esta fase se estructura un modelo de organización que recoge los resultados obtenidos en el desarrollo de esta etapa.

El desarrollo de los modelos de interacción esta sujeto a las actividades definidas en la tabla 6.3. Con estas actividades aplicadas a cada interacción entre agentes se genera un modelo por cada una de ellas.

Tabla 6.3: Actividades para la construcción de modelos de interacción del sistema

Actividad	Tipo de resultado	Referencia
Identificar los objetivos que persigue la interacción	• Un conjunto de objetivos	
Identificar la naturaleza de la interacción	• Categoría correspondiente a la interacción	Figura 6.3 pagina 131 Figura 6.4 pagina 131
Identificar los participantes	• Un conjunto de participantes	
Generar una primera especificación	• Diagramas de colaboración	

Como particularidad se debe mencionar que la construcción de modelos de interacción incorpora modelos de especificación UML en este caso en donde se modela la forma en que se desarrolla la comunicación entre los roles participantes en la interacción.

Luego de esto se deben de describir a los agentes que intervienen en la organización de la fase anterior, mediante la construcción de *modelos de agente*, cuya elaboración esta sujeta a las actividades descritas en la tabla 6.4.

Tabla 6.4: Actividades para la construcción del modelo de agente

Actividad	Tipo de resultado	Referencia
Identificar agentes utilizando el principio de racionalidad	<ul style="list-style-type: none"> Un conjunto de agentes 	
Identificar los objetivos	<ul style="list-style-type: none"> Un conjunto de objetivos 	Figura 6.12 pagina 139
Identificar funcionalidad	<ul style="list-style-type: none"> Un conjunto de roles y tareas 	Figura 6.13 pagina 140
Asociar funcionalidad con objetivos	<ul style="list-style-type: none"> Asociaciones de roles y tareas con agentes 	
Identificar aspectos de autonomía e inteligencia	<ul style="list-style-type: none"> Descripción en lenguaje natural 	Texto adjunto
Determinar cómo será el gestor y procesador de estado mental	<ul style="list-style-type: none"> Descripción en lenguaje natural 	Texto adjunto

Los objetivos y tareas encontrados se organizan en *modelos de tareas y objetivos*. El proceso para la construcción de los diagramas de tareas y objetivos, se muestran en la tabla 6.5.

Tabla 6.5: Actividades para la construcción de modelos de tareas y objetivos

Actividad	Tipo de resultado	Referencia
Identificar tareas y objetivos	<ul style="list-style-type: none"> Un conjunto de tareas y objetivos 	Referidos en otras ilustraciones
Descomponer objetivos	<ul style="list-style-type: none"> Asociaciones de descomposición entre objetivos 	Figura 6.5 pagina 132
Asociar tareas a objetivos	<ul style="list-style-type: none"> Asociaciones entre objetivos y tareas 	Figura 6.6 pagina 133
Descomponer tareas	<ul style="list-style-type: none"> Asociaciones de descomposición entre tareas 	
Establecer dependencias entre objetivos	<ul style="list-style-type: none"> Dependencias 	No es necesaria
Asociar tareas con interacciones	<ul style="list-style-type: none"> Instancias de <i>WFProduce</i> asociadas a entidades Interacción 	Figura 6.7 pagina 135
Asociar tareas con entidades producidas y consumidas	<ul style="list-style-type: none"> Instancias de <i>WFProduce</i> e instancias de <i>WFConsume</i> 	Figura 6.8 pagina 135 Figura 6.9 pagina 136 Figura 6.10 pagina 137
Asociar tareas con aplicaciones	<ul style="list-style-type: none"> Instancias de <i>WFUsa</i> asociando aplicaciones y tareas 	Figura 6.11 pagina 137

Las *aplicaciones internas*, que no son más que aplicaciones que necesitan las tareas para poder realizar su cometido. Para encontrar este modelo en la tabla 6.6 se

muestran un conjunto de actividades que ayudan a describir este entorno.

Tabla 6.6: Actividades para la optimización del modelo del entorno

Actividad	Tipo de resultado	Referencia
Identificar aplicaciones internas	• Un conjunto de instancias de aplicación interna	Figura 6.14 pagina 141
Asociar operaciones	• Operaciones sobre las aplicaciones	
Actividades del análisis de las aplicaciones	• Diagramas UML para expresar comportamiento	
Asociar aplicaciones a grupos	• Un conjunto de aplicaciones	Figura 6.14 pagina 141
Identificar recursos	• Un conjunto de recursos	

Finalmente en esta fase se refina el modelo de organización, incorporando las asociaciones con las nuevas aplicaciones y agrupando las tareas en flujos de trabajo. En la tabla 6.7 se muestran las actividades necesarias para el refinamiento de este modelo.

Tabla 6.7: Actividades para la optimización del modelo de organización del Sistema

Actividad	Tipo de resultado	Referencia
Generar miembros	• Entidades asociadas a instancias de grupo	Figura 6.15 pagina 142
Identificar flujos de trabajo	• Instancias de Flujo de trabajo asociadas a una organización	
Aplicar descomposición de flujos	• Descomposición de los flujos de trabajo	

En las sub-secciones siguientes hacemos una descripción de las fases de esta parte del desarrollo.

6.4.1 Casos de uso ampliados

Los casos de uso iniciales han sido refinados en la figura 6.2 centralizando primero el trabajo de todo el sistema en el caso de uso *AgentesMU_UTPL*, Y en segundo caso, identificando las cesiones de los estudiantes en el sistema mediante el caso de uso *Login de usuario*. Además se han identificado relaciones de extensión entre algunos casos de uso. Y finalmente se han identificado tres interacciones entre agentes que definen el trabajo de tres casos de uso, estas son: *Estilo aprendizaje*, *Activar agente*, y *Comprobar interacción*. Los demás casos de uso se desarrollan implementando tareas

que satisfacen su ejecución.

A continuación se presenta una descripción de los nuevos casos de uso:

- ◆ *AgentesMU_UTPL*.- Engloba todo el trabajo de construcción y actualización de modelos de usuario del entorno virtual de aprendizaje de la UTPL, así como los flujos y recursos que manejan el sistema.
- ◆ *Login de usuario*.- En este caso de uso se busca es monitorear los accesos de los estudiantes al sistema para poder activar los agentes que realizan la tarea de modelado de usuario de cada uno de los estudiantes.

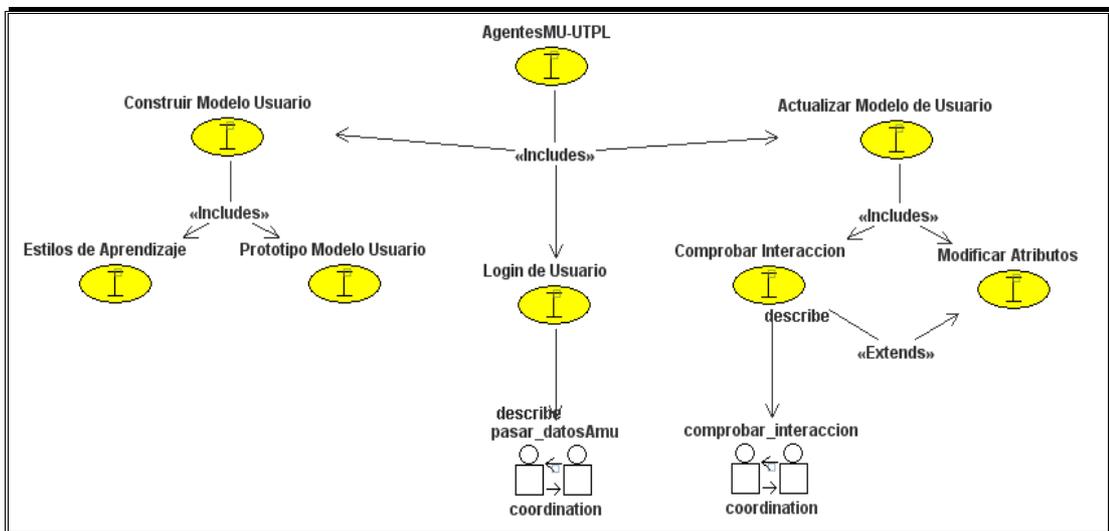


Figura 6.2: Ampliación al diagrama de casos de uso

6.4.2 Modelos de interacción

Los casos de uso principales se detallan mediante interacciones que se muestran en las figuras 6.3, 6.4, e identifican los elementos como los objetivos que estas satisfacen, así como lo roles que intervienen en las mismas.

La interacción *pasar_datosAmu* mostrada en la figura 6.3, se inicia para satisfacer el objetivo *ActivarAgentesNecesarios*, y se implementa para activar los agentes que se encargan de la creación y actualización de los modelos de usuario de los estudiantes, lo que se realiza luego de activar un agente de modelado de usuario, es asignarle los datos correspondientes al usuario en el curso que se va a modelar.

En la figura 6.3 además se muestra un modelo de colaboración para esta

interacción en la que se puede observar el flujo de mensajes por medio de los cuales, los roles identificados para esta interacción.

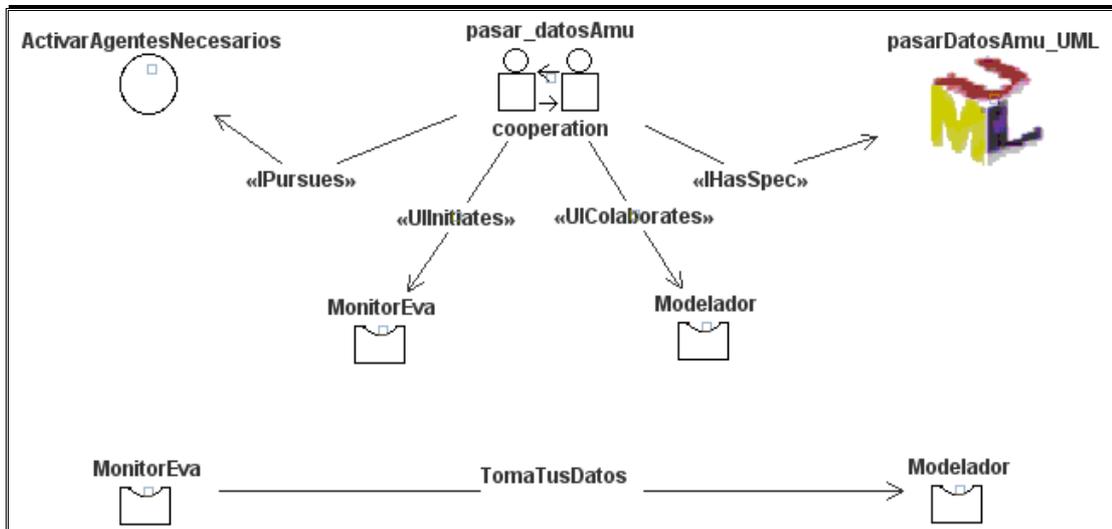


Figura 6.3: Modelo de interacción y especificación UML para la interacción activar agente

La interacción *Comprobar Interacción*, mostrada en la figura 6.4, se inicia para satisfacer el objetivo *DetectarCambiosInteraccion*, y se implementa para monitorear los cambios en la interacción de los estudiantes con el entorno virtual de aprendizaje.

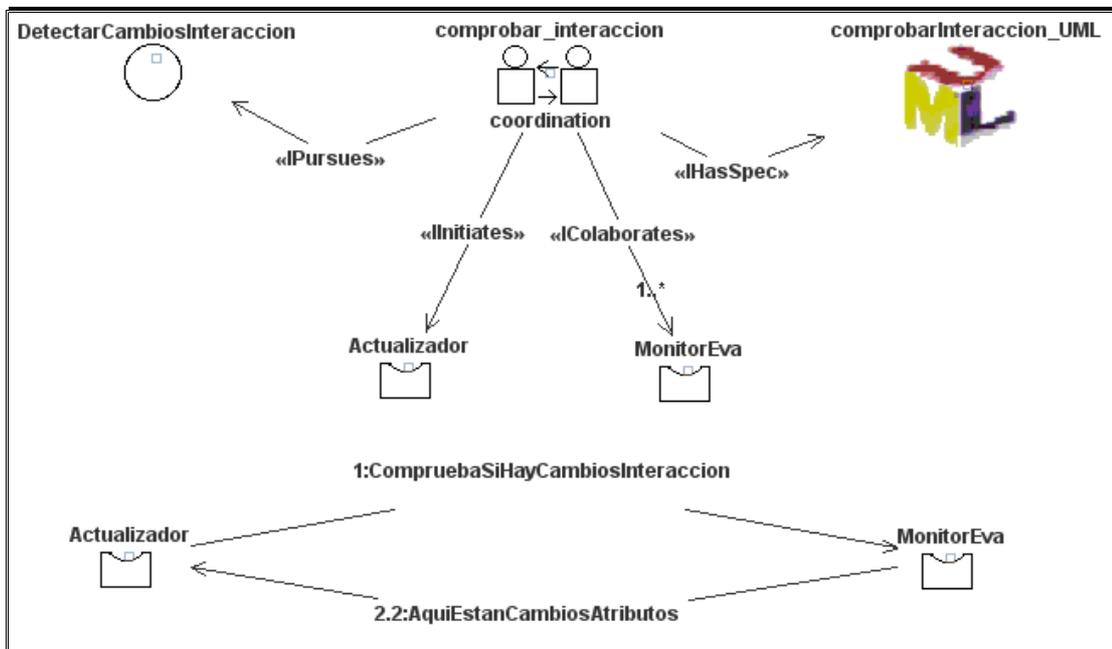


Figura 6.4: Modelo de interacción y especificación UML para la interacción comprobar interacción

La interacción se desarrolla de la forma que se muestra en el diagrama de

colaboración contenido en la figura 6.4 en donde el *Actualizador* pide al *MonitorEva* que analice los cambios en la interacción del estudiante a su cargo. El *MonitorEva* se apoya en la aplicación que enlaza al EVA con el Sistema Multiagente recopilando un conjunto de cambios en la interacción que los devuelve a *Actualizador* para que inicie sus procesos, de no encontrar cambios significativos la respuesta del *MonitorEVA* a la petición del *Actualizador* es comunicarle que no existen cambios en la interacción del usuario.

6.4.3 Descomposición de objetivos

Los objetivos encontrados a lo largo del proceso de desarrollo seguido hasta estos momentos se estructuran utilizando un modelo de objetivos y tareas.

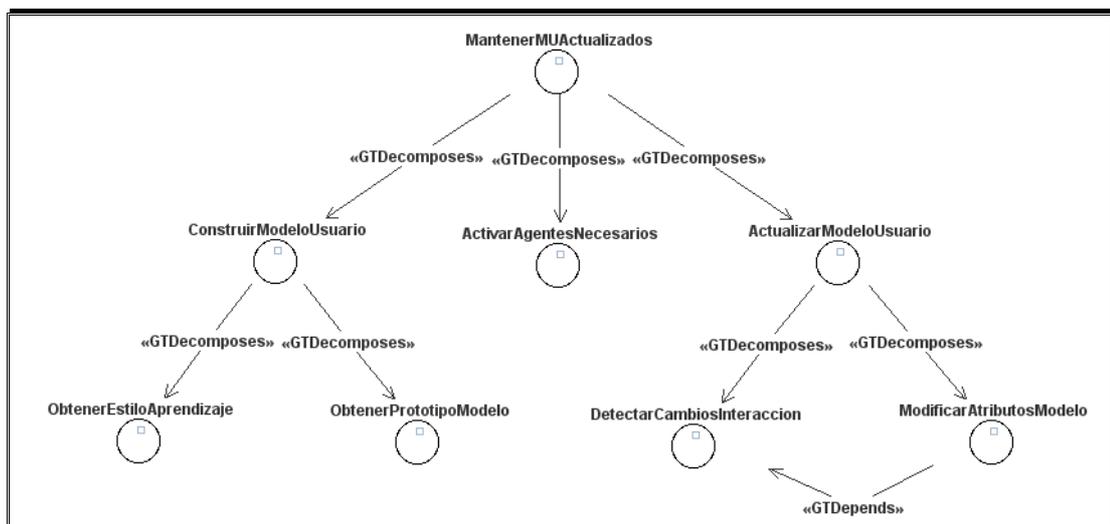


Figura 6.5: Descomposición de objetivos de sistema

Como se ve en la figura 6.5, el objetivo esencial que se persigue con el desarrollo de este sistema multiagente es el mantener los modelos de usuario de todos los estudiantes actualizados (*MantenerMUActualizados*), para cumplir este objetivo se deben cumplir un conjunto de objetivos de más bajo nivel, estos son: el construir los modelos de usuario (*ConstruirModeloUsuario*), activado agentes de modelado de usuario necesarios (*ActivarAgentesNecesarios*) para que realicen el trabajo, desarrollando actualizaciones (*ActualizarModeloUsuario*) en los modelos de usuario ya existentes. Para el desarrollo de todos estos objetivos de segundo nivel se han identificado a su vez objetivos de tercer nivel que cumplen con la satisfacción de los

objetivos del nivel anterior. Estos objetivos son: el obtener los estilos de aprendizaje del estudiante (*ObtenerEstiloAprendizaje*) base para la construcción y prototipado de modelos de usuario, otro objetivo relacionado con la construcción de los modelos de usuario y que depende el cumplimiento de obtener los estilos de aprendizaje es de obtener un prototipo del modelo de usuario (*ObtenerPrototipoModelo*). En cuanto a la actualización de los modelos de usuario los objetivos de tercer nivel necesarios para su satisfacción, incorporan el monitoreo de la acciones del estudiante en el EVA (*DetectarCambiosInteraccion*), los resultados de esto originan que se necesite el desarrollar cambios en los atributos (*ModificarAtributosModelo*) del modelo actualizado.

6.4.4 Objetivos y tareas asociadas

Como se han identificado un conjunto de objetivos que controlan el trabajo que realiza el sistema, a estos objetivos se les deben relacionar tareas que satisfacen el trabajo de cada uno de los objetivos identificados. Estas tareas y los objetivos que las generan se presentan en la figura 6.6. La forma exacta en que intervienen estas tareas se definirá más adelante. Se presenta entonces una pequeña descripción del trabajo de cada una de ellas a continuación:

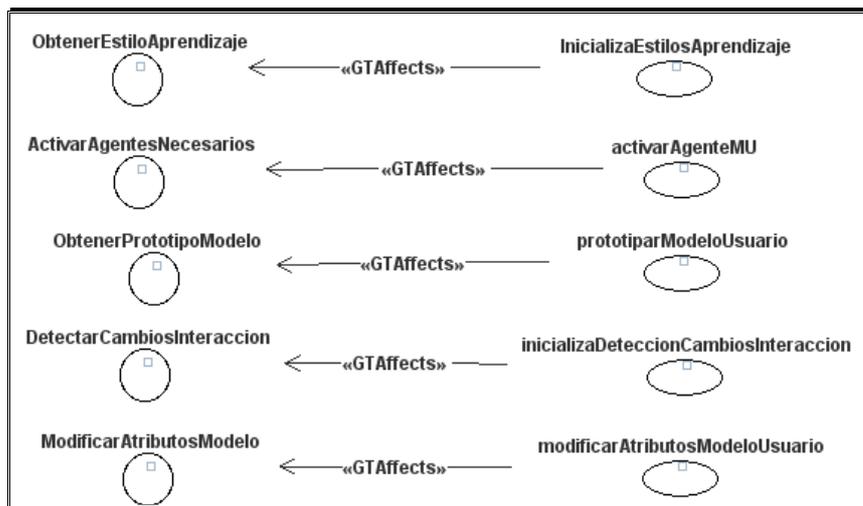


Figura 6.6: Tareas asociadas al cumplimiento de los objetivos identificados

- ◆ *Inicializa estilos aprendizaje (inicializaEstilosAprendizaje).*- El agente modelador mediante esta tarea identifica que el estudiante del que se construye un modelo de usuario ha realizado el test de estilos de aprendizaje, de no ser así

mediante la aplicación asociada al objetivo asigna al estudiante el desarrollo del mencionado test.

- ◆ *Activar agente MU (activarAgenteMU).*- Al detectar el agente monitor del EVA que un estudiante se ha identificado en el EVA, mediante esta tarea procede a activar un agente de modelado de usuario asignado a ese modelo. El agente inicializa una conversación con este agente para enviarle los datos del usuario y curso para que continúe o inicie con el trabajo de creación o actualización del modelo de usuario.
- ◆ *Prototipar modelo usuario (prototiparModeloUsuario).*- Tarea que obtenido el estilo de aprendizaje del estudiante procede mediante los prototipos creados para ese estilo a generar un modelo de usuario inicial del estudiante modelado. También se contempla en prototipado del modelo de usuario sin los estilos de aprendizaje si el usuario se tarda mucho en contestar el test.
- ◆ *Inicializa detección cambios interacción usuario (inicializaDeteccionCambiosInteraccion).*- Mediante esta tarea un agente modelador pide al agente monitor del EVA mediante una conversación que compruebe si existen cambios en la interacción del estudiante para poder desarrollar la actualización de datos del modelo.
- ◆ *Modificar atributos modelo usuario (modificarAtributosModeloUsuario).*- En esta tarea el agente modelador después de obtener los cambios en la interacción del estudiante actualiza los atributos del modelo de usuario que han sufrido cambios en la interacción o sus valores son diferentes.

6.4.5 Tareas y entidades asociadas

En esta sección se muestran las tareas con las relaciones que estas tienen con las distintas entidades del entorno y los hechos que se manejan dentro del sistema.

Para la tarea activar agente de modelado de usuario que se muestra en la figura 6.7, se consume un slot de aplicación en el que se aloja un estudiante activo en un curso (*AccesosEvaEstudiantes*) y por ende estos son los agentes que se deben activar. Además esta tarea produce la conversación pasar datos al agente de modelado de usuario (*pasar_datosAmu*) Esta tarea incorpora la utilización de la aplicación, *Gestor*

Agentes para cumplir su cometido.

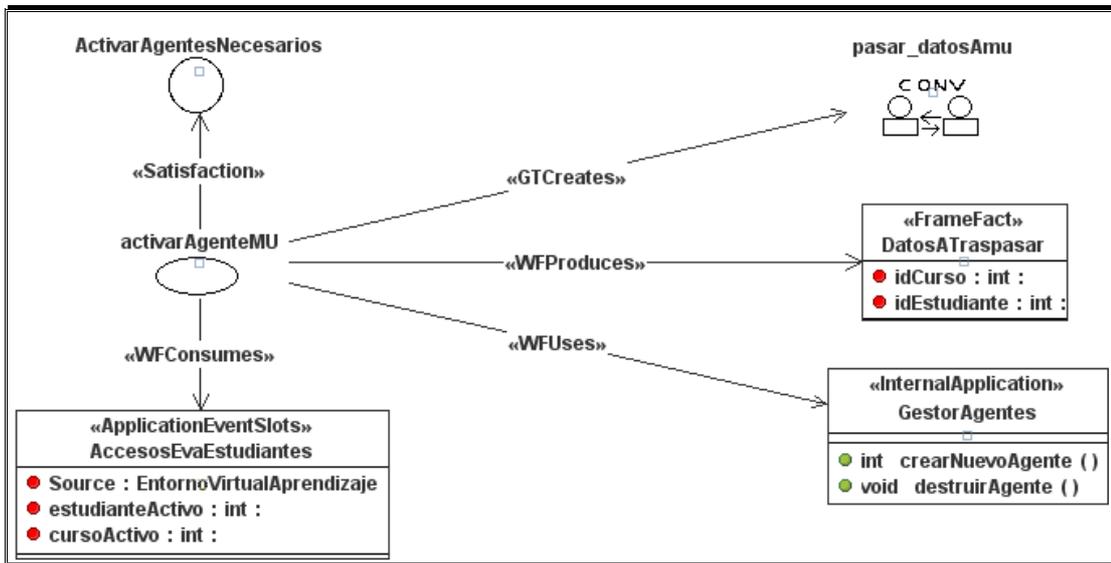


Figura 6.7: Entidades consumidas y producidas por la tarea activar agente MU

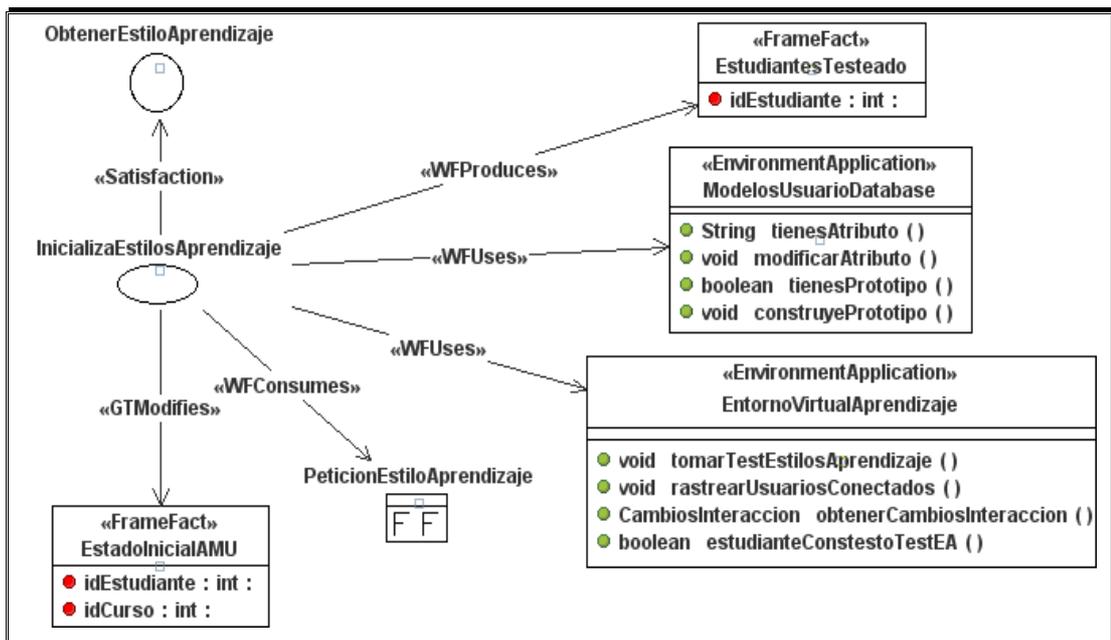


Figura 6.8: Entidades consumidas/producidas por la tarea solicitar estilos aprendizaje

En la tarea inicializa estilos de aprendizaje (*InicializaEstilosAprendizaje*) que se muestra en la figura 6.8, lo que se consume son dos estados (*EstadoInicialAMU*) y el estado (*PeticiónEstiloAprendizaje*), y utilizando la aplicación de la base de datos de modelos de usuario (*ModelosUsuarioDatabase*) se averigua si el estudiante realizó el test de estilos de aprendizaje, en el caso de no haber contestado aun el test se realiza

esta asignación mediante la aplicación del entorno virtual de aprendizaje (*EntornoVirtualAprendizaje*). Esta tarea produce un estado mental denominado (*EstudianteTesteado*).

La tarea prototipar modelo de usuario (*prototiparModeloUsuario*) que se muestra en la figura 6.9, consume un estado (*EstadoInicialAMU*) y un estado mental de (*PrototipadoModelo*), para realizar la construcción de los modelos de usuario de un estudiante en un curso, para esto usa la aplicación que accede a la base de datos de modelos de usuario (*ModelosUsuarioDatabase*). Produce como salida un estado denominado (*ModelosPrototipados*).

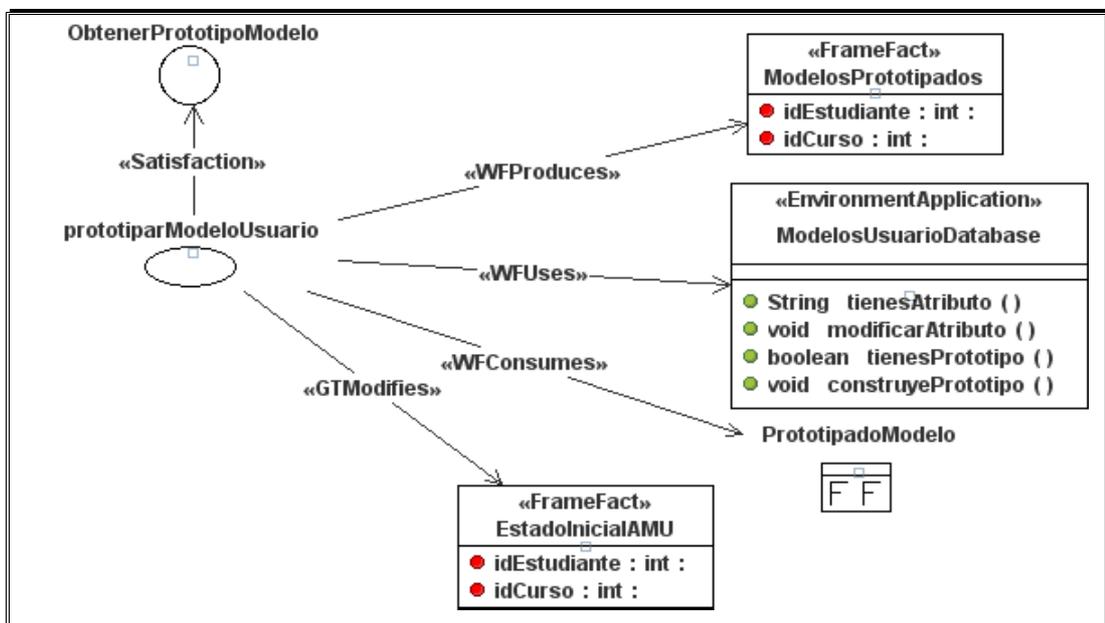


Figura 6.9: Entidades consumidas\ producidas por la tarea prototipar modelo usuario

En la figura 6.10, se muestra la tarea inicializa detección de cambios en la interacción (*inicializaDeteccionCambiosInteraccion*), que consume como entrada los estados mentales (*EstadoInicialAMU*) y un estado de (*DeteccionCambios*), para lo cual la genera un estado metal de salida denominado (*PedidoDeteccionCambios*) y para pedir al agente monitor del EVA, recolectar los cambios en la interaccion crea una conversación (*comprobar_interaccion*) para desarrollar este trabajo.

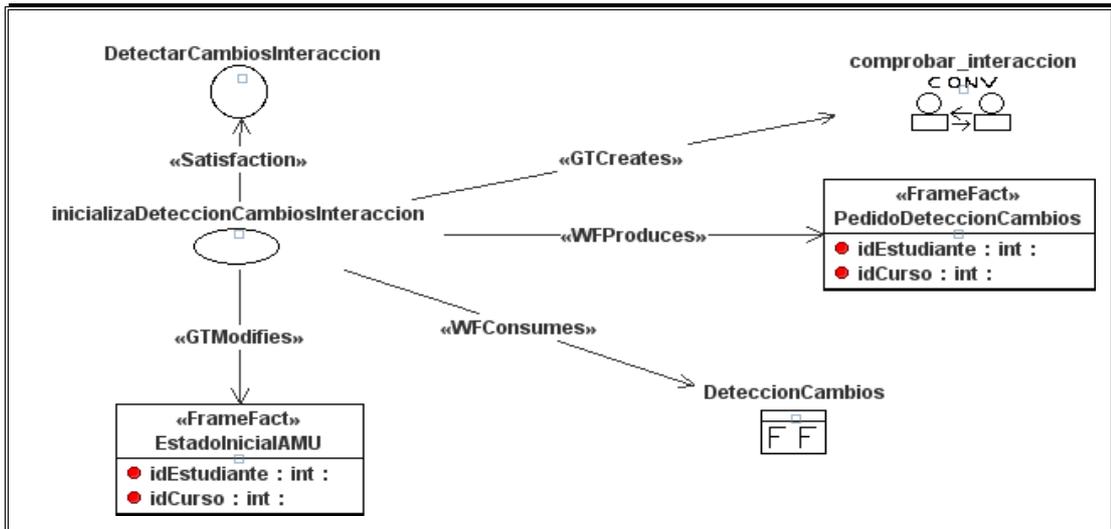


Figura 6.10: Entidades consumidas\producidas por la tarea detectar cambios interacción usuario

En la figura 6.11 se presenta la tarea modificar atributos del modelo de usuario (*ModificarAtributosModeloUsuario*), con la cual después de obtener los cambios en la interacción de los estudiantes con el EVA, se consumen peticiones de modificación de los atributos del modelo de usuario (*PedidosModificacionAtributos*) para el usuario requerido. Esta tarea necesita utilizar dos aplicaciones externas para poder desarrollar adecuadamente su trabajo, la base de datos de modelos de usuario (*Modelos de Usuario Database*), y un clasificador de atributos (*Clasificador de Atributos*), que se encarga de tomar los cambios en la interacción del sistema y los procesa mediante aprendizaje automático para poder así generar los nuevos atributos del modelo de usuario.

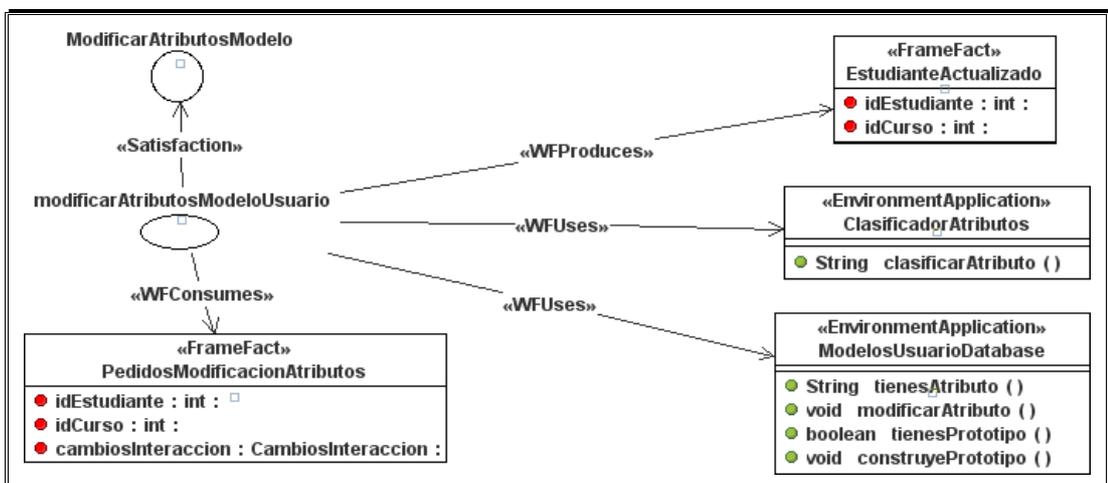


Figura 6.11: Entidades consumidas y producidas por la tarea modificar atributos modelo usuario

6.4.6 Modelos de agente

A los roles identificados se asocian a agentes que se encargan de cumplir con el trabajo de esos roles. Para el entorno virtual de aprendizaje de la UTPL se han identificado dos clases de agentes: El *Agente de modelado de Usuario*, y el *Agente monitor EVA*. Cada uno de ellos cumple diferentes roles y objetivos los cuales se describen a continuación:

El *Agente de modelado de usuario*, mostrado en la figura 6.12, desempeña el rol de *Modelador* para la creación de modelos de usuario, y para la recolección de datos de interacción y posterior actualización de modelos, esta tipología de agente cumple cinco objetivos durante su trabajo, estos objetivos son: *ConstruirModeloUsuario* objetivo necesario para realizar el prototipado de los atributos del modelo de usuario, *ObtenerEstiloAprendizaje* el estilo de aprendizaje es necesario para poder realizar el prototipado de los atributos del modelo de usuario, *MantenerMUActualizados* mediante este objetivo los agentes son capaces de rastrear cambios en la interacción para actualizar los modelos de los estudiantes, *DetectarCambiosInteraccion* para poder mantener los modelos actualizados se debe detectar las interacciones que los estudiantes han realizado en el EVA y *ActualizarModeloUsuario* luego de detectar las nuevas interacciones de los estudiantes en el entorno virtual de aprendizaje se debe de actualizar los atributos que han sufrido cambios.

Además en el gráfico se muestran las tareas que el agente cumple en el rol de modelador, estas tareas son:

- ◆ *InicializarEstilosAprendizaje*
- ◆ *inicializarDeteccionCambiosInteraccion*
- ◆ *PrototiparModelosUsuario*
- ◆ *ModificarAtributosModeloUsuario*

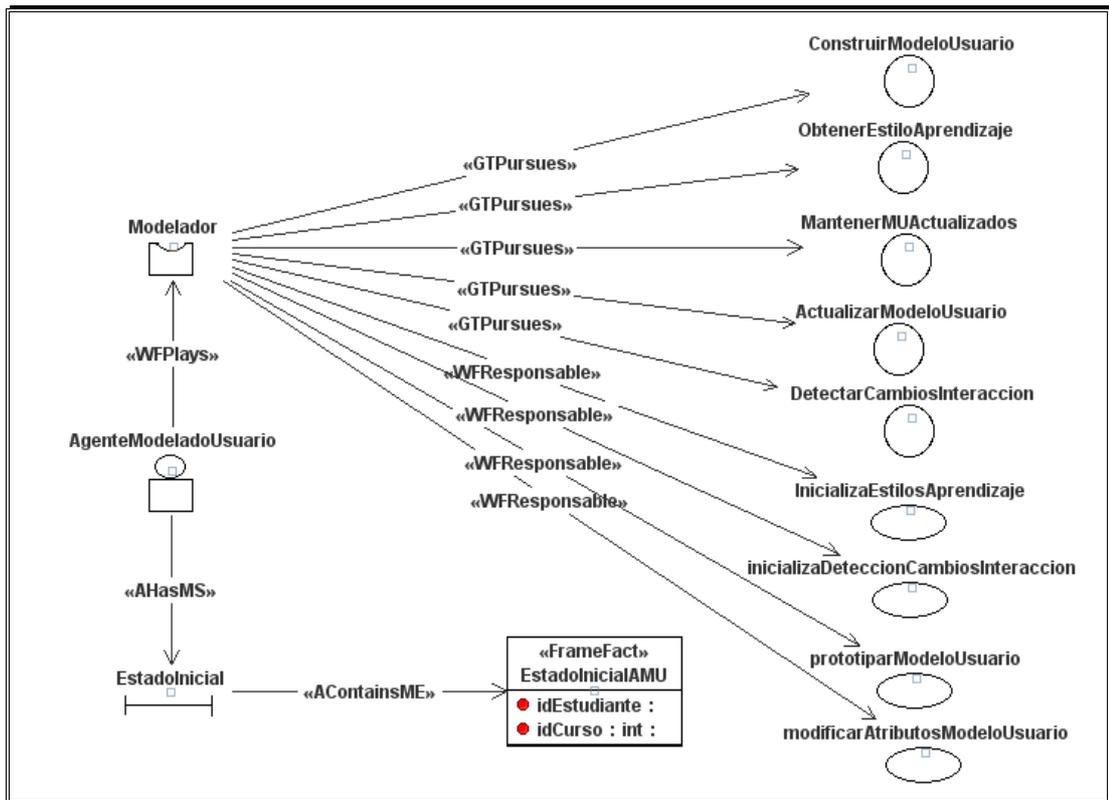


Figura 6.12: Modelo del agente de modelado de usuario

En la figura 6.13 se presenta al *Agente Monitor EVA*, el cual se encarga de rastrear los cambios y los accesos en el entorno virtual de aprendizaje y ayuda a acceder a sus herramientas. Dentro del sistema este agente cumple cinco objetivos para el desarrollo de su trabajo, el *ActivarAgentesNecesarios* para asegurar que agentes se encuentran modelando a estudiantes activos en el EVA, el *ObtenerEstiloAprendizaje* para asignar los test a los estudiantes, el *MantenerMUActualizados*. El agente monitor apoya al modelador para realizar este objetivo, el *ActualizarModeloUsuario* apoyando al agente modelador para actualizar los atributos del modelo de usuario facilitando las aplicaciones que soportan este proceso y el *DetectarCambiosInteraccion* monitoreando la interacción del estudiante a pedido del modelador. Para poder desarrollar estos objetivos desenvuelve un rol como *MonitorEVA*. Este agente desarrolla la tarea de *activarAgenteMU*.

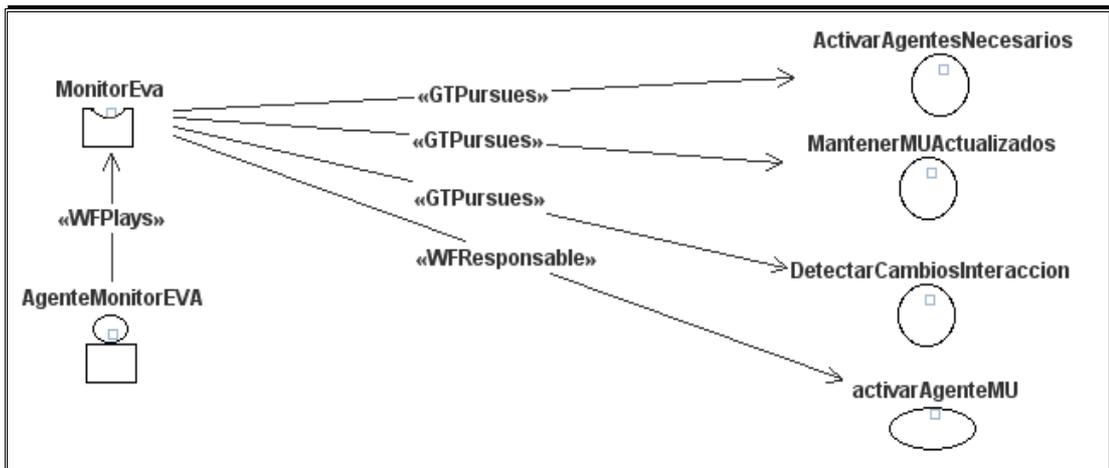


Figura 6.13: Modelo del agente monitor del EVA

El procesador de estado mental en los dos casos se basa en el algoritmo RETE. El mecanismo deliberativo utiliza reglas de producción. El gestor de estado mental tiene como restricción el que la introducción de nueva información se debe hacer escalonadamente. Sólo entra una nueva entidad de información cuando el agente ha terminado de tomar decisiones.

La inteligencia de los agentes se concreta en la forma en se adaptan a los a la interacción de los estudiantes que modelan tomando como referencia los resultados obtenidos en la deducción de los valores de atributos del modelo de usuario en los cambios en su interacción.

El grupo obtiene información de sus miembros mediante los procesos comprobación de interacción. Especialmente el agente de modelado de usuario al modelar los cambios en la interacción de los estudiantes se va adaptando a la frecuencia con que estos interactúan con el sistema y a los avances que presentan en el aprendizaje de cada contenido.

6.4.7 Nuevos elementos identificados en el entorno del sistema

De la generación de los diagramas de colaboración, en la figura 6.14, se rescatan algunas nuevas entidades que sirven como soporte a la ejecución de las interacciones descritas en cada modelo, estas entidades son: Un gestor de agentes (*GestorAgentes*), un gestor de miembros del grupo (*GestorMiembrosGrupo*) y un clasificador de atributos (*ClasificadorAtributos*).

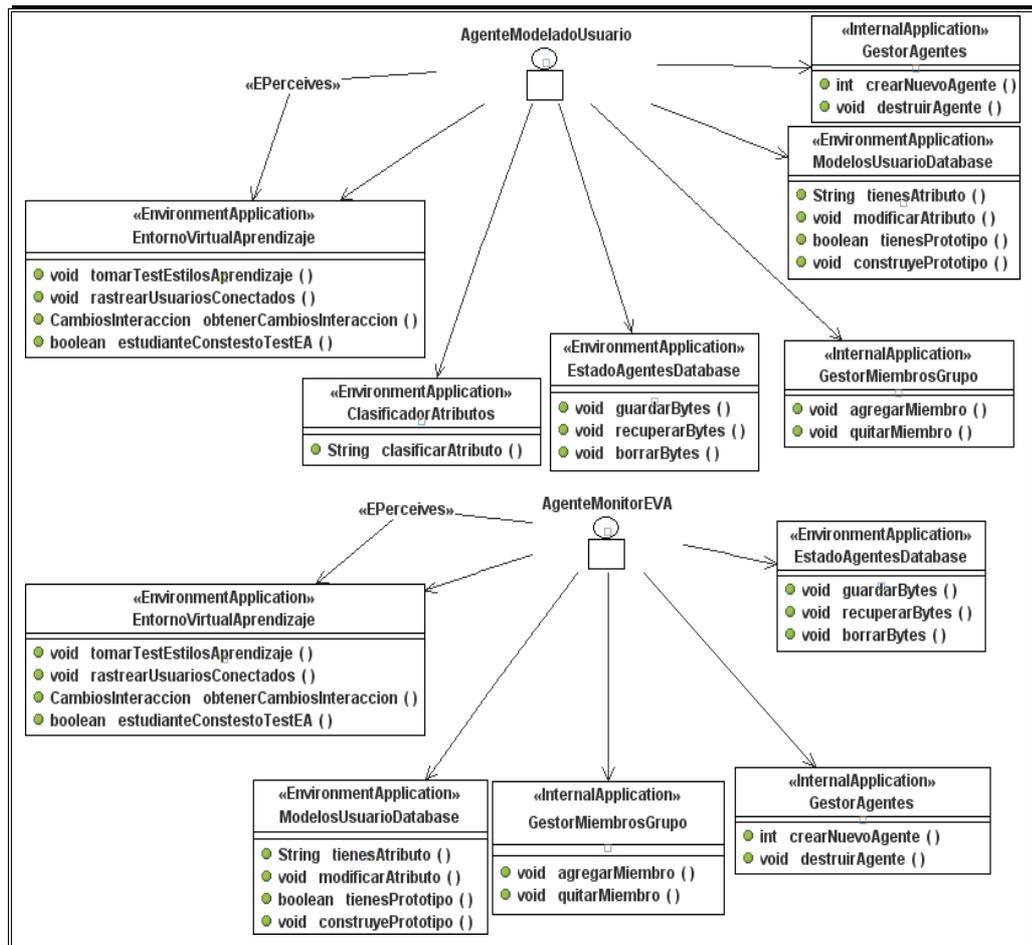


Figura 6.14: Refinación de modelo del entorno con los nuevo elementos identificados

La necesidad del primero se crea para poder localizar a los agentes en el sistema ya que ayuda en la creación y ubicación de nuevos agentes en los grupos. Y la necesidad del segundo aparece porque se lo utiliza como un mecanismo, para poder realizar la incorporación de los agentes a los grupos, así como su administración. El *Clasificador de Atributos* nace para soportar la tarea de actualización de los modelos de usuario.

En el diagrama también se implementan modelos iniciales de la percepción de los agentes hacia las entidades externas y cuáles de ellas estarán bajo su control.

6.4.8 Organización refinada del sistema

Con muchos elementos identificados dentro del sistema se hace una revisión de la arquitectura propuesta anteriormente para el sistema de modelado de usuario del

entorno virtual de aprendizaje de la UTPL, la nueva estructura se muestra en la figura 6.15.

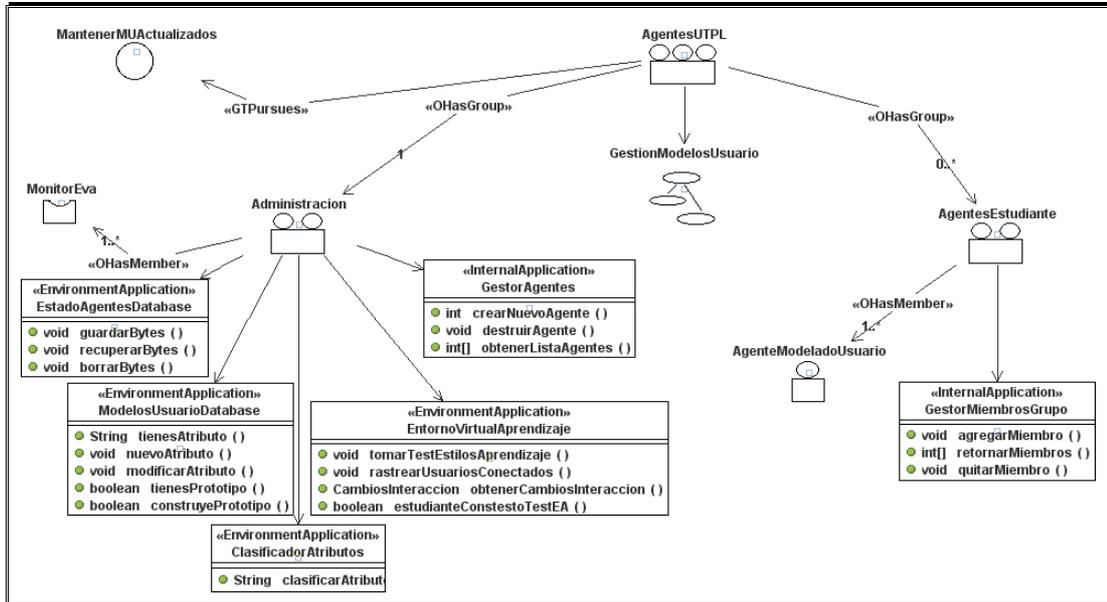


Figura 6.15: Organización del sistema multiagente de modelado de usuario del EVA UTPL

La organización se estructura de la siguiente forma: los agentes de modelado de usuario se juntan en el grupo *AgentesEstudiante* que es el más relevante, y enlaza los recursos propios del grupo. Estos recursos se utilizan para gestionar a los miembros del grupo.

El grupo Administración reúne un conjunto de recursos comunes a los agentes y que los pueden utilizar todos ellos. Lo componen, el *entorno virtual de aprendizaje*, la *base de datos de modelos de usuario*, un *clasificador de atributos*, una *base de datos del estado de los agentes* y un *gestor de agentes*, como aplicaciones dentro de la organización. Además el rol *MonitorEva* se encarga de monitorear los cambios que ocurren en el sistema.

Dentro de la organización se definen flujos de trabajo, para gestionar los modelos de usuario (*Gestión de Modelos de Usuario*) y los que gestionan el trabajo de los grupos (*Gestión de Grupos*), estos flujos se utilizan para organizar las tareas identificadas anteriormente.

6.5 Diseño – Elaboración

Se aumenta el nivel de detalle en la especificación determinando cómo se llevan a cabo los casos de uso identificados. Se refinan los flujos de trabajo y sus tareas asociadas, las interacciones, el control de los agentes y como se satisfacen los objetivos del sistema.

La especificación se refina anexando tareas a los flujos de trabajo, estas tareas se obtienen de los modelos de objetivos y tareas y de las interacciones esquematizadas en el análisis. Las tareas se unen entre sí conectando aquellas que producen entidades con aquellas que las consumen. Además se describen recursos que se consumen, detallando que tareas implican la ejecución de otras tareas en otros agentes, y quien se encarga de ejecutarlas. Estas tareas se identifican siguiendo las actividades descritas en la tabla 6.8.

Tabla 6.8: Actividades para la refinación de flujos en tareas

Actividad	Tipo de resultado	Referencia
Identificar tareas	<ul style="list-style-type: none"> Un conjunto de tareas asociadas a un flujo de trabajo 	
Conectar tareas	<ul style="list-style-type: none"> Instancias de <i>WFConecta</i> 	
Identificar tareas no locales	<ul style="list-style-type: none"> Tareas que producen interacciones 	Hecho en análisis elaboración
Identificar responsables	<ul style="list-style-type: none"> Instancias de <i>WFResponsable</i> asociando roles o agentes y tareas 	Figura 6.22 pagina 149 Figura 6.23 pagina 150
Identificar entidades mentales	<ul style="list-style-type: none"> Asociaciones de las tareas con entidades mentales mediante instancias de <i>WFProduce</i> y <i>WFConsume</i> 	Figura 6.16 pagina 146 Figura 6.17 pagina 147

Se relacionan los flujos de trabajo con las interacciones a través de entidades. Estas entidades especifican la ejecución de tareas no locales tomando como guía los diagramas de colaboración existentes. Esta especificación se muestra como un conjunto de diagramas GRASIA que parten de la especificación de los diagramas de colaboración de la etapa anterior. La conversión se inicia reemplazando los pasos de mensajes a unidades de interacción, explicando la necesidad de que el agente envíe el mensaje y que tareas se ejecutan el enviar o recibir mensajes. Finalmente estas unidades de interacción identificadas se organizan mediante un diagrama de flujo para definir en qué orden se ejecutan. En la tabla 6.9 se muestran las actividades necesarias

para especificación de estos flujos.

Tabla 6.9: Actividades para el desarrollo del detalle de los flujos del SMA

Actividad	Tipo de resultado	Referencia
Traducir mensajes	<ul style="list-style-type: none"> Diagramas GRASIA donde se asocian roles con unidades de interacción 	Figura 6.20 pagina 148 Figura 6.21 pagina 149
Establecer orden de ejecución	<ul style="list-style-type: none"> Relacionar las unidades de interacción mediante primitivas <i>UIPrecede</i>, <i>UIBifurca</i>, <i>UIItera</i>, <i>UIConcurren</i> 	
Establecer condiciones mentales	<ul style="list-style-type: none"> Instancias de patrón de estado mental asociado a unidades de interacción 	
Asociar tareas	<ul style="list-style-type: none"> Asociar tareas a instancias de unidad de interacción 	Hecho en análisis-elaboración

Para los modelos de agente se detalla como será el control de los mismos. Además se estudian los estado mentales extraídos de las interacciones y los requeridos por los flujos de trabajo, luego de obtenidos los estados mentales, se estudia cómo se pasa de un estado a otro, y se detalla cómo decide el agente que tarea ejecutar a continuación. Estas actividades se muestran en la tabla 6.10.

Tabla 6.10: Actividades para la especificación del control de los agentes

Actividad	Tipo de resultado	Referencia
Determinar cómo se pasa de un estado mental a otro	<ul style="list-style-type: none"> Referencias a diagramas GRASIA y a modelos de tareas y objetivos 	
Detallar los estados intermedios	<ul style="list-style-type: none"> Instancias de modelos de agente donde se asocia una instancia de agente concreto a instancias de estado mental 	
Detallar cómo se gestiona el estado mental	<ul style="list-style-type: none"> Instancias de modelos de tareas y objetivos donde las tareas producen, destruyen o modifican objetivos 	
Asociar los estados mentales a la ejecución de acciones	<ul style="list-style-type: none"> Instancias de modelos de tareas y objetivos donde las tareas producen, destruyen o modifican objetivos 	

Los modelos de entorno se refinan incluyendo los tipos de percepciones de los agentes y detallando cuales son los parámetros de los recursos. Los recursos se asocian con las tareas cuando sea necesario expresar necesidades relevantes, además

hay que conocer las necesidades para poder detallar los parámetros de los recursos es apropiado asociar recursos aplicaciones y tareas. Las actividades que refinan este modelo se muestran en la tabla 6.11.

Tabla 6.11: Actividades para el refinamiento del entorno y percepción la de Agentes

Actividad	Tipo de resultado	Referencia
Refinar la percepción de los agentes	<ul style="list-style-type: none"> • Instancias de <i>EPercibeNotifica</i> y <i>EPercibeMuestreo</i> asociando agentes y aplicaciones 	Figura 6.25 pagina 151
Definir los atributos propios de cada recurso	<ul style="list-style-type: none"> • Instancias de modelos de agente donde se asocia una instancia de agente concreto a instancias de estado mental 	
Asociar recursos, aplicaciones y tareas	<ul style="list-style-type: none"> • Instancias de modelos de tareas y objetivos donde las tareas producen, destruyen o modifican objetivos 	

Tabla 6.12: Actividades para el refinamiento de la satisfacción de objetivos y tareas

Actividad	Tipo de resultado	Referencia
Determinar recursos a consumir	<ul style="list-style-type: none"> • Instancias de <i>WFUsa</i> asociando tareas y recursos 	
Indicar qué recursos se restablecen	<ul style="list-style-type: none"> • Instancias de <i>WFProduce</i> asociando tareas y recursos 	
Asociar tareas con entidades mentales	<ul style="list-style-type: none"> • Instancias de <i>GTSatisface</i> y <i>GTFalla</i> 	
Refinar entidades mentales consumidas	<ul style="list-style-type: none"> • Instancias de <i>WFConsume</i> 	
Asociar tareas con operaciones de las aplicaciones	<ul style="list-style-type: none"> • Instancias de <i>WFUsaLlamada</i> 	
Refinar dependencias de objetivos	<ul style="list-style-type: none"> • Determinar dependencias Y/O ente objetivos 	Figura 6.24 pagina 150
Refinar satisfacción/fracaso de objetivos	<ul style="list-style-type: none"> • Instancias de <i>GTFallaObjetivo</i> o <i>GTSatisfaceObjetivo</i> asociando objetivos y tareas 	

Finalmente las tareas y los objetivos se estudian para definir exactamente como se satisfacen los objetivos y como se ejecutan las tareas. Para las tareas se escriben las postcondiciones y precondiciones del análisis con detalle, indicando que recursos se consumen, cuales se establecen, que ocurre con las entidades mentales requeridas, indicar si se necesitan nuevas entidades mentales, identificar las acciones a invocar de las aplicaciones. Por otro lado, los objetivos se refinan identificando las dependencias

entre ellos y como se satisfacen y fracasan. Las actividades necesarias para la ejecución de este trabajo se muestran en la tabla 6.12.

Los resultados obtenidos al ejecutar estas actividades al sistema multiagente de modelado de usuario se reflejan en las siguientes subsecciones.

6.5.1 Detalle de flujos de trabajo del sistema

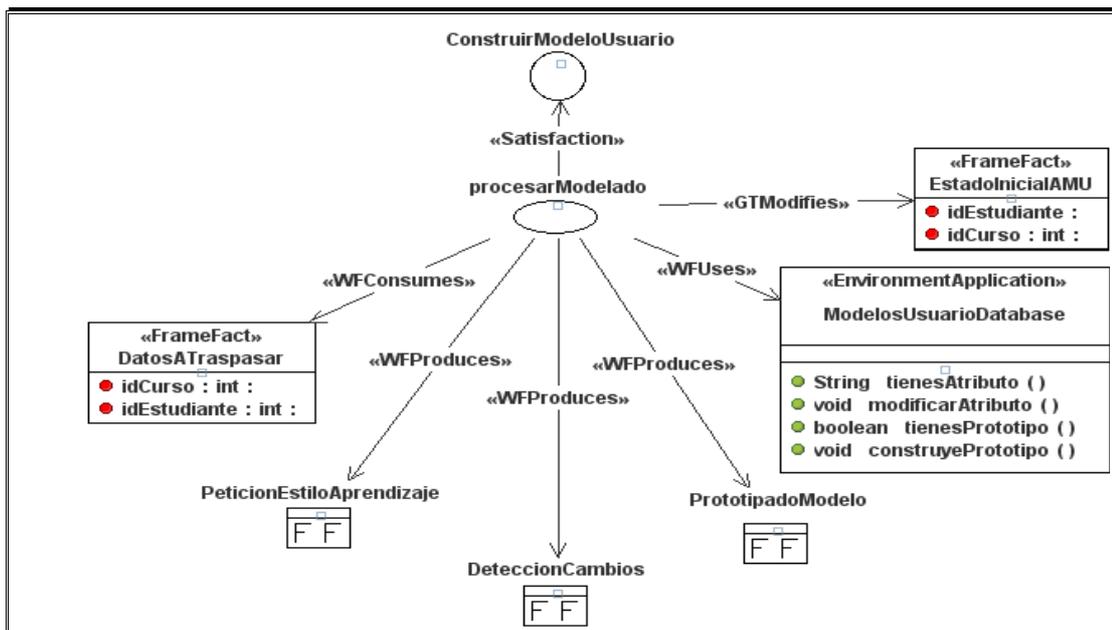


Figura 6.16: Detalle de la tarea procesar modelado

En la figura 6.16 se muestra la tarea procesar modelado (*procesarModelado*) que actúa como flujo de trabajo de la interacción (*pasar_datosAMU*) en la que el agente de modelado de usuario recibe, los datos de usuario en el curso para desarrollar el modelado de usuario. Para esto recibe la entidad de (*DatosATraspasar*) en la que le llegan los datos entregados por parte del agente monitor del EVA. Estos datos son procesados, reconociendo mediante la aplicación (*ModelosUsuarioDatabase*) el estado en que se encuentra el modelado de usuario. Dependiendo de ello genera cuatro salidas la primera es el estado inicial del agente (*EstadoInicialAMU*) quedando registrada, la propiedad del modelo al agente actual. El estado mental (*PeticionEstiloAprendizaje*) para evaluar la existencia de estilos de aprendizaje para este usuario, el estado mental (*PrototipadoModelo*) para crear el modelo de usuario si no lo está y el estado mental (*DeteccionCambios*) para que el agente inicie la

detección de cambios en la interacción del estudiante.

En la figura 6.17 se muestran las entidades mentales intercambiadas en el flujo de trabajo monitorear interacción. Sirven para la comunicación de datos entre los agentes. Las tareas encontradas en el proceso de definición de este flujo de trabajo son: (*detectarCambiosInteraccionUsuario*), (*procesaDetectarCambiosInteraccion*), (*obtenerCambiosInteraccion*) y (*respuestaDeteccionCambios*). Todas se definen para satisfacer el objetivo (*DetectarCambiosInteraccion*).

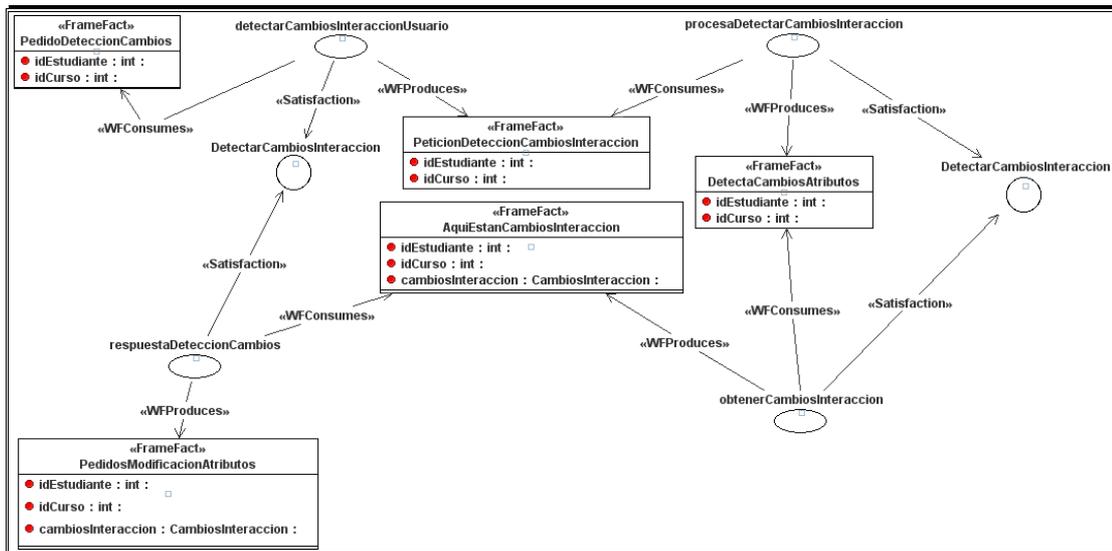


Figura 6.17: Detalle del flujo de trabajo monitorear interacción

6.5.2 Refinamiento de interacciones

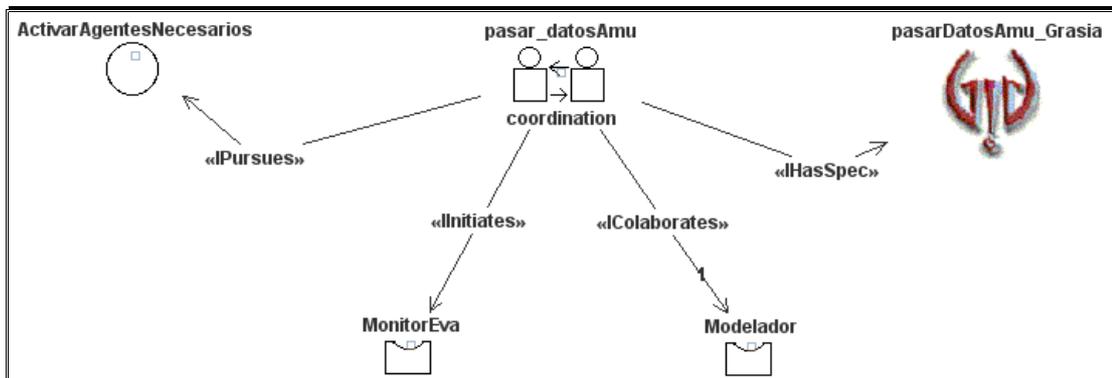


Figura 6.18: Refinamiento de la interacción pasar datos al agente de modelado de usuario

En la figura 6.18 se muestra especificación de la ejecución de las tareas asociadas a la interacción (*pasar_datosAMU*). Esto se realiza mediante la asociación del diagrama

GRACIA (*pasarDatosAmu_Gracia*) que esta especificado en la figura 6.20.

En la figura 6.19 se muestra especificación de la ejecución de las tareas asociadas a la interacción (*comprobar_interaccion*). Esto se realiza mediante la asociación del diagrama GRACIA (*comprobarInteraccion_Gracia*) que esta especificado en la figura 6.21.

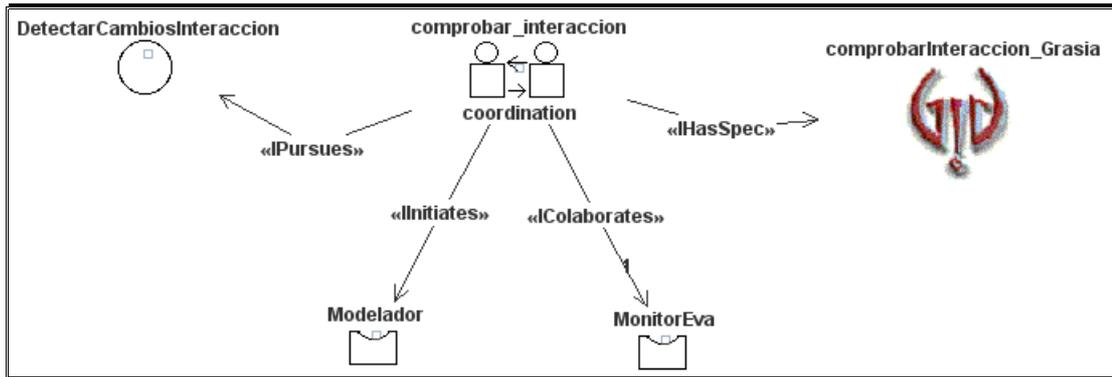


Figura 6.19: Refinamiento de la interacción comprobar interacción

6.5.3 Protocolos para el desarrollo de las interacciones

La unidad de interacción reflejada en la figura 6.20 provienen del diagrama de colaboración mostrado en la figura 6.3. Interpretando la unidad de interacción como un paso de mensaje y a cada tarea como la acción esperada por el emisor y el receptor. Además se refleja la entidad mental (*DatosATraspasar*) como parte del mensajes que recibe el modelador de parte del monitor del EVA.

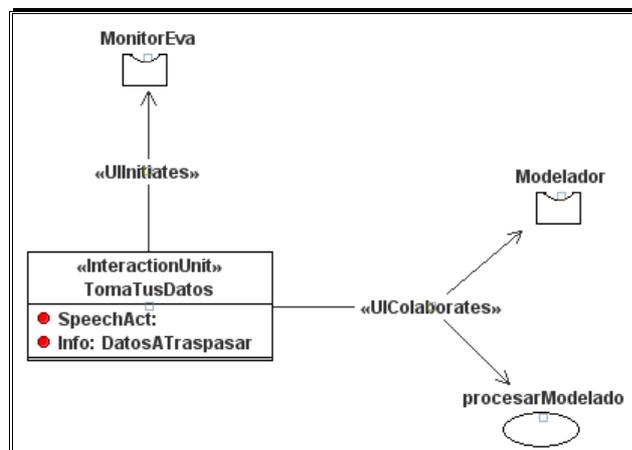


Figura 6.20: Protocolo utilizado por la interacción activar agente

Las unidades de interacción reflejadas en la figura 6.21 provienen del diagrama de colaboración mostrado en la figura 6.4. Interpretando cada unidad de interacción como un paso de mensaje y a cada tarea como la acción esperada por el emisor y el receptor. Además en la figura se muestra el ordenamiento para la ejecución de estas unidades de interacción.

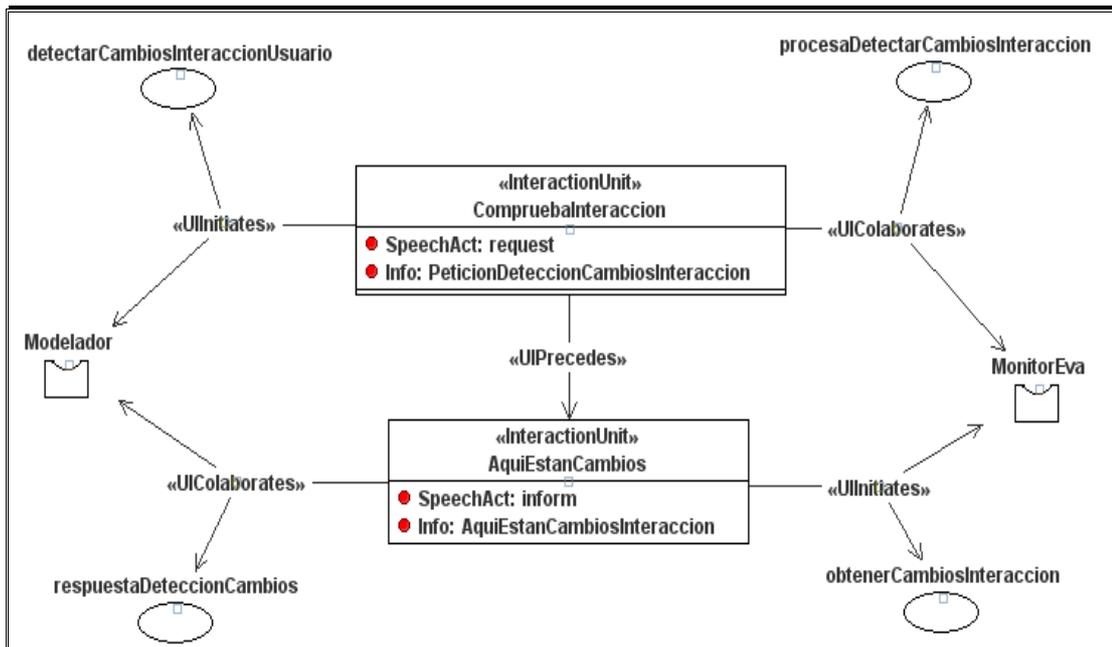


Figura 6.21: Protocolo utilizado por la interacción comprobar interacción

6.5.4 Responsabilidades en las tareas por parte de los roles

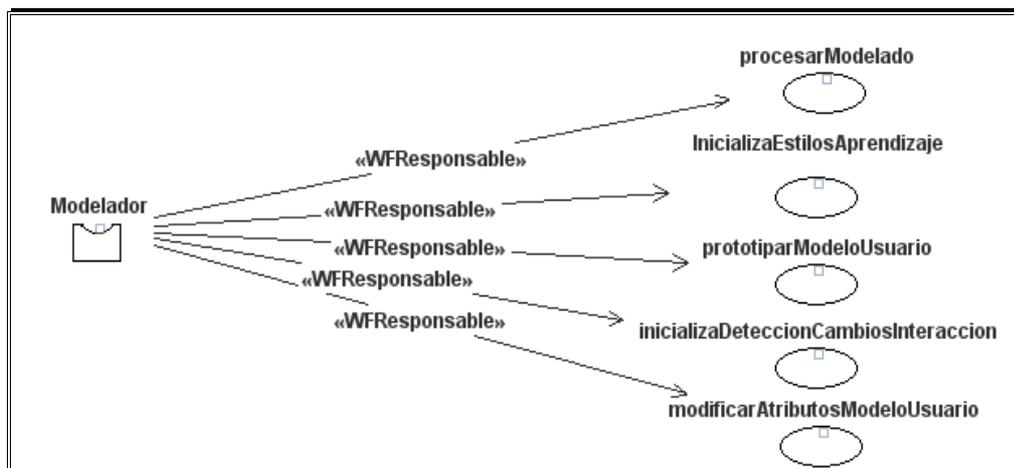


Figura 6.22: Responsabilidades de la ejecución de tareas para el agente de modelado de usuario

En la figura 6.22 se muestra la asociación de roles con las tareas en las que

interviene el agente de modelado de usuario.

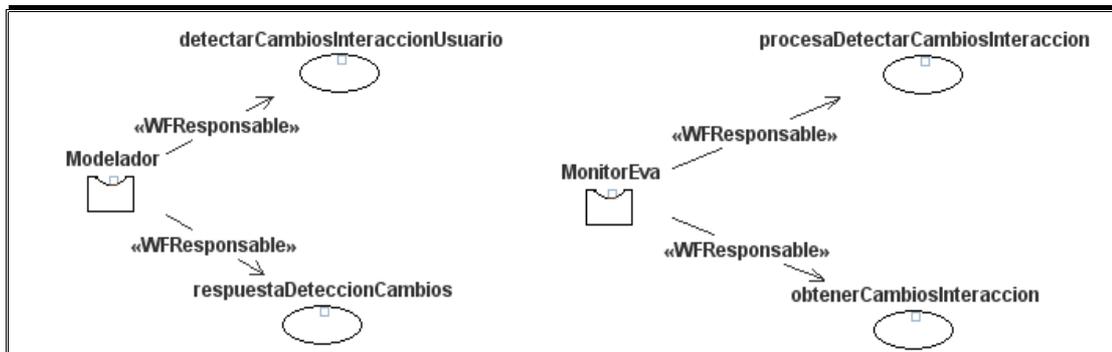


Figura 6.23: Responsabilidades de la ejecución de tareas para el flujo monitorear interacción

En la figura 6.23 se muestra la asociación de roles con las tareas indicadas en la figura 6.21.

6.5.5 Dependencias entre objetivos

La satisfacción de los objetivos del sistema no depende exclusivamente de la existencia de entidades mentales. Existiendo la posibilidad de que los objetivos puedan alcanzarse o fracasar debido a las dependencias que puedan tener respecto de otros objetivos.

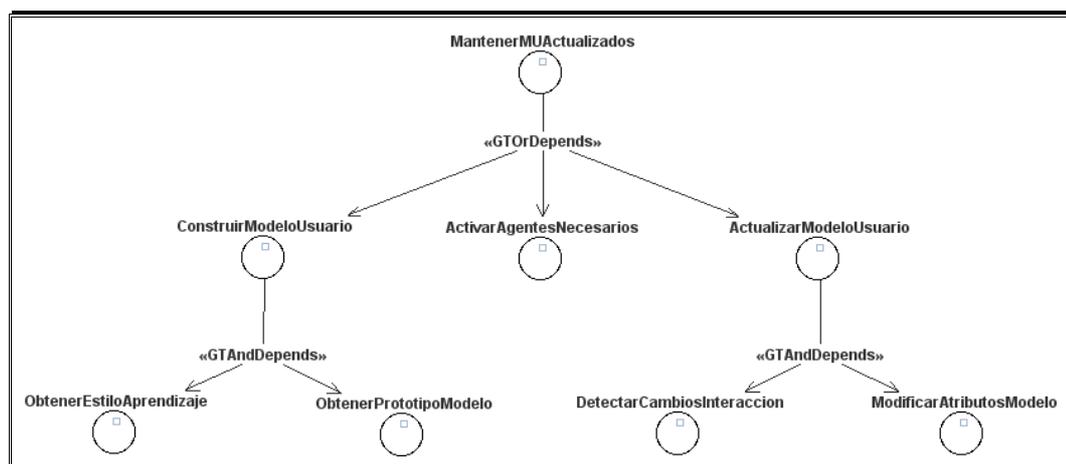


Figura 6.24: Dependencias entre objetivos

Es así que en la figura 6.24, para satisfacer el objetivo *MantenerMUActualizados* se pueden satisfacer cualquiera de sus subobjetivos, ya que esto le da opciones al sistema para que la creación y actualización de modelos de usuario se haga

consistentemente.

Los subobjetivos de este objetivo se manejan de forma diferente ya que *ConstruirModeloUsuario* y *ActualizarModeloUsuario* necesitan que se cumplan primero cada uno de sus subobjetivos. Esto se debe a la dependencia que existe entre uno u otro.

6.5.6 Percepción de los agentes

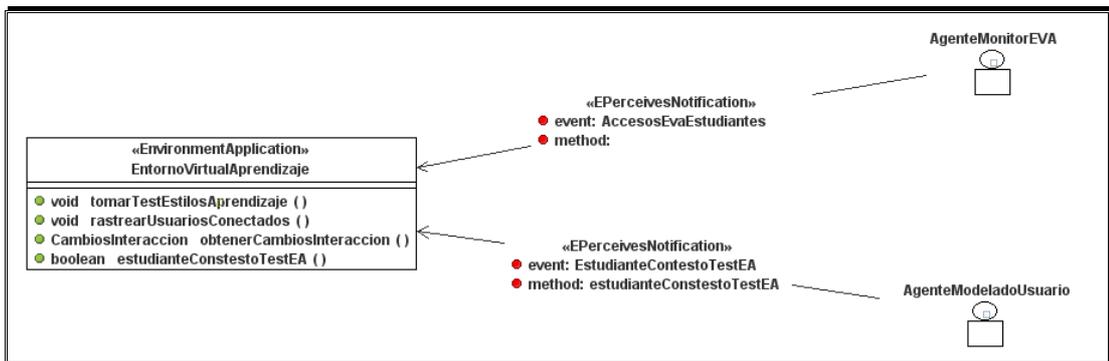


Figura 6.25: Percepción de los agentes

De la percepción del entorno son dos los agentes que realizan esta actividad, el agente monitor del EVA y el agente de modelado de usuario el primero que necesita saber los accesos de los estudiantes al entorno virtual de aprendizaje, y el segundo para reaccionar a la contestación de los estudiantes al test de estilos de aprendizaje. La especificación de la percepción de estos agentes al entorno se modela en la figura 6.25.

6.5.7 Arquitectura del Sistema

En la figura 6.26 se muestra el funcionamiento y la interacción del sistema del agente de modelado de usuario en el se presentan los módulos y los componentes que intervienen dentro del mismo, para obtener y procesar la información necesaria con la que se calcula los modelos de usuario de los estudiantes de la UTPL.

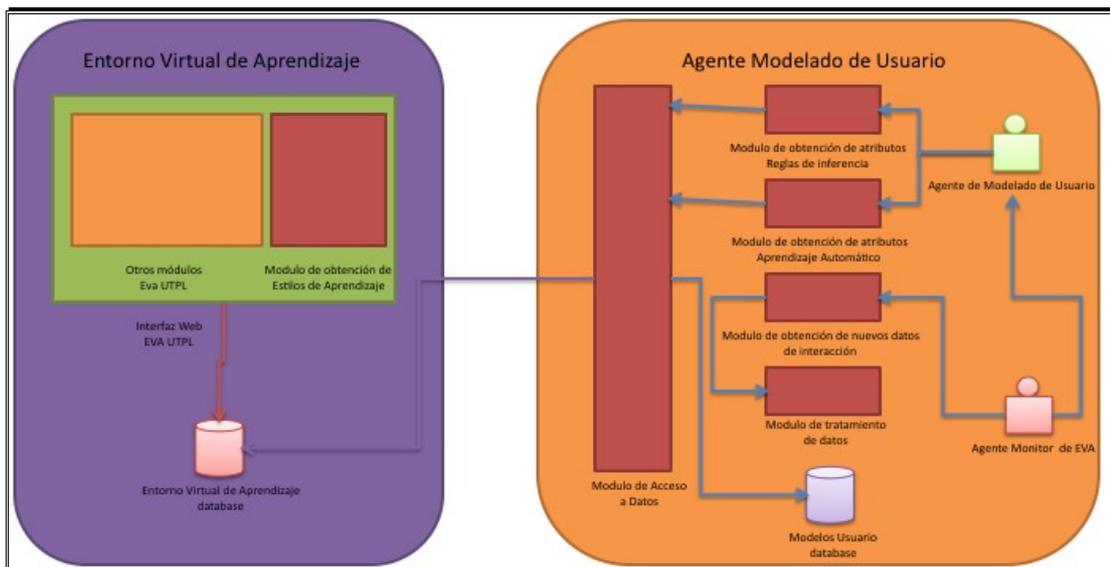


Figura 6.26: Arquitectura del agente de modelado de usuario y su interacción con el EVA UTPL

6.6 Implementación

Esta fase de Implementación arroja como resultado dos modelos en los que se obtiene los componentes de código mediante los cuales se ejecutan las aplicaciones encontradas en el proceso de análisis y diseño, y el código que ejecutan las tareas que intervienen en cada uno de los agentes que componen el sistema, este código debe ser implementado para que el sistema puede funcionar correctamente y realizar la tarea para la que ha sido diseñado y desarrollado.

6.6.1 Aplicaciones y enlaces de código fuente

El código fuente con el que se enlaza cada una de las aplicaciones encontradas en el proceso de análisis y diseño del sistema multiagente tiene que ser mapeado a archivos en los que se encuentra el mencionado código que dependerá del lenguaje empleado para la desarrollo del sistema. Este código contendrá la implementación de cada uno de los métodos referidos en cada una de las aplicaciones.

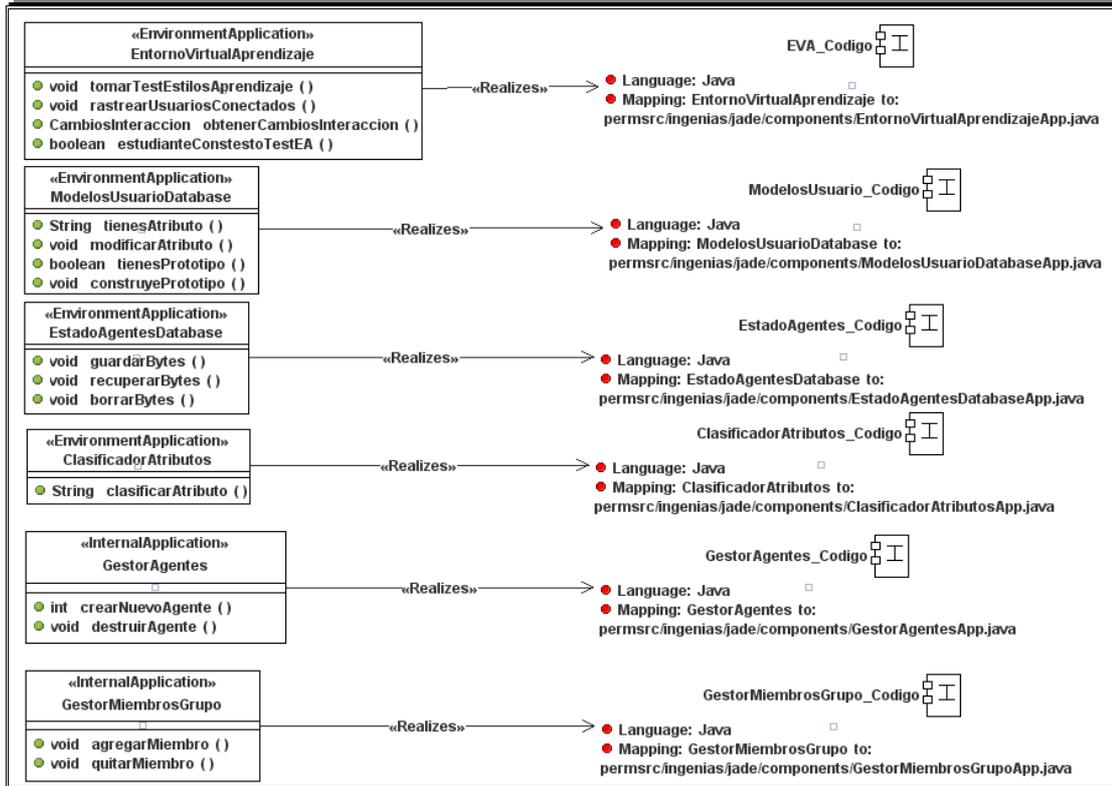


Figura 6.27: Aplicaciones y el código fuente asociado a su implementación

6.6.2 Tareas y enlaces de código fuente

En la figura 6.28 se muestran los componentes de código enlazados a cada una de las tareas que son referidas dentro del sistema, en estos componentes se debe incluir las implementaciones que deben cumplir los agentes para desarrollar cada tarea. Esta implementación, al igual que con las aplicaciones, dependerá del lenguaje elegido para el desarrollo del sistema. Además la implementación también estará influenciada por las aplicaciones que tenga enlazada cada tarea.

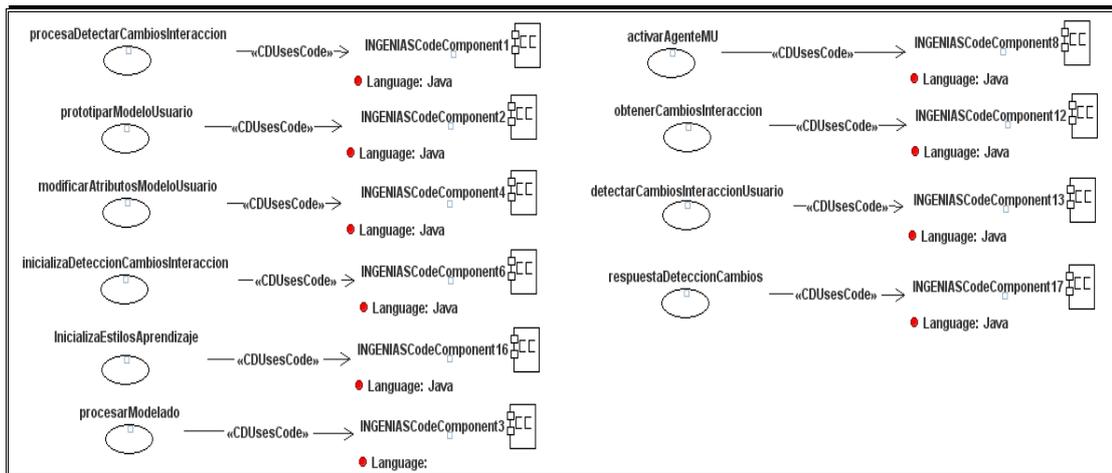


Figura 6.28: Tareas y componentes de código asociados a su implementación

En el caso de la implementación de las cinco aplicaciones encontradas en el análisis y diseño del sistema multiagente de modelado de usuario, surgió la necesidad de implementar algunos módulos adicionales al sistema que ayuden a el trabajo y acciones que no están contempladas dentro de la metodología. Para cada uno de estos módulos en las sub-secciones siguientes se presentan descripciones de su diseño e implementación.

6.6.3 Diseño e implementación del módulo de acceso a datos

El módulo de acceso a datos del sistema multiagente de modelado de usuario tiene como finalidad permitir un acceso centralizado y eficiente a las bases de datos relacionales con las que trabaja el sistema multiagente de modelado de usuario. Se necesita que el modelo de conexiones que mantenga este módulo, sea capaz de trabajar con un alto grado transaccional de acceso a los a los modelos de datos, ya que la generación y actualización de modelos de usuario de estudiantes conlleva un alto grado de acceso a los modelos de datos de los sistemas de los que se obtienen los datos y a los modelos de datos en los que guarda sus resultados.

Utilizando un enfoque orientado a objetos en la figura 6.29 se muestra el diagrama de clases del modulo de acceso a datos. Se han organizado en dos paquetes (*mud*) para el acceso a la base de datos de modelos de usuario y el (*evad*) para el acceso al base de datos del entorno virtual de aprendizaje.

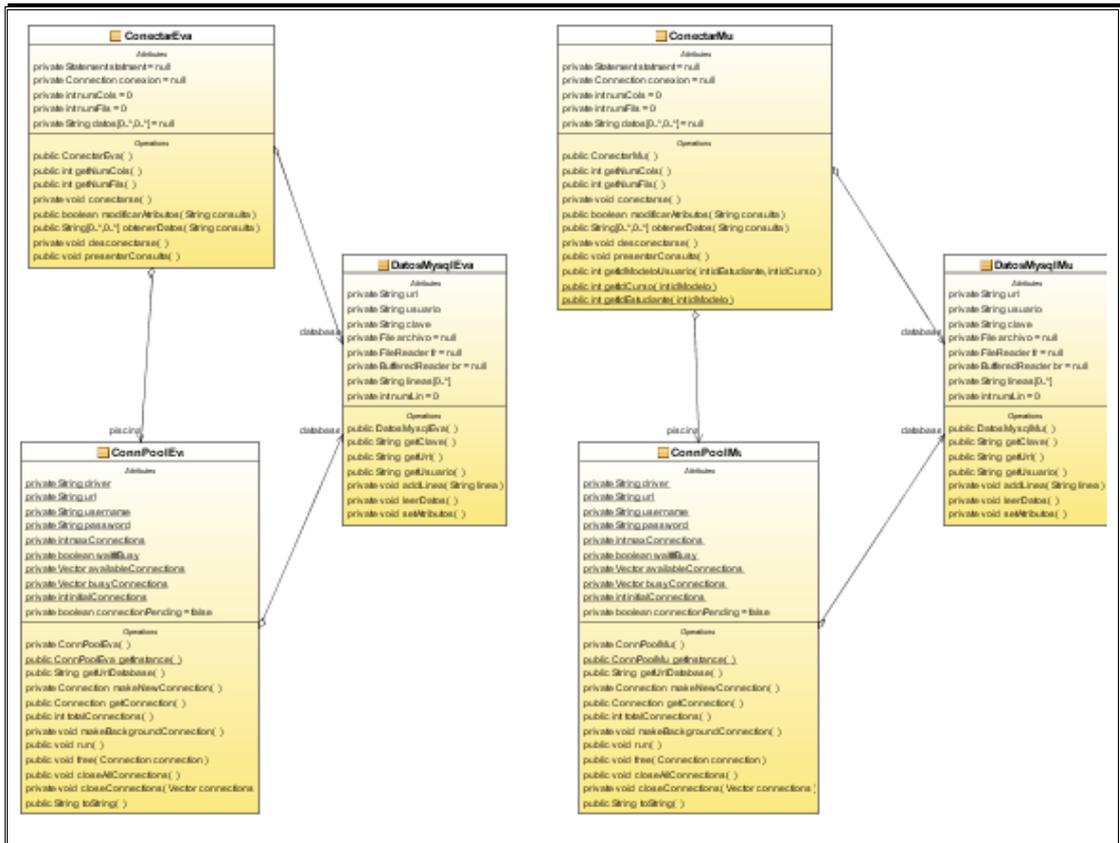


Figura 6.29: Diagrama de clases de módulo de acceso a datos

Los datos correspondientes al *usuario*, *password* y *url* de cada la base de datos se cargan al módulo por medio de archivos externos de texto plano por las clases *DatosMysqlEva* y *DatosMysqlMu* respectivamente.

En del desarrollo de este módulo se emplea el concepto de piscina de conexiones por medio de la cual se hace un acceso ordenado a los recursos de las bases de datos, por lo tanto las clases *ConnPoolEva* y *ConnPoolMu* implementan este concepto. Además como las piscinas de conexiones se crean una sola ves se necesita que en tiempo de ejecución se acceda a una solo instancia de ellas, esto se logra mediante la implementación del patrón de diseño *Singleton*.

Con el soporte mostrado en los párrafos anteriores las clases *ConectarEva* y *ConectarMu* implementan métodos para el acceso y modificación de los datos en las bases de datos correspondientes.

6.6.4 Diseño e implementación del módulo de obtención de estilos de aprendizaje del entorno virtual de aprendizaje de la UTPL

El módulo de obtención de estilos de aprendizaje se utiliza para evaluar a los estudiantes de la UTPL dentro del entorno virtual de aprendizaje (EVA). El módulo debe ser parte de la plataforma para comodidad de los estudiantes y tiene que tener la capacidad de asignar el desarrollo del mencionado test desde el sistema multiagente de modelado de usuario. El test a implementar es el de estilos de aprendizaje de Felder y Silverman elegido en la sección 3.3.2 del Capítulo 3.

Para el desarrollo de este módulo se crearon algunos archivos en el lenguaje de programación php en los que se implementaron las funciones que se despliegan el test de estilos de aprendizaje y realizan su evaluación. Además de la creación de un script en lenguaje *SQL* que incrementa algunas tablas a la base de datos, para soportar el proceso de evaluación de los estudiantes. En el Anexo E se desarrolla una descripción de los archivos creados

6.6.5 Diseño e implementación del módulo de obtención de atributos por medio de reglas de inferencia

En este módulo se implementan los métodos que permiten obtener los valores que del modelo de usuario que se calculan mediante reglas de inferencia. Se desarrolla de esta manera ya que cada atributo necesitara que se utilice una función diferente para obtener su valor. Se necesita que el módulo permita una implementación extensible ya que en un futuro se prevé la implementación de nuevos atributos de este tipo y es deseable que las nuevas formas de obtener atributos no distorsionen los atributos ya definidos anteriormente.

Mediante un enfoque orientado a objetos en la figura 6.30 se presenta el diagrama de clases del módulo de obtención de atributos por medio de reglas de inferencia. Pensando en la posterior extensión del módulo cada nuevo atributo tendrá una clase propia, que hereda datos y métodos de la clase abstracta (*Inferencia*), para obtener el valor del atributo, estas clases predefinen el método (*calcularAtributo()*) para definir sus propias formas de realizar esta tarea.

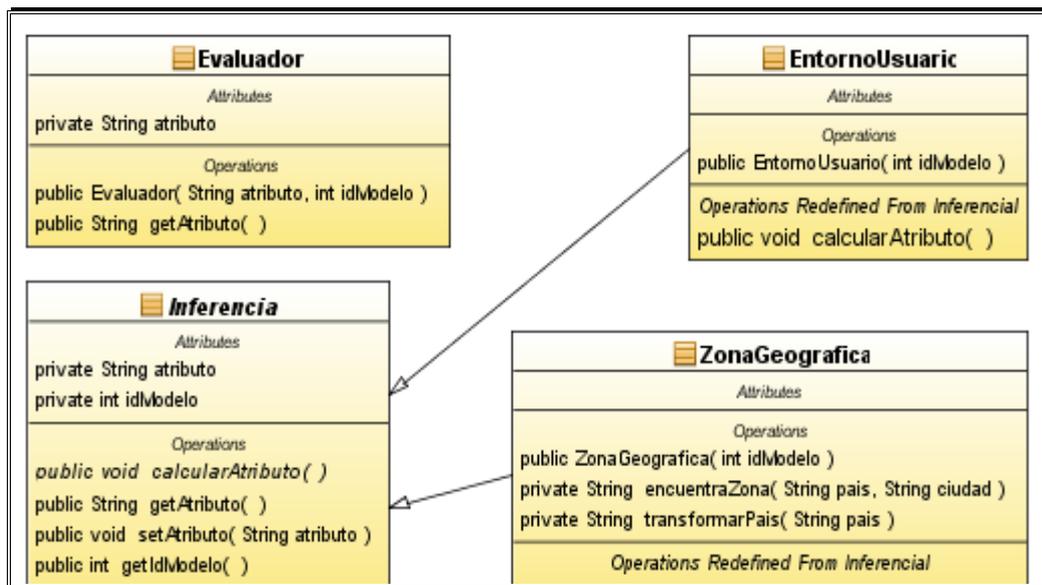


Figura 6.30: Diagrama de clases del módulo de obtención de atributos por medio de reglas de inferencia

Como son dos los atributos del modelo de usuario que son definidos mediante esta técnica en el diagrama se presentan las clases (*EntornoUsuario* y *ZonaGeografica*) que ayudan a calcularlos.

El calculo de estos atributos esta coordinado por la clase (*Evaluador*) que es la que se encarga de elegir, según el atributo del que se le pida obtener valor, que clase necesita para calcularlo. Todos las clases referidas se encuentran dentro del paquete (*inferencia*).

6.6.6 Diseño e implementación del módulo de obtención de atributos por medio de aprendizaje automático (WEKA-Util)

El módulo de obtención de atributos por medio de aprendizaje automático (WEKA-Util) fue creado para la obtener atributos mediante la utilización de modelos de aprendizaje automático que son el resultado del proceso de minería de datos realizado en el Capítulo 4, a partir del software de Data Mining WEKA de la universidad de Waikato. La idea es utilizar los algoritmos ya implementados en WEKA y con archivos de modelos de aprendizaje, más datos de entrenamiento definidos anteriormente, para clasificar nuevas instancias que el sistema necesita obtener.

Este módulo consta de cinco clases que se presentan en la figura 6.31 que permiten clasificar atributos, con ciertos parámetros de entrada.

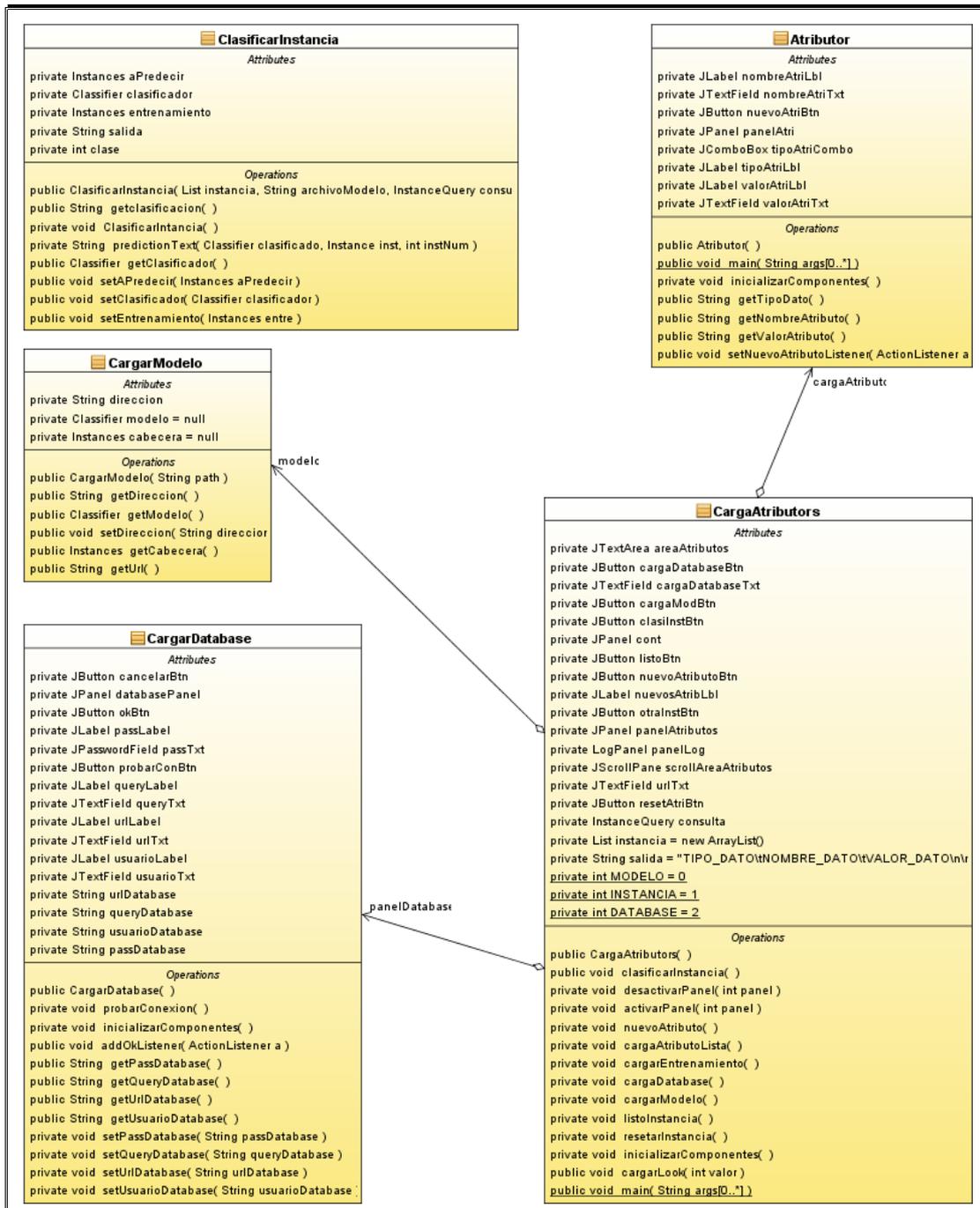


Figura 6.31: Diagrama de clases del módulo de obtención de atributos por medio de aprendizaje automático (Weka-Util)

También existen dos clases que implementan una interfaz de usuario que permite probar modelos creados en WEKA clasificando nuevos atributos, mediante esto el módulo puede funcionar independientemente del sistema de modelado de usuario.

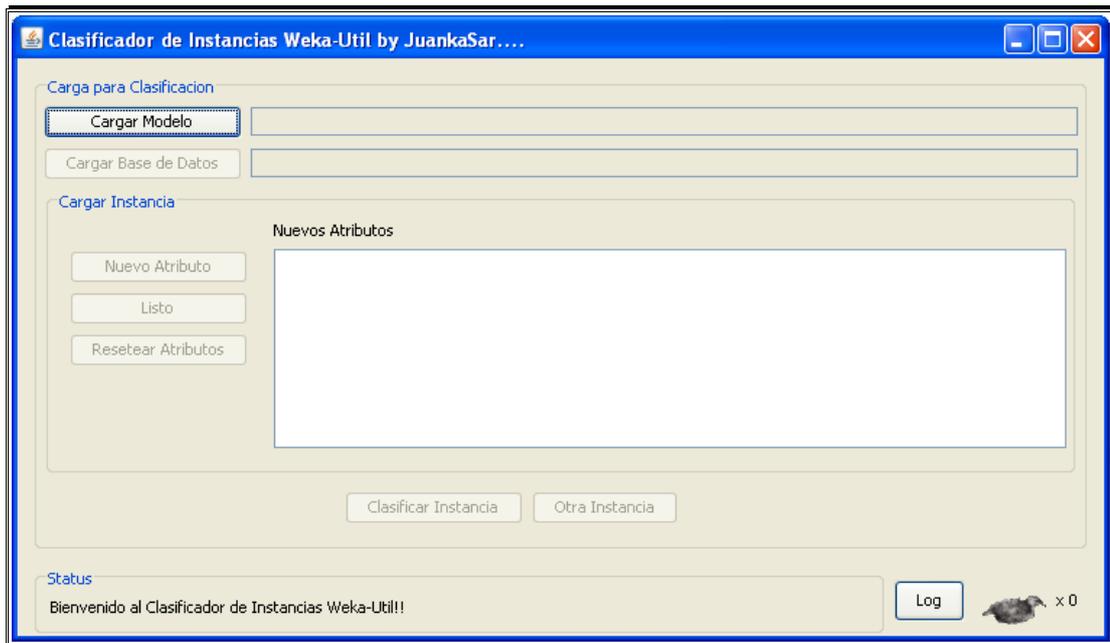


Figura 6.32: Interfaz del módulo de obtención de atributos por medio de aprendizaje automático (WEKA-Util)

En la figura 6.32 se muestra esta interfaz en donde se diferencian los parámetros que necesita el módulo para ejecutar una nueva clasificación. Necesita cargar un modelo generado anteriormente en WEKA, los datos de la base (*url*, *username*, *password*) en donde se encuentran los datos de entrenamiento del modelo, y finalmente se necesita cargar una instancia a clasificar.

Para el desarrollo de este módulo utiliza los métodos y algoritmos definidos en WEKA, contenidos en los paquetes (*classifiers*, *core*, *experiment*, *gui*). La clase denominada (*ClasificarInstancia*) es la que se encarga de recoger los valores de entrada y al crear un nuevo objeto, y mediante el método (*getclasificacion()*) se recupera el resultado obtenido al clasificar la instancia con los parámetros suministrados al construir el objeto. La clase (*CargarModelo*) se encarga de tomar una dirección en el sistema de archivos en donde se encuentra un modelo de clasificación creado a partir de algoritmos de aprendizaje automático y carga el modelo en memoria para que pueda ser utilizado por el sistema.

6.6.7 Diseño e implementación del módulo de obtención de cambios en la interacción

El sistema multiagente de modelado de usuario necesita rastrear los cambios en la

interacción de los estudiantes, para ello se hace necesaria la implementación de un módulo que permita conjuntar estos cambios de cierta forma, para que estos se puedan enviar entre los agentes que modelan a los usuarios. Es necesario que este módulo se pueda extender de forma fácil ya que se prevé que en el futuro se incorporen al sistema nuevos atributos a calcular y de esos atributos se necesitaran nuevas formas de compartir los cambios de cada atributo.

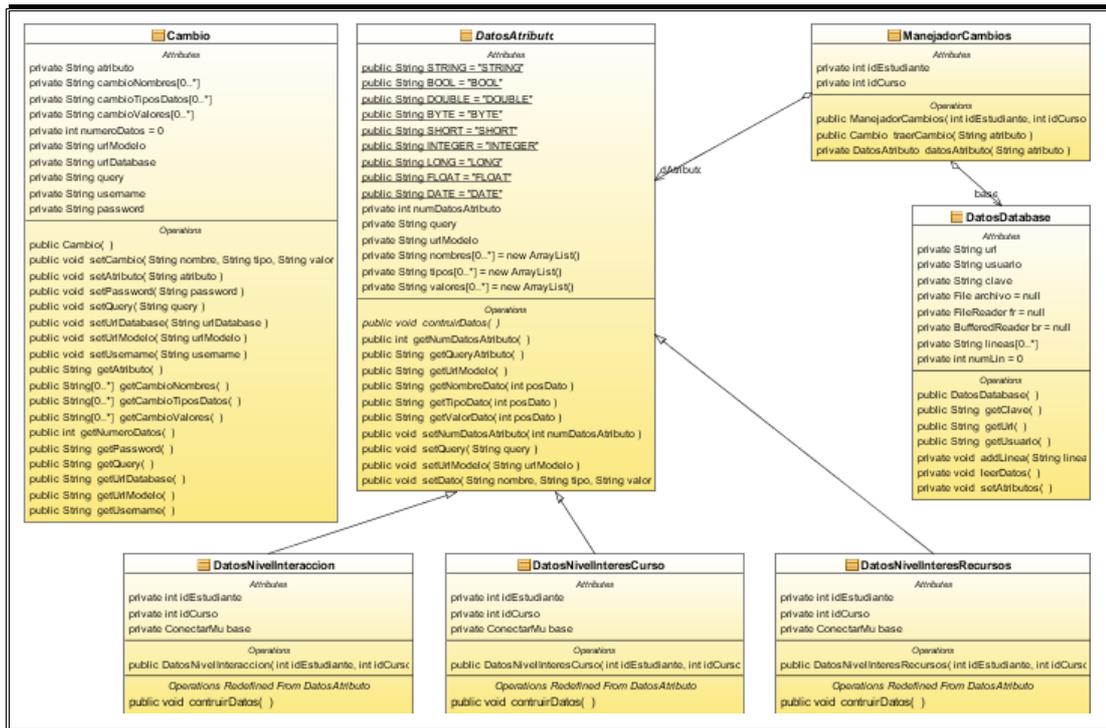


Figura 6.33: Diagrama de clases del módulo de obtención de cambios en la interacción

En la figura 6.33 se muestra el diagrama de clases para el módulo de obtención de cambios. Para la implementación de este módulo se necesita crear un objeto (*Cambio*) en el que se manejen todos los datos referentes al cambio en la interacción de valores que definen un atributo del modelo de usuario. Además se necesita de un esquema en el que se permita obtener cambios para cada atributo. Esto se logra definiendo una clase abstracta (*DatosAtributo*) que mantenga los datos comunes a todas las formas de obtener los valores para los cambios en la interacción de cada atributo, y mediante herencia calcular en las clases (*DatosNivelInteraccion*, *DatosNivelInteresCurso* y *DatosNivelInteresRecursos*), el valor del cambio para cada atributo. Toda la obtención de los cambios es coordinada por la clase (*ManejadorCambios*), que elige la clase que contiene el código preciso para calcular los valores del cambio que le a sido pedido al

objeto.

6.6.8 Diseño e implementación del módulo de actualización de estadísticas del entorno virtual de aprendizaje de la UTPL

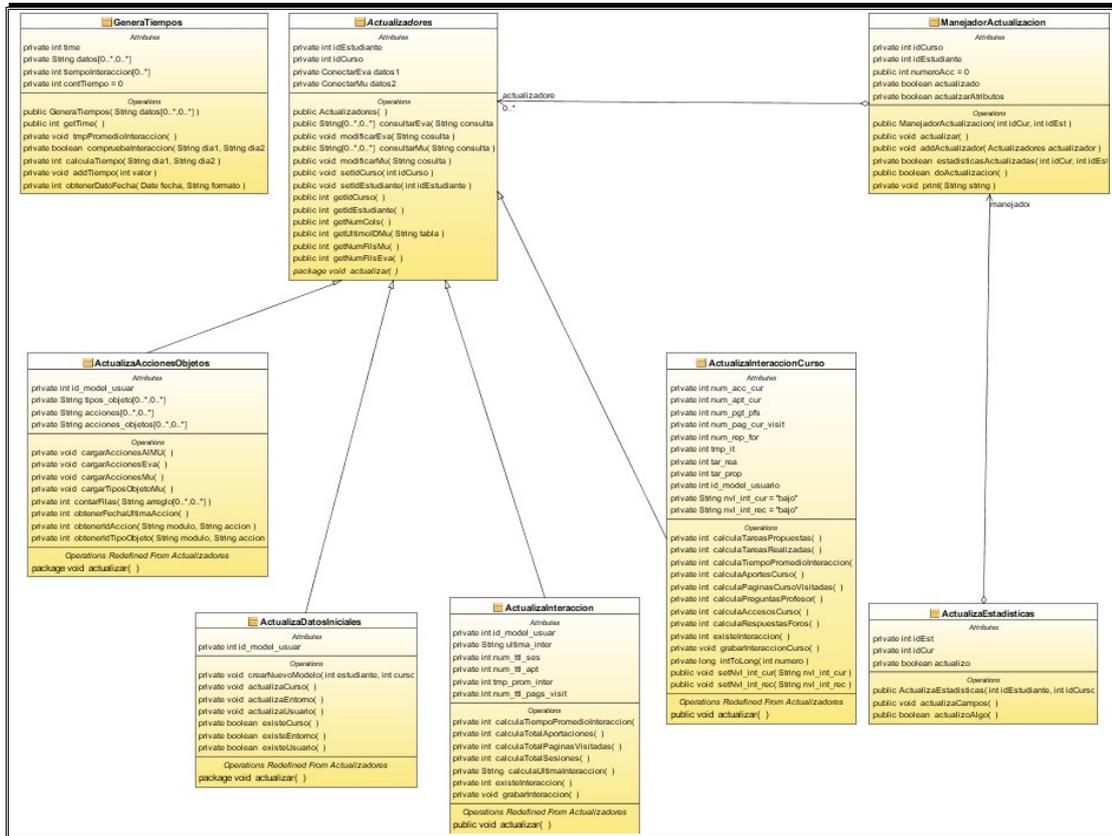


Figura 6.34: Diagrama de clases del módulo de actualización de estadísticas del EVA

Este módulo se encarga de obtener los datos del entorno virtual de aprendizaje transformarlos y convertirlos en información útil para los modelos de usuario de los estudiantes, esta información depende de los atributos que contenga el modelo de usuario, ya que cada atributo tiene enlazado un conjunto de estadísticas que ayudan a la obtención de los atributos, así que ese necesario que para cada atributo se puedan obtener sus estadísticas. El módulo tiene que poder extenderse en el futuro para incorporar las estadísticas de nuevos atributos.

En la figura 6.34 se muestra el diagrama de clases para el módulo de actualización de estadísticas. La implementación de este módulo conlleva el desarrollo de una clase abstracta (*Actualizadores*) en donde se generan las tipologías de clases que obtienen las actualizaciones de datos para un conjunto determinado de los mismos, de donde se

desprenden mediante herencia los tipos concretos de objetos para obtener las actualizaciones de estadísticas mediante las clases (*ActualizaAccionesObjetos*, *ActualizaDatosIniciales*, *ActualizaInteraccion* y *ActualizaInteraccionCurso*) Todo esta coordinado mediante la clase (*ManejadorActualizacion*) que coordina el proceso de actualización de estadísticas. Y la clase (*ActualizaEstadisticas*) es la que maneja las peticiones de actualización de estadísticas para el modelado de usuario. La clase *Tiempos* presenta funciones para manejar fechas y unidades de tiempo entre diferentes formatos.

6.7 Pruebas

La evaluación del correcto funcionamiento de cada uno de los módulos diseñados en el desarrollo del sistema multiagente de modelado de usuario, nos asegura que el sistema que se ha diseñado e implementado satisfaga los requerimientos por los cuales fue desarrollado. En los párrafos siguientes se describen los procedimientos seguidos para la evaluación de las características que presenta del sistema.

6.7.1 Plan de pruebas desarrolladas al agente de modelado de usuario

Para el agente de modelado de usuario se han diseñado un conjunto de pruebas que se cree aseguran que los modelos generados por el sistema sean fiables y representen las verdaderas características de los estudiantes de los cuales se encuentra modelando los atributos el sistema. Las pruebas se basan en las teorías de evaluación de sistemas propuestas por Pressman (2006). A continuación se listan las pruebas que se desarrollaron al sistema:

- ◆ *Pruebas unitarias de cada módulo.*- En las que se refleja el funcionamiento específico de cada unidad en el sistema
 - ◆ *Pruebas de integridad de datos.*- Para asegurar que los datos generados por el modulo son correctos.
 - ◆ *Flujo de información.*- Para ver que el camino que siguen los datos en el modulo es el más apropiado.
- ◆ *Pruebas del sistema.*- En las que se verifica que, en conjunto todo el sistema,

mantiene un funcionamiento razonablemente correcto.

- ◆ *Actualización de estadísticas.*- Para evaluar la capacidad de actualización de estadísticas de los usuarios, por parte del sistema
- ◆ *Rastreo de cambios.*- Evaluación de la capacidad de obtención de cambios en la interacción de los usuarios, por parte del sistema.
- ◆ *Generación de prototipos.*- En donde se evaluó el proceso de creación de modelos de usuario, por parte del sistema.
- ◆ *Actualización de modelos de usuario.*- Se evaluó el la capacidad de actualización de modelos de usuario, por parte del sistema.

6.7.2 Ejecución de las pruebas al agente de modelado de usuario

Las pruebas realizadas al sistema del agente de modelado de usuario, se desarrollaron en dos fases, ya que las pruebas unitarias de los módulos del sistema se desarrollaron terminada la implementación de cada módulo por separado, mientras que las pruebas de sistema, se desarrollaron al haber terminado la implementación del módulo central generado mediante las herramientas que proporciona la metodología seleccionada, y luego de pasar por un proceso de integración de todos los módulos del sistema en un un solo programa. A continuación se describen las actividades desarrolladas en cada caso:

a) Pruebas unitarias a los módulos del agente de modelado de usuario

Todos los módulos del sistema, a excepción del módulo de obtención de atributos por medio de aprendizaje automático, están diseñados para obtener sus resultados recibiendo como entradas, el *id de usuario* y el *id del curso* del estudiante del que se necesita, que sistema desarrolle el trabajo propuesto. Entonces las pruebas unitarias fueron desarrolladas utilizando funciones de ejecución de cada módulo inicializadas con estos datos.

Para evaluar la *integridad de los datos* se ejecutaron los módulos inicializados, y al terminar este proceso, se revisaron los datos arrojados por cada modulo, en el modelo de datos del sistema o en las salidas en pantalla, según corresponda al módulo, y se

verifico que las salidas sean las correctas.

En la evaluación del *flujo de información* se determino mediante código adicional implementado en las funciones de cada módulo que información se está tratando en ese momento, este código adicional no es más que sentencias de impresión en pantalla de la información tratada en ese momento, esta información permite apreciar el flujo de información que maneja el módulo, y de no ser el más óptimo permitirá diseñar estrategias alternativas de implementación de esas funcionalidades evaluadas.

b) Pruebas de sistema del agente de modelado de usuario

La evaluación de la *actualización de estadísticas*, es similar a la que desarrolla el módulo de actualización de estadísticas, la diferencia radica en que lo que se busca evaluar en este caso es la efectividad del sistema al actualizar las estadísticas de varios usuarios que acceden simultáneamente al entorno virtual de aprendizaje.

El *rastreo de cambios* se evaluó para verificar que cada cierto tiempo los datos que permiten inferir los atributos del modelo de usuario son detectados eficientemente por parte del sistema y que este proceso no es influenciado por el número de modelos que se pueden estar rastreándose simultáneamente. Para estas pruebas, por motivos de seleridad en el desarrollo de la misma, se bajo el tiempo predeterminado para detectar los cambios del sistema que normalmete se desarrolla cada semana.

Se evalua la *generación de prototipos*, para conocer si los prototipos que genera el sistema son correctos y corresponden a las características conocidas del usuario, para el desarrollo de esta prueba, se utilizaron varios usuarios con cursos activos los cuales al registrarse en el sistema desencadenan que el sistema inicialmente, desarrolle simultaneamente el prototipado de modelos de usuario para cada curso, con lo cual se puede evaluar la carga que genera el sistema en el procesamiento, y los tiempos que tarda en desarrollar el proceso de prototipado. Esto es importante ya que este proceso es el que demanda un grado mayor de utilización de recursos.

Finalmente la evaluación de la *actualización de modelos de usuario* se desarrolla para conocer la efectividad del sistema en mantener actualizados lo modelos de usuario de los estudiantes, y de evaluar los tiempos en que el sistema actualiza los

modelos de usuario.

6.7.3 Evaluación de las pruebas desarrolladas al agente de modelado de usuario

Se describe dentro de esta sección, las consecuencias que produjo la ejecución de las pruebas diseñadas para el sistema del agente de modelado de usuario y las modificaciones que sufrió el sistema al descubrir inconsistencias en la ejecución de las mismas.

a) Módulo de acceso a datos

Inicialmente este módulo fue desarrollado para acceder de manera simple a los modelos de datos a los que necesita pedir o guardar información el agente de modelado de usuario. Pero al ejecutar las pruebas de integridad de datos se descubrió que, al realizar el sistema modificaciones simultáneas a las bases de datos se producían inconsistencias en los datos que contenían las entidades de modelo, lo que permito redefinir la forma en que se acceza a los modelos de datos, eligiéndose un modelo en el que mediante piscinas de conexiones se acceza correctamente a las bases de datos y se controla la carga que sufren estas en un trabajo intenso del sistema.

b) Módulo de obtención de estilos de aprendizaje

El módulo de obtención de estilos de aprendizaje se probó inicializando en el modelo de datos, el estado que permite que el entorno virtual de aprendizaje decida plantear la contestación del test diseñado para el caso al estudiante, al ser simple la implementación del mismo, la ejecución de las pruebas de integridad de datos y de evaluación de flujos de información fueron satisfactorias, y el módulo al ser parte del EVA, no está influenciado por una carga elevada de estudiantes, tomando el test.

c) Módulo de obtención de atributos por medio de reglas de inferencia

Las pruebas desarrolladas a este módulo se realizaron escalonadamente ya que la implementación del módulo está sujeta, a que atributo se le pide al módulo que obtenga. Entonces, se procedió a probar individualmente por cada atributo pedido y además se planteo el caso en el que se piden varios atributos al modulo.

Luego de ejecutadas las pruebas de integridad de datos de y de flujo de información no se produjeron modificaciones al módulo debido a que las salidas producidas por el modulo eran correctas y la información fue procesada adecuadamente en todos los casos planteados.

d) Módulo de obtención de atributos por medio de aprendizaje automático

Para el módulo de obtención de atributos por medio de aprendizaje automático además de las pruebas planteadas se desarrollaron pruebas de interfaz de usuario ya que, el módulo al tener la capacidad de funcionar autónomamente, era necesario evaluar la facilidad con la que se desarrolla el proceso de clasificación de atributos por medio de la interfaz.

El módulo sufrió modificaciones al realizar las pruebas de flujo de información ya que la definición inicial del mismo fue concebida para procesar atributos definidos en etapa de diseño, pero al compararlos con los que se procesaban en la ejecución del modulo eran incompatibles con los definidos en WEKA, lo que conllevó una redefinición de los atributos de entrada y de la forma en que se introducían dentro de las funciones definidas por WEKA.

En cuanto a la interfaz de usuario no se obtuvieron problemas, ya que las funciones implementadas para la recolección de los parámetros necesarios para la clasificación de atributos de un modelo cargado a modulo, no presentaron fallos en la ejecución de las pruebas.

e) Módulo de obtención de cambios en la interacción

El módulo de obtención de cambios en la interacción fue definido inicialmente para obtener información para los atributos definidos dentro del modelo de usuario, y fue probado para capturar los cambios de un determinado atributo, ejecutando satisfactoriamente su trabajo. Pero dentro de la ejecución de las pruebas se planteó un caso en el que el modulo debía obtener los cambios de varios atributos simultáneamente, el cual produjo la identificación de errores de inconsistencia en los cambios obtenidos por el módulo. Lo que produjo una reestructuración del módulo para que obtenga secuencialmente los cambios para cada atributo, y además, permita

enviar los cambios para varios atributos en un solo objeto, ya que los tareas definidas en el módulo central del sistema precisan que el modulo devuelva los cambios de todos los atributos simultáneamente.

f) Módulo de actualización de estadísticas

Para este módulo la prueba principal ejecutada, es la de consistencia de los datos obtenidos del entorno virtual de aprendizaje, que dependen, del orden en que se obtengan cada uno de los tipos de datos definidos.

Luego de ejecutar las pruebas a este módulo se tuvo que reordenar la secuencia de procesos que se ejecutan ya que la definida inicialmente producía errores de inconsistencia en los datos que se reflejaban en el modelo de datos diseñado para el caso. Además que la implementación desarrollada inicialmente no permitía flexibilidad a la hora de incorporar la obtención de nuevos datos para el modelo.

g) Agente de modelado de usuario

Luego de desarrolladas la pruebas individuales de cada uno de los módulos del sistema, y de la integración de todos ellos con el módulo central del mismo, se procedió con la evaluación de las funcionalidades del sistema.

La metodología de desarrollo de sistemas multiagente INGENIAS en su ciclo de desarrollo, contempla la evaluación de los metamodelos y de la implementación generada por la herramienta IDK (*Ingenias Development Kit*), en cada una de las fases del proceso de desarrollo ya que en todo momento del análisis y diseño del sistema se puede elegir generar código fuente que permite la identificación de posibles errores en el funcionamiento del sistema. Entonces, en cada una de las fases del diseño del sistema en el IDK se procedió con la ejecución de estas pruebas lo que permitió el redefinir adecuadamente el sistema en cada una de las fases de análisis y diseño del mismo.

Ya integrado el sistema con cada uno de los módulos que apoyan el trabajo del mismo se procedió a evaluar las pruebas propuestas, para el sistema completo, en la sección 6.7.1 .

Al evaluar el prototipado de modelos de usuario, se produjeron errores a la hora de generar modelos para un mismo usuario simultáneamente debido a la forma en que se procesaba el acceso a la base de datos de modelos de usuario del sistema por parte del módulo de acceso a datos. Lo que permitió redefinir el funcionamiento del mismo como se había descrito anteriormente. Además permitió evaluar la efectividad del trabajo de los módulos de actualización de estadísticas y de obtención de estilos de aprendizaje, ya que estos procesos pueden ser parte del prototipado de modelos de usuario.

Evalutando la capacidad del sistema de actualizar modelos de usuario, se probó también el funcionamiento de la obtención de cambios en la interacción de sistema. Al realizar esta prueba, como se menciona anteriormente, se redujo el tiempo inicial de actualización de las estadísticas, lo que significaba un grado mayor de trabajo del sistema ya que la frecuencia con que se activan los agentes aumenta, lo que asegura que en parámetros normales de funcionamiento el sistema trabaja adecuadamente. En cuanto a los resultados de la prueba se puede mencionar que se obtuvieron errores de inconsistencia en la actualización de estadísticas ya que la parte del módulo que actualiza las acciones sobre los objetos del sistema redundaba la información dentro del modelo de datos del sistema. Lo que permitió redefinir la forma en que se actualizaba los datos.

También dentro de las pruebas se evaluó los tiempos de ejecución del prototipado de modelos de usuario y la carga en memoria que genera este proceso. Se obtuvo un tiempo promedio de creación de un modelo de usuario de 1.5 segundos, la carga promedio de memoria que genera el prototipado de un modelo de usuario es de 2480 Kb.

Discusión

Dentro de la recolección de datos para el modelo de usuario se enmarcaron dos procesos que son base para las demás fases de todo el trabajo desarrollado, la primera de ellas es la selección de datos de entrada para el modelo de usuario. La segunda es que de culminar con la selección de estos datos es preciso procesarlos mediante trabajo de minería de datos orientado a definir atributos calculables mediante aprendizaje automático. Los resultados obtenidos en estos procesos son descritos a continuación:

Resultados de la selección de datos para el modelo de usuario

Analizando las propuestas de distintos autores sobre, los elementos mínimos que debe contener un modelo de usuario, las tipologías a las que pertenecen esos datos, las formas en que se los obtienen y finalmente la factibilidad de adquirirlos dentro del entorno virtual de aprendizaje, se obtuvo un conjunto de datos que ayudan a definir atributos para el modelo de usuario, encontrar patrones y tendencias de esos datos.

Las dificultades de este proceso se demuestran cuando se intentan estandarizar las entradas que necesita el sistema, ya que los datos en los sistemas se encuentran dispersos y se enmarcan a los esquemas para los que fueron definidos, además obedecen a las especificaciones que imponen en las representaciones en las que encuentran implementados, por ejemplo, los datos contenidos en un esquema relacional se deben tratar de forma diferente que los que se encuentran en archivos de texto plano o binario. Superadas estas dificultades, asoman otras asociadas a las transformaciones que se debe de realizar de los datos para que puedan ser de utilidad dentro del nuevo esquema que se esta definiendo, por ejemplo, un problema común son las representaciones de fechas y horas que son diferentes para cada sistema operativo y para cada sistema de gestión de bases de datos. Lo que conlleva desfases en la sincronización de tiempos y de eventos.

Dentro de esta selección de datos de entrada se contempló la identificación de los estilos de aprendizaje de los estudiantes como un atributo del modelo de usuario. Por su constante inclusión dentro de los trabajos afines a este proyecto. Ya que ha demostrado ser un atributo crucial dentro de la definición de características de aprendizaje de los estudiantes y su implementación dentro de entornos de aprendizaje

colaborativo ha sido exitosa. Para la selección de la tipología más adecuada para la aplicación dentro del entorno virtual de aprendizaje de la UTPL en la sección 3.3.2 del Capítulo 3, se presenta una comparativa en donde se selecciona la mejor técnica para obtención de los estilos. Y además se expone la experimentación hecha con estudiantes seleccionando una clasificación de estilos de aprendizaje.

En esta experimentación las tendencias en cuanto a la pertenencia de los estudiantes a una tipología específica, tomando en cuenta que la clasificación elegida es la propuesta por Felder y Silverman (1988), la que tuvo el mayor resultado fue la tipología visual, que define, que los estudiantes evaluados tienden a asimilar conocimiento mediante medios visuales, esto facilita la aplicación y la identificación de recursos adecuados para estos estudiantes ya que la mayoría de los nuevos medios de generación de conocimiento están enmarcados dentro de los elegibles para esta tipología de estilos de aprendizaje.

Resultados de la ejecución de la minería de datos al entorno virtual de aprendizaje

Luego de la selección de datos para el modelo de usuario, se desarrolló un proceso de minería de datos, para encontrar los elementos necesarios para realizar la inferencia de atributos del modelo de usuario mediante aprendizaje automático.

En primera instancia se desarrolla un estudio de las características de la población estudiantil de la UTPL basados en estudios anteriores realizados por Rubio (2002) en donde obteniendo estadísticas del entorno virtual de aprendizaje se contabilizaron datos sobre dispersión de la población estudiantil, edades, intereses profesionales, etc. Así como una comparación entre las características que presentan los estudiantes de las dos modalidades de aprendizaje predominantes en la UTPL (presencial y a distancia). Este estudio centra su importancia en la premisa que para modelar las características de un estudiante se necesita primero entender el entorno que le rodea así como las principales motivaciones que incentivan al estudiante a incrementar su aprendizaje, lo que desde mi punto de vista es crucial si se pretende que los modelos de usuario construidos representen esa realidad que define al estudiante.

Luego se desarrolla todo el proceso de extracción de conocimiento del entorno

virtual de aprendizaje, se siguieron todos los pasos comunes para este proceso, elementos que ayudaran a la generación de atributos del modelo de usuario, estos elementos se listan a continuación:

- ◆ Un modelo de datos (*mineria_eva*) que guarda los resultados de la aplicación del proceso de extracción de conocimiento.
- ◆ Un programa que realiza la extracción, transformación y carga (ETL) de los datos necesarios para la minería de datos.
- ◆ Un conjunto de tres indicadores a evaluar, que posteriormente se transformaron en atributos del modelo de usuario estos son: el *nivel de interés en el curso*, el *nivel de interés en recursos*, y el *nivel de interacción* en el entorno virtual de aprendizaje.
- ◆ Se produjo un conjunto de instancias etiquetadas de los atributos anteriores que posteriormente se tomaron como los datos de entrenamiento de los atributos del modelo de usuario generados mediante aprendizaje automático.
- ◆ Y finalmente archivos *.model* generados de la aplicación de algoritmos de aprendizaje automático a los datos de entrada definidos para cada atributo, que permiten la inferencia de nuevos valores utilizando las instancias etiquetadas que mencionaron anteriormente.

Además de la experimentación realizada con los diferentes algoritmos de aprendizaje automático en los datos seleccionados, Se evaluaron los resultados obtenidos por los diferentes modelos creados por los algoritmos y se identifico para cada atributo cuál de estos modelos era el más consistente a la hora de encontrar el valor del atributo propuesto. Como resultado de este proceso para los atributos, nivel de interés en el curso, nivel de interés en recursos, y nivel de interacción los modelos seleccionados son: C4.5 (J48), BayesNet (K2-TAN), y Bagging (J48) .

Son grandes los avances que se han logrado en la integración de técnicas de aprendizaje automático con la generación de conocimiento para modelos de usuario y para generar adaptaciones dentro de sistemas de apoyo al aprendizaje. Después de la experimentación desarrollada en este proyecto, es tangible que el resultado es beneficioso ya que de la información que está dormida dentro de los registros que

almacenan las bases de datos relacionales de los sistemas puede ser fuente confiable para generar atributos que definen a un usuario. El problema que se presenta para la utilización de estas técnicas comúnmente, es que el costo necesario para encontrar conocimiento es elevado ya que demanda experimentar con varios algoritmos hasta encontrar el más adecuado, los datos de entrada necesarios para realizar el trabajo deben ser pre procesados y estandarizados para que no presenten problemas en el procesamiento y se necesita una gran cantidad de ejemplos etiquetados para que los resultados de la inferencia de atributos sean confiables.

Los resultados obtenidos con el caso de experimentación desarrollado en este trabajo, son alentadores, ya que comparados con los de trabajos similares presentan un grado de mejoría en los porcentajes de efectividad de los modelos generados, creemos que se debe, a una correcta identificación de datos de entrada y a la utilización de herramientas nuevas y poderosas en el desarrollo de la experimentación.

Se desarrolló el diseño del modelo de usuario de los estudiantes de la UTPL. Luego de definir cada uno de los atributos en esta sección se presentan los resultados de este proceso.

Resultados del diseño del modelo de usuario

Se realizó el diseño del modelo de usuario para los estudiantes de la UTPL definiendo tres categorías de atributos: *datos de entorno*, *datos de usuario*, y *datos de interacción*. En cada una de ellas existen atributos se son inferidos mediante la utilización de diferentes técnicas de obtención de sus valores.

Las técnicas que se implementan para la obtención de los atributos del modelo de usuario pertenecen a cuatro tipologías.

- ◆ Extracción directa del los datos del sistema
- ◆ Reglas de adquisición de atributos,
- ◆ Test electrónicos,
- ◆ y tareas de aprendizaje.

Cada una de las técnicas presenta grados de complejidad diferentes en la ejecución

de las mismas y formas diferentes de obtener el valor, pueden ser tan simples como ejecutar consultas directas al sistema (extracción directa de datos del sistema), o implementar modelos generados a través de algoritmos de aprendizaje automático (tareas de aprendizaje).

Los atributos que se han definido para el modelo de usuario, son los indispensables en la especificación de modelos de usuario. Los atributos definidos llevan una estrecha relación con los datos que se obtienen del entorno virtual de aprendizaje. El modelo definido puede ser ampliado incorporando nuevos atributos en las diferentes tipologías descritas anteriormente tomando en cuenta que datos de entrada son necesarios para inferir ese atributo, el lugar en donde se van a almacenar esos datos, y la forma en que se van a obtener dichos datos. Los atributos pueden pertenecer a otras tipologías y pueden tener formas distintas de obtenerse a las planteadas en este proyecto. Así como pueden calcularse mediante la utilización de datos de entrada existentes en otros sistemas que contengan datos, eso sí referentes de algún modo, a información del mismo usuario del que se está modelando. Más bien los atributos existentes están definidos de acuerdo a la capacidad de obtenerlos del entorno virtual de aprendizaje. Lo ideal es poder definir atributos independientemente de los datos de entrada y ocuparse posteriormente de cómo adquirir esos datos, el problema es que esto conllevará un costo mayor en la consecución del atributo diseñado.

Se realizó la implementación del modelo de usuario de los estudiantes de la UTPL y del sistema del agente de modelado de usuario. Que son los objetivos principales que persigue la realización de este proyecto de tesis. En las secciones siguientes se describen y sustentan los elementos que componen estos entregables.

Resultados de la Implementación del esquema de datos del modelo de usuario

Se definió el modelo de datos para el modelado de usuario de los estudiantes de la UTPL. Las entidades componentes del modelo de datos se pueden observar en la figura 5.1 y en la sección 5.4 se describen cada una de las entidades componentes del modelo.

El modelo de datos propuesto incorpora entidades para guardar los atributos del

modelo de usuario, y permite la incorporación de nuevos atributos en el mismo, también alberga a los datos de interacción que permiten interferir a los atributos del modelo. Y finalmente posee entidad des que apoyan al sistema del agente de modelado de usuario en el desarrollo de su trabajo. Pensando siempre en la extensibilidad de los atributos que podría poseer el modelo estas entidades pueden extenderse o modificarse para cumplir este cometido.

Existen algunas alternativas a la manipulación de modelos de usuario, por ejemplo algunos sistemas, utilizan archivos XML para el almacenaje y trato de los modelos de usuario y utilizan reglas definidas mediante lenguajes especializados para inferir los atributos del modelo, el plantear el trato de modelos de usuario mediante bases de datos relacionales, desde mi punto de vista presenta la mejores ventajas a la hora de implementar el soporte a modelos de usuario, debido a que al ser un modelo relacional exportable mediante SQL a otros motores de base de datos permite que el modelo pueda ser fácilmente implementado en otros sistema y al manejarse en un gestor de bases de datos puede ser accedido por otros sistemas que tengan acceso a su servicio, de realizarse de otra forma no sería factible realizar esto y la extensibilidad del modelo se vería diezmada.

Resultados de la definición de estereotipos

Se utiliza los estereotipos para realizar la inicialización de los modelos de usuario, este proceso necesita que existan atributos ya definidos o condiciones adecuadas para que asignar un tipo de modelo al usuario. Utilizando los estilos de aprendizaje de los estudiantes tomándose como el atributo más factible y confiable que se puede obtener inicialmente de los estudiantes en la tabla 5.18 se presenta la asignación de un estereotipo para cada tipología de estilos de aprendizaje que define la clasificación seleccionada. Y en la tabla 6.1 se describen los valores que tomaran los atributos del prototipo dependiendo de su tipología.

El prototipado permite generar modelos iniciales que luego al cambiar la interacción de los estudiantes podrán mejorarse y definir de mejor forma las características de los estudiantes. Los prototipos diseñados son básicos ya que los demás atributos del modelo de usuario se pueden obtener mediante las técnicas más

sencillas y no necesitan ser prototipados.

***Conclusiones y
Recomendaciones***

En el desarrollo de este proyecto de tesis se ha descrito el modelo de usuario, el proceso de modelado de usuario, sus implicaciones dentro de los sistemas de aprendizaje colaborativo, la descripción de sus atributos mediante varias técnicas de incluido el aprendizaje automático. De los resultados obtenidos se pueden deducir las siguientes conclusiones.

Conclusiones

- ◆ Los datos de entrada que se necesitan para clasificar atributos de los modelos de usuario no pueden ser operados directamente de las fuentes de información ya que en la mayoría de los casos necesitan transformaciones, en mayor o menor grado, dependiendo de la complejidad del atributo del modelo y la de sus valores. Es por ello que este proyecto se desarrolló un proceso de extracción de conocimiento que ayudó a formalizar estos datos de entrada utilizando la técnica de discretación.
- ◆ El estudio a las clasificaciones de estilos de aprendizaje que fueron aplicadas a los estudiantes de la UTPL, arrojaron que la clasificación que mejor se acopla a las características de los estudiantes, es la propuesta por Felder y Silverman.
- ◆ El proceso de extracción de conocimiento aplicado al entorno virtual de aprendizaje de la UTPL, definió tres atributos del modelo de usuario calculables implementando tareas de aprendizaje: el nivel de interés en el curso, el nivel de interés en recursos y el nivel de interacción.
- ◆ Los atributos del modelo de usuario definidos mediante tareas de aprendizaje, según el proceso de extracción de conocimiento aplicado al EVA, se calculan mediante los algoritmos C 4.5 (J48), Bayes Net y Bagging (J48), respectivamente.
- ◆ El algoritmo C 4.5 (j48), en la matriz de confusión resultante de su ejecución presenta 119 fallos a la hora de clasificar el atributo nivel de interés en el curso, comparado con los 120 fallos que presenta la combinación de clasificadores C 4.5 (j48), mediante el método Bagging.
- ◆ El algoritmo Bayes Net, la combinación de clasificadores C 4.5 (j48) y la combinación de clasificadores RepTree, mediante el método Bagging, en la

matriz de confusión resultante de su ejecución presentan un fallo a la hora de clasificar el atributo nivel de interés en recursos.

- ◆ La combinación de clasificadores J48, mediante el método Bagging, en la matriz de confusión resultante de su ejecución presenta un 38 fallos a la hora de clasificar el atributo nivel de interacción, comparado con los 42 fallos que presenta la combinación de clasificadores RepTree, mediante el método Bagging.
- ◆ Los resultados obtenidos en este proyecto demuestran que los modelos de usuario de los estudiantes de la UTPL se pueden crear inicialmente, mediante la utilización de los estilos de aprendizaje, evaluados mediante el test de Felder y Silverman. Estos modelos luego son actualizados ejecutando tareas de aprendizaje, que mediante el procesamiento de los datos de la interacción del estudiante infieren los nuevos valores de los atributos del modelo de usuario.
- ◆ Para la obtención de atributos del modelo de usuario mediante aprendizaje automático, en el proceso de extracción de conocimiento, se determino que se necesita obtener: modelos para inferenciar el atributo, un conjunto de datos de entrenamiento, y la instancia a clasificar.
- ◆ El modelo de usuario de los estudiantes de la UTPL contiene tres tipologías de atributos que se encuentran agrupados como, datos de entorno, datos de usuario y datos de interacción. Estos se pueden obtener mediante test electrónicos (como en el caso de los estilos de aprendizaje), extracción directa del sistema (como en el caso de los nombres y apellidos del estudiante), reglas de adquisición de atributos (como el caso de la zona geográfica del estudiante) y mediante la utilización de tareas de aprendizaje (como en el caso del nivel de interacción del estudiante).
- ◆ Los modelos de usuario de los estudiantes de la UTPL se almacenan en una base de datos relacional que permite albergar, además de los modelos de usuario, las actualizaciones en los cambios en la interacción del estudiante con la plataforma, así como los datos que apoyan el funcionamiento del agente de modelado de usuario..
- ◆ La metodología de desarrollo de sistemas multiagente INGENIAS fue

seleccionada de un grupo de siete metodologías estudiadas para el desarrollo del agente de modelado de usuario de los estudiantes de la UTPL, por demostrar un proceso de desarrollo robusto, que soporta el trabajo y la implementación de agentes inteligentes en todas las fases del desarrollo.

- ◆ Se diseñó e implementó el agente de modelado de usuario de los estudiantes de la UTPL, para que demuestre funcionalidades que permitan la construcción de modelos de usuario iniciales, así como la actualización de estos modelos. Mediante una arquitectura modular, que permite además mantener un grado de flexibilidad para la incorporación de nuevos atributos, y de las funciones de obtención que los soportan.

Recomendaciones

En relación a las conclusiones a las que se ha llegado con la realización de este proyecto de tesis se plantean las siguientes recomendaciones:

- ◆ Considerar el incremento de nuevas funcionalidades en el entorno virtual de aprendizaje ya que esto, conlleva la incorporación de nuevos datos de interacción que pueden utilizarse como datos de entrada, que permitan la generación de nuevos atributos para el modelo de usuario.
- ◆ La implementación e incorporación de nuevas tareas de aprendizaje a las existentes, que mediante aprendizaje automático modelen atributos relacionados, con el nivel rendimiento académico, y la identificación de las acciones del usuario sobre la interfaz del sistema.
- ◆ La implementación de un módulo de administración del agente de modelado de usuario que permita incrementar las opciones de configuración, y una mejor gestión de las funcionalidades del sistema.
- ◆ Para las tareas de aprendizaje existentes y las nuevas, es recomendable la experimentación con diferentes algoritmos de aprendizaje automático ya que dependiendo de los datos de entrada los resultados con un mismo algoritmo pueden variar drásticamente entre atributos.
- ◆ Considerar, para futuras extensiones del agente de modelado de usuario, las nuevas versiones de la metodología INGENIAS, que soporten el trabajo con

más de un desarrollador, ya que si la magnitud del proyecto creciera sería difícil el soporte al proceso de desarrollo con la versión actual de la metodología.

ANEXOS

Anexo A

Test de estilos de aprendizaje

A.1 Tests de estilos de aprendizaje

En este anexo se muestra el cuestionario de los test de estilos de aprendizaje ejecutado a un grupo de estudiantes de la UTPL, para con los resultados obtenidos evaluar cual clasificación y test es el más apropiado para su ejecución dentro del entorno virtual de aprendizaje de la UTPL (EVA).

A.2 Test de Felder y Silverman

1. Entiendo algo de una mejor manera luego que yo:
 - a. Lo pruebo
 - b. Lo pienso
2. Me gustaría ser considerado:
 - a. Realista
 - b. Innovativo
3. Cuando pienso en lo que hice ayer, estoy impulsado a hacer:
 - a. Un dibujo
 - b. Un escrito
4. Yo tiendo a:
 - a. Entender los detalles de un tema pero puede estar confundido en la estructura completa de la misma.
 - b. Entender los la estructura completa de un tema pero puedo estar confundido en los detalles.
5. Cuando estoy aprendiendo algo nuevo, ello me ayuda a:
 - a. Hablar acerca de ello
 - b. Pensar acerca de ello
6. Si yo fuera un profesor, me gustaría enseñar en un curso de:
 - a. Que trata sobre hechos y situaciones de la vida real
 - b. Que trata con ideas y teorías
7. Prefiero obtener información nueva en:
 - a. Dibujos, diagramas, gráficos, o mapas
 - b. Orientaciones escritas o información verbal
8. Cuando entiendo:
 - a. Entiendo cómo se relacionan todas las partes, para entender el todo
 - b. Entiendo el todo, para luego entender cómo se relacionan sus partes
9. En un grupo de estudio en que trabaja sobre un material difícil, me gusta más:
 - a. Sobresalir y contribuir con ideas
 - b. Esperar y escuchar
10. Para mi es más fácil:
 - a. Aprender hechos
 - b. Aprender conceptos
11. En un libro con muchos dibujos y gráficas, me gusta:
 - a. Observar los dibujos y gráficas cuidadosamente
 - b. Enfocarme en el texto escrito

12. Cuando soluciono problemas de matemáticas:
 - a. Normalmente trabajo con mi manera de obtener las soluciones paso a paso
 - b. A menudo observo las soluciones pero tengo que detenerme a pensar en los pasos para obtenerla
13. En las clases que he tomado:
 - a. Normalmente he tenido que conocer a muchos de los estudiantes
 - b. Rara vez he tenido que conocer a muchos de los estudiantes
14. En los textos que no son de ficción, prefiero:
 - a. Algo que me enseña nuevos hechos o me dice como hacer algo
 - b. Algo que me da nuevas ideas para pensar sobre ello
15. Me gustan los profesores:
 - a. Que hacen varios diagramas en el tablero
 - b. Que gastan mucho tiempo explicando
16. Cuando estoy analizando una historia o novela:
 - a. Pienso sobre los incidentes y trato de colocarlos juntos para adivinar el desenlace
 - b. Conozco el desenlace solo cuando finalizo la lectura y luego tengo que regresar a encontrar los incidentes que lo demuestran
17. Cuando me enfrento a un problema o tarea, me gusta más:
 - a. Comenzar a trabajar en la solución inmediatamente
 - b. Tratar de comprender el problema primero
18. Prefiero la idea de:
 - a. Certeza
 - b. Teoría
19. Recuerdo mejor:
 - a. Lo que veo
 - b. Lo que escucho
20. Para mi es más importante que un instructor:
 - a. Marque el material de clase en pasos secuenciales bien claros
 - b. Me dé una visión global y relacione el material de clase con otras área
21. Prefiero estudiar:
 - a. En un grupo de estudio
 - b. Solo
22. Me gusta que me consideren como:
 - a. Cuidadoso sobre los detalles de mi trabajo
 - b. Creativo sobre cómo hago mi trabajo
23. Cuando tengo la dirección de un nuevo lugar, yo prefiero:
 - a. Un mapa
 - b. Instrucciones escritas
24. Yo aprendo:
 - a. A un paso bastante regular. Si estudio duro, “lo conseguiré”
 - b. De manera intermitente. Estaré totalmente confuso y luego repentinamente todo “se relaciona”
25. Me gusta primero:
 - a. Hacer las cosas
 - b. Pensar acerca de cómo voy a hacerlas

26. Cuando leo por entretenimiento, me gusta que los escritores:
 - a. Digan claramente lo que quieren expresar
 - b. Digan cosas de manera creativa, interesante
27. Cuando veo un diagrama o boceto en clase, me gusta más recordar:
 - a. El dibujo
 - b. Lo que el instructor dijo acerca de el
28. Cuando considero un paquete de información, me gusta más:
 - a. Enfocarme en los detalles y olvidar la visión general
 - b. Tratar de entender la visión general antes que entender los detalles
29. recuerdo más fácilmente:
 - a. Algo que he hecho
 - b. Algo que he pensado mucho
30. Cuando tengo que realizar un trabajo, yo prefiero:
 - a. Especializar una manera de hacerlo
 - b. Empezar con nuevas formas de hacerlo
31. Cuando alguien me muestra datos, yo prefiero:
 - a. Diagramas o gráficos
 - b. Texto resumiendo los resultados
32. Cuando escribo un artículo, me gusta más:
 - a. Trabajar en (pensar sobre escribir) el principio del artículo e ir progresando hacia adelante.
 - b. Trabajar en (pensar sobre escribir) diferentes partes del artículo y luego ordenarlas
33. Cuando tengo que trabajar en un proyecto en grupo, primero quiero:
 - a. Tener “Lluvia de ideas del grupo” donde cada uno contribuye con ideas
 - b. Lanzar ideas individualmente luego reunirme en grupo para comparar las ideas
34. Considero que es un gran elogio llamar a alguien:
 - a. Sensible
 - b. Imaginativo
35. Cuando conozco gente en una fiesta, me gusta recordar más:
 - a. Como lucían
 - b. Que dijeron sobre ellos mismos
36. Cuando estoy aprendiendo una nueva área, yo prefiero:
 - a. Mantenerme enfocado en el área, aprendiendo tanto como pueda de ella.
 - b. Tratar de hacer conexiones entre el área y otras áreas relacionadas
37. Me gusta más que me consideren como:
 - a. Expresivo
 - b. Reservado
38. Prefiero los cursos que enfatizan:
 - a. El material concreto (hechos, datos)
 - b. El material abstracto (conceptos, teorías)
39. Como entretenimiento, a mí me gusta:
 - a. Ver televisión
 - b. Leer un libro
40. Algunos profesores comienzan sus lecciones con un vistazo de lo que cubrirán. Tales intervenciones son:

- a. De algún modo poderosas para mi
 - b. Muy provechosas para mi
41. La idea de hacer trabajos en grupo, con una calificación para todo el grupo:
- a. Me atrae
 - b. No me atrae
42. Cuando estoy haciendo cálculos extensos:
- a. Tiendo a repetir todos mis pasos y verificar mi trabajo cuidadosamente
 - b. Encuentro la verificación de mi trabajo muy tediosa y tengo que esforzarme para hacerlo
43. Tiendo a recordar los lugares cuando he estado
- a. Fácilmente y exactamente ubicado
 - b. Con dificultad y sin muchos detalles
44. Cuando soluciono problemas, me gusta más:
- a. Pensar en los pasos del proceso de solución
 - b. Pensar en las posibles consecuencias o aplicaciones de la solución en un amplio rango de áreas

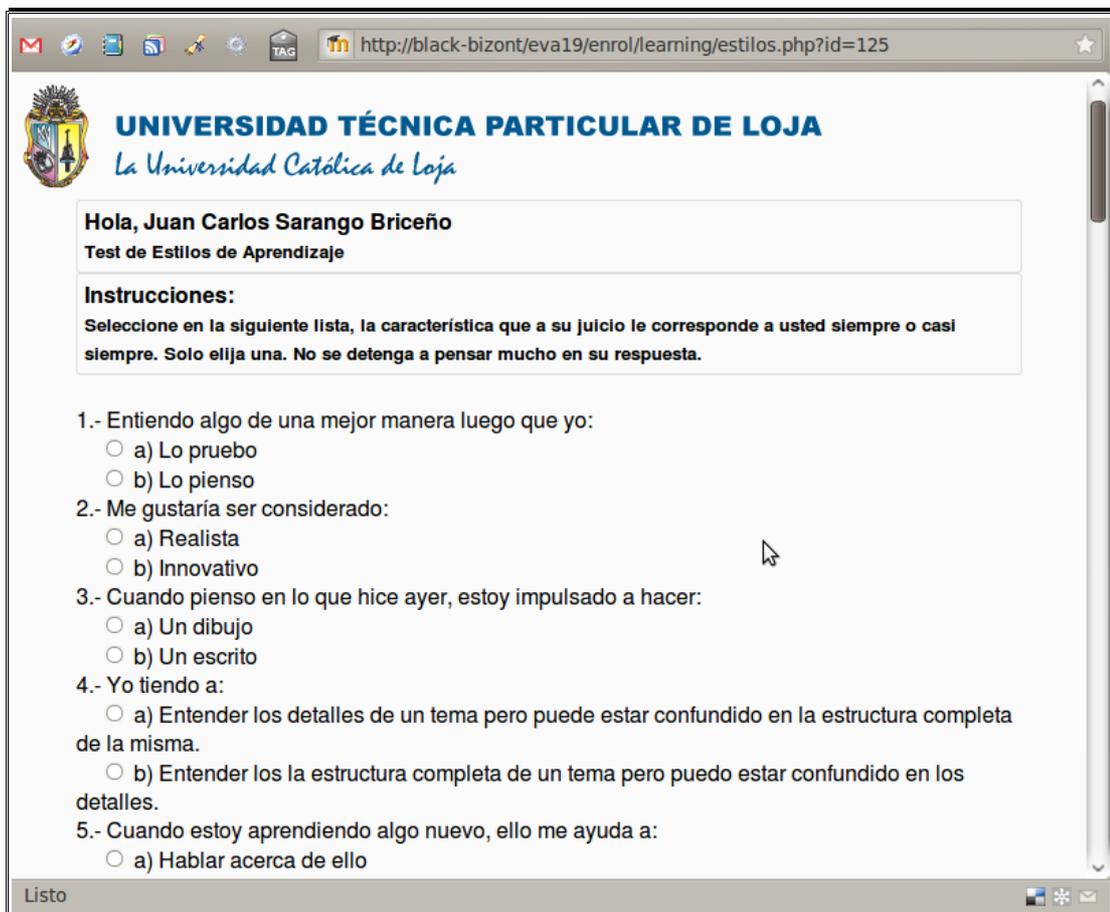


Figura A.1: Test de estilos de aprendizaje implementado en la plataforma

En la figura A.1 se muestra una captura del test de estilos de aprendizaje de Felder y Silverman implementado como módulo del entorno virtual de aprendizaje de la UTPL.

Anexo B

*Elementos de navegación y tracking
obtenibles y no obtenibles del EVA
UTPL*

B.1 Trazas de navegación y elementos de tracking obtenibles del EVA UTPL

Dentro de la investigación desarrollada en este proyecto de tesis se utilizan algunos datos de entrada que ayudan a entender algunas acciones que los usuarios han desarrollado dentro del entorno virtual de aprendizaje de la UTPL, estos datos nacen del análisis de los logs de sucesos ocurridos en el sistema y son conocidos como trazas de navegación. Según Kobsa, Koenemann, y Pohl (1999) estas trazas de navegación, dependiendo de que interpretación reciban, pueden ser utilizadas para medir diferentes aspectos de la interacción del usuario en el sistema, como los niveles de interés en los objetos propuestos, el desconocimiento de los estudiantes de un termino y las preferencias de los mismos por los objetos mostrados dentro del los links ofrecidos en las trazas de navegación.

En el anexo presente se describen las trazas de navegación que se pueden obtener del entorno virtual de aprendizaje, y que se utilizaron dentro de la investigación para desarrollar la experimentación dentro de la fase de minería de datos. El entorno virtual de aprendizaje, almacena el log de navegación de los usuarios en una tabla de su base de datos denominada *mdl_log*, esta tabla almacena registros de cada una de las acciones que los usuarios del EVA han desarrollado en el sistema.

#	Field	Type	Null	Key	Default
1	id	bigint(10) unsigned	NO	PRI	<NULL>
2	time	bigint(10) unsigned	NO	MUL	0
3	userid	bigint(10) unsigned	NO	MUL	0
4	ip	varchar(15)	NO		
5	course	bigint(10) unsigned	NO	MUL	0
6	module	varchar(20)	NO		
7	cmid	bigint(10) unsigned	NO	MUL	0
8	action	varchar(40)	NO	MUL	
9	url	varchar(100)	NO		
10	info	varchar(255)	NO		

Figura B.1: Campos que componen la entidad *mdl_log*

En la figura B.1 se muestran los campos que componen a la entidad *mdl_log*, del análisis de los campos y de la información que contiene esta entidad, permitió estructurar secuencias de acciones que el estudiante ha desarrollado dentro del sistema, mientras que la agrupación y sumariación de los registros permite

contabilizar los elementos de interacción que derivan los atributos del modelo de usuario.

Para la obtención de estos datos se aplicaron dos técnicas, según la naturaleza de los mismos, la primera mediante consultas de lenguaje SQL que permiten la agrupación y la sumarización de valores relacionados a la interacción, y la segunda que permite el calculo de trazas de navegación, tomando el cuenta los tiempos en que se realizan las acciones. A continuación se describen estas secuencias y agrupaciones así como la forma en que fueron obtenidas.

B.1.1 Número de accesos al curso

Este parámetro identifica el número de veces que el usuario a accedido al curso en que se encuentra matriculado, se consideran como accesos al curso a todas las veces que e estudiante a mirado una de las paginas referentes al curso, esto incluye las veces que ha mirado dentro de los recursos pertenecientes a él. La forma en que se puede obtener este valor es mediante la aplicación de una consulta SQL, esta consulta se muestra en la figura B.2, en donde el número “21152” representa el id de usuario en el sistema, y el número “10121” representa el id del curso del que se esta obteniendo la interacción.

```
1 select count(*)
2 from mdl_log
3 where action='view' and
4 (module='course' || module='resource' || module='blog' || module='quiz' ||
5 module='assignment' || module='wiki' || module='glossary' || module='notes' ||
6 module='choice' || module='chat' || module='scorm')
7 and userid=21152 and course=10121
```

Figura B.2: Consulta para la obtención del número de accesos al curso de un estudiante

B.1.2 Número de aportes al curso

Es aquel parámetro que contabiliza el número de veces que un estudiante ha realizado una acción que agrega valor a su aprendizaje, para ello se consideran un conjunto de acciones que al ser completadas por el estudiante han dejado un un conocimiento o alguna destreza nueva. La forma en que se puede obtener este valor es mediante la aplicación de una consulta SQL, esta consulta se muestra en la figura B.3, en donde el número “125” representa el id de usuario en el sistema, y el número

“11893” representa el id del curso del que se esta obteniendo la interacción.

```

1  select count(*)
2  from mdl_log
3  where ((module='forum' and action='add') ||
4         (module='forum' and action='add post') ||
5         (module='resource' and action='add') ||
6         (module='resource' and action='add_planutpl') ||
7         (module='quiz' and action='add') || (module='forum' and action='add') ||
8         (module='assignment' and action='add') ||
9         (module='assignment' and action='upload') ||
10         (module='blog' and action='add') || (module='blog' and action='comment') ||
11         (module='choice' and action='add') || (module='glossary' and action='add') ||
12         (module='glossary' and action='add comment') ||
13         (module='glossary' and action='add entry') ||
14         (module='wiki' and action='add') || (module='wiki' and action='attachments') ||
15         (module='notes' and action='add') || (module='survey' and action='add') ||
16         (module='survey' and action='submit') || (module='chat' and action='add') ||
17         (module='chat' and action='talk') || (module='scorm' and action='add') ||
18         (module='upload' and action='upload') ||
19         (module='message' and action='write'))
20  and userid = 125 and course = 11893

```

Figura B.3: Consulta para la obtención del número de aportes al curso de un estudiante

B.1.3 Número de preguntas al profesor

Mediante este parámetro se acumula el número de veces que el estudiante se ha contactado con el tutor que le esta impartiendo el curso del que esta participando para pedir información acerca del curso y de los conocimientos que se esta compartiendo en la asignatura en que se encuentra participando. La forma en que se puede obtener este valor es mediante la aplicación de una consulta SQL, esta consulta se muestra en la figura B.4, en donde el número “466” representa el id de usuario en el sistema, y el número “3542” representa el id del curso del que se esta obteniendo la interacción. Cabe mencionar que esta consulta contiene un sub-consulta, mediante la cual se obtiene el id del profesor que da la clase en el curso del estudiante.

```

1  select count(*)
2  from mdl_message
3  where useridfrom=466 and useridto =
4         (
5           select userid
6           from mdl_enrol_utpl
7           where courseid=3542 and rol = 3
8         )
9

```

Figura B.4: Consulta para la obtención del número de preguntas al profesor en un curso de un estudiante

B.1.4 Número de paginas por curso visitadas

Este parámetro contabiliza el número de visitas de un estudiante a las paginas

pertenecientes al curso al que esta participando, entendiéndose como visitas las veces que el estudiante selecciono acceder a un contenido perteneciente a un mismo curso activo para el estudiante en cuestión. La forma en que se puede obtener este valor es mediante la aplicación de una consulta SQL, esta consulta se muestra en la figura B.5, en donde el número “125” representa el id de usuario en el sistema, y el número “9042” representa el id del curso del que se esta obteniendo la interacción.

```
1 select count(*)
2 from mdl_log
3 where userid=125 and course=9042 and action like '%view%'
```

Figura B.5: Consulta para la obtención del número de paginas por curso visitadas por el estudiante

B.1.5 Número de respuestas en los foros

Una cantidad que representa en número de veces que un estudiante a contestado o ha colgado un nuevo hilo en un foro perteneciente al curso en el cual esta participando. La forma en que se puede obtener este valor es mediante la aplicación de una consulta SQL, esta consulta se muestra en la figura B.6, en donde el número “125” representa el id de usuario en el sistema, y el número “9042” representa el id del curso del que se esta obteniendo la interacción.

```
1 select count(*)
2 from mdl_log
3 where userid=125 and course=9042 and module='forum'
4 and (action='add discussion' || action='add post')
```

Figura B.6: Consulta para la obtención del número de respuestas en los foros de un estudiante en un curso

B.1.6 Tiempo promedio de interacción

El tiempo promedio de interacción es un parámetro que se obtiene al promediar un conjunto de interacciones del estudiante con el sistema, estas interacciones se calculan fijando un punto de partida, en el cual se selecciona un registro inicial y un segundo, y al comparar los dos, si se encuentra que los dos sucesos se produjeron en la misma hora se dice entonces que pertenecen a la misma interacción, luego de esto el tiempo

en segundos entre las dos interacciones se acumula en una variable, paso siguiente se compara el segundo registro con el siguiente para identificar si el ultimo pertenece a la misma interacción o pertenece a una nueva y se procede de la misma forma como con las dos primeras.

B.1.7 Número de tareas realizadas

Parámetro que enmarca el número de tareas, posts y archivos subidos por un estudiante en un curso en el que éste se encuentra participando. La forma en que se puede obtener este valor es mediante la aplicación de una consulta SQL, esta consulta se muestra en la figura B.7, en donde el número “125” representa el id de usuario en el sistema, y el número “9042” representa el id del curso del que se esta obteniendo la interacción. Cabe mencionar que se incluye una sub-consulta mediante la cual se obtiene primero las asignaciones, es decir, las tareas que tiene propuesto un estudiante.

```
1 select count(*)
2 from mdl_log a,
3 (select distinct userid from mdl_role_assignments where roleid = 5) b
4 where (action like "%upload%" || action like "%add%")
5 and a.userid = b.userid and a.userid=125 and a.course=9042
```

Figura B.7: Consulta para la obtención del número de tareas realizadas por un estudiante en un curso

B.1.8 Número de tareas propuestas

Representa el número de asignaciones que un tutor a introducido como tarea para un grupo de estudiantes. La forma en que se puede obtener este valor es mediante la aplicación de una consulta SQL, esta consulta se muestra en la figura B.8, en donde el número “14455” representa el id del curso del que se esta obteniendo la interacción.

```
1 select *
2 from mdl_event
3 where courseid = 14455
```

Figura B.8: Consulta para la obtención del número de tareas propuestas por un tutor en un curso

B.2 Información de tracking que no se puede obtener del EVA UTPL

Para construir modelos de usuario es necesaria la correcta selección de atributos que van a formar parte del mismo, esta selección esta ligada directamente a la calidad de los elementos de entrada que se seleccionen para producir estos atributos, por otro lado las técnicas para obtener los elementos de entrada, dependen mucho de su facilidad de implementación dentro del sistema del que se encuentra modelando la interacción de los usuarios, pero es en esta parte donde asoman las dificultades ya que los sistemas no están pensados para recoger secuencias de interacción de los usuarios, entonces se hace necesaria la incorporación de componentes de código adicionales que ayuden con esta tarea. En esta sección se describen un conjunto de parámetros que tienen un alto grado de importancia para la generación de atributos de un modelo de usuario, pero que por su complejidad de implementación no se han podido incorporar a la representación actual.

B.2.1 Patrones de navegación

Los patrones de navegación representan la secuencia típica de revisión de los elementos disponibles en la representación de un sistema que es visible para el usuario, estos patrones se obtienen con el tiempo ya que para su construcción se necesita ir obteniendo varias secuencias de navegación e ir las comparando para observar a que contenidos comúnmente el usuario accede, e ir armando el patrón, la complejidad de implementación dentro del EVA UTPL es alta debido a que no se tiene control de los elementos presentados a los usuarios, lo cual conllevaría una estandarización de las vistas en pantalla de los contenidos a los que accede un usuario concreto.

B.2.2 Preferencias de selección

Para encontrar las preferencias en los contenidos a los cuales accede un usuario dentro del EVA UTPL se necesita que se tenga control de los links de contenido o de la información y recursos que se presentan a los usuarios en la paginas del sistema, ya que para dar ponderación a un elemento que el usuario selecciono se necesita saber que elementos tenia disponibles en la interfaz. El saber las preferencias en la selección

de contenidos del usuario es crucial para la recomendación de contenidos similar o que pueda ser de ayuda significativa en un tema que el usuario considera importante.

B.2.3 Monitorear el scroling y el enlarging

Las acciones de scroling (acción de desplazarse en un documento usando la rueda central de un dispositivo apuntador) y enlargin (acción de redimensionar una ventana mediante el uso de un dispositivo apuntador), se miden para detectar algunos comportamientos de los usuarios ya que cuando se navega en la web mediante el dispositivo apuntador se hacen vistas rápidas de los contenidos, pero en los que llaman la atención de los usuarios estos se detienen a observarlos, leerlos, o maximizarlos, para apreciarlos de mejor manera. Entonces nace la necesidad de incorporar métodos que ayuden a la captura de estos comportamientos dentro del sistema ya que estos podrían ser utilizados para encontrar preferencias de los usuarios de los contenidos pertenecientes a un curso

B.2.4 Monitorear selección de elementos de ayuda

Los elementos de ayuda en un sistema sirven para introducir en el ambiente en el que trabaja el sistema al que accede un usuario, para definir los elementos que componen al sistema o a su interfaz, y para guiar al usuario en las funcionalidades que ofrece el sistema para su trabajo. El monitoreo de la selección de estos elementos es utilitario para reconocer patrones de desconocimiento de los usuarios en ciertos temas de su interés, referentes al desenvolvimiento de la asignatura en la que se encuentra participando, o al funcionamiento del sistema con el que trabaja. La dificultad para la implementación de este tipo de monitoreo es, la poca información de ayuda existente, y que este tipo de información no es diferenciada del contenido perteneciente a la asignatura.

B.2.5 Etiquetado y rating de objetos de aprendizaje

Dentro de un entorno educativo el ordenamiento de la información es crucial para enmarcar cada contenido o recurso educativo en el contexto al que pertenece, es por esto que se hace necesaria la incorporación métodos de etiquetado de contenidos. Y un componente importante que ha tenido algunos casos de acierto es de incorporar

ratings en los contenidos que se imparten en las asignaturas mediante esto los estudiantes podrían calificar que relevancia a tenido un contenido en su aprendizaje, se gana con estos métodos un ordenamiento de la información contenida en el sistema, que facilita las búsquedas de contenido, así como identificar los contenidos que son mas relevantes en las asignaturas ya que se podrían comparar con ratings de cursos anteriores similares y obtener los contenidos más relevantes en cada asignatura, etc. Pero en este momento se dificulta la incorporación de este tipo de métodos de ordenamiento ya que conllevan un gran esfuerzo de ingeniería para incorporarlos en el EVA UTPL

B.2.6 Monitorear acciones con objetos de aprendizaje

Otra forma de obtener información acerca de la preferencia de los usuarios a los objetos de aprendizaje disponibles en el curso en el que se encuentra participando un estudiante, se basa en el monitoreo de las acciones que el usuario realiza con estos objetos, la pre-visualización, la descarga, la reproducción, el etiquetar, el clasificar etc., se utilizan como indicadores de la relevancia que han tenido para el usuario estos objetos de aprendizaje, la dificultad radica en que para la obtención de esta información se necesita de técnicas complejas, ya que al integrar diferentes módulos del EVA UTPL, que han sido diseñados individualmente, para que representen de una forma común este tipo de información, conlleva la re-ingeniería de estos módulos para incorporar los elementos faltantes que faciliten la integración.

Anexo C

Base de datos de modelos de usuario

C.1 Inicialización de la base de datos de modelos de usuario

Dentro del trabajo del sistema multiagente de modelado de usuario se necesita que algunas de las entidades que conforman el modelo de datos contengan datos iniciales que son utilizados para definir tipologías, y estados. Para estas entidades, se han definido un conjunto de datos que ayudan al sistema a cumplir sus objetivos, los datos que se han definido obedecen en algunos casos a parámetros ya definidos en capítulos anteriores. A continuación se describen las definiciones de estos datos y las entidades que los soportan:

- ♦ *mu_valores_estereotipos* .- Los valores que contempla esta entidad se asignan según el estereotipo al que pertenecen los datos. Fueron definidos en la tabla 5.10 del Capítulo 5. Los datos que intervienen en la entidad se muestran en la figura C.1 de a continuación:

VES_ID_VALORES_ESTEREOTIPOS	TMU_ID_TIPO_MODELO_USUARIO	VES_ATRIBUTO_MU	VES_VALOR_ESTEREOTIPO
1	1	Nivel de Interes en el Curso	bajo
2	1	Nivel de Interes en Recursos	bajo
3	1	Nivel de Interaccion	principiante
4	2	Nivel de Interes en el Curso	medio
5	2	Nivel de Interes en Recursos	medio
6	2	Nivel de Interaccion	medio
7	3	Nivel de Interes en el Curso	alto
8	3	Nivel de Interes en Recursos	alto
9	3	Nivel de Interaccion	avanzado

Figura C.1: Datos iniciales de la entidad valores de estereotipos

- ♦ *mu_asignaciones_estereotipos* .- Los valores iniciales que almacena esta entidad son asignados según el estilo de aprendizaje que posee el estudiante a prototipar. Fueron definidos en la tabla 5.9 del Capítulo 5. Los datos iniciales definidos para esta entidad se muestran en la figura C.2 de a continuación:

AES_ID_ASIGNACIONES_ESTEREOTIPOS	TMU_ID_TIPO_MODELO_USUARIO	AES_ESTILO_APRENDIZAJE
9	1	sensitivo
2	1	verbal
3	2	activo
4	2	global
5	3	reflexivo
6	3	intuitivo
7	3	visual
8	3	secuencial

Figura C.2: Datos iniciales de la entidad asignaciones de estereotipos

- ♦ *mu_tipos_modelos_usuario* .- Los valores de esta entidad fueron definidos en la tabla 5.6 del Capítulo 5. y corresponden a los niveles que puede alcanzar un estudiante dependiendo de su estilo de aprendizaje o de nivel de interacción con el entorno virtual de aprendizaje. Los valores que toma la entidad se muestran en la figura C.3:

TMU_ID_TIPO_MODELO_USUARIO	TMU_NOMBRE_TIPO_MODELO	TMU_DESCRIPCION_TIPO_MODELO
1	principiante	modelo inicial
2	medio	modelo intermedio
3	avanzado	modelo superior

Figura C.3: Datos iniciales de la entidad tipos de modelos de usuario

- ♦ *mu_acciones* .- Los valores de contenidos en esta entidad son asignados según el modulo del EVA al que pertenecen y se muestran en la figura C.4:

ACC_ID_ACCION	ACC_NOMBRE_ACCION	ACC_DESCRIPCION_ACCION	ACC_MODULO_EVA	TOB_ID_TIPO_OBJETO
1	upload		assignment	1
2	view		assignment	1
3	view all		assignment	1
4	comment		blog	1
5	view		blog	1
6	update		blog	1
7	add		chat	1
8	talk		chat	1
9	update		chat	1
	view		chat	1
	view all		chat	1

Figura C.4: Datos iniciales de la entidad acciones

- ♦ *mu_tipos_objeto* .- Los tipos de objeto son determinados según la característica

principal que presenta el recurso al cual están enlazados. Los valores iniciales que contiene esta entidad se muestran en la figura C.5.

TOB_ID_TIPO_OBJETO	TOB_NOMBRE_TIPO_OBJETO	TOB_DESCRIPCION_TIPO_OBJETO
1	Texto	
2	Recurso	

Figura C.5: Datos iniciales de la entidad tipos de objeto

- ◆ *mu_atributos_modelo_usuario* .- Los valores iniciales de esta entidad son los atributos definidos para el modelo de usuario. Y de los cuales hay que encontrar sus valores mediante el proceso de modelado de usuario. Los valores iniciales definidos para esta entidad se muestra en la figura C.6.

AMU_ID_ATRIBUTO_MODELO_USUARIO	AMU_NOMBRE_ATRIBUTO	AMU_DESCRIPCION_ATRIBUTO	AMU_TIPO_ATRIBUTO	AMU_TECNICA_OBTENCION
1	Estilo de Aprendizaje	Metodo de aprendizaje por medi...	Datos de Usuario	Test
2	Zona Geografica	Atributo que describe la localiza...	Datos de Entorno	Inferencia
3	Entorno de Usuario	Atributo que describe el medio ...	Datos de Entorno	Inferencia
4	Nivel de Interes en el Curso	Atributo que define el grado int...	Datos de Interaccion	Clasificacion
5	Nivel de Interes en Recursos	Atributo que describe el nivel en...	Datos de Interaccion	Clasificacion
6	Nivel de Interaccion	Este atributo define los niveles ...	Datos de Interaccion	Clasificacion

Figura C.6: Datos iniciales de la entidad atributos del modelo de usuario.

Anexo D

*Documentación de las fases de
desarrollo*

D.1 Resultados de la fase de Análisis-Inicio

De aplicar los pasos y actividades descritas en la sección 6.4 del Capítulo 6, para el agente de modelado de usuario se han obtenido los siguientes resultados.

D.1.1 Casos de uso asociados al problema

El primer objetivo a cumplir en la especificación es definir un conjunto de casos de uso iniciales. En la Figura D.1 se presenta el diseño de los caso de uso asociados a la creación de agente de modelado de usuario de los estudiantes de la UTPL, el cual se ha dividido en dos partes principales; la parte referente al agente de modelado de usuario propiamente dicha y a un modulo de administración del mismo.

Para cada uno de los casos de uso definidos en el diagrama propuesto, se muestra una pequeña descripción acerca de su objetivo dentro del agente.

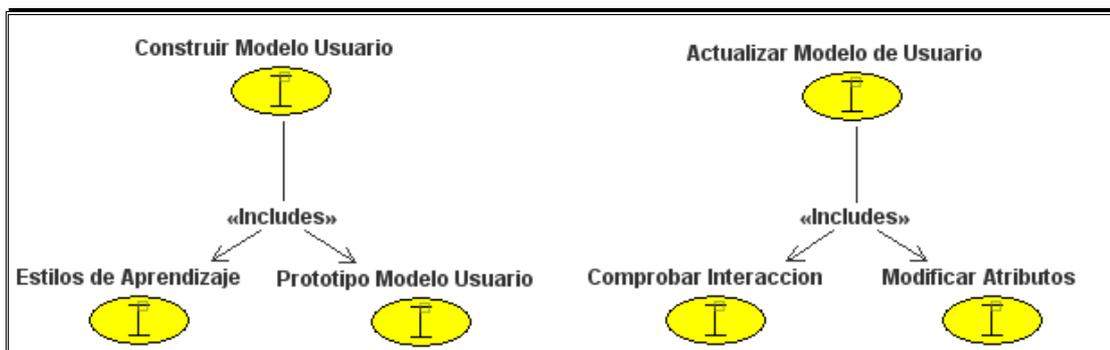


Figura D.1: Diagrama de casos de uso iniciales para el agente de modelado de usuario

- ◆ *Construir Modelo de Usuario.*- El agente de modelado de usuario necesita poder construir modelos de usuario iniciales a partir de clasificar al estudiante en un estilo de aprendizaje, y mediante un proceso de prototipado de los atributos que lo necesiten.
 - ◆ *Estilos de Aprendizaje.*- Los estilos de aprendizaje del estudiante se obtienen mediante la aplicación de un test propuesto y seleccionado en la sección 3.3.2 del Capítulo 3 de esta tesis y se utilizan como base en el prototipado del estudiante.
 - ◆ *Prototipo de Modelo de Usuario.*- Los prototipos del modelo de

estudiante fueron definidos en la sección 5.5 del Capítulo 5 de esta tesis y se utilizan para dar valores iniciales a los atributos del modelo de usuario.

- ◆ *Actualizar Modelo de Usuario.*- El agente de modelado de usuario necesita poder actualizar los modelos construidos anteriormente, y para ello accede a los cambios que ha tenido la interacción de los estudiantes con el sistema.
 - ◆ *Comprobar Interacción.*- El agente de modelado de usuario debe estar capacitado para poder obtener los cambios en la interacción de los usuarios con el sistema a lo largo de su estancia en el curso asignado.
 - ◆ *Modificar Atributos.*- El agente de modelado de usuario modifica los atributos del modelo de usuario si detecta cambios significativos en los datos de interacción del estudiante con el sistema.

D.1.2 Modelo de entorno

La definición del modelo del entorno del sistema es otro de los objetivos a conseguir en la etapa inicial de análisis del desarrollo para lo cual se siguen las actividades descritas en la tabla 6.1 del Capítulo 6.

En la figura D.2 se presenta el modelo del entorno para el sistema del agente de modelado de usuario, los elementos principales de este entorno se describen a continuación:

- ◆ *Entorno Virtual de Aprendizaje.*- Es la plataforma de soporte en el aprendizaje, por medio del cual los estudiantes de la UTPL interactúan de forma electrónica con los cursos en los cuales se encuentran matriculados.
- ◆ *Modelos de Usuario (Database).*- Base de datos relacional en la que se guardaran los modelos de usuario generados en cada curso por el agente de modelado de usuario.

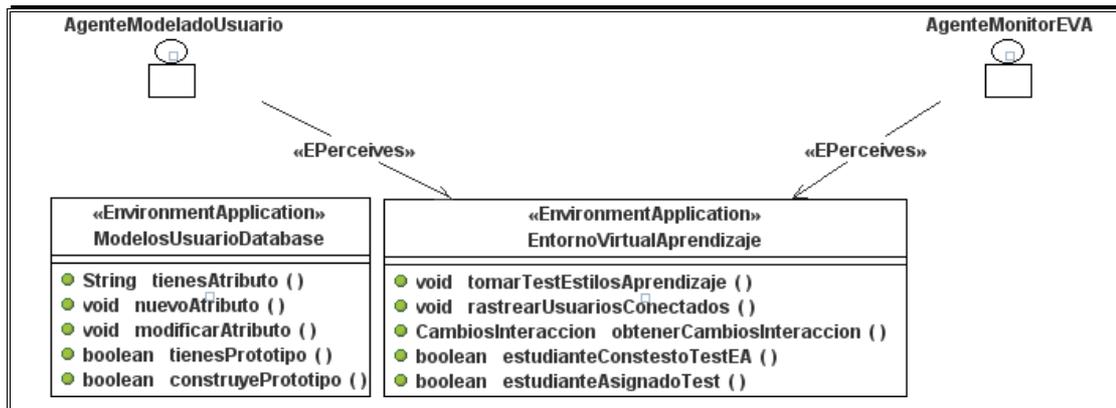


Figura D.2: Modelo de entorno para el agente de modelado de usuario

D.1.3 Modelo de organización sistema

La elaboración del modelo de organización del sistema mostrado en la Figura D.3 es el resultado de la aplicación de las actividades mencionadas en la tabla 6.2 del Capítulo 6.

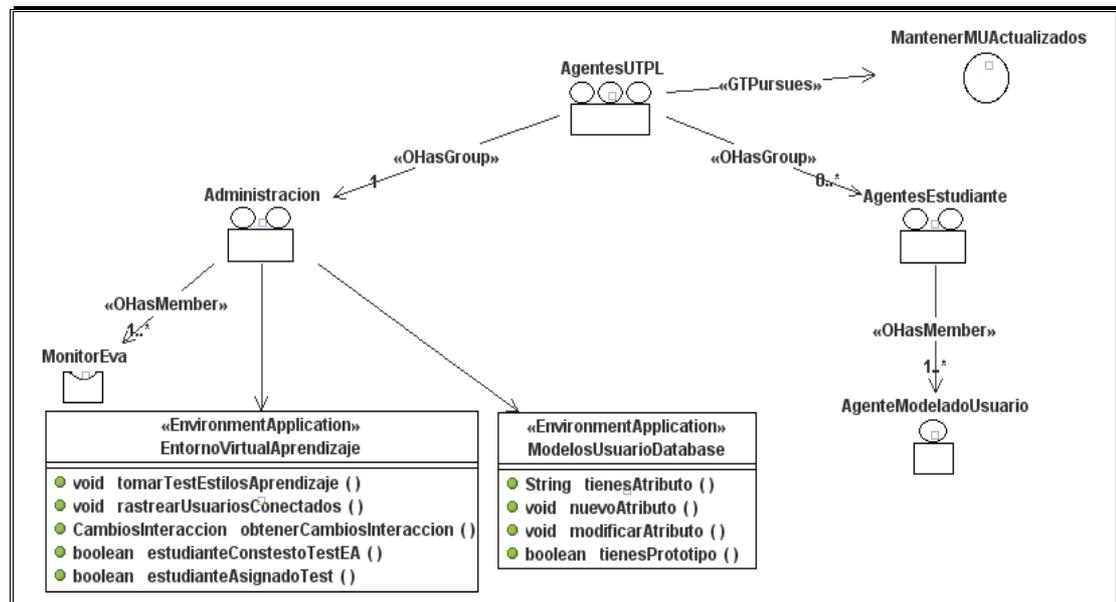


Figura D.3: Modelo de organización del sistema de agentes

Las entidades que componen este modelo se enumeran y describen a continuación:

- ◆ *Organización Agentes UTPL.*- Sirve para delimitar el espacio de desenvolvimiento del sistema, esta organización persigue el objetivo de mantener actualizados los modelos de usuario el (*MantenerMUActualizados*) el

cual lleva implícita la construcción de los modelos de usuario de los estudiantes.

- ◆ *Grupo Agentes Estudiante.*- Dentro de la organización UTPL puede haber un número determinado de Agentes de Estudiante que se organizan en grupos, que necesitan trabajar en contacto para generar los modelos de cada estudiante. En cada grupo de agentes de estudiante existen un conjunto de agentes *Constructores* de modelos, que caducan cuando el modelo para el que fueron instanciados fue creado. También existen un conjunto de agentes *Actualizadores* que trabajan constantemente en mantener los modelos de cada estudiante actualizados.
- ◆ *Grupo Administración.*- En este grupo gestiona recursos comunes a los grupos de agentes de estudiante y que serán utilizados a lo largo de todo el trabajo del sistema.

Anexo E

*Archivos del modulo de obtención de los
estilos de aprendizaje del EVA*

E.1 Archivos del modulo

Los archivos que componen el modulo de obtención de estilos de aprendizaje del EVA están contruidos en el lenguaje de programación php, en el modulo se encuentran un conjunto de archivos que sirven para el despliegue y la evaluación del test de estilos de aprendizaje de Felder y Silverman. Las modificaciones y desarrollo de estos archivos se muestran a continuación:

a) Modificación al archivo index.php del EVA

El archivo index.php del EVA se encuentra en el directorio raíz del sistema y es el que se encarga de desplegar las los diferente módulos del entorno virtual de aprendizaje para que el sistema pueda hacer uso del modulo se necesitan añadir a este archivo las siguientes modificaciones.

1. Abrir del archivo index.php
2. En la sección de carga de librerías colocar linea

```
require_once($CFG->dirroot . '/enrol/learning/lib.php');
```

Quedando de la siguiente forma:

```
require_once('config.php');  
require_once($CFG->dirroot . '/course/lib.php');  
require_once($CFG->dirroot . '/lib/blocklib.php');  
require_once($CFG->dirroot . '/enrol/utpl/lib.php');  
require_once($CFG->dirroot . '/enrol/learning/lib.php');
```

3. En la sección del SUBMENU UTPL añadir el siguiente código dentro de la sección *if* que averigua que el usuario que accede al sistema solo tiene el rol de estudiante dentro del EVA.

```
$action_link .= "&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;";  
//Estilos de Aprendizaje  
$tomarTest = tengoTest($USER->id);  
if ($tomarTest){  
    $action_link .= link_to_popup_window("$CFG->  
>wwwroot/enrol/learning/estilos.php?id=$USER->id",  
    'learning', 'Test de Estilos de Aprendizaje', 600, 800,
```

```
'Test de Estilos de Aprendizaje', 'none', true, 'command');  
$action_link .= "&nbsp;|&nbsp;";  
}
```

Quedando de la siguiente forma:

4. Finalmente salvar el archivo *index.php*.

b) Archivo estilos.php

Es el archivo principal del modulo, porque es el que despliega el test de estilos de aprendizaje al estudiante que lo va a contestar utiliza la una función denominada *imprimirTest()*, que se encuentra en la librería de *lib.php* del modulo, para mostrar las preguntas del test de estilos de aprendizaje.

c) Archivo lib.php

El archivo *lib.php* implementa funciones que desarrollan el trabajo de presentación y toma del test de estilos de aprendizaje a los estudiantes, accediendo a la base de datos del entorno de aprendizaje (EVA). Las funciones que implementa este archivo se describen a continuación:

- ◆ *tengoTest()* .- Función que obtiene si el estudiante que esta registrado en el sistema tiene asignado el contestar el test de estilos de aprendizaje.
- ◆ *ImprimirTest()* .- Función que presenta el formulario de en que se encuentran las preguntas del test de estilos de aprendizaje.
- ◆ *obtenerPreguntas()* .- Función que obtiene las preguntas del test de estilos de aprendizaje de la base de datos del EVA.
- ◆ *obtenerOpcionesPregunta()* .- Función que recupera de la base de datos del entorno virtual de aprendizaje las opciones para contestar cada una de las preguntas del test de estilos de aprendizaje.

d) Archivo ejecutar_test.php

El archivo *ejecutar_test.php* contiene una implementación que permite recoger los resultados de los valores contestados por los estudiantes en el test de estilos de

aprendizaje. Luego de realizar esto procede a evaluar estos valores, para finalmente dar un veredicto acerca de a que tipología de estilos de aprendizaje pertenece el estudiante.

Además de esto, el resultado final obtenido así como los resultados que se obtuvieron en las otras tipologías son guardados en la base de datos del entorno virtual de aprendizaje.

En cuanto a la interfaz de usuario, luego de contestar y evaluar el test de estilos de aprendizaje se presenta el resultado, así como una descripción breve sobre la tipología de estilos de aprendizaje a la que pertenece.

E.2 Modificación a la base de datos

Para el desarrollo de la implementación del modulo obtención de estilos de aprendizaje del EVA se creó un esquema de datos adicional que soporta el trabajo del modulo. Consta de cinco entidades que se muestran en la figura E.1 de las cuales se realiza una descripción a continuación:

a) **mdl_learning_asignaciones**

En esta entidad es en donde se guardan las asignaciones que el agente de modelado de usuario realiza para que el estudiante realice el test de estilos de aprendizaje, y dependiendo del valor que contenga el campo estado el modulo tomara o no el test al estudiante.

- ◆ *id* (int(10)) .- Número que identifica al registro, es único, auto-incrementable y se toma como llave primaria del mismo.
- ◆ *userid* (int(10)) .- Número que representa la relación entre esta entidad y la denominada como mdl_user. Se toma como llave foránea.
- ◆ *estado* (varchar(100)) .- El estado en el que se encuentra el test de estilos de aprendizaje para el usuario asignado.

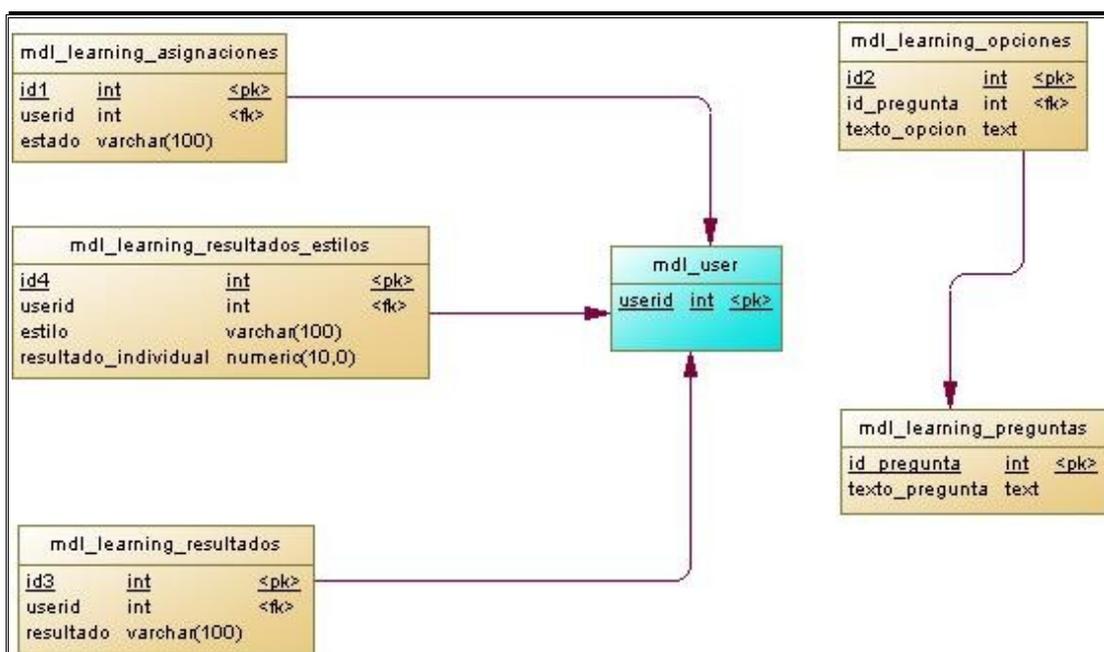


Figura E.1: Modelo de datos de modulo de obtención de estilos de aprendizaje

b) mdl_learning_resultados_estilos

En esta entidad se almacenan los resultados que el estudiante ha alcanzado en todas las tipologías de estilos de aprendizaje que se presentan en el test propuesto.

- ♦ *id* (bigint(10)) .- Número que identifica al registro, es único, auto-incrementable y se toma como llave primaria del mismo.
- ♦ *userid* (bigint(10)) .- Número que representa la relación entre esta entidad y la denominada como mdl_user. Se toma como llave foránea.
- ♦ *estilo* (varchar(100)) .- El estilo de aprendizaje al que pertenece el resultado.
- ♦ *resultado_individual* (decimal(10,0)) .- El resultado que obtuvo el estudiante por la tipología de estilos de aprendizaje descrita.

c) mdl_learning_resultados

En esta entidad se guarda el resultado que obtuvo el estudiante luego de responder el test de estilos de aprendizaje.

- ♦ *id* (bigint(10)) .- Número que identifica al registro, es único, auto-incrementable y se toma como llave primaria del mismo.

- ◆ *userid* (bigint(10)) .- Número que representa la relación entre esta entidad y la denominada como mdl_user. Se toma como llave foránea.
- ◆ *resultado* (varchar(100)) .- El resultado que obtuvo el estudiante al resolver el test de estilos de aprendizaje.

d) **mdl_learning OPCIONES**

Esta entidad sirve para guardar las opciones de respuesta que presentan las preguntas del test de estilos de aprendizaje.

- ◆ *id* (bigint(10)). - Número que identifica al registro, es único, auto-incrementable y se toma como llave primaria del mismo.
- ◆ *id_pregunta* (bigint(10)) .- Número que representa la relación entre esta entidad y la denominada como mdl_learning_preguntas. Se toma como llave foránea.
- ◆ *texto_opcion* (text) .- En el que se guarda el texto del que esta conformado la opción.

e) **mdl_learning PREGUNTAS**

En la entidad mdl_learning_preguntas se almacenan las preguntas que se hacen al estudiante.

- ◆ *id* (bigint(10)) .- Número que identifica al registro, es único, auto-incrementable y se toma como llave primaria del mismo.
- ◆ *texto_pregunta* (text) .- el texto que en que se encuentra contenida una pregunta del test de estilos de aprendizaje.

Referencias Bibliográficas

Akoulchina, I., y Ganascia, J. (1997). SATELIT-Agent: an adaptive interface based on learning interface agents technology. En A. Jameson, C. Paris, y C. Tasso (Eds.). *User Modeling: Proceedings of the Sixth International Conference*. Springer-Verlag (pp. 21–32).

Alonso, C. (2001). *Estilos de aprendizaje y estudiantes universitarios*. En Gallego D, Alonso C. Estilos de Aprender y Estilos de Enseñar. Curso de Doctorado: UNED; 2001-2002. (p. 39)

Arias, F., Ovalle D. A. y Cadavid, J. (2008). *Modelo para la Selección de Objetos de Aprendizaje Adaptados a los Estilos de los Estudiantes*. GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial de la Escuela de Ingeniería de Sistemas de la Universidad Nacional de Colombia.

Baldiris, S., Santos O. C., Boticario J. G. y Fabregat R. (2007). *Modelado de Cursos Mediante el Uso intensivo de Estándares Educativos Para dar Soporte a Adaptaciones Dinámicas*. Universidad de Girona y UNED. España.

Botía, J., González, J., Gómez, J. y Pavón, J. (2008). The Ingenias Project: Methods And Tool For Developing Multiagent Systems. En *IEEE Latin America Transactions*, Vol 6, no 6.(pp. 529-534)

- Botti, V. (2000). Agentes Inteligentes: El Siguiete Paso en la Inteligencia Artificial. En *NOVATICA Especial de Aniversario*. Universidad Politécnica de Valencia. (pp. 95-99)
- Cabrera, J. y Fariñas, G.(2005). El estudio de los estilos de aprendizaje desde una perspectiva vigostkiana: una aproximación conceptual. *Revista Iberoamericana de Educación*, 42. (pp. 11-19).
- Chandrasekaran, B., Josephson, J., y Benjamins, V. (1999). Ontologies: What are they? why do we need them? En *Special Issue on Ontologies* (p.20–26). IEEE Intelligent Systems.
- Cooper, G. y Herskovits, E. (1992). A Bayesian Method for the Induction of Probabilistic Networks from Data. En *Machine Learning*, 9, (pp 309-347).
- Debevc, M., Meyer, B., Donlagic, D., & Svecko, R. (1996). Design and evaluation of an adaptive icon toolbar. En *User Modeling and User-Adapted Interaction* (pp. 1–21).
- DeLoach, S. (2001). Analysis and Design using MaSE and agentTool. En *Proceedings of the 12th Midwest Artificial Intelligence and and Cognitive Science Conferece* (MAICS).
- Demazeau, Y. (1995) From cognitive interactions to collective behaviour in agent-based systems. En *European Conference on Cognitive Science*.
- Duda, R. y Hart, P. (1973). *Pattern classification and scene analysis*. New York: John Wiley & Sons.

- Durán E. y Costaguta R. (2007). Minería de Datos para Descubrir Estilos de Aprendizaje. En OEI (Eds.), *Revista Iberoamericana de Educación* 42, (pp. 11-21).
- Fayyad, U., M., Piatetsky-Shapiro, G., y Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. *Advantages in Knowledge Discovery and Data Mining*, AAAI/MIT Press, (pp. 1-34).
- Felder, R y Silverman, L. (1988): “*Learning and Teaching Styles in Engineering Education Application*”. *Engr. Education*, 78 (7), (pp. 674-681).
- Fernández, R. (2008). *Glosario básico inglés-español para usuarios de internet*. Informática A. (Eds.). Obtenido de <http://www.ati.es/novatica/glointv4.html>
- Fink, J. (2001). User Modeling Servers: Requirements, Design, and Evaluation. En U. o. Essen, *Department of Mathematics and Computer Science*. Germany.
- Fontalvo H., Iriarte F., Domínguez E., Ricardo C., Ballesteros B., Muñoz V, y Campo J. (2007). Diseño de Ambientes Virtuales de Enseñanza Aprendizaje y Sistemas de Hipermedia Adaptativos Basados en Modelos de Estilos de Aprendizaje. En *Zona Proxima*, 8. Universidad del Norte de Colombia, Barranquilla, Colombia, (pp. 42-61).
- Franklin, S., y Graesser, A. (1996). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. En *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag.
- Friedman, N., Geiger, D. y Goldszmidt, M. (1997). Bayesian Networks Classifiers. En *Machine Learning*, 29. (pp 131-163)

- García, E., y Pascual, F. (1994). Estilos de aprendizaje y cognitivos. En: Puente A., *Estilos de aprendizaje y enseñanza*. Madrid: CEPE, S.A. (pp. 30).
- Gaudioso E. (2002). *Contribuciones al Modelado del Usuario en Entornos Adaptativos de Aprendizaje y Colaboración a través de Internet mediante técnicas de Aprendizaje Automático*. Tesis Doctoral. Universidad Complutense de Madrid, Madrid, España.
- Gómez, J. y Fuentes, R. (2002). The INGENIAS Methodology. *Fourth Iberoamerican Workshop on Multi-Agent Systems Iberagents*.
- Gómez, J. (2003). Metodologías para el Desarrollo de Sistemas Multi-Agente. En *Revista Iberoamericana de Inteligencia Artificial*, 18, (pp. 51-63)
- Gómez, J. y Pavón, J. (2004). INGENIAS Methodology and Tools. En *AL3 PROMAS TFG*. Universidad Complutense de Madrid.
- Hassan, Y., Martín, F., y Iazza, G. (2004). *Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información*. Recuperado mes día, año de <http://www.hipertext.net/web/pag206.htm#Modelado%20del%20usuario>
- Hernández, J., Ramirez, M., J., y Ferri, C. (2004). *Introducción a la Minería de Datos*. España: Pearson Education.
- Keefe, J. (1988). *Profiling and Utilizing Learning Style*, Reston, Virginia, NASSP, p. 48
- Kinny, D., Georgeff, M., y Rao, A. (1997). *A Methodology and Modelling Technique for Systems of BDI Agents*.

Kobsa, A., Koenemann, J., y Pohl, W. (1999). *Personalised hypermedia presentation techniques for improving online customer relationships*. German National Research Center for Information Technology. St. Augustin, Germany: Technical Report 66.

Kobsa, A., (2002). User modeling, Privacy and Security. Invited talk. En *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, 2347*, en Lecture Notes in Computer Science (LNCS), Malaga, España, Springer Verlag.

Linton, F., y Schaefer, H. (2000). Recommender Systems for Learning: Building User and Expert Models through Long-Term Observation of Application Use. En Kluwer Academic Publishers (Eds.), *User Modeling and User-Adapted Interaction 10*. (pp. 181-207).

Mitchell, T., Caruana, R., Freitag, D., McDermott, J., y Zabowski, D. (1994). Experience With a Learning Personal Assistant. En *Communications of the ACM*, 37. pp. 81–91.

Nwana, H. (1996). *Software Agents: An Overview*. *Intelligent Systems Research*. AA&T, BT Laboratories, Ipswich, United Kingdom

Nwana, H., Ndumu, D, Lee, L., y Collis, J. (1999). ZEUS: A Toolkit for Building Distributed Multi-Agent Systems, En *Applied Artificial Intelligence Journal*, vol. 1, no. 13, (pp. 129-185).

Ortega, J., (1995). Exploiting Multiple Existing Models and Learning Algorithms, En *AAAI 96 - Workshop in Induction of Multiple Learning Models*.

- Peña, C., Marzo, J., de la Rosa, J. y Fabregat, R. (2002). Un sistema de Tutoría Inteligente Adaptativo Considerando Estilos de Aprendizaje. Universitat de Girona, España.
- Pressman, R. (2006). *Ingeniería del Software. Un Enfoque Practico*. Sexta Edición, D.F., México, McGraw-Hill Interamericana.
- Quinlan, J.(1993). *C4.5: Programs for machine learning*, Morgan Kaufmann Pub.
- Ricordel, P. (2001). *Programmation Orientée Multi-Agents , Développement et Déploiement de Systèmes Multi-Agents Voyelles*. Tesis doctoral. INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE.
- Rubio M. (2002). *Orientacion y Metodología para la Educación a Distancia*. Loja, Ecuador: CEDIB.
- Rubio B., Delgado G., Toledo J., Sánchez J., Delgado D., Rubio E., Ocón A., Galán M. y Álvarez L. (2008). Desarrollo de un Módulo para Detectar el Estilo de Aprendizaje (Istest) En *Adaptaciones de Moodle a los Requerimientos de un plan de Formación on-line Personalizado*. (pp. 1-2)
- Russell, S., y Norving, P., (2004). *Inteligencia Artificial. Un Enfoque Moderno*. Madrid, España: Pearson Education.
- Seung, S. (2002). *Multilayer perceptrons and backpropagation learning*.
- Schwab, I., Kobsa, A., y Koychev, I. (2000). Learning about Users from Observation. En Park M. (Eds.), *2000 AAAI Spring Symposium*. (pp. 102–106).
- Sierra, B. (2006). *Aprendizaje Automático: conceptos básicos y avanzados. Aspectos*

prácticos utilizando el software WEKA. Madrid, España, Pearson Education.

Tansley, D. y Hayball, C. (1993). *Knowledge Based systems Analysis and Design a KADS developer's handbook*. Prentice Hall.

Valdiviezo, P., Jara, I. y Pesantez, R. (2009). *Educación Adaptativa en la Web: Estado del Arte*. Tesis de Grado, Universidad Técnica Particular de Loja, Loja, Ecuador.

Webb, G., Pazzani, M., y Billsus, D. (2001). Machine Learning for User Modeling. En Kluwer Academic Publishers (Eds), *User Modeling and User-Adapted Interaction II*. (pp. 19-29).

Witten, I., Frank, E., Trigg, L., Hall, M., Holmes, G. y Cunningham, S. (2000). *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*. Department of Computer Science, University of Waikato, New Zealand.

Wooldridge, M., Jennings, N., y Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design, En *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 15.

Zytkon, J., y Klosgen, W. (2001). *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press.