

2004

Adaptación dinámica al usuario en un sistema de enseñanza mediante aprendizaje por refuerzo

Iglesias Maqueda, Ana María

<http://hdl.handle.net/10016/652>

Descargado de e-Archivo, repositorio institucional de la Universidad Carlos III de Madrid



Universidad Carlos III de Madrid
Escuela Politécnica Superior

Tesis Doctoral
Programa de Doctorado en Ingeniería Informática

Adaptación Dinámica al Usuario en un Sistema de
Enseñanza mediante Aprendizaje por Refuerzo

AUTORA: Ana María Iglesias Maqueda
DIRECTORES: Paloma Martínez Fernández
Ricardo Aler Mur

29 de octubre de 2004

Agradecimientos

No es sencillo resumir en tan pocas líneas todo el apoyo y cariño que he recibido durante la realización de la tesis doctoral. Aún así, no me gustaría cerrar esta fase sin que quede plasmado en este documento mi agradecimiento, aunque quizás no sea capaz de transmitir ni la infinitésima parte.

En primer lugar me gustaría agradecer a mis tutores, Paloma Martínez y Ricardo Aler, el apoyo que me han ofrecido durante la realización de la tesis. Gracias.

A todos los que han participado en la elaboración y/o evaluación del sistema RLATES. Gracias.

A mis compañeros y amigos les he de agradecer el estar siempre ahí cuando se les necesita, el “*¡ánimo, ya queda poco!*” en el momento más oportuno; el “*¿en qué te puedo ayudar?*” cuando más agobiada me veían. Gracias Fernan, Susi, Rafa, Agustín, Agapito, Yolanda, Anabel, Luis, Germán, Susana, Javier Carbó, Loli, Elena, Javi Calle, César, Harith, Lourdes, Conchi, Josele, Manu ...

Cómo olvidar a los amigos de Villacañas, Toledo, Burguillos, Calera, a los amigos de la Complutense y de la Carlos III... gracias por ser como sois, por animarme en las horas bajas y disfrutar conmigo los días felices. Gracias.

He de agradecerle a toda mi familia todo el cariño que me han dado, el empuje y confianza que me han proporcionado, el día a día. Gracias Papá, Mamá, Chule, Marién, Juanete, Mamen, Luismi, Abuelitas Ramona y Ana y al chiquitín: Miguel; y no se me puede olvidar mi nueva familia: Lola, Fernando, Virgi, Luis, Joaquín, Estíbaliz, y las chiquitinas: Alba y Marta. Gracias a toda la familia: tíos, primos y allegados.

Y qué decir de Fernan, mi mejor amigo, mi shesy, mi marido. Cómo explicar en tres líneas todo lo que me das, lo que me haces sentir, el apoyo, ánimo, confianza, paciencia... Miles de gracias. Sabes que esta tesis también es tuya.

Resumen

En los últimos años los sistemas de educación a distancia han mejorado abriendo nuevas perspectivas a planteamientos diferentes de enseñanza. La utilización de Internet como herramienta para estos sistemas de enseñanza ha favorecido el acceso a la educación, evitando barreras físicas de movilidad de los alumnos a las aulas y la falta de comunicación informática debido a diferencias entre plataformas de estos sistemas y los ordenadores de los alumnos.

Uno de los problemas más importantes de los sistemas de educación a distancia es personalizar la enseñanza a cada estudiante, adaptando su política pedagógica dependiendo de las necesidades de aprendizaje que tengan los estudiantes. La aplicación de técnicas de inteligencia artificial a los sistemas de educación permiten que estos sistemas se adapten de forma *inteligente* a los estudiantes, representando de forma explícita o implícita las políticas pedagógicas, habilidades o conocimiento de los expertos.

Los *Sistemas de Educación Adaptativos e Inteligentes en Web* (del inglés *Web-based Adaptive and Intelligent Educational Systems*) (SEAI en *Web*) son sistemas de educación basados en Internet donde se aplican técnicas de inteligencia artificial con el objetivo de adaptar el contenido del sistema a los estudiantes según sus necesidades pedagógicas. Definir políticas pedagógicas efectivas en estos sistemas es uno de los principales problemas de los SEAI en *Web*, decidiendo qué, cómo y cuándo mostrar el contenido del curso a los estudiantes. En este trabajo de tesis doctoral se ha utilizado el modelo de *Aprendizaje por Refuerzo* para resolver el problema de soporte adaptativo a la navegación a través del contenido del sistema y de presentación de dicho contenido (eligiendo el *formato* de presentación del contenido). Al aplicar el modelo de *Aprendizaje por Refuerzo* en el módulo pedagógico del sistema, éste es capaz de aprender automáticamente las mejores tácticas pedagógicas a aplicar para cada estudiante individualmente, basado únicamente en la experiencia adquirida con otros estudiantes de características de aprendizaje similares, como hace el tutor en las aulas.

Con esta tesis doctoral se muestra cómo la definición de tácticas pedagógicas de los SEAI en *Web* se puede modelar como un problema de *Aprendizaje por Refuerzo*. También se muestra cómo el sistema es capaz de aprender a enseñar

IV

desde un punto de vista teórico, realizando experimentos simulando el comportamiento de los estudiantes. Finalmente, se muestra el comportamiento del sistema a través de Internet, con estudiantes reales que asisten a asignaturas impartidas en la Universidad Carlos III de Madrid , desde un punto de vista práctico.

Abstract

Last years, distance educational systems have been improved opening new perspectives on different ways to teach. The use of Internet as a tool for the educational systems helps the students and avoid the physical barriers in the access to the classrooms and the incompatibilities due to the different students system platforms.

One of the most important issues in distance educational systems is to personalize the teaching to each student, adapting its pedagogical policy according to the pedagogical student needs. The application of artificial intelligence techniques to educational systems allows the systems to adapt in an intelligent way to the students, representing implicitly or explicitly the pedagogical policies, abilities and expert knowledge.

The Web-based Adaptive and Intelligent Educational Systems (Web-based AIES) use artificial intelligence techniques in order to adapt the content to the students according to their pedagogical needs. One of the most important issues in these systems is to define effective pedagogical strategies for tutoring students according to their needs. The pedagogical strategies define *what*, *how* and *when* to show the system content to the students. In this PhD. Thesis we propose to use a pedagogical knowledge representation based on the Reinforcement Learning (RL) model. Using this model, the system is able to automatically provide adaptive navigation support and presentation support to the students (choosing the best presentation format for the content). The system learns which is the best pedagogical way to teach each student individually based only on acquired experience with other students with similar learning characteristics, like a human tutor does.

In this dissertation we show how the definition of the pedagogical policies in the AIESs can be considered as a *Reinforcement Learning* problem from a theoretical point of view. Next, we show how the system is able to learn to teach from a theoretical point of view, using simulated students. Finally, we show how the system is able to teach in a practical point of view, interacting with real students of our University through Internet.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Fundamentos	3
1.3. Objetivos	5
1.4. Organización de la Memoria	7
2. Estado de la Cuestión	9
2.1. Programas de Enseñanza	9
2.1.1. Repaso Histórico	10
2.1.2. Sistemas de Educación Adaptativos e Inteligentes en <i>Web</i> .	12
2.1.3. Sistemas Hipermedia Adaptativos en la Enseñanza	13
2.1.4. Tutores Inteligentes	16
2.2. Políticas Pedagógicas	18
2.2.1. Secuenciar el Contenido	18
2.2.2. Taxonomía de la Tarea de Secuenciar el Contenido	20
2.2.3. Elección de la Mejor Política de Aprendizaje	21
2.3. IA en Sistemas de Educación	23
2.3.1. Aplicación de Inteligencia Artificial en Programas de Enseñanza	23
2.3.2. Aprendizaje por Refuerzo Aplicado a Programas de Enseñanza	25
2.4. Aprendizaje por Refuerzo	27
2.4.1. Componentes	28
2.4.2. Procesos de Decisión de Markov	29
2.4.3. Puntos Clave en el Aprendizaje por Refuerzo	30
2.4.4. Clasificación de Algoritmos	31
2.4.5. Funciones de Utilidad	32
2.4.6. Q-learning: un Algoritmo de Aprendizaje por Refuerzo	33
2.4.7. Exploración vs Explotación	35
2.5. Discusión	38

3. Propuesta	41
3.1. Motivación	42
3.2. Adaptación Mediante RL	43
3.2.1. Descripción del Modelo	43
3.2.2. Componentes del Modelo	45
3.2.3. Exploración vs. Explotación	48
3.2.4. Ejemplo de Aprendizaje en RLATES	49
3.2.5. Fases de Funcionamiento de RLATES	54
3.3. Diseño e Implementación del Sistema RLATES	56
3.3.1. Módulo del Estudiante	56
3.3.2. Módulo del Dominio	57
3.3.3. Módulo Pedagógico	58
3.3.4. Módulo del Interfaz	60
3.3.5. Implementación	60
3.4. Interfaz de Usuario	61
3.4.1. Guía Directa y Adaptación de la Presentación	62
3.4.2. Soporte Adaptativo a la Navegación de forma Indirecta	66
4. Aspectos Previos a la Experimentación	69
4.1. Conceptos Teóricos de una Evaluación	70
4.2. Método de Evaluación	71
4.3. Modelos del Dominio en Experimentos	74
5. Experimentación Simulada	79
5.1. Motivación de la Experimentación	80
5.2. Diseño de la Experimentación	80
5.3. Parámetros de Evaluación	86
5.4. Resultados Obtenidos	87
5.4.1. Estrategia de Exploración/Explotación E-GREEDY	88
5.4.2. Estrategia de Exploración/Explotación de BOLTZMANN	98
5.4.3. Inicialización de las Tácticas Pedagógicas	109
5.5. Conclusiones	113
6. Experimentación Real	115
6.1. Motivación de la Experimentación	115
6.2. Diseño de la experimentación	116
6.2.1. Entorno en la Experimentación	116
6.2.2. Versiones Implementadas del Sistema	119
6.2.3. Inicialización del Sistema	123
6.2.4. Evaluación de la Usabilidad del Sistema y Grado de Conocimiento del Estudiante	123

6.3.	Parámetros de Evaluación	125
6.4.	Resultados Obtenidos	126
6.4.1.	Resultados de la Interacción Estudiante-RLATES	126
6.4.2.	Evaluación de la Usabilidad del Sistema	141
7.	Conclusiones y Trabajos Futuros	153
7.1.	Conclusiones	153
7.1.1.	Resultados Obtenidos	153
7.1.2.	Principales Aportaciones	158
7.1.3.	Difusión de Resultados	161
7.2.	Trabajos Futuros	165
7.2.1.	Modelado y Clasificación de Estudiantes	165
7.2.2.	Mejorar la Evaluación del Conocimiento del Estudiante	166
7.2.3.	Reducción del Número de Estados y Acciones	166
7.2.4.	Inicialización de Tácticas Pedagógicas con Conocimiento de un Experto	167
7.2.5.	Desarrollo y Evaluación del Sistema <i>RLATES mixto</i>	168
7.2.6.	Aplicación del Modelo de <i>Aprendizaje por Refuerzo</i> a la Elección de las Estrategias Pedagógicas	168
A.	Ejemplos de Contenidos de RLATES	181

Índice de figuras

2.1. Arquitectura de un Tutor Inteligente.	17
2.2. Modelo de Aprendizaje por Refuerzo.	28
2.3. Algoritmo <i>Q-learning</i>	34
3.1. Esquema de Aprendizaje por Refuerzo aplicado a los SEAI	44
3.2. Algoritmo <i>Q-learning</i> adaptado al dominio de los Sistemas de Educación Adaptativos e Inteligentes en <i>Web</i>	52
3.3. Ejemplo de Aprendizaje en RLATES. Modelo del dominio.	52
3.4. Ejemplo de Aprendizaje en RLATES. MDP del estudiante.	53
3.5. Ejemplo de Aprendizaje en RLATES. Tabla Q inicializada a cero	53
3.6. Ejemplo de Aprendizaje en RLATES. Tabla Q. Paso 1	53
3.7. Ejemplo de Aprendizaje en RLATES. Tabla Q. Paso 2	53
3.8. Fases Funcionales de RLATES	54
3.9. Arquitectura de RLATES	56
3.10. Modelo del Dominio. Estructura general de un árbol de conoci- miento	58
3.11. Ejemplo de los valores de un estado de estudiante	59
3.12. Interfaz de RLATES. Soporte Adaptativo a la Navegación y a la Presentación. Guía Directa.	63
3.13. Interfaz de RLATES. Opciones de Siguiente Página a Visitar (Boltz- mann).	64
3.14. Interfaz de RLATES. Página <i>Web</i> mostrada después de elegir un contenido concreto a ser mostrado.	65
3.15. Interfaz de RLATES. Soporte Adaptativo a la Navegación. Guía Indirecta.	67
3.16. Interfaz de RLATES. Colores de los Enlaces en el Árbol de Con- cimiento.	68
4.1. RLATES interaccionando con estudiantes simulados	72
4.2. RLATES interaccionando con estudiantes reales	73
4.3. Modelo del dominio A	75

4.4.	Modelo del dominio B	76
5.1.	Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del <i>grupo1</i> cuando se les enseña el modelo del dominio A	82
5.2.	Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del <i>grupo2</i> cuando se les enseña el modelo del dominio A	83
5.3.	Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del <i>grupo3</i> cuando se les enseña el modelo del dominio A	83
5.4.	Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del <i>grupo1</i> cuando se les enseña el modelo del dominio B	84
5.5.	Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del <i>grupo2</i> cuando se les enseña el modelo del dominio B	84
5.6.	Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del <i>grupo3</i> cuando se les enseña el modelo del dominio B	85
5.7.	Planificación de los experimentos simulados	87
5.8.	SIMUL-1. Curvas de aprendizaje de RLATES variando el parámetro e de la estrategia de exploración <i>e-greedy</i> para estudiantes pertenecientes al <i>grupo2</i> , $\alpha=0,1$. Se ha utilizado el modelo del dominio A.	89
5.9.	SIMUL-1. Desviación típica del sistema variando el parámetro e de la estrategia de exploración <i>e-greedy</i> para estudiantes pertenecientes al <i>grupo2</i> , $\alpha=0,1$. Se ha utilizado el modelo del dominio A para estos experimentos.	90
5.10.	SIMUL-2. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al <i>grupo2</i> y modelo del dominio A.	90
5.11.	SIMUL-2. Desviación típica del sistema variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al <i>grupo2</i> y modelo del dominio A.	91
5.12.	SIMUL-3. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, $e=0,9$ y modelo del dominio A.	92
5.13.	SIMUL-3. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, $e=0,9$ y modelo del dominio A.	92

5.14. SIMUL-4. Curvas de aprendizaje de RLATES variando el parámetro e de la estrategia de exploración *e-greedy* para estudiantes pertenecientes al *grupo2*, $\alpha=0,5$ y modelo del dominio *B*. 93

5.15. SIMUL-4. Desviación Típica de RLATES variando el parámetro e de la estrategia de exploración *e-greedy* para estudiantes pertenecientes al *grupo2*, $\alpha=0,5$ y modelo del dominio *B*. 94

5.16. SIMUL-5. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al *grupo2* y modelo del dominio *B*. 95

5.17. SIMUL-5. Desviación típica del sistema variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al *grupo2* y modelo del dominio *B*. 95

5.18. SIMUL-6. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, $e=0,9$ y modelo del dominio *B*. 96

5.19. SIMUL-6. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, $e=0,9$ y modelo del dominio *B*. 97

5.20. SIMUL-7. Curvas de aprendizaje de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interactuando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *A*. 99

5.21. SIMUL-7. Desviación típica de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interactuando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *A*. 100

5.22. SIMUL-8. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interactuando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *A*. 100

5.23. SIMUL-8. Desviación típica de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interactuando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *A*. 101

5.24. SIMUL-9. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es el *A*. 101

- 5.25. SIMUL-9. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es el A. 102
- 5.26. SIMUL-10. Curvas de aprendizaje de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*. 103
- 5.27. SIMUL-10. Desviación típica de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*. 103
- 5.28. SIMUL-11. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*. 104
- 5.29. SIMUL-11. Desviación típica de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*. 105
- 5.30. SIMUL-12. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es el *B*. 105
- 5.31. SIMUL-12. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es el *B*. 106
- 5.32. Curvas de aprendizaje de RLATES comparando las estrategias de exploración/explotación de *e-greedy* (su parámetro *e* se ha fijado a 0,9) y *Boltzmann* (su parámetro de temperatura (τ) se ha fijado a 0,01). El ratio de aprendizaje (α) se ha fijado a 0,5 y se ha trabajado con estudiantes del *grupo2*. El modelo del dominio utilizado es el *B*. 107

5.33. Desviación típica de RLATES comparando las estrategias de exploración/explotación de <i>e-greedy</i> (su parámetro e se ha fijado a 0,9) y <i>Boltzmann</i> (su parámetro de temperatura (τ se ha fijado a 0,01)). El ratio de aprendizaje (α) se ha fijado a 0,5 y se ha trabajado con estudiantes del <i>grupo2</i> . El modelo del dominio utilizado es el <i>B</i>	107
5.34. Curvas de aprendizaje de RLATES comparando el efecto de la variación en la temperatura de la estrategia de <i>Boltzmann</i> y plasmando el efecto de la inicialización del sistema. El ratio de aprendizaje (α) se ha fijado a 0,5 y el modelo del dominio utilizado es el <i>B</i> . Se ha trabajado inicialmente con estudiantes del <i>grupo3</i> y posteriormente con estudiantes del <i>grupo2</i>	110
5.35. Curvas de aprendizaje de RLATES comparando el efecto de la variación en la temperatura de la estrategia de <i>Boltzmann</i> y plasmando el efecto de la inicialización del sistema. El ratio de aprendizaje (α) se ha fijado a 0,5 y el modelo del dominio utilizado es el <i>B</i> . Se ha trabajado inicialmente con estudiantes del <i>grupo2</i> y posteriormente con estudiantes del <i>grupo3</i>	111
5.36. Curvas de aprendizaje de RLATES analizando las ventajas de la inicialización.	112
6.1. Interfaz de IGNATES.	120
6.2. Interfaz de RLATES1 y RLATES2. Opciones para continuar la navegación	121
6.3. Interfaz de RLATES1 y RLATES2. Pestañas de contenido	122
6.4. Planificación de la experimentación con estudiantes reales.	127
6.5. REAL-1: Número de páginas web que visitó cada estudiante al completar una interacción con el sistema en cada una de las versiones (RLATES e IGNATES).	129
6.6. REAL-2: Tiempo que tardó cada estudiante en completar una interacción con el sistema.	130
6.7. REAL-4: Número de páginas web que visitó cada estudiante al completar una interacción con el sistema en cada una de las versiones (RLATES e IGNATES).	133
6.8. REAL-5: Tiempo que tardó cada estudiante en completar una interacción con el sistema.	133
6.9. REAL-7: Número de páginas web que visitó cada estudiante al completar una interacción con el sistema en cada una de las versiones (RLATES e IGNATES).	136
6.10. REAL-8: Tiempo que tardó cada estudiante en completar una interacción con el sistema.	137

A.1.	Contenido de RLATES. Un elemento de tipo <i>Introducción</i> en formato <i>Texto</i> del tema Entidad	182
A.2.	Contenido de RLATES. Un elemento de tipo <i>Definición</i> en formato <i>Imagen</i> del tema Entidad	183
A.3.	Contenido de RLATES. Un elemento de tipo <i>Definición</i> en formato <i>Texto</i> del tema Entidad	184
A.4.	Contenido de RLATES. Un elemento de tipo <i>Ejercicio</i> en formato <i>Imagen</i> del tema Entidad	185
A.5.	Contenido de RLATES. Un elemento de tipo <i>Ejercicio</i> en formato <i>Texto</i> del tema Entidad	186
A.6.	Contenido de RLATES. Ejemplo 1 de Test del tema Entidad . . .	187
A.7.	Contenido de RLATES. Ejemplo 2 de Test del tema Entidad . . .	188

Índice de tablas

4.1. Acciones en el modelo del dominio <i>A</i>	76
4.2. Acciones en el modelo del dominio <i>B</i>	77
5.1. Grupos de estudiantes simulados contruidos para los experimentos.	81
6.1. Comparación de los resultados obtenidos en la experimentación con estudiantes simulados y reales cuando el sistema utilizaba el modelo del dominio <i>A</i>	131
6.2. Comparación de los resultados obtenidos en la experimentación con estudiantes simulados y reales cuando el sistema utilizaba el modelo del dominio <i>B</i>	140
6.3. Evaluación del sistema: Conocimiento en Modelos de Diseño . . .	142
6.4. Evaluación del sistema: Conocimiento en Sistemas de Enseñanza .	143
6.5. Evaluación de sistema: UTILIDAD. ¿Qué opinión te merece el sistema con respecto a las siguientes características?	143
6.6. Evaluación del sistema: INTERFAZ. ¿Qué opinión te merece EL INTERFAZ DEL SISTEMA con respecto a las siguientes carac- terísticas?	144
6.7. Evaluación del sistema: INTERFAZ. Indica tu grado de satisfac- ción global con respecto al sistema	144
6.8. Evaluación del sistema: INTERFAZ. ¿Crees que se ajusta al mo- delo Entidad Interrelación?	145
6.9. Evaluación del sistema: CONTENIDO. ¿Piensa que ha visitado más o menos páginas de las necesarias para aprender los contenidos?	146
6.10. Evaluación del sistema: INTERFAZ. ¿Ha visitado páginas que no ha entendido?	146
6.11. Evaluación del sistema: CONTENIDO. ¿Ha entendido mejor los contenidos cuando iban acompañados de imágenes?	146
6.12. Evaluación del sistema: CONTENIDO. ¿Qué tipo de elemento le ha resultado más conveniente para comprender los contenidos? . .	147

6.13. Evaluación de RLATES: BOTÓN SIGUIENTE. ¿Le gusta que el sistema le guíe a través del botón de “siguiente” o preferiría que le diera más autonomía (navegar libremente en el árbol de conocimientos)?	148
6.14. Evaluación de RLATES: BOTÓN SIGUIENTE. Cuando pulsa el botón de ”siguiente”, ¿le gusta que el sistema le de varias opciones, o preferiría que le mostrara siempre un contenido directamente?148	
6.15. Evaluación de RLATES: BOTÓN SIGUIENTE. Al pulsar el botón de ”siguiente”,¿Ha elegido la opción que el sistema le sugería en primer lugar?	148
6.16. Evaluación de RLATES: TESTS. ¿Le ha resultado incómodo realizar todos los tests (exámenes) del sistema?	149
6.17. Evaluación de RLATES: ÁRBOL DE CONOCIMIENTO. ¿Le ha resultado útil el uso de colores en el árbol de conocimiento? . . .	149
6.18. Evaluación del sistema: ÁRBOL DE CONOCIMIENTO. Indica tu grado de satisfacción sobre el acceso al conocimiento mediante el árbol de conocimiento:	150
6.19. Evaluación del sistema: ÁRBOL DE CONOCIMIENTO. ¿Le gusta que el sistema proporcione autonomía (navegar libremente en el árbol de conocimientos) o preferiría que le guiara más por dicho árbol ofreciéndole sugerencias o pistas sobre qué es mejor que aprenda a continuación y en qué formato presentarlo?	150
6.20. Evaluación del sistema: ÁRBOL DE CONOCIMIENTO. ¿Le ha resultado útil el uso de colores en el árbol de conocimiento? . . .	150

Capítulo 1

Introducción

Gracias al auge que ha experimentado en los últimos años la aplicación de las nuevas tecnologías a la enseñanza y, en concreto, la aplicación de los servicios de Internet a la educación a distancia, se ha avanzado enormemente en el proceso de facilitar el acceso a la enseñanza para los estudiantes, mediante sistemas de educación via *Web*.

En este capítulo se presentará tanto la motivación como los objetivos de la presente tesis doctoral, donde se plantea la necesidad de desarrollar sistemas de educación adaptativos a través de Internet, de forma que actualicen sus políticas pedagógicas dependiendo de las características de aprendizaje de cada usuario. Así mismo, se presenta la estructura organizativa del documento.

1.1. Motivación

En los últimos años se ha contemplado un auge en la formas de enseñanza, derivadas principalmente de la investigación en nuevas tecnologías. Internet proporciona grandes beneficios frente a la enseñanza tradicional (en las aulas). Uno de los beneficios más importantes es que los alumnos no tienen que trasladarse a un aula física, sino que pueden aprender desde sus propias casas. Además, la enseñanza a distancia mediante Internet puede ser independiente de la plataforma software y hardware del ordenador desde donde se conectan los estudiantes al sistema.

Los *Sistemas de Educación Adaptativos e Inteligentes en Web* son sistemas de enseñanza a distancia mediante Internet que se adaptan de forma inteligente a las necesidades de aprendizaje de cada estudiante en cada instante de interacción con el sistema. Este tipo de sistemas pueden utilizar técnicas de inteligencia artificial para representar el conocimiento del experto e interactúan con los estudiantes con el objetivo de que éstos lo aprendan [VanLehn, 1988]. En el caso ideal de adap-

tación, el sistema se ha de comportar como si fuera un tutor humano enseñando a un único alumno en cada momento, donde el proceso de enseñanza es mucho más efectivo que enseñar un temario a un conjunto de alumnos como suele ocurrir en una clase magistral.

Para lograr la adaptación del sistema, éste almacena información sobre la materia de enseñanza (módulo del dominio), sobre las características personales y de aprendizaje del estudiante (módulo de estudiante), sobre la interacción con el estudiante (módulo del interfaz) y sobre las estrategias y tácticas pedagógicas que se pueden tomar para enseñar a los estudiantes en cada momento (módulo pedagógico) [Burns and Capps, 1988].

Mejorar la comunicación entre el sistema y el estudiante no es tarea sencilla, sino que implica la aplicación de diferentes técnicas y métodos de adaptación a los estudiantes, según sus características de aprendizaje. Normalmente, las técnicas y métodos de adaptación en este tipo de sistemas se basan en la aplicación de inteligencia artificial en cada uno de los módulos del sistema. En el modelo del dominio se han utilizado desde *modelos simbólicos* en [Fernández-Castro, 1989]) hasta *redes bayesianas* [VanLehn and Zhendong, 2001] para describir el conocimiento que el sistema ha de enseñar a los estudiantes. El comportamiento pedagógico del sistema se ha modelado habitualmente mediante técnicas de *planificación* [André *et al.*, 1993] y *sistemas basados en reglas* de producción [Anderson and Reiser, 1985] [Prentzas *et al.*, 2002]. En el modelo del interfaz se han aplicado técnicas de *procesamiento del lenguaje natural* [Alevén and Koedinger, 2000], técnicas de anotación en el interfaz [Stern, 2001] y técnicas de búsquedas de temas dentro del modelo del dominio [Brusilovsky *et al.*, 1996]. Por último, en el modelo de estudiantes se han utilizado desde *estereotipos* [Rich, 1979] y *modelos overlay* [Carr and Goldstein, 1977], hasta *redes bayesianas* [VanLehn and Zhendong, 2001] [Martin and VanLehn, 1993] entre otros.

Uno de los problemas más importantes en los sistemas de educación a distancia es secuenciar el contenido del sistema y presentar el contenido del curso de forma diferente según el estudiante, dependiendo de sus características de aprendizaje. La consecución de este tipo de adaptación al estudiante no es tarea sencilla, ya que el sistema ha de definir la estrategia pedagógica a seguir en cada momento, dependiendo del estudiante con el que interactúe. Las estrategias pedagógicas definen qué tipo de realimentación proporcionar en cada instante del proceso de enseñanza, cuándo mostrar el contenido del curso y cómo mostrarlo, cuándo explicar o resumir el contenido del sistema, etc. [Murray, 1999b]. Es decir, se entiende como estrategia pedagógica toda acción que realiza el tutor con el propósito de facilitar la formación y el aprendizaje de las disciplinas en los estudiantes.

La elección de la mejor estrategia pedagógica a seguir en cada instante de la interacción también es una tarea complicada, siendo necesario definir distintas estrategias pedagógicas según las necesidades de cada usuario y especificar para

cada una de ellas cuándo utilizarlas, sus diferencias, y si, tras elegir una determinada estrategia se comprueba que falla la elección, analizar por qué falló [Khan and Yip, 1996].

A lo largo de la historia de los sistemas de educación a distancia se han utilizado diversas técnicas de definición de estrategias pedagógicas (planificación, heurísticas, etc.), pero normalmente presentan dos problemas fundamentales:

- el primero de ellos es que se tienen que definir muchísimas estrategias pedagógicas para lograr adaptarse completamente a las necesidades de los estudiantes, por lo que definir cada una de ellas y codificarlas en el sistema es muy costoso
- el segundo problema viene derivado de la dificultad de incorporar el conocimiento de los tutores humanos mediante reglas, ya que los expertos puede que no sepan describir cómo debería razonar el sistema con ese conocimiento.

Por ello, en este trabajo se propone evitar la predefinición de las estrategias pedagógicas, evadiendo el costoso trabajo de incorporar el conocimiento del experto en el sistema de educación.

1.2. Fundamentos

Como se ha mencionado anteriormente, uno de los principales retos de los sistemas de educación a distancia es adaptar el curso a las necesidades de cada estudiante en cada momento. El problema fundamental es la gran diversidad y heterogeneidad de los alumnos que podrían interaccionar con el sistema. Esto implicaría diferentes características de aprendizaje y, por lo tanto, diferentes necesidades pedagógicas en el proceso de interacción con el sistema educativo.

Los sistemas de educación más avanzados utilizan técnicas de inteligencia artificial para proporcionar adaptación y personalización del sistema según las características de aprendizaje del estudiante. Además, si el sistema está basado en tecnología *Web*, éste puede ser utilizado por los estudiantes sin necesidad de desplazarse a ningún aula física, pudiendo utilizar cualquier tipo de plataforma hardware y software.

Tras el estudio de los sistemas de aprendizaje actuales, se ha comprobado que hasta el momento casi todos los sistemas de educación en *Web* tratan de encontrar estrategias adecuadas a cada usuario, definiendo estrategia como un *plan* de acción para mostrar los contenidos del tutor [André *et al.*, 1993].

El trabajo que aquí se presenta propone que el tutor aprenda a enseñar sin estrategia aparente, evitando el problema de la definición de estrategias predefinidas

de antemano detectado por [Beck, 2001]. Para ello se formula la hipótesis de que la definición de este problema como un problema de *Aprendizaje por Refuerzo* [Mitchell, 1997] en el módulo pedagógico del sistema de educación puede evitar la costosa predefinición de las tácticas de enseñanza. La utilización de aprendizaje por refuerzo permite definir tácticas de comportamiento óptimas basadas únicamente en experiencias adquiridas anteriormente con otros alumnos de características de aprendizaje similares al alumno que interacciona en cada momento con el sistema.

El *Aprendizaje por Refuerzo* plantea la cuestión de cómo un agente autónomo puede aprender a elegir las mejores acciones a ejecutar en cada instante con el objetivo de llegar a un estado meta [Mitchell, 1997]. Se trata de un modelo bastante genérico que cubre una gran cantidad de problemas de aprendizaje, como controlar un robot móvil, aprender a optimizar operaciones en fábricas, etc. Cada vez que el agente ejecuta una acción en su entorno, se le proporciona una señal de refuerzo (positiva, negativa o neutra) que indica lo bueno que ha resultado ejecutar esa acción cuando el agente se encontraba en un determinado estado. La tarea fundamental del agente será aprender de forma indirecta, a partir de los refuerzos recibidos, a elegir la secuencia óptima de acciones a ejecutar.

El modelo de *Aprendizaje por Refuerzo* aprende a enseñar a cada alumno basándose únicamente en la experiencia al enseñar a otros alumnos de características de aprendizaje similares. La aplicación de este modelo de aprendizaje suprime la tediosa tarea del experto en la materia (normalmente un tutor humano) de definir cada una de las estrategias pedagógicas que ha de seguir el sistema cuando interacciona con cada uno de los estudiantes, en cada situación que se pueda encontrar. Además, proporciona más flexibilidad al sistema, pudiendo encontrar nuevas tácticas de enseñanza que al tutor humano no se le habían ocurrido, decidiendo no sólo cuándo hay que utilizarlas, sino también cómo hacerlo y cómo solucionar el problema cuando la estrategia pedagógica seguida falla.

El sistema de educación definido como un problema de *Aprendizaje por Refuerzo*, por tanto, aprenderá a enseñar a los estudiantes mediante un proceso de prueba y error de forma paralela al aprendizaje del estudiante. Este método se apoya en la enseñanza tradicional, donde los tutores humanos aprenden a hacer frente a los alumnos a través de un largo proceso de prueba y error [Rea *et al.*, 2000]. En la enseñanza tradicional, [ReinSmith, 1992] comprobó que el tutor humano puede observar al estudiante a través de las expresiones de su cara, lenguaje del cuerpo, entonación y otros gestos. Es decir, los tutores humanos aprenden a hacer frente a los alumnos a través de un largo proceso de prueba y error, que sirven para encontrar un gran rango de estrategias de enseñanza. El sistema que se desea conseguir intenta, al igual que el sistema tradicional de enseñanza, aprender de sus errores proporcionando, tras un número suficiente de interacciones con usuarios, un comportamiento óptimo de enseñanza de conocimiento (una estrate-

gia óptima). El proceso de prueba y error en esta primera aproximación se basa en un modelo conductivista del usuario, es decir, se evalúa el conocimiento del mismo a partir de sus respuestas a unos tests de evaluación y su comportamiento histórico en el sistema.

1.3. Objetivos

La tesis doctoral que aquí se presenta tiene como objetivo principal la aplicación del modelo de *Aprendizaje por Refuerzo* para definir de forma implícita las tácticas pedagógicas en los *Sistemas de Educación Adaptativo e Inteligente en Web* (SEAI; del inglés *Adaptive and Intelligent Educational Systems: AIESs*).

Se entiende como táctica pedagógica la secuencia de acciones que realiza el tutor con el propósito de formar a los estudiantes, definiendo, entre otras cosas, cuándo mostrar el contenido del curso y cómo mostrarlo, cuándo explicarlo o resumirlo, cuándo presentar realimentación, etc. [Murray, 1999b]. En este trabajo se entiende como táctica pedagógica la secuencia de tareas (definiciones, introducciones, ejemplos, resúmenes, etc.) que forman parte del contenido del sistema de educación.

Al aplicar *Aprendizaje por Refuerzo* en los SEAI en *Web*, el sistema de educación aprenderá las tácticas pedagógicas de forma implícita, decidiendo qué tarea mostrar en cada momento y en qué formatos de presentación. Será capaz de adaptarse a las necesidades pedagógicas de cada estudiante en cada momento de la interacción con el sistema, solventando uno de los problemas más importantes y costosos en todo sistema de enseñanza: el análisis y la definición de las estrategias pedagógicas a seguir al interactuar con los usuarios para lograr una adaptación dinámica. El modelo de aprendizaje por refuerzo elimina el concepto de *estrategia pedagógica* que habitualmente se describen como reglas IF-THEN-ELSE, aprendiendo a comportarse basándose únicamente en la interacción con los usuarios.

En concreto, este proyecto presenta el módulo pedagógico del tutor como un problema de *Aprendizaje por Refuerzo*, centrándose en la tarea de proporcionar *Soporte Adaptativo a la Navegación*, incluyendo las tareas de *Secuenciar el Contenido* según las necesidades de aprendizaje de los estudiantes, guiando su navegación a través de las páginas *Web* de forma directa (a través de botones de navegación) y *Presentar el Contenido* según los formatos de presentación que mejor se adaptan al estudiante.

Se propone, por tanto, definir el problema de secuenciar los contenidos en los sistemas de educación a distancia como un problema de *Aprendizaje por Refuerzo*, de forma que el sistema aprende a enseñar de forma óptima a cada alumno, basándose únicamente en la experiencia adquirida con otros alumnos con carac-

terísticas de aprendizaje similares. Este tipo de adaptación a los estudiantes tiene ventajas con respecto a sistemas anteriores de educación a distancia, ya que, al adaptar el curso a cada estudiante de forma individual, éste mejorará su aprendizaje obteniendo mejores resultados en un tiempo menor de enseñanza. Además, el sistema aprenderá a comportarse mediante un proceso de prueba y error de forma paralela al aprendizaje del estudiante, tal y como aprenden los tutores humanos en la enseñanza tradicional [Rea *et al.*, 2000].

También será necesario validar la propuesta, comprobando la utilidad de la aplicación del modelo de *Aprendizaje por Refuerzo* en sistemas informáticos educativos, tanto de forma teórica como práctica (interaccionando en las aulas, con alumnos reales). Una vez comprobada la aplicabilidad de la propuesta, es necesario ajustar los parámetros de aprendizaje del sistema según las necesidades pedagógicas en un sistema de educación *Web* determinado.

En este trabajo se pretende principalmente dar respuesta a las siguientes cuestiones:

1. ¿Se pueden representar las tácticas pedagógicas del sistema mediante el modelo de aprendizaje por refuerzo? Para ello, se comprobará si cada uno de los componentes del modelo de aprendizaje por refuerzo se pueden adaptar al dominio de los sistemas de educación a distancia, definiendo las características de cada componente.
2. ¿Se puede abordar el sistema desde un punto de vista teórico? Para poder demostrar si el problema es abordable desde un punto de vista teórico se comprobará si el sistema es capaz de aprender tácticas pedagógicas buenas para cada estudiante con el que interactúa, simulando un comportamiento predefinido de cada tipo de estudiante.
3. ¿Se puede abordar de forma práctica? Finalmente, para comprobar que el problema se puede abordar desde el punto de vista práctico (comprobar si el sistema puede implantarse en sustitución de las aulas físicas), se realizarán experimentos con estudiantes reales, analizando si el sistema es capaz de converger a buenas tácticas pedagógicas interaccionando con un número no muy grande de estudiantes.

Como resumen, por tanto, se fijarán los siguientes objetivos de la tesis doctoral:

- Definir el problema de *secuenciar y elegir el formato de presentación* del contenido del sistema de educación como un problema de *Aprendizaje por Refuerzo*. Para ello, se comprobará la aplicación del problema en cada uno de los componentes del modelo de *Aprendizaje por Refuerzo*. Esta definición deberá ser general, independiente del dominio de enseñanza del sistema de educación.

- Que el sistema de educación en *Web* se comporte de forma *inteligente*, sin necesidad de predefinir estrategias pedagógicas. Es decir, que el tutor aprenda a partir de la interacción con los distintos estudiantes a mejorar la forma de mostrarles la materia dependiendo de sus necesidades en cada instante. Para ello, ha de proporcionar al sistema la capacidad de aprender por sí mismo tácticas pedagógicas para cada estudiante en cuanto a la navegación a través de los contenidos del tutor y en cuanto a la elección del formato de presentación de dichos contenidos.
- Validar la propuesta realizada basándose principalmente en dos cuestiones: ¿el soporte adaptativo a la navegación en sistemas de enseñanza se puede definir como un problema de aprendizaje por refuerzo de forma teórica? y, una vez comprobado que se puede aplicar de forma teórica ¿es aplicable en la realidad, en las aulas, cuando interaccionan alumnos con el sistema?
- Estudiar el efecto de los parámetros del sistema en el aprendizaje cuando estudiantes simulados interaccionan con un sistema de educación determinado. En concreto, se estudiará la propuesta sobre un sistema de enseñanza de *Diseño de Bases de Datos*.

En este punto, es conveniente destacar que no forman parte de los objetivos de la presente tesis la investigación en los módulos del estudiante, del dominio, ni del interfaz de usuario. En esta tesis se presenta una propuesta de representación implícita de las tácticas pedagógicas de un sistema de educación mediante el modelo de aprendizaje por refuerzo. Con el objetivo de validar el sistema se implementarán todos y cada uno de los módulos del sistema de educación, utilizando técnicas y métodos previamente validados por otros autores en cada uno de los módulos.

1.4. Organización de la Memoria

Esta sección presenta la organización de la memoria, que se ha dividido en varios capítulos. El capítulo 2 revisa del estado de la cuestión sobre los *Sistemas de Educación Adaptativos e Inteligentes en Web*, métodos y técnicas de adaptación que se han aplicado hasta el momento y sobre la técnica de inteligencia artificial que se aplicará posteriormente en el sistema: el *Modelo de Aprendizaje por Refuerzo*.

En el capítulo 3 se presenta la propuesta de la tesis doctoral, definiendo el problema de *Soporte Adaptativo a la Navegación* de los SEAIs como un problema de *Aprendizaje por Refuerzo*.

El capítulo 4 trata de aspectos previos a la presentación de la experimentación, donde se definirán conceptos introductorios a la evaluación de un sistema de estas características, y se presentarán el método de evaluación y los módulos del dominio que se utilizarán para realizar la validación del sistema. Posteriormente, el capítulo 5 analiza los experimentos realizados con estudiantes simulados con el objetivo de probar si la propuesta es aplicable de forma teórica, ajustar los parámetros de aprendizaje del sistema y exponer las ventajas de la inicialización del sistema con información sobre el dominio. En el capítulo 6, tras haber demostrado en el capítulo anterior que la propuesta es aplicable desde el punto de vista teórico, se validará la propuesta en las aulas, interaccionando con estudiantes reales.

Por último, en el capítulo 7 se presentan las conclusiones y principales aportaciones de la tesis doctoral, así como líneas futuras de investigación que podrían ampliar este trabajo.

Capítulo 2

Estado de la Cuestión

En este capítulo se presenta el estado de la cuestión sobre programas de enseñanza por ordenador, centrándonos principalmente en los *Sistemas de Educación Adaptativos e Inteligentes en Web*, que surgieron a partir de los *Sistemas Tutores Inteligentes* y los *Sistemas Hipermedia Adaptativos*. A continuación, se dedica especial atención a la tarea de *Secuenciar el Contenido* en el módulo pedagógico de los sistemas educativos. Posteriormente se estudian métodos y técnicas de inteligencia artificial aplicadas a sistemas de enseñanza, profundizando en los sistemas en los que se ha aplicado anteriormente el modelo de *Aprendizaje por Refuerzo*. Por último, se presenta el estado de la cuestión del modelo de *Aprendizaje por Refuerzo*.

2.1. Programas de Enseñanza

Gracias a las nuevas tecnologías, los planteamientos educativos han ido evolucionando a lo largo del tiempo, utilizando cada vez más los ordenadores como herramientas de apoyo a la enseñanza, permitiendo incluso ofrecer enseñanza a distancia.

Uno de los principales problemas de los programas de enseñanza es la heterogeneidad de los alumnos que interactúan con el sistema. Es necesario que el estudiante se sienta en todo momento cómodo con el sistema y, para ello, se plantea un aprendizaje centrado en el alumno, adaptando los contenidos del programa de enseñanza a sus características de aprendizaje. De igual forma, se debería evitar que el alumno pierda un tiempo excesivo en el aprendizaje de la herramienta, planteando un interfaz sencillo, pero potente en su uso.

2.1.1. Repaso Histórico

Desde 1960 se están estudiando nuevos métodos de enseñanza o de ayuda a la enseñanza apoyándose en las nuevas tecnologías (ordenadores, comunicaciones, etc.). Los programas de soporte a la enseñanza mediante ordenadores han avanzado de la siguiente forma hasta la actualidad [Sison, 2001]:

- En los años 60 surgieron los *Tutores Basados en Ordenadores* (CBT: Computer-Based Tutors) [Corbett and Anderson, 1992]. Estos programas automatizaban las instrucciones y cursos que se le mostraban a los usuarios. Es decir, se predefinía la secuencia de definiciones, explicaciones, ejemplos, ejercicios o preguntas relacionadas con el dominio particular en el que se enmarcaba el curso. Estos conocimientos se mostraban siempre en el mismo orden y de la misma forma, independientemente del usuario y sin ajustarse a sus necesidades en cada instante.
- Posteriormente, en los años 70, aparecieron los *Entornos de Aprendizaje Basados en Ordenadores* (CBLE: Computer Based Learning Environments) [Dillenbourg, 1999], que permitían al usuario aprender conceptos o habilidades mediante descubrimientos. En este caso, estos programas más que ofrecer instrucciones o cursos, presentaban micromundos de objetos (del inglés *microworlds*), que se comportaban de acuerdo con las reglas pre-programadas y que los usuarios podían manipular para jugar con ellas. De esta forma permitían que cada estudiante tomara parte del proceso de enseñanza, decidiendo él mismo en qué orden deseaba aprender los conceptos. Este tipo de sistemas permitían que el estudiante se sintiera más interesado por el curso, ya que iba descubriendo poco a poco todo el conocimiento según se le antojaba; pero también tenía desventajas como, por ejemplo, que el estudiante pudiera perderse en el micromundo, intentando aprender conocimientos muy complicados sin conocer primero los conceptos más simples. Esto solía llevarle al desánimo, que normalmente acababa en el abandono del curso.
- En los 80 aparecieron dos tipos de sistemas, los *Tutores Inteligentes* y los *Entornos de Aprendizaje Inteligente*:
 1. *Tutores Inteligentes* (ITS: Intelligent Tutoring Systems) [Sleeman and Brown, 1982] [Burns *et al.*, 1991] [Brusilovsky *et al.*, 1996] [Murray, 1999a]. La diferencia principal con respecto a los CBTs es que éstos permiten diagnosticar y remediar errores de sus alumnos, es decir, almacenan propiedades de cada estudiante, por ejemplo grados de certeza del conocimiento y conocimiento mal aprendido, objetivos, preferencias, intereses, aptitudes de aprendizaje, idiosincrasias, etc. del

estudiante. A estas representaciones se las denominan *Modelos de Estudiante*. Gracias a esta información el sistema posee la capacidad de diagnóstico que permite, a través del uso de aproximación de funciones cualitativas, posiblemente parciales, la adaptación a cada usuario en cada instante.

2. *Entornos de aprendizaje Inteligente* (ILE: Intelligent Learning Environment) [White and Frederiksen, 1990]. Al igual que los ITS, utilizan los modelos de estudiante para determinar qué retro-alimentación (*feedback*) o asistencia proporcionar al estudiante. Pero, a diferencia de los ITSs, el papel del los ILE es crear un entorno efectivo de aprendizaje, donde el estudiante “descubre” de forma guiada el nuevo conocimiento, sin adoptar el papel de un tutor. Los ILEs también se suelen ver como entrenadores, pero no como tutores.
- En los 90, gracias a las nuevas tecnologías de comunicación (sobre todo basado en Internet), han surgido varias aplicaciones basadas en Internet: los *Entornos de Aprendizaje Virtual* y los *Tutores Inteligentes Basados en Internet*.
1. Los *Entornos de aprendizaje Virtual* (VLE: Virtual Learning Environment) [Landon, 2000] son colecciones de herramientas (e-mail, grupos de discusión, facilidades de conversación (chats), editores, librerías/repositorios, formularios de exámenes/resúmenes, contadores de frecuencia, motores de búsqueda, etc.) utilizados para la presentación *online* de cursos y para la comunicación entre distintos ordenadores (CMC: Computer-Mediated Communication), y, por lo tanto, comunicación entre distintos estudiantes. Se diferencian de los ILE en que los VLE necesitan profesores humanos para que dirijan y monitoricen el aprendizaje, mientras que los ILE y los ITS son totalmente autónomos en ese aspecto.
 2. Los Sistemas Inteligentes de Aprendizaje basados en Internet (IILS: Internet based Intelligent Learning Systems) [Sison, 2001], también denominados *Tutores Inteligentes en Web*, surgen a partir de los ITS, facilitando a cada estudiante, individualmente, el aprendizaje de cursos a través de Internet. En estos sistemas, los estudiantes aprenden mediante instrucciones o guiados por descubrimiento, con o sin ayuda de tutores humanos (ellos eligen). En estos programas existen dos tipos de tutores, el software y el humano, que podrían descubrir nuevo conocimiento sobre el dominio o sobre métodos pedagógicos efectivos sobre ese dominio en particular. Estos sistemas se diferencian de los VLE en que, al igual que los ILE, no necesitan tutores humanos en

su funcionamiento (pueden funcionar de forma autónoma en este aspecto), aunque sí los pueden utilizar si se desea. Uno de los primeros ITS basados en *Web* es CALAT [Nakabayashi *et al.*, 1997], que incluía un entorno de simulación como parte del proceso de enseñanza, animación y presentaciones de los contenidos en texto.

- En la actualidad se están estudiando los *Sistemas de Educación Adaptativos e Inteligentes Basados en Web* (AIES: Web-based Adaptive and Intelligent Educational Systems), que son sistemas no del todo nuevos, ya que se derivan históricamente de los *Tutores Inteligentes en Web* (IILS) y de los *Sistemas Hipermedia Adaptativos* (AHS: Adaptive Hipermedia Systems) [Brusilovsky, 1999]. Estos sistemas son los más completos hasta el momento, ya que se estudia por la parte de IILS el conocimiento sobre el dominio, el estudiante y sobre las estrategias de enseñanza que soporten la individualización de la enseñanza y la flexibilidad; y por la parte de AHS, la aplicación de las diferentes formas de modelos de usuarios para adaptar el contenido y los enlaces de las páginas hipermedia que se han de mostrar al usuario (presentación adaptativa y soporte a la navegación adaptativa).

2.1.2. Sistemas de Educación Adaptativos e Inteligentes en *Web*

Como se ha introducido en el apartado anterior, los AIES surgen a partir de los Tutores Inteligentes y de los Sistemas Hipermedia Adaptativos, por lo que las características y problemas asociados a este nuevo tipo de sistemas se derivan de las características y problemas de sus antecesores. De este modo, los desarrolladores de AIES pueden confiar en las tecnologías utilizadas por los ITS y los AHS [Brusilovsky, 1999]. Es por ello que en los siguientes apartados se trata de dividir el desarrollo de los AIES estudiando individualmente los Tutores Inteligentes en la sección 2.1.4 y los Sistemas Hipermedia Adaptativos en la sección 2.1.3.

Los límites de los AIES no son claros [Brusilovsky, 1999], ya que sistemas *online* como, por ejemplo, una enciclopedia utilizada para la educación [Signore *et al.*, 1997] o un sistema de filtrado adaptativo aplicado a un contexto educativo [Nomoto *et al.*, 1997] son considerados por algunos autores como sistemas educativos. Sin embargo, otros autores no, pidiendo algún tipo de inteligencia y adaptación en el sistema.

Los primeros sistemas AIES basados en *Web* aparecieron por primera vez en 1995 y surgieron a partir de Tutores Inteligentes, como por ejemplo, ELM-ART [Brusilovsky *et al.*, 1996], CALAT [Nakabayashi *et al.*, 1998], WITS [Ito *et al.*, 1998] o Belvedere [Suthers and Jones, 1997]. Posteriormente surgieron varias líneas de investigación, como la adaptación del contenido de las páginas a cada

usuario (Adaptación Hipermedia) [Kay and Kummerfeld, 1997], [Bra and Calvi, 1998], [Stern, 2001] y adaptación colaborativa en *Web* [Leigh *et al.*, 1997], [Verdejo *et al.*, 2002], [Boticario *et al.*, 2001], entre otros.

Los beneficios de la educación a distancia utilizando Internet son claros [Brusilovsky, 1999]. Por un lado, una de las principales ventajas es que la enseñanza se imparte fuera de un aula física (lo único que necesita cada estudiante es un ordenador conectado a Internet, independientemente del lugar físico en el que se encuentre), evitando así los inconvenientes del desplazamiento de los estudiantes y tutores. Por otro lado, al ser una enseñanza por Internet, es independiente también de la plataforma (hardware o software) en la que se encuentre cada ordenador.

Por lo tanto, de esta forma, para la enseñanza de un determinado curso o habilidad se necesita una única aplicación instalada y mantenida en un único sitio, que podrá ser utilizada por varios usuarios al mismo tiempo a lo largo de todo el mundo, equipados con distintos ordenadores conectados a Internet.

Los objetivos principales de los sistemas de enseñanza basados en *Web* son la *adaptación* individual a cada usuario en cada instante y la *interactividad*. Hasta llegar a ellos, la mayoría de los cursos formativos en *Web* solían ser simplemente páginas estáticas de hipertexto, sin que se *adaptan individualmente* a cada usuario que esté aprendiendo y no solían permitir *interactividad*, siendo el sistema el que tomaba siempre la iniciativa del aprendizaje. Al permitir la interactividad, el usuario se suele sentir más atraído hacia la información que puede ofrecer el sistema, evitando así que se sientan aburridos.

2.1.3. Sistemas Hipermedia Adaptativos en la Enseñanza

Los Sistemas Hipermedia Adaptativos (AHS: Adaptive Hipermedia Systems) son sistemas que proporcionan una progresión no lineal a través del material. En los sistemas Hipermedia anteriores, esta propiedad implicaba el problema de que los estudiantes se solían perder en el hiperespacio sin saber qué estudiar a continuación y cómo recordar dónde habían estado [Stern, 2001]. Los nuevos Sistemas Hipermedia se diferencian de los anteriores en que añaden un componente inteligente para adaptarse a los usuarios y ayudarlos. Su principal dominio de investigación se basa en la adaptación de:

1. **Los contenidos** de las páginas *Web* según el usuario que interacciona en cada momento. Las técnicas de adaptación del contenido más utilizadas son:
 - **Texto Condicional:** donde se incluyen partes del texto en la página que se muestra sólo cuando se cumple una condición. Sistemas como C-Book [Kay and Kummerfeld, 1997] y el curso adaptativo de Hipertexto AHA! [Bra and Calvi, 1998], utilizan este tipo de técnicas.

- **Variantes:** donde se tienen múltiples formas de describir un ítem determinado. El modelo de usuario será capaz de distinguir entre estas variantes y elegir la correcta en el momento adecuado. Los sistemas ANATOM-TUTOR [Beaumont, 1998] y AVANTI [Fink *et al.*, 1998], por ejemplo, utilizan esta técnica.
 - **Texto extendido (del inglés *Stretchtext*):** técnica que reemplaza una palabra o frase clave por información adicional de esa frase. Se modifica en base al modelo de usuario. iManic [Stern, 2001] y MetaDoc [Boyle and Encarnación, 1998], por ejemplo, utilizan esta técnica.
 - **Generación de lenguaje natural:** Se trata de un área de investigación dentro de la adaptación de los contenidos. Se trata de incorporar el modelo de usuario en decisiones de cómo generar el texto. Sistemas como ILEX-0 [Knott *et al.*, 1996], iManic [Stern, 2001], el sistema de O'Donnell [O'Donnell, 1997] o ARIANNA [Carolis *et al.*, 1998] utilizan esta técnica. Otra forma de generar el contenido es individualizar los exámenes que se realizan a los alumnos, como hace, por ejemplo, el sistema TANGOW [Carro *et al.*, 2001] o InterBook [Brusilovsky *et al.*, 1998]. Otros sistemas como Medtec [Eliot *et al.*, 1997] son capaces de generar resúmenes adaptativos de capítulos de libros. También existen sistemas que proporcionan advertencias (del inglés *warnings*) adaptativas sobre el estado educativo de una página (por ejemplo, cuando una página ya ha sido mostrada anteriormente, o no se puede mostrar aún).
2. **La navegación** a través de las páginas hipermedia según el usuario que interactúa con el sistema, por ejemplo, cambiando la apariencia de los enlaces visibles, ordenándolos o escondiéndolos parcialmente para hacer más sencilla la elección del siguiente enlace. Habitualmente, la navegación que se suele hacer en los diferentes sistemas son decisiones muy simples del tipo “si el usuario ha visto el tema x , entonces está listo para aprender el tema y ”. Pero esto no es correcto, habría que distinguir entre “ver” y “conocer” un tema [Stern, 2001]. El soporte adaptativo a la navegación (ANS: *Adaptive Navigation Support Technology*) puede considerarse como una extensión de la tecnología de *secuenciar el contenido* en los tutores inteligentes, ya que comparten la misma meta: ayudar a los estudiantes a encontrar un “camino óptimo” a través del material de aprendizaje. Pero hay que tener en cuenta que ANS tiene más opciones que secuenciar de forma tradicional. Puede guiar al estudiante mediante:
- **Guía directa:** Guía al estudiante de forma explícita, indicándole dinámicamente cuál es el siguiente enlace (generalmente denominado “*next*”)

que debería pulsar si quisiera seguir el “mejor camino”. Similar a secuenciar el contenido tradicional. Sistemas como InterBook [Brusilovsky *et al.*, 1998] y ELM-ART [Brusilovsky *et al.*, 1996] utilizan esta técnica de adaptación.

- **Guía indirecta:** Trata de guiar al estudiante implícitamente mediante el color o la ordenación de los enlaces en cada página. Las técnicas de guía indirecta más utilizadas son:
 - **Anotación:** Es la técnica más popular, donde se puede anotar mediante distintas formas (textual, iconos, colores, tamaños de fuente o tipos de fuente) si se ha visitado ya un enlace o no. Sistemas como InterBook [Brusilovsky *et al.*, 1997], AST [Specht *et al.*, 1997], ADI [Schöch *et al.*, 1999], ACE [Specht and Oppermann, 1998], ELM-ART [Brusilovsky *et al.*, 1996], ISIS-Tutor [Brusilovsky and Pesin, 1994], iManic [Stern, 2001], entre otros, utilizan esta técnica.
 - **Ordenación:** Trata de ordenar físicamente en la página los enlaces según el modelo del usuario (normalmente basados en el conocimiento o en el comportamiento). Por ejemplo, poniendo los enlaces más relevantes al principio de la página. Los sistemas ISIS-Tutor [Brusilovsky and Pesin, 1994], InterBook [Brusilovsky *et al.*, 1998] e Hypadapter [Hohl *et al.*, 1996] e iManic [Stern, 2001], por ejemplo, utilizan esta técnica.
 - **Ocultación:** Restringe el espacio navegable a las páginas no relevantes, que no tenga que ver con los objetivos del usuario actual, o que contenga materiales que todavía el usuario no está preparado para entender. Esta técnica está implementada en sistemas como AHA! [Bra and Calvi, 1998], ISIS-Tutor [Brusilovsky and Pesin, 1994], Hypadapter [Hohl *et al.*, 1996], el sistema “Remedial Multimedia System” [Anjaneyulu, 1997] y Albatros [Lai *et al.*, 1995]. Otra aproximación de esta técnica es esconder documentos (en lugar de enlaces), utilizada por iManic.
 - **Añadir enlaces dinámicos:** Añade nuevos enlaces a las páginas de forma dinámica, basándose en las diferencias individuales de los alumnos. Los sistemas de Kushniruk y Wang [Kushniruk and Wang, 1994], DynaWeb [Bodner *et al.*, 1997], KBS Hyperbook [Henze and Nejd, 1999] e iManic [Stern, 2001] utilizan esta técnica.
 - **Guía:** Proporciona al usuario una vista del hiperespacio, indicando los enlaces que existen entre las diferentes páginas, de forma que él mismo pueda elegir dónde ir. Se suele generar dinámica-

mente una secuencia de enlaces o caminos de acuerdo al usuario actual. Sistemas como TANGOW [Carro *et al.*, 1999], ADAPTS [Brusilovsky and Cooper, 1999], ACE [Specht and Oppermaun, 1998] o iManic [Stern, 2001] utilizan esta técnica de adaptación al usuario, almacenando información sobre el estudiante y sus preferencias sobre el lenguaje, interfaz, anotaciones personales y diálogo directo, sobre las unidades aprendidas, etc.

La educación es una de las principales áreas de investigación de los AHS, pero no la única. Existen también otros sistemas, como AVANTI [Fink *et al.*, 1998], PUSH [Höök, 1997] o IfWeb [Asnicar and Tasso, 1997], que permiten la adaptación del contenido de las páginas *Web* según el usuario que interactúa a lo largo del tiempo, pero no se dedican al área de la enseñanza.

Aunque todas estas técnicas se han probado sobre todo en laboratorios, un gran número de sistemas de prueba han demostrado claramente los beneficios de tecnologías adaptativas e inteligentes [Brusilovsky, 1999]

2.1.4. Tutores Inteligentes

Un tutor inteligente (ITS: *Intelligent Tutoring System*) es un programa de ordenador que utiliza técnicas de inteligencia artificial para representar el conocimiento e interactuar con los estudiantes [Clancey, 1987] [VanLehn, 1988].

Los ITS son un área tradicional de investigación de problemas de los AIES [Burns and Capps, 1988]. El objetivo de los ITS es la utilización de conocimiento sobre el dominio, sobre el estudiante y sobre estrategias de enseñanza para soportar aprendizaje y enseñanza flexible e individual a cada estudiante. Se distinguen varios tipos de problemas típicos de los ITS, algunos de los cuales se citan a continuación [Burns and Capps, 1988] [Brusilovsky, 1998]:

- Secuenciar el contenido (*del inglés Curriculum Sequencing*)
- Modelado de estudiantes (*del inglés Student Modelling*)
- Análisis inteligente de las soluciones de los estudiantes (*del inglés Intelligent analysis of student's solutions*)
- Soporte a la resolución interactiva de problemas (*del inglés Interactive problem solving support*)
- Soporte a la resolución de problemas basada en ejemplos (*del inglés Example based problem solving support*)

La estructura típica de un tutor inteligente se compone de cuatro módulos bien diferenciados [Burns and Capps, 1988] (ver Figura 2.1):

- **Módulo de estudiante.** Contiene toda la información importante sobre el usuario en su proceso de aprendizaje: grado de conocimiento sobre el dominio del ITS, características personales, comportamiento histórico, objetivos, preferencias de presentación del contenido, aptitudes de aprendizaje, etc.
- **Módulo de interfaz.** Facilita la comunicación entre el sistema y el estudiante. Para ello, ha de cumplir ciertas características, como, por ejemplo, ser fácil de utilizar, ser intuitivo y no ambiguo, etc.
- **Módulo del dominio.** Denominado también por muchos autores como *módulo experto* debido a su semejanza con los sistemas expertos. Contiene todas las características del conocimiento que se va a enseñar (*materia (topics)* a enseñar, *tareas (items de conocimiento)*, relaciones entre ellos, grado de dificultad, etc.).
- **Módulo pedagógico.** Decide qué, cómo y cuándo enseñar los contenidos del tutor, adaptando sus decisiones pedagógicas a las necesidades del estudiante. Algunos autores le denominan *módulo tutor*, ya que es el encargado de comparar las características de los estudiantes con el contenido a enseñar y elegir la mejor forma de tomar las decisiones pedagógicas oportunas, adaptándose en cada momento al usuario.

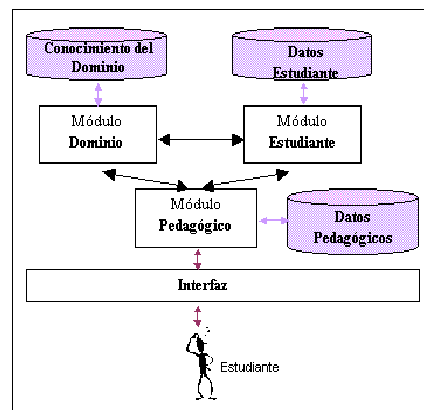


Figura 2.1: Arquitectura de un Tutor Inteligente.

La tesis que se presenta en este documento se centra en la mejora del primer problema citado anteriormente: cómo secuenciar el contenido del tutor inteligente adaptándose en cada momento al usuario que interactúa con el sistema, que es la principal función del módulo pedagógico. Los siguientes apartados, por tanto, se centran en este módulo, explicando la tarea de secuenciar el contenido del sistema.

2.2. Políticas Pedagógicas

Es importante en esta memoria dedicarle especial interés al módulo pedagógico, ya que el objetivo de la tesis doctoral trata de mejorar la adaptación del sistema según las necesidades individuales de aprendizaje de cada estudiante. Una forma de adaptar la enseñanza al estudiante trata de la elección de una buena estrategia de enseñanza (estrategia pedagógica), que especifica cómo se secuencian el contenido, qué tipo de realimentación se proporciona en la enseñanza, cuándo y cómo se muestra, explica o resume el contenido del tutor (problemas, definiciones, ejemplos, etc.) [Murray, 1999a].

La elección de una buena política de enseñanza no es tarea sencilla. Es necesario definir distintas estrategias según las necesidades de cada usuario en cada instante y especificar en qué situaciones se diferencian [Khan and Yip, 1996], con la problemática añadida de si falla en la elección, saber por qué.

A lo largo de la historia de los tutores inteligentes se han realizado muchas investigaciones sobre la elección de la mejor estrategia a seguir en cada instante, pero el coste de hallarlas ha sido siempre muy elevado. Desde el sistema PMS [Sleeman, 1977], donde el usuario es el responsable de su propia enseñanza, eligiendo la siguiente acción a tomar; hasta sistemas donde se han incorporado técnicas de inteligencia artificial [Sison and Shimura, 1998] para resolver los problemas derivados de la elección de las estrategias pedagógicas, como por ejemplo, utilización de redes semánticas en MENO-TUTOR [Woolf, 1987], redes neuronales en UNIMEM [Lebowitz, 1987], etc., o incluso aprendizaje por refuerzo para el modelado de usuarios en el sistema ADVISOR [Beck, 2001].

Es imprescindible para la elección de políticas pedagógicas la continua evaluación del nivel de conocimiento del estudiante, mediante análisis de pruebas de evaluación de conocimiento (tests) que se realizan durante la interacción con el sistema. Estos tests permiten observar la evolución del estudiante al adquirir los conocimientos impartidos.

Pero también es importante en la elección de las políticas pedagógicas cuidar que el usuario se encuentre a gusto con el sistema, motivado con la materia enseñada y, sobre todo, que note que pueda aprender del tutor. Está demostrado [Shah, 1997] que uno de los motivos principales por los que un usuario no se encuentra a gusto con un tutor inteligente es cuando siente que no posee el control, sino que lo tiene la máquina. Por ello, es imprescindible que el usuario pueda participar activamente en la elección del contenido que desea ver a continuación.

2.2.1. Secuenciar el Contenido

Se entiende como contenido o *curriculum* de un sistema de educación a todos las unidades de conocimiento de estudio ofrecidos por el sistema. El objetivo

de la tarea de *Secuenciar el Contenido* (del inglés *Curriculum Sequencing*) es proporcionar la mayor adaptabilidad individual posible al estudiante en cuanto a la enseñanza de las unidades de conocimiento que desea aprender [Brusilovsky, 1998]. A esta tarea también la denominan algunos autores *Planificación Instruccional* (*Instructional Planning*).

Es, por tanto, objetivo de esta tecnología encontrar para cada estudiante la mejor manera de mostrarle la información que desea aprender, lo que significa, en otras palabras, encontrar la secuencia de *conocimiento* (*materia*, *conceptos del dominio* o *topics*) que mejor se adapte a sus necesidades, así como la mejor secuencia de *tareas de aprendizaje* (*ítem de conocimiento*: ejemplos, cuestiones, problemas, etc.) para trabajar con ellas. Concluyendo, ayuda al estudiante a encontrar el *camino óptimo* a través del aprendizaje del material que se dispone, motivándolos de forma que crean que pueden superar el curso, sin aburrirlos o hacerles perder el tiempo mostrándoles conocimiento que ya poseen [Cho, 2000].

Los términos *materia*, *ítem de conocimiento*, etc., son términos generales elegidos a propósito, ya que estas técnicas de planificación (o secuencias) se pueden aplicar a todo tipo de dominios (bases de datos, matemáticas, cualquier lenguaje de programación, etc.) o cualquier tipo de conocimiento (declarativo, procedimental, etc.).

La tecnología de *secuenciar el contenido* puede ser considerada la más antigua y popular de las tecnologías de los ITS, ya que todos los sistemas han de mostrar el contenido adaptándose al usuario y, por tanto, se implementó ya en los primeros tutores inteligentes. Sin embargo, se le ha prestado poca atención [Brusilovsky *et al.*, 1996]. Las investigaciones en los ITS se han centrado casi siempre alrededor de las tecnologías de la resolución de problemas, modelos de usuarios y creación de modelos de contenidos. En un principio, se pensaba que secuenciar el contenido no dependía del sistema, sino que dependía más del conocimiento externo (del experto en la materia) que normalmente tenía el profesor real, por lo que no se estudiaba en profundidad. En la actualidad, a la hora de hablar de tutores en *Web*, esta tarea vuelve a adquirir importancia para que el estudiante pueda ser guiado a través del hiperespacio, evitando así que se pierda con tanta información como existe en la *Web*.

Por ello, en esta tesis se retoma el problema, intentando aplicar la mejor secuencia del contenido a cada estudiante. En esta sección se estudian las distintas clasificaciones y aproximaciones para secuenciar el conocimiento realizadas por otros autores y la problemática de la elección de las distintas políticas pedagógicas.

2.2.2. Taxonomía de la Tarea de Secuenciar el Contenido

La tarea de secuenciar el material a mostrar al estudiante se puede clasificar fijándose en varios parámetros. Las clasificaciones más habituales son las que distinguen entre metas de aprendizaje conocidas o desconocidas y niveles de secuencia en el árbol de conocimiento.

El sistema BIP [Bart *et al.*, 1976] distingue dos formas diferentes de secuenciar el contenido:

- De forma **Activa**: Implica una *meta de aprendizaje* conocida (subconjunto de temas que se han de enseñar). Los sistemas que realizan este tipo de secuencias pueden construir el mejor camino individual para conseguir la meta requerida. Dentro de estos sistemas, tiene sentido distinguir los sistemas que poseen metas de aprendizaje fijas o ajustables. La mayoría de los sistemas los guían hacia metas fijas (que suele coincidir con la totalidad de los conceptos del sistema). Sólo muy pocos sistemas permiten la meta de aprendizaje ajustable, permitiendo decidir al estudiante o al profesor el subconjunto de conceptos que necesita aprender. Al secuenciar de forma activa, el sistema se guía normalmente por el conocimiento de los estudiantes (por diferencia con el conocimiento meta que desean adquirir).
- De forma **Pasiva**: También se la denomina **atrasada** (*remediation*). Se trata de una tecnología reactiva y no requiere un objetivo de aprendizaje activo. Comienza cuando un usuario no es capaz de resolver un problema o cuestión de forma correcta. Su meta es ofrecer al usuario un subconjunto de material de aprendizaje que pueda ayudar al usuario a comprender su error.

La mayoría de los sistemas que secuencian de forma activa (*ELM-ART-II* [Weber and Specht, 1997], *AST* [Specht *et al.*, 1997], *ADI* [Schöch *et al.*, 1999], *ACE* [Specht and Oppermann, 1998], etc.), y solamente algunos sistemas, como *InterBook* [Brusilovsky *et al.*, 1998], *PAT-InterBook* [Brusilovsky *et al.*, 1997], *CALAT* [Nakabayashi *et al.*, 1997] y *Remedial Multimedia System* [Anjaneyulu, 1997] secuencian de forma pasiva.

Otra clasificación que se suele hacer de la tarea de secuenciar el contenido es la distinción entre los posibles niveles de secuencia:

- **Alto nivel**, denominado también **secuenciadores de conocimiento** o **Planificadores de Currículum** (del inglés *Curriculum Planning*). Se encargan de elegir el siguiente conocimiento (topic/s, concepto/s o lección/es) a enseñar: la siguiente sub-meta a alcanzar (aprender) [Cho *et al.*, 1999] [Cho, 2000]. Se encargan, sobre todo, de guiar al usuario para que realice de forma correcta distintos problemas.

- **Bajo nivel**, denominado también **secuenciadores de tareas** o **Planificadores del Discurso** (del inglés *Discourse Planning*). Encargados de seleccionar y secuenciar el conocimiento cuando el usuario ya está realizando el problema. Controlan la presentación de los ítems de conocimiento (la siguiente tarea de aprendizaje que se ha de realizar: problemas, ejemplos, tests, etc.) a los estudiantes [Freedman, 1999].

En la mayoría de los tutores inteligentes se suele implementar de forma inteligente una única forma de secuenciar [Brusilovsky, 1998]. Por ejemplo, el estudiante elige el siguiente concepto a aprender, mientras que el sistema se adapta seleccionando las tareas a enseñar. Si el tutor inteligente es capaz de manipular el orden en el que el estudiante ha de realizar las tareas según el tipo (problema, ejercicio, test, etc.) o formato (texto, video, imagen, etc.), se dice que este sistema es capaz de **secuenciar problemas o cuestiones**. Tres ejemplos de sistemas de este tipo son el sistema DCG [Vassileva, 1997], que permite secuenciar de forma estática el material educativo, adaptándose a la meta de aprendizaje antes de que el estudiante comience a trabajar; el sistema SIETTE [Rios *et al.*, 1999], que es un sistema de adaptación a exámenes, donde el único tipo de tarea es la pregunta del examen, generando una secuencia de preguntas según el conocimiento del usuario; y el sistema MEDTEC [Eliot *et al.*, 1997], que permite que el usuario elija un tema, pero genera una secuencia adaptativa de problemas a resolver.

Únicamente unos pocos sistemas permiten secuenciar a dos niveles. Por ejemplo, MANIC [Stern and Woolf, 1998] permite secuenciar a bajo nivel (eligiendo la actividad a realizar) sin que el usuario elija ningún tema de conocimiento en especial.

2.2.3. Elección de la Mejor Política de Aprendizaje

Para realizar una buena elección de política pedagógica, lo primero que nos hemos de plantear es por qué se necesitan distintas políticas y en qué situaciones se diferencian. En general existen cuatro situaciones [Khan and Yip, 1996]:

1. *Enseñanza a un alumno de nuevo conocimiento*. Dependiendo de la materia, el material a enseñar puede ser diferente (p.e., en lingüística se puede enseñar tanto la teoría, como conocimiento práctico sobre la causalidad). Esto necesita distintas estrategias de enseñanza para facilitar el aprendizaje del estudiante.
2. *Enseñanza a distintos alumnos del mismo ítem de conocimiento*. Por ejemplo, puede ocurrir que el estudiante *A* prefiera un manual, y el estudiante *B* prefiera ideas intuitivas. El sistema tiene que tener en cuenta la diferencia entre los estudiantes, proponiéndoles distintos métodos de aprendizaje.

3. *Enseñanza a un alumno de nuevo conocimiento y estrategias para enseñar a diferentes estudiantes el mismo conocimiento (mezcla de las anteriores).* Un ITS normalmente interacciona con más de un alumno al mismo tiempo, por lo tanto, se puede deducir que se necesitan distintas estrategias para la enseñanza del mismo conocimiento al mismo tiempo.
4. *El estudiante progresa a partir del discurso de la enseñanza.* Al principio podría necesitar una estrategia de enseñanza tradicional (mediante lectura), pero posteriormente, mediante el discurso, el alumno podría ser retado a pensar antes de solicitar la ayuda. Estas situaciones implican que la selección debería ser capaz de proporcionar diferentes estrategias para tutorar, de acuerdo con los niveles de conocimiento del estudiante que se encuentran en continuo cambio.

Por lo tanto, no es tarea fácil realizar una buena selección de políticas pedagógicas para cada tutor inteligente, con la problemática añadida de, si falla en la elección, decidir por qué falla.

Hasta ahora, los pocos sistemas que detectan cuándo falla una política pedagógica se basan en los resultados de los tests realizados a los estudiantes. Cuando un estudiante falla repetidamente en distintos tests, esto indica que la política utilizada no es la correcta.

En general, los resultados de los tests se pueden clasificar en tres categorías [Tong and Angelides, 1999]:

1. **Mayoría de las respuestas correctas.** Parece que es una política pedagógica adecuada, aunque también podría ser que el alumno recuerda y comprende la información, que es demasiado fácil para él, que el alumno adivina las respuestas sin conocerlas, que el alumno ha recibido tantas veces esa información (demasiado repetitiva la estrategia) que no puede olvidarlo, etc. Es decir, puede existir mucho ruido en los resultados de los tests, de forma que parece que el resultado es constructivo, mientras que es todo lo contrario.
2. **Mayoría de las respuestas Incorrectas.** Esta vez sí que es una indicación muy fiable de que la política pedagógica falla. También podría ocurrir que el usuario hubiera pulsado de forma errónea una tecla, que esté aburrido de la materia, etc.
3. **Más o menos 50 % de aciertos.** Indica que el estudiante progresa a velocidad constante, aunque esta posibilidad podría derivar rápidamente a cualquiera de los dos extremos anteriores.

2.3. Técnicas de Inteligencia Artificial Aplicadas a Sistemas de Educación

Los ordenadores se han utilizado en la educación desde hace más de 40 años. Los primeros sistemas informáticos utilizados para la enseñanza no se adaptaban a las necesidades del estudiante (ver sección 2.1.1). Posteriormente se comenzaron a utilizar técnicas de inteligencia artificial para permitir que el sistema fuera más flexible a la hora de presentar el contenido del sistema, así como el orden de presentación del curso.

En el siguiente apartado se repasan brevemente los métodos y técnicas de inteligencia artificial más utilizados en los programas de enseñanza. Además, se dedica especial atención a cómo se ha aplicado *Aprendizaje por Refuerzo* en los sistemas de educación.

2.3.1. Aplicación de Inteligencia Artificial en Programas de Enseñanza

Habitualmente los sistemas de enseñanza se basan en modelos formales (ver sección 2.1.4) que contienen información acerca de las características de aprendizaje del estudiante (modelo de estudiante), de las características de la materia a enseñar (modelo del dominio), de las estrategias pedagógicas a seguir (modelo pedagógico) y sobre la interacción con los distintos estudiantes (modelo del interfaz). Los sistemas de educación inteligentes utilizan estos modelos para representar internamente el entorno y obtener la información necesaria para emular el comportamiento de un profesor en la enseñanza tradicional, con la ventaja de que la interacción se puede reducir a un único alumno.

El desarrollo y mantenimiento de estos modelos formales ha sido hasta hoy una de las principales metas de la aplicación de las técnicas de inteligencia artificial en los sistemas de enseñanza.

El modelo del dominio se suele describir como parte de un sistema experto, donde se han utilizado frecuentemente técnicas de inteligencia artificial relacionadas con el *modelado simbólico* (por ejemplo, los *marcos* [Fernández-Castro, 1989]). También se suelen utilizar *redes bayesianas* [VanLehn and Zhendong, 2001] para describir el modelo del dominio.

Por otro lado, el comportamiento pedagógico del sistema se ha modelado habitualmente mediante técnicas de *planificación*, como el sistema WIP [André *et al.*, 1993] y *sistemas basados en reglas* de producción, basadas en el comportamiento de un experto, como los sistemas LISP [Anderson and Reiser, 1985] y el sistema de [Prentzas *et al.*, 2002], que combina reglas con computación neuronal.

En el modelo del interfaz se aplican técnicas de inteligencia artificial para

lograr adaptar la interacción al usuario. Por ejemplo, se están aplicando técnicas de *procesamiento del lenguaje natural* para interactuar mediante diálogo en sistemas como PACT Geometry Tutor [Alevén and Koedinger, 2000]. También se utilizan técnicas de anotación en el interfaz (ver apartado 2.1.3) y técnicas de búsquedas de temas dentro del modelo del dominio en muchos sistemas, por ejemplo, ELM-ART [Brusilovsky *et al.*, 1996].

Se han utilizado muchos métodos de representar la información en el modelo de estudiante. Quizás los tres más comúnmente utilizados sean los *estereotipos* [Rich, 1979], los *modelos overlay* [Carr and Goldstein, 1977] y las *redes bayesianas* [VanLehn and Zhendong, 2001].

Cuando se utiliza el modelo overlay para representar al estudiante, el conocimiento del estudiante se considera como un subconjunto de un sistema experto. Posteriormente se han estudiado técnicas de *detección de errores* extendiendo el *modelo overlay*, donde se almacena información no sólo del conocimiento correcto del estudiante (parte del sistema experto), sino también de los errores del estudiante y el conocimiento mal entendido (en inglés “*buggy*” *knowledge*). El sistema DEBUGGY [Burton, 1982] es uno de los sistemas que utiliza este tipo de técnicas para modelado de usuario.

Las redes bayesianas razonan de forma probabilística sobre el estado de conocimiento del estudiante basado en sus interacciones con el tutor. Los sistemas de [Villano, 1992] y [Martin and VanLehn, 1993], por ejemplo, utilizan redes bayesianas para representar el conocimiento del estudiante.

En el modelado del estudiante, ha sido muy común la utilización de técnicas de aprendizaje automático, permitiendo adaptarse dinámicamente a las necesidades de los alumnos a partir de datos recogidos al interactuar con otros alumnos [Sison and Shimura, 1998]. Se define *aprendizaje* como la *inducción* de nuevo conocimiento o la *compilación* de conocimiento existente, de forma que pueda mejorar el desarrollo de una tarea. La inducción de nuevo conocimiento puede ser supervisada, donde se aprende a partir de objetos etiquetados a priori, o no supervisado [Mitchell, 1997].

Se han utilizado técnicas de *aprendizaje inductivo supervisado* sobre todo para inducir modelos de usuario a partir de múltiples comportamientos de estudiantes, por ejemplo, en los sistemas DEBUGGY, ACM [Langley and Ohlsson, 1984], THEMIS [Kono *et al.*, 1994] y ASSERT [Baffes and Mooney, 1996].

También se han utilizado estas técnicas con el objetivo de construir y extender librerías de errores al mismo tiempo que se inducen modelos de estudiantes parciales, por ejemplo, en el sistema MEDD [Sison *et al.*, 2000].

Se han aplicado técnicas de *aprendizaje deductivo supervisado* para extender librerías de errores en los modelos de estudiante a partir de la compilación de conocimiento existente. Sistemas como PIXIE [Sleeman and Brown, 1982] y Hoppe [Hoppe, 1994] utilizan este tipo de técnicas.

El modelo de *aprendizaje por refuerzo* también ha sido aplicado a tutores inteligentes por [Beck, 2001]. En la siguiente sección se habla de las ventajas de la aplicación del modelo de aprendizaje por refuerzo en tutores inteligentes, detallando el trabajo de Beck.

2.3.2. Aprendizaje por Refuerzo Aplicado a Programas de Enseñanza

Casi todos los sistemas de enseñanza que se adaptan de forma inteligente a los estudiantes poseen un conjunto de reglas de producción pedagógicas de tipo “if-then”. Este tipo de módulos pedagógicos tienen dos problemas principalmente:

1. Existen demasiadas reglas, por lo que es demasiado costoso codificarlas en el sistema.
2. Es difícil incorporar el conocimiento de los tutores humanos mediante reglas, ya que los expertos no pueden describir cómo debería razonar el sistema con ese conocimiento.

La aplicación del modelo de aprendizaje por refuerzo en sistemas de educación adaptativos e inteligentes en *Web* evita el problema de la definición de las estrategias pedagógicas para cada estudiante en cada instante. Esta metodología permite a los tutores inteligentes tomar estrategias de decisión a alto nivel basándose en información de bajo nivel del estudiante.

Además, el modelo de aprendizaje por refuerzo es flexible ante ruido en los datos, ya que aprende del siguiente estado y no necesita un sistema experto en el modelo del dominio. Pero este modelo tiene un punto débil: necesita una gran cantidad de aprendizaje inicial para conseguir alcanzar con seguridad una estrategia de comportamiento óptima.

La investigación de Beck [Beck, 2001] surge a partir de un tutor inteligente de enseñanza de aritmética decimal y fraccional. Este sistema posee heurísticas que ayudan a determinar si el estudiante necesita ayuda en ciertos temas, ha olvidado alguno, necesita que se muestre un ejemplo, etc. El problema encontrado en el sistema inicial era la falta de adaptación individual a cada alumno, existiendo una secuencia de contenido general para todos los alumnos. Empíricamente era posible generar un conjunto de reglas, sin embargo, se trataba de una tarea demasiado costosa y no se conseguía la adaptación a cada usuario según sus características (por ejemplo, aunque dos usuarios parecían tener características similares, era posible que uno aprendiera mejor mediante resolución de problemas y otro mejor mediante texto).

Para solucionarlo, Beck utiliza *Aprendizaje por Refuerzo* (del inglés *Reinforcement Learning: RL*) como modelo de aprendizaje del sistema a secuenciar el

contenido del tutor a alto nivel (temas o *topics* de conocimiento) [Beck, 2001], indicando que tener un modelo que razona sobre el conocimiento del usuario a bajo nivel (grano-fino: tareas o items de conocimiento) es menos efectivo que tratar de encontrar las características del usuario en grano-grueso (temas de conocimiento).

Para ello, define el estado del agente como el modelo del estudiante del tutor inteligente y las acciones a tomar como *topics* a mostrar. Estas acciones las define como estocásticas (indeterministas o que se ejecutan de forma incorrecta). Utiliza la exploración egoísta (*e-greedy*). Además, propone un refuerzo inmediato, de forma que si, al ejecutar una acción y examinar cómo afecta al estudiante en los problemas futuros la acción fue efectiva, el sistema recibe un refuerzo positivo inmediato, ya que esperar hasta el final del aprendizaje del estudiante para proporcionar el refuerzo tiene varios problemas:

1. No se sabe cuándo el estudiante llegará el final de su aprendizaje.
2. El refuerzo retardado produce mucha más varianza en los valores de la tabla Q durante el aprendizaje, por lo que la convergencia hacia valores óptimos es más lenta.

Recoge en el modelo de estudiante información de ayuda ante las decisiones pedagógicas, como el grado de habilidad del estudiante (conocimiento sobre la materia y probabilidad de aprendizaje de cada tema), pero hace notar que estos datos tan simples a veces no son suficientes, necesitándose un razonamiento más complejo (revisión de temas aprendidos, ofrecerle problemas para resolver, ofrecerle el mismo conocimiento pero expresado de otra forma, etc.).

La arquitectura de su sistema en cuanto al proceso de ejecución se reduce a dos únicos módulos:

- **Modelo de estudiante** (*Population Student Model (PSM)*), que aprende el modelo de cómo los estudiantes utilizan el tutor.
- **Agente Pedagógico** (*Pedagogical Agent (PA)*), utilizando para entrenar al agente del modelo de aprendizaje por refuerzo responsable de conseguir el objetivo de enseñanza.

Para solventar el problema de la gran cantidad de aprendizaje inicial que ha de llevar a cabo el sistema para comenzar a enseñar de manera óptima, propone una metodología de aprendizaje off-line utilizando datos simples, estudiantes simulados y pequeñas cantidades de conocimiento del experto.

La utilización de estudiantes simulados se ha analizado anteriormente, probando su eficacia. [VanLehn *et al.*, 1996] fue uno de los primeros trabajos que utilizaron estudiantes simulados. Beck construye modelos de estudiante “típicos” no completos a partir del análisis de las tareas cognitivas de estudiantes reales que

interaccionan con el sistema, de forma que aprenda a partir de ellos. Mediante esta inicialización, el coste del aprendizaje del sistema decrece.

Como conclusiones a su trabajo, se pueden distinguir las siguientes ventajas:

1. El coste de construir y validar el sistema es menor que en los sistemas anteriores, donde había que definir una a una las estrategias pedagógicas de cada estudiante en cada instante, las diferencias que existían entre ellas y, si fallaban, por qué y cómo solucionarlo.
2. El sistema es capaz de adaptarse a las necesidades de cada estudiante individual, automatizando la toma de decisiones del tutor inteligente aprendiendo el comportamiento de los estudiantes observando a conjuntos de usuarios.

Como puntos débiles en el trabajo de Beck se observa que:

- Aplica *aprendizaje por refuerzo* al modelo del estudiante, donde existen muchas características propias de cada estudiante que el sistema ha de tener en cuenta. Al tener en cuenta tantas características, el proceso de aprendizaje del sistema será muy lento.
- Beck aplica el modelo de *aprendizaje por refuerzo* con el objetivo de secuenciar el contenido del sistema a alto nivel en el árbol de conocimiento (a nivel de tema), sin tener en cuenta el orden de presentación de las tareas o ítems de conocimiento de cada tema.
- Utiliza una política de exploración/explotación *e-greedy*, sin estudiar si la aplicación de otras políticas de exploración/explotación pudieran ser ventajosas en sistemas de educación a distancia.

Todos estos puntos débiles encontrados en el trabajo de Beck, se analizan en profundidad en el apartado 2.5. En el siguiente apartado se estudiará en detalle la técnica de *Aprendizaje por Refuerzo*.

2.4. Profundizando en el Modelo de Aprendizaje por Refuerzo

El objetivo del aprendizaje por refuerzo [Kaelbling *et al.*, 1996] es que un agente siga un comportamiento óptimo al tratar de alcanzar sus objetivos, mediante la ejecución de acciones elegidas mediante prueba y error, dependiendo únicamente de la percepción que posee del entorno y el estado en el que se encuentra. El entorno es el encargado de proporcionar refuerzos al agente para guiar

su aprendizaje. El refuerzo aplicado será positivo si se ejecuta una acción adecuada en el estado en el que se encuentra el agente, y negativo o nulo en otro caso.

El propósito del agente es obtener el mayor refuerzo positivo en cada instante y, de esta forma, indirectamente, realizar un comportamiento óptimo global en la consecución de sus objetivos.

Este tipo de aprendizaje se utiliza en muchos y muy variados tipos de problemas, desde aprender a controlar robots móviles, hasta controlar centrales químicas, pasando por aprender a jugar al ajedrez, sobre todo en sistemas donde la interacción con el entorno es muy activa y dinámica, como es el caso de la propuesta que se realiza en este trabajo.

Una de las principales ventajas del aprendizaje por refuerzo se enuncia en el principio de optimalidad de Bellman [Bellman, 1957]: “Una política óptima tiene la propiedad de que sea cual sea el estado inicial y la decisión inicial, el resto de las decisiones deben constituir una política óptima para alcanzar el estado final a partir de la primera decisión”. Es decir, sea cual sea el estado inicial del entorno, mediante mecanismos de aprendizaje por refuerzo, el sistema llega a la solución del problema.

2.4.1. Componentes

La definición de aprendizaje por refuerzo según [Kaelbling *et al.*, 1996] es la siguiente: sea un agente que, encontrándose en un estado concreto, s , que percibe según una entrada, i , realiza una acción, a . Esta acción produce un cambio de estado en el entorno, que proporciona al agente una señal de refuerzo, r (ver Figura 2.2):

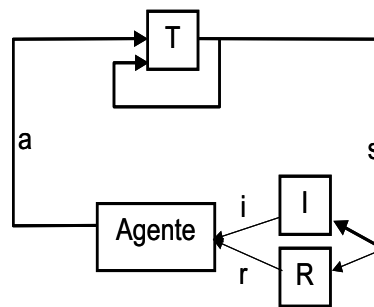


Figura 2.2: Modelo de Aprendizaje por Refuerzo.

Por lo tanto, existen los siguientes componentes en el modelo de aprendizaje por refuerzo:

- S : conjunto de estados posibles del entorno.
- A : conjunto de acciones que realiza el agente.
- I : función de entrada al agente que indica cómo percibe el agente el estado actual del entorno.
- R : conjunto de señales de refuerzo que percibe el agente al cambiar el estado del entorno tras la ejecución de una acción.
- T : función que realiza las transiciones de estado.

2.4.2. Procesos de Decisión de Markov

El modelo más utilizado para definir los problemas de refuerzo retardado es el de *los Procesos de Decisión de Markov (MDP: Markov Decision Problems)* [Bellman, 1957], que define la dinámica del entorno del problema mediante las funciones de transición de estado, T , y de refuerzo, R :

- $R : S \times A \rightarrow \mathcal{R}$, donde para cada estado y acción se produce un refuerzo (esperanza del refuerzo en el siguiente instante):

$$R(s, a) = R_{ss'}^a = E\{r_{t+1} | s_t = s, a_t = a\} \quad (2.1)$$

- $T : S \times A \rightarrow \Pi(S)$, donde para cada estado y acción existe una distribución de probabilidad sobre el conjunto S , de forma que $T(s, a, s')$ es la probabilidad de que se realice la transición a s' , encontrándose en el estado s y ejecutando la acción a .

$$T(s, a, s') = P_{ss'}^a = Pr\{s_{t+1} = s' | s_t = s, a_t = a\} \quad (2.2)$$

Por lo tanto, el objetivo del agente para aprender un comportamiento óptimo es encontrar la política π que maximice el refuerzo obtenido en la consecución de sus objetivos.

Una de las propiedades más importantes de los *procesos de decisión de Markov (MDP)* es que no tienen en cuenta por qué estados se ha pasado anteriormente, sino que a partir de estado actual, el agente puede decidir qué acción tomar. Para poder tener en cuenta los estados pasados, existen, por tanto, señales de estado que resumen toda la información anterior relevante para el problema, denominadas funciones *Markovianas*. Estas funciones indican que la probabilidad de que un agente se encuentre en un estado, recibiendo un determinado refuerzo, depende

únicamente del estado en el que se encontraba en el instante anterior y la acción ejecutada. Estas propiedades se definen como:

$$\begin{aligned} Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\} = \\ Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, r_{t-1}, \dots, s_0, a_0, r_0\} \end{aligned} \quad (2.3)$$

Se denomina *proceso de decisión de Markov* a todo problema de aprendizaje que satisface esta propiedad. Además, si el número de estados y acciones definidos en el problema son finitos, se le denomina *MDP finito*.

2.4.3. Puntos Clave en el Aprendizaje por Refuerzo

A la hora de definir cualquier problema de aprendizaje por refuerzo, es necesario tener en cuenta los siguientes aspectos del problema a tratar [Mitchell, 1997] [Fernández, 1999]:

1. **Aplicación del refuerzo.** Es necesario decidir en qué momento se ha de aplicar la señal de refuerzo sobre el agente. En algunos casos es necesario aplicarlo inmediatamente después del cambio de estado percibido por la acción realizada, ya que en cada estado se conoce lo bueno o malo que es haberlo alcanzado. Sin embargo, en otros casos no se conoce la bondad del estado inmediatamente después de la ejecución de la acción elegida, siendo necesaria la ejecución de varias, o incluso siendo necesario llegar al objetivo final del problema.
2. **Exploración versus Explotación.** También es necesario definir qué tipo de estrategias de exploración se van a utilizar. Las estrategias de explotación tratan la utilización de conocimiento adquirido previamente, al generar planes de ejecución. Sin embargo, las estrategias de exploración tratan de buscar nuevas alternativas con las que obtener eficazmente los objetivos. La calidad de la estrategia elegida será determinante en la eficiencia y eficacia del proceso de aprendizaje.
3. **Parcialidad en la observación de los estados.** En algunos casos, el agente es incapaz de percibir el estado completo en el que se encuentra el entorno. Debido a este problema, el agente puede no distinguir la diferencia existente entre estados.
4. **Indeterminismo en acciones y refuerzos.** En algunos entornos, la ejecución de una acción desde un estado concreto puede que lleve al agente a varios estados finales diferentes. Este indeterminismo puede deberse a problemas en la percepción del estado en el que se encuentra (ruido en los

sensores), o debido a que las acciones que ejecuta no son deterministas. Por lo tanto, el refuerzo que obtiene el agente al realizar una acción puede variar, ya que la bondad del estado al que se ha llegado puede ser distinta.

5. **Dominios continuos o de gran tamaño.** En el mundo real, la mayoría de los dominios son continuos y, por lo tanto, los estados en los que se puede encontrar el entorno son infinitos. También ocurre a menudo que el conjunto de acciones válidas que puede realizar un agente son infinitas. Cuando esto ocurre, es necesario un mecanismo de discretización de los valores que pueden adoptar los estados y/o acciones. A este problema se le denomina *generalización de estados y acciones*.
6. **Integración de habilidades.** Las acciones que pueden realizar los agentes podrían agruparse según ciertas características, de manera que se reduce el número de acciones que puede realizar el agente en un determinado estado.

2.4.4. Clasificación de Algoritmos

Existen un gran número de clasificaciones de los algoritmos de aprendizaje por refuerzo. Una de ellas, que se explica a continuación, se basa en el conocimiento que se posee del dominio:

- Si se conoce toda la dinámica del entorno (estados (S), acciones (A) y funciones de transición (T) y refuerzo (R)), se pueden aplicar **técnicas de programación dinámica** [Bellman, 1957].
- En el caso de que se conozcan los estados y acciones, pero no sus efectos en el entorno, no se puede aplicar programación dinámica, al no conocer todos los parámetros del algoritmo y, por lo tanto, se han de aplicar otras técnicas de aprendizaje. Este tipo de problemas se pueden subdividir a su vez en dos, dependiendo de la forma de realizar el aprendizaje:
 - **Métodos basados en el modelo:** algunas veces es interesante aprender primero el modelo del dominio (los parámetros que no se conocían del modelo) para posteriormente aplicar programación dinámica sobre el mismo [Sutton, 1990] [Sutton, 1991].
 - **Métodos libres de modelo:** en otros casos no interesa aprender el modelo, ya que, entre otras cosas, necesita de mucha exploración y coste computacional [Watkins, 1989].

2.4.5. Funciones de Utilidad

La mayoría de los algoritmos de aprendizaje por refuerzo se basan en las llamadas *funciones de valor*, que estiman lo bueno que es encontrarse en un determinado estado y lo bueno que es ejecutar una acción desde ese estado. La función de *valor-estado*, $V^\pi(s)$, es el refuerzo que se espera obtener si el agente se deja guiar por una política de acción, π , desde el estado, s , hasta el infinito, donde γ es el parámetro de descuento de las futuras acciones (ver ecuación 2.4):

$$V^\pi(s) = E_\pi\{R_t | s_t = s\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right\} \quad (2.4)$$

donde $E_\pi\{\}$ denota el valor esperado si el agente sigue la política π .

Existe también otra función en la que se basan este tipo de algoritmos, la función de *valor-acción*, $Q^\pi(s, a)$, que estima el refuerzo que se espera obtener si el agente se deja guiar por una política de acción, π , desde el estado s hasta el infinito, comenzando por ejecutar la acción a (ver ecuación 2.5):

$$Q^\pi(s, a) = E_\pi\{R_t | s_t = s, a_t = a\} = E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\} \quad (2.5)$$

Estas funciones de valor nos proporcionan un buen método de comparación de calidad de políticas (π). Por ejemplo, se podría decir que la política más adecuada es la que proporciona una ganancia mayor, tal y como se define en la ecuación 2.6:

$$\pi \geq \pi' \text{ si y sólo si } V^\pi(s) \geq V^{\pi'}(s) \forall s \in \mathcal{S} \quad (2.6)$$

Se denomina política óptima, π^* , a la política que proporciona una función de *valor-estado* óptima, definida en la ecuación 2.7:

$$V^*(s) = \max_{\pi} V^\pi(s) \forall s \in \mathcal{S} \quad (2.7)$$

y una función de *valor-acción* óptima (ver ecuación 2.8):

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a) \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \quad (2.8)$$

Por lo tanto, se puede decir que la búsqueda de una política óptima se basa en la búsqueda de la función de valor óptima, dado que la política Π^* puede ser obtenida a partir de Q^* , tal y como se define en la ecuación 2.9:

$$\Pi^*(s) = \arg_a \max Q^*(s, a) \quad (2.9)$$

2.4.6. Q-learning: un Algoritmo de Aprendizaje por Refuerzo

El algoritmo Q-learning [Watkins, 1989] [Watkins and Dayan, 1992] implementa un método de aprendizaje por refuerzo libre del modelo que utiliza técnicas de aprendizaje por diferencia temporal (*TD: Temporal-Difference learning*). Este tipo de técnicas aprenden directamente de la experiencia, es decir, no se basan en ningún modelo de dinamismo del entorno, basándose únicamente en la función de *valor-acción*, $Q(s, a)$, para aprender.

Este algoritmo se basa en la actualización iterativa de la función $Q(s, a)$, tal y como define la ecuación 2.10 para el caso determinista:

$$Q(s, a) = R(s, a) + \gamma \max_{a'} Q(s', a') \quad (2.10)$$

En entornos no deterministas, como ocurre en el entorno de los tutores inteligentes, la ecuación 2.10 no se puede aplicar, ya que no tiene en cuenta las probabilidades de cambiar o no de estado, por lo que la ecuación de evaluación $Q(s, a)$ se redefine según la ecuación 2.11. Al redefinir la ecuación, se tienen en cuenta los casos en los que la ejecución de una acción desde un estado puede llevar al agente a estados distintos con la probabilidad α . El parámetro *alpha* mide cuánto se tiene en cuenta el nuevo valor estimado para $Q(s, a)$ y cuánta importancia se da al valor antiguo, influyendo en la velocidad del aprendizaje del sistema.

$$Q_t(s, a) = (1 - \alpha_t)Q_{t-1}(s, a) + \alpha_t[R(s, a) + \gamma \max_{a'} Q_{t-1}(s', a')] \quad (2.11)$$

Esta definición recursiva de la función de *estado-acción* proporciona la base para algoritmos que aproximan Q de forma iterativa, por ejemplo, el algoritmo *Q-learning* presentado en esta sección.

En la Tabla 1 se puede observar el algoritmo Q-learning para procesos de decisión de Markov no deterministas, donde se han asumido acciones y refuerzos no deterministas, y el factor de descuento en futuras acciones, γ , será una constante tal que $0 \leq \gamma < 1$, siendo α el parámetro de aprendizaje del algoritmo, cuyos valores varían entre $0 \leq \alpha < 1$.

Una de las principales características de este algoritmo es que no requiere que el agente ejecute secuencias óptimas para converger a una política óptima [Mitchell, 1997], pudiendo aprender la función Q , representada habitualmente de forma tabular y, por lo tanto, la política óptima, ejecutando en cada paso acciones elegidas de forma aleatoria, siempre que los pares estado-acción sean visitados un número suficiente de veces.

En este momento, se puede intuir la complejidad de este algoritmo. Como se ha dicho anteriormente, para converger, es necesario visitar todos los posibles pares estado-acción varias veces, que en determinados dominios puede ser un gran

Algoritmo Q-learning

- Inicializar $Q(s, a)$ arbitrariamente
 - Repetir (para cada episodio)
 - Inicializar s
 - Repetir
 - Seleccionar una acción a a partir de s usando una política, π , derivada de Q
 - Ejecutar la acción a observando el refuerzo inmediato recibido, r , y el siguiente estado, s'
 - Actualizar la entrada de la tabla, $Q(s, a)$ con la ecuación:

$$Q_t(s, a) = (1 - \alpha_t)Q_{t-1}(s, a) + \alpha_t [R(s, a) + \gamma \max_{a'} Q_{t-1}(s', a')] \quad (2.12)$$
 - $s = s'$
- Hasta que s sea un estado final
-

Figura 2.3: Algoritmo *Q-learning*.

inconveniente. Un gran número de estados y/o de acciones diferentes son poco eficientes, siendo necesario en la mayoría de los casos acotar la búsqueda del comportamiento óptimo. Es necesario en este punto advertir al lector de las desventajas y peligros que pueden existir al acotar esta búsqueda, donde, por ejemplo, se puede caer en mínimos locales si no se realiza toda la exploración sobre el dominio. Es decir, se podría encontrar un buen comportamiento, pero no el óptimo.

Una estrategia para el tratamiento de grandes cantidades de información, es tratarla de forma jerárquica. En muchos casos, las soluciones jerárquicas realizan una ligera mejora en la optimización del desarrollo del sistema, pero potencialmente introducen una gran eficiencia en tiempo de ejecución, de aprendizaje y de espacio [Kaelbling *et al.*, 1996], implicando reducciones exponenciales del coste computacional al encontrar políticas óptimas [Dietterich, 2000]. La descomposición en subproblemas tiene muchas ventajas, como por ejemplo, las políticas aprendidas en los subproblemas pueden ser compartidas (o reutilizadas) por muchas tareas de los padres de la jerarquía; los valores de las funciones aprendidas en subproblemas pueden ser compartidas, de forma que el aprendizaje total se acelera; y la abstracción de estados puede ser aplicada de forma que la función

de valor final puede ser representada como una suma de funciones de valor que dependen sólo del subconjunto de las variables de estado. Todo esto implica que se necesita menos datos para aprender y, por lo tanto, el aprendizaje será más rápido. Existen distintos algoritmos de aprendizaje que se podrían utilizar, como *Q-learning Feudal* (del inglés *Feudal Q-learning*) [Albus, 1981] [Barto *et al.*, 1995], *Q-learning Composicional* (del inglés *Compositional Q-learning*) [Singh, 1992] [Singh, 1993], *Q-learning con distancia jerárquica a la meta* (del inglés *HDG: Hierarchical Distance Goal*) [Kaelbling *et al.*, 1993], etc.

2.4.7. Exploración vs Explotación

Una de las características más importantes que distinguen el aprendizaje por refuerzo de otro tipos de aprendizaje es que utiliza información de entrenamiento y evalúa las acciones tomadas, y no tanto las instrucciones que proporcionan acciones correctas. Por ello se crea la necesidad de que el agente que aprende mediante aprendizaje por refuerzo explore de forma activa el entorno, buscando explícitamente las mejores acciones a ejecutar en cada instante mediante prueba y error con el objetivo de conseguir buenos comportamientos.

Existen dos tipos de realimentación (del inglés *feedback*) que se han usado en problemas de aprendizaje:

- La realimentación puramente evaluativa indica cómo de buena es la acción tomada, pero no si es la mejor o la peor acción posible. Normalmente los métodos de optimización de funciones se basan en este tipo de realimentación. Depende únicamente de la acción ejecutada por última vez.
- La realimentación puramente instructiva, por otro lado, indica cuál es la acción correcta a ejecutar a continuación, independientemente de la acción ejecutada por última vez. Este tipo de realimentación sirve como base normalmente al aprendizaje supervisado, incluyendo clasificación de patrones, redes neuronales artificiales y sistemas de identificación. Este tipo de realimentación es totalmente independiente de la acción ejecutada en última instancia.

En aprendizaje por refuerzo se utiliza la realimentación evaluativa, basándose típicamente en la optimización de las funciones de *valor-acción* ($Q(s, a)$), por lo que este apartado se centra en este tipo de realimentación (ver apartado 2.4.5).

En un sistema de aprendizaje, realiza *Exploración* y *Explotación* del entorno cuando utiliza el conocimiento que se posee sobre el valor de las acciones. Si elige una de las acciones no *avariciosas* (entendiendo por *avariciosas* las acciones cuyo valor estimado es mayor), se dice que se *explora* el entorno porque permite mejorar el valor de las acciones no avariciosas. Por otro lado, si elige alguna de las

acciones *avariciosas* para ejecutar, se dice que el sistema *explota* el conocimiento del entorno.

Mediante *Explotación*, el sistema maximiza el refuerzo de una determinada interacción completa con el sistema, pero si se realiza exploración, se producirá el mejor refuerzo total de todas las interacciones. Si es mejor explorar o explotar depende de una gran cantidad de variables, como los valores de las estimaciones de las acciones, inexactitudes en las estimaciones, el número de interacciones que faltan para concluir, etc.

La mejor forma de mejorar la interacción del sistema será, en primera instancia, afinar el valor estimado de las acciones del entorno, realizado *exploración*. Una vez afinados los valores estimados de las mejores acciones, lo mejor será *explotar* este conocimiento. De esta manera se alcanzaría el mejor refuerzo total en el sistema.

Existen muchos métodos sofisticados para balancear exploración y explotación [Sutton and Barto, 1998], normalmente aplicadas a formulaciones matemáticas particulares de problemas dados. Sin embargo, la mayoría de estos métodos asumen ciertos datos estacionales y un determinado conocimiento inicial que normalmente es imposible de verificar.

Como se comentó anteriormente en el capítulo 2.4.5, la mayoría de los sistemas basan su aprendizaje en la función de *valor-acción* ($Q(s, a)$), donde $Q^*(s, a)$ es el valor estimado de la función $Q(s, a)$ óptima.

La regla de selección más simple en este caso es seleccionar la acción con mayor valor estimado de $Q(s, a)$: $Q_t^*(s, a) = \max_a Q_t(s, a)$. Este método siempre *explota* el conocimiento actual para maximizar el refuerzo inmediato; no explora otras acciones de aparentemente menor valor, aunque estas podrían ser realmente mejores. A esta estrategia de selección de acciones se la denomina *estrategia de explotación avariciosa* (del inglés *greedy*).

Una pequeña variación a esta estrategia es comportarse de forma avariciosa, pero con una pequeña probabilidad de no elegir la acción con mejor valor. A esta estrategia de exploración/explotación se la denomina *e-greedy* (también denominada regla de selección *casi-avariciosa*: del inglés *near-greedy*). Esta estrategia de selección de acción garantiza que cuando la exploración se lleva a infinito para todas las acciones, $Q_t(s, a)$ converge a $Q^*(s, a)$.

Si se comparan numéricamente las dos estrategias de exploración/explotación, el método *greedy* hace que el sistema converja a tácticas de comportamiento buenas de forma un poco más rápida que si se hubiera utilizado *e-greedy*, pero alcanza una política de comportamiento significativamente peor. Esto se debe a que *e-greedy* con frecuencia toma acciones sub-óptimas al no explorar por completo el dominio, por lo que optimiza el refuerzo del sistema de forma inmediata, pero no a largo plazo (sumando el refuerzo de todas las interacciones). Normalmente, si la varianza de los refuerzos es un valor alto, es mejor realizar al principio más ex-

ploración, hasta que el sistema encuentra la acción óptima (utilizando *e-greedy*), pero si la varianza no es muy grande, se podría utilizar la estrategia *greedy*.

Aunque la política de selección de acción *e-greedy* es efectiva y popular para balancear exploración y explotación en el aprendizaje por refuerzo, un problema de esta política es cuando explora (no ejecuta la acción avariciosa) y escoge por igual entre todas las acciones. Es decir, podría escoger de igual forma una acción que parece ser la peor, como una acción que parece ser la siguiente a la mejor acción. En tareas donde la peor acción es muy mala este tipo de elección no es en ningún caso satisfactoria.

La solución más obvia es variar las probabilidades de acción mediante una función que estime el valor de una acción, de forma que la acción con mayor valor de Q (acción avariciosa) continúe siendo la que posea una probabilidad mayor de ser elegida, pero el resto de acciones posean un peso estimado de acuerdo a su función de valor acción. A este tipo de reglas de selección se las denomina reglas *Softmax*, y los métodos más comunes que utilizan este tipo de reglas utilizan una distribución de *Gibbs* o *Boltzmann*.

En la estrategia de exploración/explotación de *Boltzmann*, la probabilidad de ejecutar una acción viene dada por la ecuación 2.13. En esta función, n es el número total de acciones que puede ejecutar el agente y el parámetro τ posee un valor positivo y se le denomina *temperatura*. En este método, cuando la temperatura es alta (cercana a 1), todas las acciones del entorno son casi equiprobables. Sin embargo, a bajas temperaturas, la probabilidad de las acciones difiere en su valor estimado. En el límite cuando τ tiende a cero, la estrategia de selección de acción de *Softmax* se comporta igual que la estrategia de selección de acción *greedy*.

$$P(a) = \frac{\epsilon^{\frac{Q(a_i)}{\tau}}}{\sum_{a_j=1}^n \epsilon^{\frac{Q(a_j)}{\tau}}} \quad (2.13)$$

Si la estrategia de selección de acción de *Softmax* es mejor que la estrategia *e-greedy* o viceversa no está claro, dependerá de las tareas a llevar a cabo y de factores humanos.

Los métodos discutidos anteriormente dependen en alguna medida de la estimación de la función de valor-acción inicial ($Q_0(a)$). En lenguaje estadístico, estos métodos son parciales en sus estimaciones. Por ejemplo, los métodos de media simple, la parcialidad desaparece una vez que todas las acciones se han seleccionado al menos una vez.

En la práctica, este tipo de parcialidad a veces no es un problema, sino que algunas veces puede ser de ayuda.

Inicializar los valores de acción puede usarse como una manera muy sencilla de llevar a cabo la exploración en el sistema. Supongamos que en lugar de ini-

cializar los valores de todas las acciones a cero, lo inicializaremos al valor más alto (por ejemplo 1). Esta inicialización es optimista. Sea cual sea la acción seleccionada en primer lugar, el refuerzo es menor o igual que lo que inicialmente se estimó. El resultado es que todas las acciones se prueban varias veces antes de que el valor estimado converja. El sistema realiza más exploración, incluso cuando se selecciona la acción mediante el método *greedy*.

Al utilizar el método de *inicialización optimista*, los sistemas convergerán más tarde, ya que realizan más exploración, pero llegarán con una probabilidad mayor a ejecutar la acción óptima. Este método es muy simple, pero bastante efectivo para problemas estacionarios, pero no se puede generalizar como útil para todos los problemas de exploración. Por ejemplo, si se trata de un problema no estacionario no sirve, ya que su exploración es inherentemente temporal. Si la tarea cambia, se crearía una necesidad nueva de exploración y este método no podría ayudar.

2.5. Discusión

En este capítulo se ha revisado el estado de la cuestión de los sistemas informáticos educativos, reparando principalmente en las ventajas que ofrece la aplicación de métodos y técnicas de inteligencia artificial proporcionando adaptación individualizada a cada estudiante que interactúa con el sistema. Además, se ha realizado una introducción al aprendizaje por refuerzo y sus principales elementos.

En esta tesis se propone la aplicación del modelo de *Aprendizaje por Refuerzo* en sistemas de enseñanza, planteando diferencias significativas al trabajo de [Beck, 2001].

La aplicación de aprendizaje por refuerzo en los sistemas de enseñanza solventa uno de los mayores problemas de este tipo de sistemas: la definición de las estrategias pedagógicas a seguir, dependiendo de cada estudiante en cada momento. Se entiende como estrategia pedagógica a la secuencia de tareas (definiciones, introducciones, ejemplos, resúmenes, etc.) que realiza el tutor del curso con el propósito de enseñar la materia a los alumnos.

Hasta el momento, la mayoría de los sistemas de educación se construían a partir de sistemas expertos basados en reglas. Estas reglas eran costosas de codificar, por la existencia de una gran cantidad de reglas para lograr una mayor adaptación al usuario, y por la dificultad de incorporar el conocimiento de los expertos (tutores humanos) en el sistema.

Otra característica importante del modelo de aprendizaje por refuerzo es la flexibilidad ante el ruido. Por ejemplo, hay situaciones en las que no es fácil determinar el conocimiento que tiene un estudiante sobre un tema dado, por lo que es sencillo cometer errores en la evaluación de su conocimiento.

Puesto que los únicos trabajos que han utilizado *aprendizaje por refuerzo* en este tipo de sistemas son los de [Beck, 2001], a continuación se detallan las principales diferencias encontradas:

1. **Aplicación de *Aprendizaje por Refuerzo*.** La diferencia principal se basa en la aplicación del modelo de *Aprendizaje por Refuerzo*. Beck aplica esta técnica en el modelo del estudiante, definiendo el *estado* del sistema como todas las características de aprendizaje del estudiante, incluyendo el conocimiento que tiene el estudiante sobre los temas principales descritos en el módulo del dominio entre otras muchas características del usuario. Beck, por lo tanto, aplicaba la inteligencia artificial al modelo del estudiante, adaptándose a cada uno de ellos dependiendo de todas sus características de aprendizaje. Esta solución tiene un problema fundamental, y es que las características de aprendizaje del estudiante a tener en cuenta son muchas y muy variadas, creciendo el espacio de estados y dificultando el aprendizaje. Además, el sistema aprende muy despacio, ya que el programa aprende a comportarse a partir de la interacción con otros estudiantes de exactamente las mismas características de aprendizaje. Por ello, en esta tesis se propone la aplicación del modelo de *Aprendizaje por Refuerzo* principalmente en el módulo pedagógico del programa de enseñanza, donde el *estado* se define como únicamente el conocimiento que tiene el estudiante sobre los temas del dominio. La principal ventaja de nuestro sistema en este punto es que el sistema es capaz de adaptarse de forma dinámica a los estudiantes que interactúan con el sistema, sin necesidad de contemplar como características del estudiante otras características distintas a su conocimiento sobre los temas del dominio. Una mejora a nuestro sistema sería realizar un modelado de estudiante detallado y una posterior clasificación de los mismos dependiendo de sus características de aprendizaje. Una vez clasificados, al interactuar con nuestro sistema, éste permitiría una mayor y más rápida adaptación a los estudiantes.
2. **Adaptación a bajo nivel en el árbol de conocimiento.** En el trabajo de Beck se aplica el modelo de *Aprendizaje por Refuerzo* a los temas del árbol de conocimiento (a alto nivel), aprendiendo secuencias de contenidos de temas. En nuestro trabajo se propone aplicar este mismo modelo a las tareas de cada uno de los temas, secuenciando el contenido del sistema a bajo nivel en el árbol del dominio (*definiciones, introducciones, ejercicios, etc.*), siendo capaz también de decidir en qué formato presentar los contenidos a los estudiantes según sus preferencias cuando estas tareas se encuentran almacenadas en el sistema en distintos formatos.
3. **Política de aprendizaje.** También se estudiará la importancia de la elec-

ción de la política de aprendizaje del sistema desde el punto de vista del algoritmo de *Aprendizaje por Refuerzo*, donde se evaluarán las ventajas e inconvenientes de utilizar estrategias de exploración y explotación a la hora de elegir el siguiente conocimiento a mostrar al estudiante. Beck utiliza en su sistema una política avariciosa (*e-greedy*), mientras que otros tipos de políticas podrían ser ventajosas a la hora de presentar el conocimiento al estudiante, como la política de *Boltzmann*.

Capítulo 3

Propuesta

En el capítulo anterior se han expuesto las ventajas que existen al aplicar técnicas y métodos de inteligencia artificial en los sistemas de educación a distancia. Gracias a la aplicación de inteligencia artificial a este tipo de sistemas informáticos, estos sistemas son capaces de adaptarse a cada estudiante que interactúa con ellos individualmente, según las necesidades pedagógicas que tengan en cada instante.

En este capítulo se expone la propuesta de aplicar *Aprendizaje por Refuerzo* en el dominio de los sistemas informáticos de educación a distancia, a través de Internet, representando de forma implícita la política pedagógica que adopta el sistema con cada estudiante.

Como se comentó en el apartado anterior, hasta el momento, la mayoría de los sistemas educativos se basan en sistemas expertos donde es necesario definir cada estrategia pedagógica a seguir cuando se interacciona con cada estudiante en cada instante de la interacción. La definición de cada una de las estrategias pedagógicas era difícil y costosa de llevar a cabo.

Al definir el problema de adaptación de secuencias de contenidos como un problema de *Aprendizaje por Refuerzo*, se evita tener que definir una a una cada estrategia pedagógica. Además, el sistema es capaz de filtrar pequeñas ocurrencias de ruido en el dominio como, por ejemplo, algunos errores cometidos al evaluar el grado de conocimiento de los estudiantes ante un determinado tema del curso que se desea que aprenda. El modelo propuesto se ha denominado RLATES (Reinforcement Learning in Adaptive and Intelligent Educational Systems).

En este capítulo, en la sección 3.1, se expone la motivación de la propuesta; en la sección 3.2 se exponen los detalles de la aplicación del modelo de *Aprendizaje por Refuerzo* representando de forma implícita las tácticas pedagógica del sistema educativo; posteriormente, en la sección 3.3, se detallan los procesos de diseño e implementación del sistema RLATES, definiendo cada uno de sus módulos e introduciendo la implementación del sistema; para finalizar, en la sección 3.4 se

presentan detalles de adaptación del interfaz de usuario derivados de la aplicación del modelo de *Aprendizaje por Refuerzo*.

3.1. Motivación

Un *Sistema de Educación a Distancia* es un sistema software que utiliza técnicas de inteligencia artificial para representar el conocimiento e interactúa con los estudiantes con el objetivo de que éstos lo aprendan [VanLehn, 1988]. El aprendizaje por parte del alumno supone el desarrollo de nuevo conocimiento y comprensión a través de una asimilación individual [Rea *et al.*, 2000]. Esto implica que debe existir una comunicación clara y efectiva que facilite la interacción entre el estudiante y el tutor de forma que se mantenga la atención del alumno, adaptándose dinámicamente en cada momento a sus necesidades.

Uno de los principales problemas en los sistemas educativos es precisamente éste: adaptarse a las necesidades del usuario con el que interactúa en cada instante. Para conseguir esta adaptación dinámica, el sistema ha de definir, entre otras cosas, la política de enseñanza a seguir (secuencias del contenido del curso, realimentación en la enseñanza, formato en el que se muestra el contenido, etc.) para cada interacción con el estudiante. Un medio de adaptación al usuario es mediante las llamadas estrategias pedagógicas, que especifican cómo se secuencia el contenido, qué tipo de realimentación se proporciona en la enseñanza, cuándo y cómo se muestra, explica o resume el contenido del tutor (problemas, definiciones, ejemplos, etc.) [Murray, 1999b].

La elección de una estrategia de enseñanza no es tarea sencilla, planteándose la necesidad de definir distintas estrategias según las necesidades de cada usuario y especificando en qué situaciones se diferencian [Khan and Yip, 1996], con la problemática añadida de, si falla en la elección, encontrar el por qué. En la bibliografía se pueden encontrar diversas técnicas de definición de políticas pedagógicas (planificación, heurísticas, etc.), pero éstas suelen introducir problemas difíciles de resolver (ver sección 2.3.1).

En este trabajo se propone definir el problema de los sistemas educativos como un problema de *Aprendizaje por Refuerzo*, de forma que el sistema aprende a enseñar de forma óptima a cada alumno, basándose únicamente en la experiencia adquirida con otros alumnos con características de aprendizaje similares a las suyas. El sistema será capaz de aprender la secuencia de tareas (o acciones) en el formato adecuado para cada estudiante, dependiendo de sus características de aprendizaje. De esta forma, se evita tener que definir una a una las posibles estrategias pedagógicas a seguir. Para ello, se propone la utilización de técnicas de *Aprendizaje por Refuerzo* en el módulo pedagógico del tutor. Esta técnica permite definir tácticas de comportamiento basadas en el conocimiento del alumno en

cada instante, en lugar de utilizar tácticas predefinidas de antemano.

El aprendizaje del tutor inteligente se realiza mediante prueba y error de forma paralela al aprendizaje del estudiante. Este método se apoya en la enseñanza tradicional, donde los tutores humanos aprenden a hacer frente a los alumnos a través de un largo proceso de prueba y error [Rea *et al.*, 2000].

3.2. Adaptación Dinámica Mediante Aprendizaje por Refuerzo

En este apartado se propone una representación para la definición del módulo pedagógico de un sistema de educación *Web* desde el punto de vista del *Aprendizaje por Refuerzo*. Para realizar una representación completa, a continuación se definen cada una de las características de los problemas de *Aprendizaje por Refuerzo*, aplicándolas al entorno de los sistemas de enseñanza. Al modelo propuesto se le ha llamado RLATES (Reinforcement Learning in Adaptive and Intelligent Educational Systems).

3.2.1. Descripción del Modelo

En este apartado se realiza una descripción a alto nivel de la forma en la que se ha aplicado el modelo de *Aprendizaje por Refuerzo* al dominio de los *Sistemas Educativos Adaptativos e Inteligentes en Web*, describiendo en un nivel conceptual cada componente que participa en el modelo.

En la Figura 3.1 se presentan los componentes del modelo de *Aprendizaje por Refuerzo* aplicados a los SEAI en *Web*. En el dominio de los sistemas de educación a distancia, los estudiantes interactuarán con el sistema de educación (en nuestro caso RLATES) para aprender los contenidos del curso almacenados en el sistema. El sistema almacena toda la información del curso en un módulo denominado *módulo del dominio* (ver sección 2.1.4). El contenido del curso se almacenará en forma de conceptos que pueden resultar interesantes para el estudiante.

Antes de interactuar con el sistema, cada estudiante se encontrará en un determinado estado de conocimiento (s) sobre el material del curso que imparte RLATES. Este estado de conocimiento el sistema lo percibe evaluando el grado de conocimiento que posee el alumno sobre cada uno de las unidades de conocimiento que contiene en su módulo del dominio (mediante exámenes o tests de evaluación). Al evaluar el grado de conocimiento del estudiante, el sistema puede percibir datos de evaluación ruidosos, percibiendo un estado de conocimiento del estudiante distinto al que realmente posee. Esto se puede deber a muy diversos

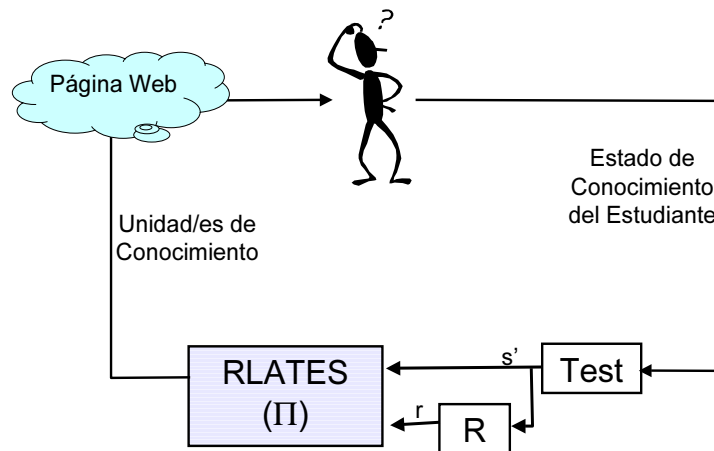


Figura 3.1: Esquema de Aprendizaje por Refuerzo aplicado a los SEAI

factores: por ejemplo, si el examen es de tipo test, el estudiante podría acertar la respuesta correcta por casualidad; o podría ocurrir que el estudiante no leyera bien la pregunta del examen y contestara de forma errónea conociendo la respuesta correcta; o el estudiante podría despistarse al realizar el examen y no contestar adecuadamente a las preguntas; etc. Al estado que percibe el sistema tras evaluar el conocimiento del estudiante se le denomina s' .

Una vez que el sistema ha percibido el estado de conocimiento en el que se encuentra el estudiante, éste evalúa cómo de bueno es el estado actual del estudiante. Normalmente, el estudiante desea alcanzar un objetivo que en sistemas de educación suele coincidir con aprender todo el conocimiento del curso. En este caso, el sistema supondrá que cuando el estudiante conoce todos los conceptos de su modelo del dominio, éste ha llegado a su objetivo, encontrándose en un estado realmente bueno para el estudiante actual. Siempre que el estudiante modifica su estado de conocimiento, el sistema percibe una señal de refuerzo (r) indicando la bondad del estado de conocimiento nuevo. El sistema tendrá en cuenta esta señal de refuerzo para seguir una política de comportamiento adecuada para el estudiante (II).

Según el estado de conocimiento percibido por el sistema, éste tomará una decisión: si el estudiante no ha llegado a su objetivo, RLATES elegirá un concepto nuevo para mostrar al estudiante; mientras que si el estudiante ha llegado a su estado meta, RLATES finaliza la interacción con el estudiante actual.

Cuando el estudiante estudia los conceptos que el sistema le ha proporcionado (creando dinámicamente una página web si se trata de un sistema de educación *Web*), este estudiante podrá cambiar o no de estado de conocimiento sobre el dominio, dependiendo de si los conceptos mostrados por el sistema eran pe-

dagógicamente adecuados para que el estudiante aprendiera en ese instante de la interacción. En este punto se repetiría el ciclo: el sistema tendría que volver a evaluar el conocimiento del estudiante, mostrarle nuevos conceptos, y continuar de esta manera hasta que el alumno llegara a un estado meta.

La elección de los siguientes conceptos a mostrar al estudiante en RLATES se basa en el conocimiento que posee de previas interacciones con otros estudiantes de las mismas características de aprendizaje. Este conocimiento en sistemas de aprendizaje por refuerzo se puede almacenar en varias funciones de aprendizaje (ver sección 2.4.5). En RLATES se ha utilizado la función de *valor-acción* ($Q(s, a)$). El aprendizaje de la función $Q(s, a)$ puede llevarse a cabo de diversas formas, desde programación dinámica [Bellman, 1957] hasta métodos libres del modelo [Sutton, 1988]. En RLATES el algoritmo de aprendizaje por refuerzo que aprende la función $Q(s, a)$ es el algoritmo *Q-learning* [Watkins, 1989] (ver sección 2.4.6). Este algoritmo permite que el sistema aprenda directamente de la experiencia, basándose en la función de *valor-acción* ($Q(s, a)$) para definir su política de acción. Una de las características principales de este algoritmo es que no necesita que el sistema ejecute secuencias óptimas para converger [Mitchell, 1997], siendo capaz de aprender la función Q y, de esta forma, la política de comportamiento óptima. Esta característica es muy importante en los sistemas de enseñanza, ya que el aprendizaje está basado en la interacción con los usuarios y esto implica un alto coste.

El algoritmo Q-learning adaptado al dominio de los sistemas de enseñanza adaptativos e inteligentes en *Web* (adaptado al sistema RLATES) se describe en la Figura 3.2. Este algoritmo muestra cómo el sistema genera dinámicamente la tabla de *valor-acción* (tabla Q), que permite a RLATES seguir potencialmente una política de comportamiento óptima. Esto quiere decir que, en cada momento, al alumno se le enseña el conocimiento más apropiado en el formato más apropiado. En la función de actualización de $Q(s, a)$, el parámetro γ modifica la importancia de las futuras acciones con respecto a las actuales, y el parámetro α es el ratio de aprendizaje del sistema, que indica cómo de rápido aprende el sistema.

3.2.2. Componentes del Modelo

En este apartado se definen en detalle los componentes del modelo de *Aprendizaje por Refuerzo* adaptados al dominio de los Sistemas de Educación en Web Adaptativos e Inteligentes (SEAI en *Web*) mostrados en la Figura 3.1. Dichos componentes son:

1. **Conjunto de estados (S):** Se define un estado del modelo como la descripción del conocimiento del estudiante y se representa mediante un vector de tantos elementos como ítems de conocimiento se desea que aprenda

el usuario (número de temas del modelo del dominio). Cada componente del vector almacena valores representativos del conocimiento que posee el usuario del tema al que se refiere, habiendo numerado los temas del árbol de conocimiento en *pre-orden*. Estos valores pueden ser discretos o continuos, teniendo en cuenta que los valores continuos aumentan la complejidad del método de aprendizaje, al existir un número de estados infinito y tener que aplicar mecanismos de discretización (generalización) de estados [Santamaría *et al.*, 1998], pero aporta una mayor aproximación a la solución óptima del problema. El sistema percibe el estado de conocimiento del estudiante mediante evaluaciones (que habitualmente se realizan mediante exámenes o tests).

2. **Conjunto de acciones (A):** Las acciones que puede realizar el sistema de educación son las de mostrar cada uno de los temas del modelo del dominio. Un sistema de educación en Internet mostrará este conocimiento a través de páginas web que muestran las diferentes tareas de los temas (definiciones, introducciones, ejemplos, etc.). Una ventaja del sistema sería que éste permitiera que el programa de educación pudiera mostrar varios ítems de conocimiento al mismo tiempo (en la misma página web) si lo creyera oportuno (de esta forma no se perdería generalidad limitando el número de ítems de conocimiento mostrados en cada instante). Se define una macro-acción al conjunto de varias tareas del árbol de conocimiento que serán mostradas a los estudiantes al mismo tiempo. De nuevo hay que tener en cuenta en este punto que el conjunto de acciones puede tener un tamaño demasiado grande para que el problema sea computacionalmente resuelto, requiriendo la utilización de mecanismos de generalización (integración de acciones) para solucionarlo.
3. **Percepción del estado actual ($Test : S \rightarrow S'$):** Esta Función indica la manera en la que el agente (sistema de educación) percibe cada uno de los estados en los que se puede encontrar un estudiante. Percibir las consecuencias de la ejecución de una acción dada cuando el estudiante se encuentra en un determinado estado no es tarea fácil. Como se indicó anteriormente, los estados representan el conocimiento que posee el alumno de la materia que se pretende que aprenda. Realizar la evaluación del conocimiento de una persona (por ejemplo, mediante exámenes, tests, etc.) es una de las tareas más complicadas en el mundo de la enseñanza, existiendo siempre una probabilidad de que los resultados de la evaluación no sean absolutamente fieles. La función I permite definir probabilidades de que el estado del entorno que se percibe sea cierto o varíe en alguno de sus componentes. De esta forma se permite realizar inferencias sobre el conocimiento del usuario.

4. **Refuerzo** ($R : SxA \rightarrow R$): Esta función define el conjunto de señales de refuerzo proporcionadas por el entorno. Esta señal de refuerzo la proporciona el entorno cuando se realiza una transición de estados provocada por la ejecución de una acción. Estas señales de refuerzo le dan más valor a unas transiciones que a otras dependiendo de la acción que se ejecute en un determinado estado. La función de refuerzo que se vaya a utilizar ha de dar un refuerzo máximo al llegar al objetivo del tutor. Por ejemplo, un objetivo válido del tutor podría ser que el usuario haya aprendido todos los ítems de conocimiento que se le mostraron. Se puede determinar, de esta forma, que el objetivo final del sistema de educación será maximizar la señal de refuerzo a lo largo del tiempo. La aplicación de la señal de refuerzo es muy importante en el modelo de *Aprendizaje por Refuerzo*, ya que decidir cuándo y cómo aplicar el refuerzo será crucial para el aprendizaje del sistema. El aprendizaje del sistema de educación es un problema de refuerzo positivo retardado, dado que es imposible determinar la bondad del estado actual de un estudiante inmediatamente después de ejecutar una acción. En algunos casos, el sistema no sabe si la acción ejecutada cuando el estudiante se encontraba en un determinado estado era buena o no hasta ejecutar alguna acción más y evaluar posteriormente si el estudiante ha llegado a un estado de conocimiento positivo según la meta planteada por el estudiante, por ejemplo, llegando al estado objetivo. De hecho, se proporcionará un mayor refuerzo en las transiciones de estado que hagan que el estudiante aprenda mejor y en menor tiempo (ejecutando un número menor de acciones). El comportamiento del sistema será elegir las acciones que tiendan a maximizar la suma de los valores de la señal de refuerzo, eligiendo de esta forma las tácticas pedagógicas óptimas (qué, cuándo y cómo enseñar; la mejor secuencia de contenidos y cómo enseñarlos) para enseñar al estudiante actual.

5. **Función de Valor-Acción** ($Q : SxA \rightarrow R$) Esta función estima la utilidad de ejecutar una determinada acción a , (mostrar al estudiante el contenido de una hoja del árbol de conocimiento) cuando el alumno se encuentra en un determinado estado, s . La función proporciona un método de evaluación para distintas políticas de acción, $\pi(s)$, al sistema de educación, definidas como $\pi(s) = \arg \max_{ai} Q(s, ai)$. De hecho, el objetivo del proceso de aprendizaje será encontrar la política de acción que maximice esta función $Q(s, a)$, es decir, encontrar la función de valor-acción óptima ($Q^*(s, a)$) tal que $Q^*(s, a) \geq Q(s, a) \forall a \in A, s \in S$.

3.2.3. Exploración vs. Explotación

Como se ha expuesto en el apartado anterior, el algoritmo de *Q-learning*, en un momento dado, ha de seleccionar una acción de entre todas las que existen en el entorno (selecciona conocimiento del modelo del dominio a mostrar a continuación al estudiante). Para realizar esta selección, el sistema se basa en una política de acción (π) derivada de la función de *valor-acción* ($Q(s, a)$).

Elegir la siguiente acción a ejecutar en el algoritmo *Q-learning* es muy importante para la convergencia del algoritmo. La calidad de las estrategias de exploración/explotación es determinante en la eficiencia y eficacia en alcanzar los objetivos fijados. En este trabajo se ha comparado la eficiencia y eficacia del sistema con las estrategias de exploración/explotación de *Boltzmann* y *e-greedy*, introducidas en la sección 2.4.7.

En este punto es importante analizar las ventajas de utilizar una u otra estrategia de exploración/explotación en el dominio de los *Sistemas de Educación Adaptativos e Inteligente en Web*. Para ello se tendrán en cuenta sus diferencias, ventajas y desventajas en dicho dominio.

En primer lugar, hay que tener en cuenta que cuando no se trata de la política *I-greedy*, la probabilidad que introduce el parámetro ϵ en esta política hace que no siempre se elija la acción que mayor valor $Q(s, a)$ posee, eligiendo de forma totalmente aleatoria otra acción del entorno. Al elegir otra acción de forma totalmente aleatoria, podría ocurrir que se tomara alguna cuya $Q(s, a)$ sea realmente mala. Sin embargo, al utilizar la estrategia de exploración/explotación de *Boltzmann*, se proporciona una probabilidad de ejecutar una acción dada de forma normalizada. De esta forma, en el caso en el que no se deseara ejecutar la acción con mayor probabilidad (para seguir una política que no fuera demasiado avariciosa y realizar una mayor exploración por el dominio de las acciones), existe un baremo de probabilidades que nos sugiere la bondad de ejecutar una determinada acción en un momento dado, pudiendo elegir la siguiente acción a ejecutar en base a esta probabilidad.

Además, en el dominio de los sistemas de enseñanza, la estrategia de exploración/explotación de *Boltzmann* podría añadir una ventaja frente a la estrategia de explotación/exploración de *e-greedy*: dejar que los estudiantes sean los que decidan qué acción ejecutar a continuación (qué contenidos del curso se verán en la siguiente página web). Si a los estudiantes se les proporcionara una medida de bondad de las acciones, ellos mismos podrían decidir qué acción ejecutar a continuación. La probabilidad de ejecución de acciones de la política de *Boltzmann* puede proponerse como esta medida de bondad. De esta forma, a los estudiantes se les puede proporcionar varias acciones para ejecutar a continuación y dejar que sean ellos los que decidan qué estudiar en la siguiente página web, en lugar de proporcionarles una sola página web, como haría la estrategia de ex-

ploración/explotación de *e-greedy*. Al presentar las siguientes acciones posibles a ejecutar, también se les puede proporcionar la probabilidad que estima el sistema de que la acción sea ejecutada, proporcionando al estudiante una medida que facilite la elección de la acción a tomar.

En los sistemas de educación a distancia (como RLATES), es imprescindible mantener la atención de los estudiantes, proporcionándoles la sensación de que es el estudiante el que posee el control de la interacción. En RLATES se ha propuesto la utilización de la estrategia de exploración/explotación de *Boltzmann* para conseguir estas características en el sistema, proporcionando al estudiante, como se verá en el apartado 3.4, varias acciones como opción a la siguiente página a visitar. Será el propio estudiante el que decida en cada momento qué contenidos del curso visitar a continuación.

3.2.4. Ejemplo de Aprendizaje en RLATES

En este apartado se presenta un ejemplo de modificación de la tabla Q en el proceso de aprendizaje del sistema. El modelo del dominio que se desea que los estudiantes aprendan se presenta en la Figura 3.3. En esta figura se puede observar que se pretende que el estudiante aprenda tres temas, y que de cada uno de ellos se cuenta con una introducción y una definición.

Para simplificar el ejemplo en este apartado, se supondrá que el comportamiento de los estudiantes que interactúan con el sistema es totalmente determinista. En la Figura 3.4 se muestra una parte del *Proceso de Decisión de Markov* (del inglés *Markov Decision Process*, MDP) que describe el comportamiento que se ha simulado del estudiante (las posibles transiciones de estado del estudiante cuando interacciona con el sistema). El MDP es un modelo de comportamiento del estudiante.

El estado S muestra cómo el sistema aún no conoce el tema E/R ¹ (contienen un 0 en la posición del vector), pero sí conoce el tema *Entidad*² y el tema *Atributo*³ (contiene un 1 en la posición del vector). Por otro lado el estado *Meta* es el estado objetivo del estudiante, que se corresponde cuando éste conoce todos los temas del árbol de conocimiento (posee un 1 en cada posición del vector).

En el MDP de la figura 3.4 se muestra que la única forma de aprender el tema E/R cuando el estudiante se encuentra en el estado S es cuando se ejecuta la acción AI (le muestra la página web que contiene la *Definición1* propia de este tema). Ésta será la única forma en la que el estudiante pueda llegar al estado *Meta* pasando por el estado S . Si el sistema ejecuta cualquier otra acción distinta

¹Tema dedicado al modelo *Entidad/Interrelación* (E/R), utilizado en el análisis conceptual de bases de datos, dominio de enseñanza sobre el que se ha implementado RLATES.

²Elemento del modelo *Entidad/Interrelación*.

³Elemento del modelo *Entidad/Interrelación*.

a la acción $A1$ cuando el estudiante se encuentra en el estado S , el estudiante permanecerá en el mismo estado.

Se supondrá en este ejemplo, entonces, que el estudiante se encuentra en el estado inicial S . Además, se supone que el ratio de aprendizaje del sistema (α) es igual a 0,9, es decir, el sistema aprende muy rápido. El parámetro de descuento en futuras acciones, γ , también se ha fijado a 0,9 y la señal de refuerzo (r) que proporciona el sistema será igual a 1 cuando el estudiante llega a un estado meta y cero en cualquier otro caso. La estrategia de exploración/explotación utilizada será la de *Boltzmann*.

Por otro lado, se inicializa la tabla Q a cero, para comprobar cómo se modifican los valores de esta tabla según el estudiante interacciona con el sistema. En la Figura 3.5 se muestra parte de la tabla Q del sistema, cuando los estados del estudiante son S y $Meta$. Por definición del algoritmo de aprendizaje por refuerzo, $Q(Meta, a_i)$ es igual a cero para toda acción (a_i) del entorno. El resto de valores de la tabla Q también se han inicializado a cero.

Supongamos en este punto que el sistema, aleatoriamente, elige la acción $A6$ para ser ejecutada. Se ha elegido de forma aleatoria ya que todas las acciones poseen el mismo valor en la tabla Q y, por tanto, tienen la misma probabilidad de *Boltzmann* de ser elegidas por esta estrategia de exploración/explotación.

Según el grafo de transiciones de la Figura 3.4, cuando a un estudiante que se encuentra en el estado S se le muestra la acción $A6$, éste no cambia de estado, pero se modificará la tabla Q en la posición $Q(S, A6)$ siguiendo la función de modificación del algoritmo *Q-learning* (ver ecuación 3.1):

$$Q(S, A6) = 0, 1 * 0 + 0, 9(0 + 0, 9 \text{ máx}(0, 0, 0, 0, 0, 0)) = 0$$

En este caso, la tabla Q no varía con respecto a la inicial después de haber modificado la posición de la tabla $Q(S, A6)$. En la Figura 3.6 se muestra la tabla tras la primera ejecución de una acción (la acción $A6$).

Supongamos en este instante que el sistema elige ejecutar la acción $A1$. En este caso, según el grafo de transiciones del estudiante, éste, tras haber mostrado la información de la acción $A1$, cambia de estado al estado $Meta$. De nuevo, la tabla Q se modifica en la posición $Q(S, A1)$ según la función de *Q-learning*:

$$Q(S, A1) = 0, 1 * 0 + 0, 9(1 + 0, 9 \text{ máx}(0, 0, 0, 0, 0, 0)) = 0, 9$$

Por lo tanto, la tabla Q queda en este momento como se muestra en la Figura 3.7 y finaliza la interacción del estudiante actual al llegar a un estado meta. Es importante resaltar que el siguiente estudiante con las mismas características de aprendizaje que el anterior que interactúe con el sistema partirá de la tabla Q de la Figura 3.7. Por lo tanto, cuando el siguiente estudiante se encuentre en el estado S , la acción $A1$ tendrá una mayor probabilidad de ser elegida según la estrategia de exploración/explotación de *Boltzmann*.

Es importante resaltar en este punto que, como se ha podido comprobar en el ejemplo, si se inicializa la tabla Q a cero, inicialmente el sistema elegirá de

forma totalmente aleatoria las acciones a ejecutar en cada instante. En RLATES los refuerzos se van transmitiendo hacia atrás, por lo que los primeros estudiantes que interaccionen con el sistema no conseguirán interaccionar de la forma más adecuada según sus características de aprendizaje hasta que el sistema consiga una actualización de la tabla Q apropiada. Por ello, se propone inicializar la tabla Q del sistema para que los primeros estudiantes no sufran la falta de política pedagógica en el sistema. En el capítulo 5 se evalúa la posibilidad de realizar la inicialización y en el capítulo 6 se realizan experimentos con estudiantes reales habiendo inicializado la tabla Q previamente.

Por último, hay que recordar que el MDP del estudiante (mostrado para este ejemplo en la Figura 3.4) es desconocido en la realidad, lo que hace requerir el proceso de aprendizaje mediante prueba y error.

Algoritmo Q-learning adaptado a los SEAI en Web

- Inicializar $Q(s, a)$ (pudiendo incluir información sobre el modelo del dominio)
- Repetir (para cada alumno)
 - Evaluar el estado de conocimiento del estudiante actual (s) mediante exámenes o tests.
 - Repetir
 - Seleccionar una acción a (elemento de conocimiento del modelo del dominio a enseñar al alumno) a partir de s usando una política, π , derivada de Q y siguiendo una estrategia de exploración apropiada.
 - Ejecutar la acción a (creando una página web que contenga el conocimiento seleccionado y mostrar al estudiante). Observar el refuerzo inmediato recibido, r , y el siguiente estado, s'
 - Actualizar la entrada de la tabla, $Q(s, a)$ con la ecuación:

$$Q_t(s, a) = (1 - \alpha_t)Q_{t-1}(s, a) + \alpha_t[R(s, a) + \gamma \max_{a'} Q_{t-1}(s', a')] \quad (3.1)$$

- $s = s'$

Hasta que s sea un estado final (el alumno aprenda todo el conocimiento)

Figura 3.2: Algoritmo *Q-learning* adaptado al dominio de los Sistemas de Educación Adaptativos e Inteligentes en Web.

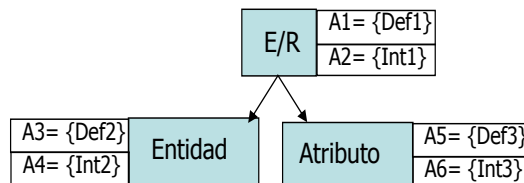


Figura 3.3: Ejemplo de Aprendizaje en RLATES. Modelo del dominio.

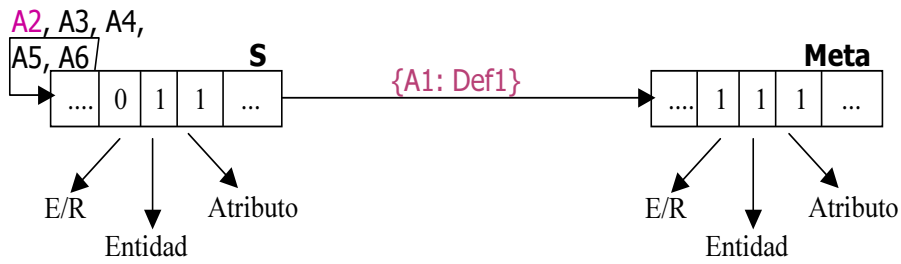


Figura 3.4: Ejemplo de Aprendizaje en RLATES. MDP del estudiante.

Q(s,a)	A1	A2	A3	A4	A5	A6
S	0	0	0	0	0	0
Meta	0	0	0	0	0	0

Figura 3.5: Ejemplo de Aprendizaje en RLATES. Tabla Q inicializada a cero

Q(s,a)	A1	A2	A3	A4	A5	A6
S	0	0	0	0	0	0
Meta	0	0	0	0	0	0

Figura 3.6: Ejemplo de Aprendizaje en RLATES. Tabla Q. Paso 1

Q(s,a)	A1	A2	A3	A4	A5	A6
S	0.9	0	0	0	0	0
Meta	0	0	0	0	0	0

Figura 3.7: Ejemplo de Aprendizaje en RLATES. Tabla Q. Paso 2

3.2.5. Fases de Funcionamiento de RLATES

Es necesario para el correcto funcionamiento de RLATES y para que se adapte mejor a cada estudiante en cada momento de la interacción, que se completen sus tres fases de funcionamiento: la agrupación de estudiantes, el entrenamiento del sistema y el uso del sistema. La Figura 3.8 muestra las tres fases de funcionamiento del sistema.

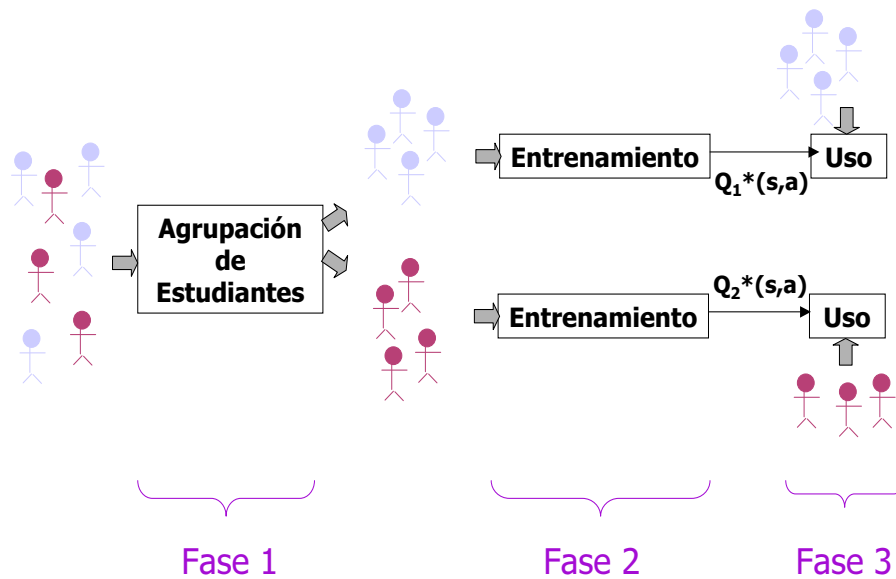


Figura 3.8: Fases Funcionales de RLATES

1. **Agrupación de los Estudiantes.** RLATES es capaz de adaptarse a cada estudiante individualmente, modificando su función de *valor-acción* ($Q(s, a)$) según la interacción que realiza cada estudiante con el sistema. Si el sistema mantuviera una sola tabla Q para todos los estudiantes que interactúan con el sistema, RLATES adaptaría esta tabla Q al conjunto de todos los estudiantes. Sin embargo, los alumnos podrían tener distintas características de aprendizaje y la adaptación del sistema a cada uno de ellos podría no ser muy buena. Si se realizara una previa agrupación de los estudiantes según sus características de aprendizaje (por ejemplo, asignándole un *estereotipo* a cada estudiante como se hizo en el sistema GRUNDY [Rich, 1979]) y el sistema mantuviera una tabla Q para cada grupo de estudiantes, esto permitiría que el sistema se adaptara más rápido y mejor a cada grupo de estudiantes y, por lo tanto, a cada estudiante individualmente. Esta fase no es necesaria, pero se recomienda para conseguir una mejor adaptación del sistema a

cada tipo de estudiantes. Posteriormente en este trabajo, en el capítulo 5, se analiza el comportamiento del sistema al interactuar con distintos grupos de estudiantes, variando la homogeneidad de las características de aprendizaje de los estudiantes de cada grupo.

2. **Entrenamiento del Sistema.** En esta fase, el sistema explora nuevas alternativas pedagógicas para llegar al objetivo, mostrando el conocimiento del curso a los estudiantes en diferente orden de secuencia. Al mismo tiempo que el sistema interactúa con los estudiantes, éste modifica apropiadamente su tabla Q . De esta forma, el sistema es capaz de aprender una política pedagógica adecuada basándose únicamente en interacciones previas del sistema con otros estudiantes de sus mismas características de aprendizaje (del mismo grupo de estudiantes).
3. **Uso del Sistema.** Una vez que el sistema ha convergido a una buena política pedagógica, es el momento de utilizar esta información para enseñar a otros estudiantes con las mismas características de aprendizaje de la manera adecuada. Estos estudiantes alcanzarán sus objetivos de aprendizaje mostrándoles la información del contenido del curso en la secuencia que el sistema cree más adecuada.

Aunque la interacción de los estudiantes con el sistema se ha dividido en dos fases (fase de *entrenamiento* y fase de *uso*), el sistema nunca finaliza su proceso de aprendizaje, adaptando continuamente su función de *valor-acción* ($Q(s, a)$) en cada momento de la interacción con cada uno de los estudiantes. Esta división se ha realizado en este trabajo para poder distinguir en el proceso de aprendizaje del sistema cuándo RLATES realiza más exploración (descubriendo la utilidad de ejecutar nuevas acciones cuando el estudiante se encuentra en un determinado estado de conocimiento) o más explotación (aprovechando el conocimiento anterior al ejecutar acciones). El sistema realizará más exploración en la fase de *entrenamiento* y más explotación en la fase de *uso* del sistema, una vez que considera que la secuencia de contenidos que muestra a los estudiantes es la más adecuada según sus características de aprendizaje.

La situación ideal en un sistema de estas características es que su fase de *entrenamiento* se redujera el máximo posible. Para minimizar esta fase de *entrenamiento* se han realizado unos experimentos que se analizan en profundidad en el capítulo 5.

3.3. Diseño e Implementación del Sistema RLATES

En RLATES se propone implementar la arquitectura típica de un sistema *Tutor Inteligente* (ver sección 2.1.4), componiéndose de cuatro módulos bien diferenciados, mostrados en la Figura 3.9, el módulo del estudiante, el módulo del dominio, el módulo pedagógico y el módulo del interfaz. Además, como se puede comprobar en la figura, los estudiantes pueden acceder al sistema a través de cualquier navegador *Web*, lo que hace que el sistema sea accesible desde cualquier ordenador con acceso a Internet, independientemente de su plataforma, arquitectura o sistema operativo. Como se ha comentado anteriormente, el trabajo realizado en la presente tesis doctoral se centra en el módulo pedagógico, realizando una propuesta de utilización del modelo de *Aprendizaje por Refuerzo* en este módulo de forma que RLATES se pueda adaptar a los diferentes alumnos. Por lo tanto, en el resto de los módulos se han utilizado técnicas que en sistemas de educación anteriores se ha comprobado su buen funcionamiento.

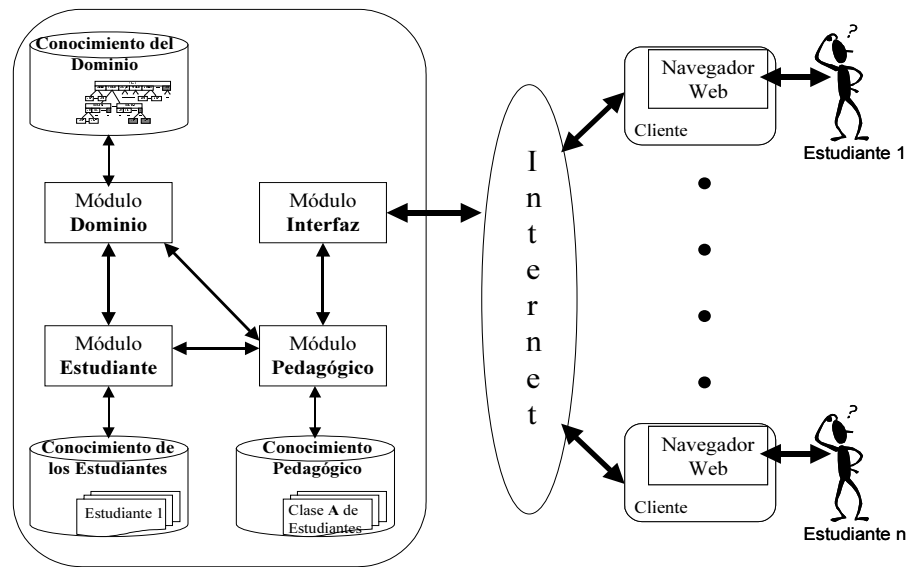


Figura 3.9: Arquitectura de RLATES

3.3.1. Módulo del Estudiante

El módulo del estudiante contiene toda la información importante sobre el estudiante en su proceso de aprendizaje: objetivos, conocimiento anterior sobre la materia que se le desea enseñar, características personales, comportamiento histórico, etc. Todas estas características pueden clasificarse en *datos de usuario*

(por ejemplo, los datos personales), *datos de uso* (por ejemplo, páginas visitadas o conocimiento del temario) y *datos de entorno* (por ejemplo, la versión del navegador que utiliza).

Los sistemas adaptativos basan sus tareas de adaptación en la construcción de un modelo de usuario. Se define modelo de usuario como *una representación explícita de la propiedades de un usuario específico* y se utiliza para razonar, entre otras cosas, sobre el comportamiento futuro del estudiantes, sus preferencias o necesidades [Gaudioso, 2002].

Como se ha comentado anteriormente, en este trabajo no se pretende proporcionar un modelo de usuario completo, sino que sólo se considerarán algunas características de usuario imprescindibles para que el sistema funcione. Igualmente, la representación elegida para el modelo de usuario quizás no sea la más novedosa y/o más adecuada: sólo se pretende, eligiendo un modelo de usuario base, ser capaces de comprobar si el problema de secuenciar del conocimiento en el módulo pedagógico de un sistema de enseñanza se puede representar como un problema del *Aprendizaje por Refuerzo*.

El modelo de usuario y, en concreto el grado de conocimiento que posee el alumno sobre los contenidos del curso, se ha representado en el sistema mediante el *modelo superpuesto* (en inglés *overlay*) [Carr and Goldstein, 1977]. El método de representación *overlay* duplica o superpone el modelo del dominio del sistema de enseñanza, indicando si el estudiante conoce o no un determinado tema del árbol de conocimiento del dominio. El módulo pedagógico basará principalmente su adaptación utilizando el grado de conocimiento que posee cada estudiante sobre el dominio del sistema, como se verá en el apartado 3.3.3.

3.3.2. Módulo del Dominio

El módulo del dominio contiene toda la información que se desea enseñar a los estudiantes. Se ha realizado una estructuración jerárquica del conocimiento del sistema, ya que es ventajosa a la hora de definir una política pedagógica en un sistema de educación. El conocimiento del curso se ha dividido en temas (en inglés *topics*) y sub-temas. Esta estructura implica la relación *es-padre* entre temas. Al mismo tiempo, cada nodo interior del árbol de conocimiento (tema) puede estar compuesto de diversos conceptos (tareas; en inglés *tasks*). Se considera como *concepto* a cualquier unidad de información que el sistema educativo considera interesante para guiar al alumno: *definiciones, introducciones, problemas, ejercicios*, etc. en varios formatos de presentación (*texto, video, imagen*, etc.). Las tareas son los nodos *hoja* del árbol de conocimiento y serán mostrados a los estudiantes en forma de páginas web.

Hay que tener en cuenta en este punto que en nuestro modelo del dominio no es necesario introducir más información del sistema experto (como relaciones *es-*

prerrequisito o están-relacionados), ya que el sistema será capaz de adaptarse a los estudiantes únicamente con el conocimiento básico de temas y tareas de cada tema.

Para evaluar el conocimiento de los estudiantes sobre los temas del árbol de conocimiento, es necesario contar con pequeños exámenes o tests de evaluación para cada uno de los temas (nodos intermedios del árbol de conocimiento). En la Figura 3.10 se muestra la estructura jerárquica de la información llevada a cabo en el módulo del dominio, destacando los tests sombreados en la figura como si fueran cualquier otra tarea del árbol de conocimiento.

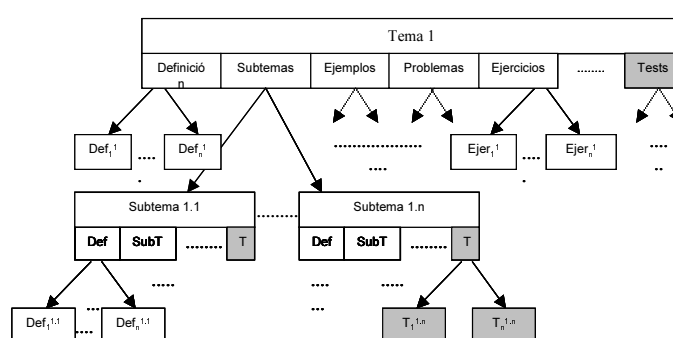


Figura 3.10: Modelo del Dominio. Estructura general de un árbol de conocimiento

3.3.3. Módulo Pedagógico

El módulo pedagógico en el sistema de enseñanza es el encargado de encontrar la mejor forma para enseñar a los estudiantes los items de conocimiento (encontrar la mejor secuencia de items de conocimiento). La secuencias de conocimientos será diferente cuando alumnos de distintas características de aprendizaje interactúan con el sistema, ya que RLATES adapta la secuencia de contenidos a sus necesidades pedagógicas. Este módulo decidirá qué, cómo y cuándo mostrar el contenido del sistema a cada uno de los estudiantes.

En la sección 3.2 se ha analizado la posibilidad de definir el problema de secuenciar el contenido como un problema de *Aprendizaje por Refuerzo*, definiendo de forma general cada uno de los componentes del modelo en el dominio de los sistemas de enseñanza. En este apartado se describen estos componentes teniendo en cuenta la implementación que se ha realizado en RLATES. El objetivo de esta tesis es comprobar que los sistemas de enseñanza se pueden definir como un

problema del modelo de *Aprendizaje por Refuerzo*, por ello la implementación se simplificará teniendo en cuenta los siguientes valores para los componentes del sistema.

1. **Conjunto de estados (S):** Para representar el estado de conocimiento del estudiante en RLATES, se supondrá que los valores del estado de un estudiante se definen en el conjunto $\{0,1\}$. El valor 0 en un elemento del vector, i indica que el estudiante aún no conoce el tema al que representa ese elemento del vector (tema i). Mientras que el valor 1 indica que el estudiante ha aprendido correctamente el tema. En la Figura 3.11 se muestra un posible estado para un estudiante que aprende el modelo del dominio de la Figura 3.10. En este ejemplo se puede ver cómo el estudiante conoce el *Tema 1* y el *Tema 1.n*, pero aún no conoce el *Tema 1.1*. El sistema percibe el estado de conocimiento del estudiante en cada momento cuando éstos contestan a los exámenes que el sistema les propone (denominados *Tests* en la Figura 3.10).

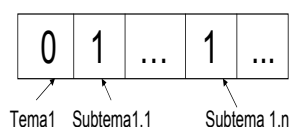


Figura 3.11: Ejemplo de los valores de un estado de estudiante

2. **Conjunto de acciones (A):** RLATES ejecuta una acción mostrando un determinado ítem de conocimiento al estudiante en páginas web. Este conocimiento se mostrará en forma de tareas (o conceptos: *definiciones, introducciones, ejemplos*, etc.).
3. **Percepción del entorno ($I : S \rightarrow S'$):** En RLATES, cada tema en el modelo del dominio del sistema tiene asociados un conjunto de pruebas (exámenes, tests, etc.) que servirán para evaluar cuánto conoce el estudiante sobre ese tema (representados en la Figura 3.10 como rectángulos sombreados).
4. **Refuerzo ($R : S \times A \rightarrow R$):** En RLATES, se proporciona una señal de refuerzo igual a 1 cuando el estudiante consigue llegar a un estado meta, y la señal de refuerzo será igual a 0 en otro caso.
5. **Algoritmo de aprendizaje:** En RLATES se ha implementado el algoritmo de aprendizaje *Q-learning*, basándose en la función de *valor-acción* ($Q(s, a)$). Además, se ha realizado una inicialización no optimista de la función Q , ya que inicializar la tabla con valores optimistas implicaría una

mayor exploración del dominio, característica no recomendable en sistemas que interactúan con estudiantes reales. Por ello, en los capítulos 5 y 6 se ha analizado la posibilidad de inicializar la función de *valor-acción* con información derivada de la interacción con estudiantes simulados.

6. **Estrategia de Exploración/Explotación:** En RLATES se han implementado las estrategias de exploración/explotación *e-greedy* y *Boltzmann*, evaluando en el capítulo 5 las ventajas y desventajas de ambas estrategias al interactuar con estudiantes simulados. Posteriormente, en la evaluación con estudiantes reales se ha utilizado la estrategia de exploración/explotación de *Boltzmann*.

3.3.4. Módulo del Interfaz

Para finalizar, el módulo del interfaz facilita la comunicación entre el sistema de educación y el estudiante. Este módulo aplica técnicas inteligentes y adaptativas con el objetivo de adaptar el contenido y la navegación a los estudiantes, basándose en información proveniente del módulo pedagógico.

En la sección 3.4 se trata en profundidad el tipo de adaptación que se ha realizado en este módulo derivadas siempre de características del módulo pedagógico, proporcionando vistas y explicaciones de la estructura y algunas páginas web del sistema.

3.3.5. Implementación

El sistema RLATES es un sistema *Web* de tipo cliente/servidor, donde los clientes (alumnos) únicamente necesitan un navegador *Web* para poder interactuar con el sistema. Los recursos software utilizados se pueden resumir en:

- Lenguajes Java y JavaScript: Con los que se implementó la lógica del sistema y creación del interfaz.
- Sistema Gestor de Bases de Datos (SGBD) Oracle 9i: Donde se han creado las bases de datos necesarias para almacenar toda la información de los distintos módulos del sistema educativo.
- Servidor de páginas Web. Se ha utilizado el servidor *Web* Apache 3.2, incluido en el paquete del SGBD elegido.
- Java Server Pages (JSP) y Java Data Base Connectivity (JDBC), que nos permitirán realizar conexiones desde *Web* a las bases de datos de Oracle 9i.

- Navegador Internet Explorer (versión 5.0 o superior), que permitirá a los estudiantes interactuar con el tutor inteligente.

3.4. Interfaz de Usuario

Este apartado se centra en la descripción del módulo del interfaz de RLATES. Como se describió en el apartado 3.3, el módulo del interfaz de usuario facilita la comunicación entre el sistema de educación adaptativo e inteligente y el estudiante. Este modelo aplica técnicas inteligentes y adaptativas con el objetivo de adaptar el contenido del curso y la navegación a través del sistema dependiendo de las necesidades pedagógicas de los estudiantes.

El trabajo realizado en esta tesis se basa en la aplicación del módulo de *Aprendizaje por Refuerzo* en un sistema de educación a distancia. Por lo tanto, en este apartado solamente se tratarán las características de adaptación que se presentan en el interfaz de usuario derivadas directamente del módulo pedagógico de RLATES.

Uno de los principales problemas de los sistemas hipermedia es que los estudiantes tienden a perderse en el hiperespacio, sin saber qué estudiar a continuación y cuál fue la última página visitada (ver sección 2.1.3). RLATES proporciona al estudiante una progresión no lineal a través del material del curso, donde la información de los contenidos del curso se presentan a los estudiantes mediante páginas hipermedia conectadas mediante enlaces.

El sistema, gracias a la aplicación de *Aprendizaje por refuerzo* en su módulo pedagógico, proporciona al estudiante soporte adaptativo en las tecnologías de *Navegación y Presentación de Contenido*, eligiendo el formato de presentación del contenido (ver sección 2.1.3). Mediante la técnica *Soporte Adaptativo a la Navegación* (del inglés *Adaptive Navigation Support: ANS*) el sistema ayuda a los estudiantes a que no se pierdan entre las páginas hipermedia. Éste soporte lo realiza mediante la adaptación de la presentación y funcionalidad de las páginas web según los objetivos, conocimiento actual y otras características de aprendizaje de cada estudiante. Por otro lado, mediante la técnica de *Presentación Adaptativa* (del inglés *Adaptive Presentation*) los sistemas adaptan el contenido hipermedia de las páginas introduciendo inteligencia artificial al sistema.

La tecnología de *Soporte Adaptativo a la Navegación* comparte el mismo objetivo que la técnica de *Creación de Secuencias de Contenido* (del inglés *Curriculum Sequencing*) en los Sistemas Tutores Inteligentes (del inglés *Intelligent Tutoring Systems*): proporcionar a los estudiantes una secuencia de contenidos del curso adaptándose a sus necesidades pedagógicas en cada momento (ver sección 2.1.4). Pero la tecnología ANS añade algunas ventajas gracias al contexto hipermedia del sistema: puede guiar a los estudiantes directamente (como los sistemas

tutores inteligentes), pero también indirectamente (mediante la presentación de los enlaces en las páginas).

RLATES proporciona soporte a la navegación al estudiante mediante *Guía Directa* al mismo tiempo que proporciona soporte a la presentación de los contenidos, todo esto derivado directamente del modelo pedagógico del sistema. Para ello, en el módulo pedagógico se almacenan implícitamente las tácticas pedagógicas a tomar para cada estudiante en cada momento de la interacción, almacenando también información sobre los formatos de sus páginas (texto, video, imagen, etc.).

3.4.1. Guía Directa y Adaptación de la Presentación

El sistema es capaz de guiar al estudiante a través del material del curso mediante *Guía Directa* cuando el estudiante navega por el interfaz de RLATES a través del botón *Siguiente*. En la Figura 3.12 se muestra el botón siguiente con comentarios en color azul. Las páginas principales del interfaz del sistema se han dividido verticalmente en dos marcos. El marco izquierdo presentará, entre otra información importante para el usuario, una tabla de contenidos del curso, que, gracias a la estructura del modelo del conocimiento del sistema, se ha presentado fácilmente en forma de árbol de conocimiento. Este árbol de conocimiento posee un enlace por cada unidad de conocimiento del sistema (tema). En el marco derecho se presentan los botones de navegación de *Guía Directa* (*Anterior*, *Página Principal* y *Siguiente*) y el contenido del curso propiamente dicho (diferentes tareas o elementos: *definiciones*, *ejemplos*, etc.), acompañado normalmente por pestañas de tests que evalúan el conocimiento del estudiante. En concreto, en la Figura 3.12 se estudia actualmente el tema *Interrelación*⁴ (coloreado en rojo en el árbol de conocimiento del marco izquierdo del interfaz). Por ello, en el marco derecho del interfaz se muestran diferentes tareas de ese tema.

Cuando el estudiante pulsa en el botón *Siguiente* para continuar con su proceso de aprendizaje, el sistema le proporcionará ayuda sobre el/los *mejor/es* contenido/s del curso a visitar a continuación según sus características de aprendizaje (conocimiento actual, objetivos, preferencias de formato, etc.), proporcionándole al mismo tiempo los contenidos que se corresponden con las preferencias de formato que necesita el estudiante en ese instante (*Adaptación de la Presentación*).

Como se comentó en el apartado 3.2.3, el sistema, mediante la estrategia de exploración/explotación de *Boltzmann*, estima la probabilidad de elegir la siguiente acción a ejecutar (el contenido de la siguiente página web a visitar en el formato adecuado) según la función de *Boltzmann* y la función de *valor-acción* ($Q(s, a)$).

El sistema RLATES ha implementado la técnica de *Guía Directa* proponiéndole al estudiante más de una posibilidad de contenidos para visitar a continuación

⁴Elemento del modelo *Entidad/Interrelación*.



Figura 3.12: Interfaz de RLATES. Soporte Adaptativo a la Navegación y a la Presentación. Guía Directa.

(normalmente las 5 opciones con mayor utilidad). El estudiante es el encargado de elegir el contenido de la siguiente página web que va a visitar, teniendo en cuenta las sugerencias que le proporciona el sistema para cada uno de los contenidos (el sistema le proporciona la probabilidad de *Boltzmann* de ser mostrado cada uno de los contenidos en forma de porcentaje).

En la Figura 3.13 se puede observar cómo, cuando el estudiante pulsa en el botón *Siguiente* de la aplicación, el sistema le proporciona varias opciones de contenido para continuar con su aprendizaje. Para cada opción, RLATES le proporciona también la probabilidad de *Boltzmann* de ser ejecutada esa acción en formato de porcentaje. El estudiante podrá, de esta forma, elegir los contenidos que desea aprender a continuación basándose en las sugerencias del sistema.

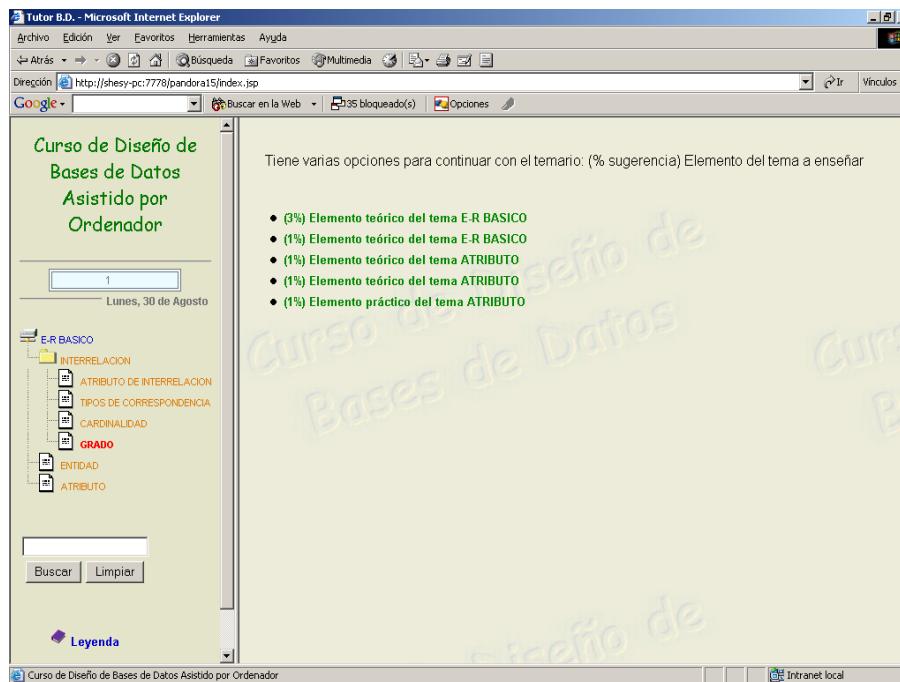


Figura 3.13: Interfaz de RLATES. Opciones de Siguiente Página a Visitar (Boltzmann).

Una vez elegida una de las opciones, por ejemplo, un elemento teórico del tema *Atributo*⁵ el sistema mostrará al estudiante un interfaz con el aspecto de la Figura 3.14, donde en el marco derecho de la interfaz solamente se muestra el contenido seleccionado en el formato adecuado y un test para evaluar el conocimiento del estudiante tras haber estudiado esa información.

⁵Elemento del modelo *Entidad/Interrelación*.



Figura 3.14: Interfaz de RLATES. Página Web mostrada después de elegir un contenido concreto a ser mostrado.

3.4.2. Soporte Adaptativo a la Navegación de forma Indirecta

Aunque la aplicación de *Aprendizaje por Refuerzo* en el módulo pedagógico de RLATES permite proporcionar al estudiante soporte adaptativo en la navegación y en la presentación del contenido en cuanto a la elección del *formato* de presentación, en RLATES también se han implementado otro tipo de ayudas a la navegación, con dos objetivos:

1. **Apoyo a la Navegación.** Al cambiar el aspecto de los enlaces visibles en las páginas (color, colocación, etc.), se le proporciona ayuda al estudiante para orientarse y navegar entre las páginas web del sistema (ver sección 2.1.3). Por ello, este tipo de adaptación también es útil al estudiante en su proceso de aprendizaje.
2. **Evaluación del Sistema.** Es importante, a la hora de evaluar el sistema, analizar si la adaptación proporcionada es útil para el estudiante. Por ello, se realizarán experimentos de dos tipos con los alumnos: interaccionando con el sistema aplicando *Aprendizaje por Refuerzo* e interaccionado con el sistema sin aplicar la propuesta. Cuando los estudiantes interactúen con el sistema sin aplicar *Aprendizaje por Refuerzo* en el módulo pedagógico, éstos podrían llegar a perderse en el hiperespacio si no se les proporciona ningún tipo de adaptación a la navegación. Por ello, se han implementado técnicas básicas de adaptación a la navegación mediante *Guía Indirecta*.

RLATES es capaz de guiar a los estudiantes de forma indirecta a través del contenido del curso cambiando la apariencia de los enlaces en la página (color u orden de los enlaces).

En la Figura 3.15 se puede ver la interfaz de RLATES, dividida en dos marcos. En el marco izquierdo se muestra el nombre del curso, el nombre del estudiante actual, la fecha actual y el árbol de conocimiento del curso. También se permite buscar ítems de conocimiento según su nombre. Por otro lado, en el lado derecho aparecen distintas tareas del tema actual que está estudiando el alumno (coloreado en rojo en el árbol de conocimiento del marco izquierdo). Concretamente, en la página de la figura el estudiante actualmente está estudiando el tema *Interrelación*, por lo que en el marco derecho aparecen *definiciones, introducciones, ejemplos, problemas y tests* para este tema en particular.

Las técnicas de *Guía Indirecta* que se usan en el sistema RLATES se pueden observar en la Figura 3.15 con comentarios coloreados de amarillo y se detallan a continuación:

1. **Anotación** (del inglés *Annotation*): Es la técnica más popular dentro del *Soporte Adaptativo a la Navegación*. El sistema RLATES cambia el color



Figura 3.15: Interfaz de RLATES. Soporte Adaptativo a la Navegación. Guía Indirecta.

de los enlaces del árbol de contenido según el comportamiento histórico del estudiante (ver Figura 3.16): el enlace es de color negro cuando no se ha visitado nunca; el color del enlace es azul si la página se ha visitado previamente; el enlace es de color rojo y el formato de la letra es negrita si se trata del tema actual que está siendo visitado por el estudiante; el color del enlace es verde cuando el estudiante ha estudiado el tema, ha hecho un test, y lo ha aprobado; finalmente, el color del enlace es naranja si el estudiante suspendió el test tras haber estudiado el tema.

2. **Enlaces Escondidos** (del inglés *Hiding Links*). Esta técnica esconde los temas que en un determinado momento no son relevantes para el estudiante. Por ejemplo, en la Figura 3.15 el estudiante está estudiando el tema *Interrelación*, por ello se ven los enlaces de sus sub-temas, pero sus sub-temas no aparecerán desplegados en el árbol de conocimiento hasta que para el estudiante no sean relevantes (cuando esté estudiando otro tema del árbol de conocimiento).
3. **Enlaces Ordenados** (del inglés *Sorting Links*). Esta técnica ordena los enlaces en la página según las necesidades del estudiante (normalmente basados en el conocimiento del estudiante o en el comportamiento del estudiante).

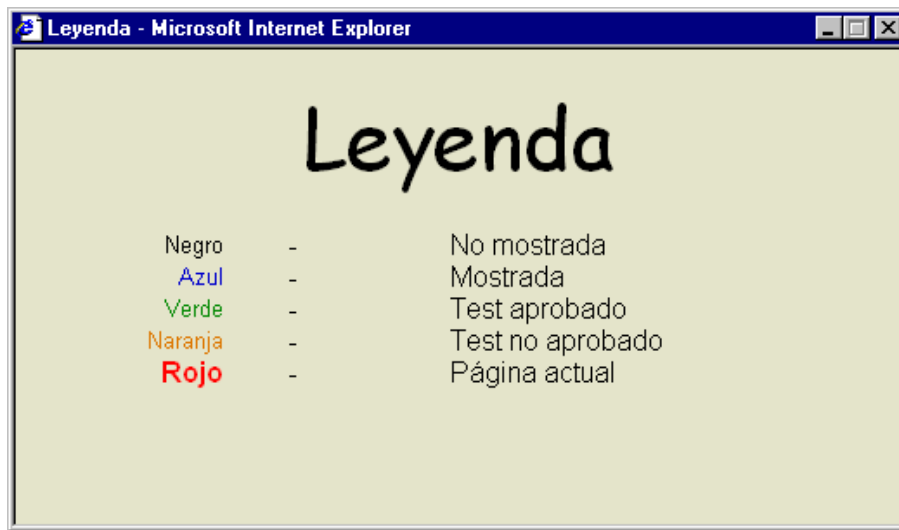


Figura 3.16: Interfaz de RLATES. Colores de los Enlaces en el Árbol de Concimiento.

En RLATES, los enlaces del árbol de conocimiento se ordenan según la estructura arborescente del modelo del dominio del sistema (teniendo en cuenta las relación *es-padre* entre temas).

4. **Añadir/Borrar Enlaces Dinámicamente** (del inglés *Adding/Removing Links*). Esta técnica añade o borra links dinámicamente a la página web según las necesidades del estudiante en un momento dado. Trata de prevenir que aparezcan en la página enlaces que no son relevantes para el alumno en un momento determinado. RLATES añade y elimina enlaces del marco izquierdo del interfaz de la aplicación según las tareas (elementos de un determinado tema) almacenadas en la base de datos para un tema dado. Por ejemplo, para el tema *Interrelación* existen *definiciones*, *introducciones*, *ejemplos* y *tests*, pero no *problemas*, por lo que el enlace de la tarea *problemas* ha desaparecido.

Capítulo 4

Aspectos Previos a la Experimentación

Hasta el momento se ha presentado la propuesta de este trabajo, basada en la aplicación del modelo de aprendizaje por refuerzo en el módulo pedagógico de los sistemas de enseñanza para adaptar de forma inteligente la secuencia y el formato de los contenidos mostrados a los usuarios. En los siguientes capítulos se pretende comprobar que la adaptación del sistema es útil para el estudiante, ayudándole a alcanzar sus objetivos. En trabajos previos se ha hablado de la dificultad de evaluar un sistema adaptativo [Gaudioso, 2002], mostrando la necesidad de validar cada una de las fases del proceso de desarrollo descrito en apartados anteriores.

En este capítulo se exponen aspectos previos a la experimentación. En primer lugar, en la sección 4.1, se tratan conceptos teóricos de una evaluación; posteriormente, en la sección 4.2, se describe del método de evaluación, que diseña el tipo de experimentación que se va a realizar, basado principalmente en dos cuestiones: primero si *RLATES funciona en teoría (interaccionando con estudiantes simulados)* y, una vez comprobado que el sistema funciona correctamente con estudiantes simulados, nos cuestionaremos si *el sistema también es útil con estudiantes reales* o se comporta de forma diferente; finalmente, en la sección 4.3 se presentan los modelos del dominio sobre los que se ha desarrollado la experimentación de este trabajo.

Posteriormente, en el capítulo 5, se analizarán los experimentos llevados a cabo sobre estudiantes simulados; y, para finalizar con la experimentación, en el capítulo 6 se detallarán los experimentos realizados en el aula, con estudiantes reales¹.

¹Los estudiantes reales que han participado en la experimentación han sido alumnos matriculados en asignaturas del Departamento de Informática de la Universidad Carlos III de Madrid

4.1. Conceptos Teóricos de una Evaluación

En este trabajo se ha realizado una *evaluación empírica*, entendiendo como tal al proceso de evaluación de un sistema mediante unos determinados experimentos. Es necesario para realizar una buena experimentación, realizar el diseño e implementación apropiados de los experimentos, definiendo cuáles son los parámetros de evaluación, e intentando que éstos se puedan evaluar de forma individual, sin que interfieran en el desarrollo del experimento.

Inicialmente, en un proceso de experimentación, se han de definir cuales son las *variables independientes* de la experimentación, entendiendo como *variables independientes* aquellas cuyo valor no depende de los valores de otras variables, pudiendo variar independientemente de otras variables. De esta forma, se entienden como *variables dependientes* aquellas cuyo valor depende del valor de otras variables.

Un experimento perfecto sería aquel en el que únicamente variarían las *variables independientes*, fijando las *variables dependientes*. Sin embargo, este tipo de experimentos es difícil de conseguir cuando se trata de experimentación con estudiantes reales en sistemas de educación adaptativos, ya que las características propias de cada estudiante (inteligencia, habilidades, etc.) o las condiciones de entorno del experimento (hora, lugar, ruido, distracciones, capacidad de cómputo de los ordenadores, velocidad de acceso a Internet, etc.) influirán en las variables dependientes muchísimo más que lo que podrían influir los cambios en las variables independientes. Por ejemplo, por la tarde normalmente los estudiantes se encuentran mucho más cansados que por la mañana, por lo que podría ocurrir que necesitaran más tiempo, más ejemplos, etc. para asimilar el conocimiento.

Para paliar estos problemas, los participantes han de ser asignados en grupos de forma aleatoria, de manera que al realizar la media de los experimentos se atenúen los efectos de 'ruido' sobre las variables dependientes. Es necesario, en este caso, trabajar con muchos estudiantes, para que la media dé un valor significativo. También se han utilizado medidas estadísticas, como la desviación típica, para determinar si las diferencias en las variables dependientes entre grupos se deben a los cambios de valor en las variables independientes o a fluctuaciones aleatorias.

Para evitar tener que trabajar con tantos estudiantes, mejorando la sensibilidad de los experimentos, en algunos casos se utilizaban *experimentos cruzados* (del inglés *crossed designs*) [Heck *et al.*, 2000], donde se utilizan los mismos participantes en múltiples condiciones de variables dependientes. Por ejemplo, el mismo participante trabajará con el sistema en dos modos: con y sin adaptación al usuario.

También se han utilizado otras técnicas para evitar que los alumnos participaran de forma subjetiva sobre el sistema. Estas técnicas se denominan *ciegas*

(del inglés *blind*) o *doblemente-ciegas* [Kayama and Okamoto, 2001] (del inglés *double-blind*). En los experimentos *ciegos*, los estudiantes no conocen si el software se adapta o no a sus necesidades, y en los experimentos *doblemente-ciegos* ni siquiera la persona que lleva a cabo los experimentos conoce qué versión del sistema le entrega al estudiante, con o sin adaptación al usuario.

En cuanto a la evaluación de los sistemas adaptativos aplicados a la educación, por lo general, los sistemas educativos se evalúan comparando los resultados obtenidos con dos versiones del sistema (una con adaptación, y la otra sin adaptación) [Gaudioso, 2002], aunque se han encontrado muchos casos en los que no se han realizado experimentación empírica alguna, a pesar de la importancia de este proceso en el desarrollo de un sistema de educación [Chin, 2001]. Inicialmente, se podría comprobar la precisión del modelo de estudiante, por ejemplo, comparando las acciones que se predijeron con las acciones que realmente realiza el estudiante [Corbett *et al.*, 1993], o hallando el porcentaje de errores reconocidos por el sistema [Sison *et al.*, 1998].

Por otro lado, los sistemas que utilizan métodos de aprendizaje automático (del inglés *machine learning*) para adquirir los modelos de usuario, podrían evaluar el modelo adquirido mediante medidas estándar de aprendizaje automático que compara el modelo de usuario con los modelos de usuarios que se utilizaron para entrenar el sistema [Webb *et al.*, 2001] [Zukerman and Albrecht, 2001].

En los sistemas de reconocimiento de planes (del inglés *plan recognition*) [Albrecht *et al.*, 1998] y de interacción de iniciativa mixta (del inglés *mixed-initiative interaction*) [Green and Carberry, 1999] se suelen comparar la secuencia de acciones [Chiu and Webb, 1998] o respuesta [Virvou and du Bulay, 1999] de los estudiantes que se predijo o lo que esperaría una persona experta en el tema con la secuencia de acciones o respuesta de los estudiantes actuales.

Por último, los sistemas con interfaz de usuario o ayuda al usuario (del inglés *User interfaces/help systems*) normalmente evalúan sus sistemas de forma subjetiva, evaluando la satisfacción del usuario y la velocidad con la que los estudiantes completan las tareas del sistema [M. Debevc and Svecko, 1996]. También se suele evaluar el ratio de error o la calidad de los estudiantes al alcanzar sus objetivos [Krause *et al.*, 1993]

4.2. Método de Evaluación

En cuanto a la manera de evaluar RLATES, el proceso se centra en las siguientes cuestiones:

1. ¿Funciona RLATES en teoría (cuando estudiantes simulados interactúan con el sistema)? Una vez que se ha comprobado que el dominio de los sistemas de enseñanza inteligentes y adaptativos es representable desde el punto

de vista del modelo de *Aprendizaje por Refuerzo* (ver capítulo 3), es importante plantearnos inicialmente si el sistema aprende a enseñar mejor según interactúa con los estudiantes simulados, llegando a producir tácticas de enseñanza adecuadas para los estudiantes. Para comprobar que RLATES funciona teóricamente, se crearán varios modelos de estudiantes predefinidos, es decir, se predefinirán varios estilos de comportamiento de estudiantes, con diferentes características de aprendizaje (ver sección 5.2) o, lo que es lo mismo, se representará a los estudiantes como Procesos de Decisión de Markov con distintas probabilidades de transición entre estados. Así se comprobará si el sistema es capaz de aprender tácticas pedagógicas adecuadas cuando interacciona con estos estudiantes simulados (ver Figura 4.1).

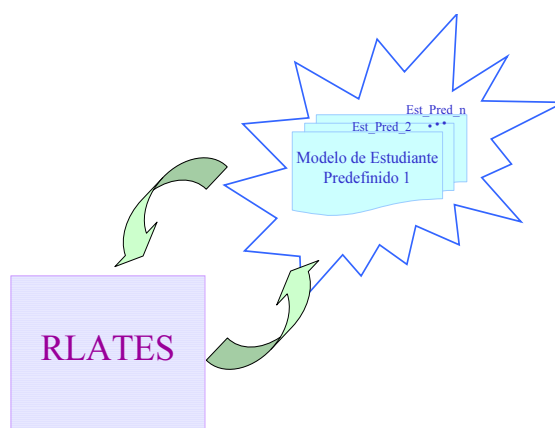


Figura 4.1: RLATES interactuando con estudiantes simulados

2. ¿El sistema es útil para su uso en las aulas? Dado que el sistema representado mediante el modelo de *Aprendizaje por Refuerzo* es capaz de aprender a enseñar a estudiantes simulados, es momento de plantearnos si RLATES es abordable desde el punto de vista práctico. Esto es necesario ya que quizás los estudiantes reales hacen que el sistema se comporte de forma diferente, impidiendo que el sistema converja a una política de comportamiento buena para los estudiantes. Esto equivale a comprobar si, en verdad, un grupo de estudiantes refleja un MDP similar a los utilizados con los estudiantes simulados. También habrá que tener en cuenta el coste computacional que necesita el sistema para converger a tácticas pedagógicas buenas. Se comprobará si el coste es lo suficientemente bajo como para que sea útil, o si los estudiantes, en su interacción normal con el sistema, no noten el cálculo

que realiza el sistema. En la Figura 4.2 se muestra el sistema interactuando con estudiantes reales.

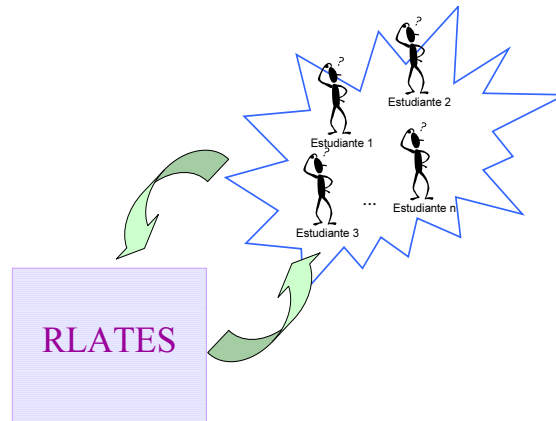


Figura 4.2: RLATES interactuando con estudiantes reales

Por tanto, se van a realizar dos tipos de pruebas para comprobar las cuestiones planteadas:

- **Experimentación simulada.** Con estudiantes simulados a partir de comportamientos de estudiantes reales, al igual que hacen [VanLehn *et al.*, 1996] y [Beck, 2001] en sus sistemas. Esta primera evaluación será determinante para ver la eficacia y eficiencia del tutor inteligente al aplicar técnicas de aprendizaje por refuerzo en su módulo pedagógico. En este bloque de pruebas se evaluará si el modelo de aprendizaje por refuerzo es aplicable en tiempo real, si el algoritmo elegido (Q-learning [Watkins, 1989]) es adecuado para este tipo de problemas, qué estrategia de exploración/explotación es más adecuada para los sistemas de enseñanza en Internet, y se realizará un análisis exhaustivo del efecto de los parámetros del modelo de aprendizaje por refuerzo propuesto en el proceso de aprendizaje del sistema.
- **Experimentación real.** Tras haber instanciado los parámetros de aprendizaje del algoritmo elegido, se realizará la evaluación del sistema con estudiantes reales. Esta prueba es más costosa que la anterior, dado que se requiere de estudiantes reales. Por ello no se van a realizar todas las evaluaciones con estudiantes reales, sino que esta experimentación se basa en los resultados obtenidos en la experimentación con estudiantes simulados. Además,

se asumirá que todos los estudiantes poseen características de aprendizaje similares, eliminando la fase de agrupación definida en la sección 3.2.5, y asumiendo que el conjunto de estudiantes es modelable mediante un único MDP desconocido.

4.3. Modelos del Dominio Utilizados en la Experimentación

Para llevar a cabo la experimentación, tanto con estudiantes simulados como con estudiantes reales, se han utilizado dos modelos distintos del dominio. El objetivo de utilizar dos modelos distintos es observar la escalabilidad del problema.

Los dominios que se han tomado como ejemplo se basan en el '*Diseño de Bases de Datos*', ya que esta tesis forma parte del proyecto de investigación *PANDORA*² [Castro *et al.*, 2002], herramienta CASE cuyo objetivo principal es definir métodos y técnicas para el desarrollo de bases de datos. Uno de los módulos de este sistema trata sobre un sistema de enseñanza en *Web*, para lo que se ha creado el sistema RLATES.

Los modelos del dominio utilizados se basan en la metodología propuesta en [Teorey *et al.*, 1986], que define tres fases principales en el diseño de bases de datos: el modelo conceptual, el modelo lógico y el modelo físico (*modelo Entidad-Interrelación: modelo E/R, modelo relacional y modelo físico* para bases de datos relacionales). Para simplificar el ejemplo, la experimentación se ha centrado en el modelo conceptual explicado por [Chen, 1976].

El modelo del dominio *A* se muestra en la Figura 4.3, y posee 3 temas (topics) y 16 acciones (tareas), concretamente dos *definiciones*, dos *introducciones* y 2 *ejemplos* por tema, excepto el primer tema, que sólo posee dos *definiciones* y dos *introducciones*. En la Tabla 4.1 se muestran las acciones que contiene cada tema del modelo del dominio. La tabla muestra en primer lugar, el número de acción, a continuación, el tema al que pertenecen (sobre qué tema trata esa acción), a continuación, el tipo de elemento (o tarea) y, para finalizar, el formato de la página web que se le mostrará al estudiante cuando el sistema ejecute la acción. En el anexo A se han incluido ejemplos del contenido del curso en forma de páginas web.

Por otro lado, el modelo del dominio *B* se muestra en la Figura 4.4 y posee 8 temas y 52 acciones, concretamente dos *definiciones*, una *introducción* y cuatro *ejemplos* por tema, excepto el primer tema, que sólo posee dos *definiciones* y una *introducción*. En la Figura 4.4 no se detallan las tareas del modelo, para facilitar

²Proyecto CICYT (TIC99-0215). Se trata de una plataforma CASE de diseño de bases de datos y su enseñanza vía Internet

su lectura. Como se hizo en el modelo del dominio A, la Tabla 4.2 muestra las acciones que contiene cada uno de los temas del modelo.

En los modelos del dominio también se han incluido relaciones *es-prerrequisito* existentes entre temas a la hora de definir el comportamiento de los estudiantes simulados (ver apartado 5.2). Estas relaciones indican cuando un tema (*a*) es prerrequisito de otro tema (*b*): cuando al estudiante no se le puede intentar enseñar conocimiento del tema *b* si no conoce previamente el tema *a*. En la figura 4.3 y 4.4 se han representado las relaciones *es-prerrequisito* mediante flechas, donde, por ejemplo, se puede observar que todo tema es prerrequisito de sus subtemas (por ejemplo, el tema *E-R Básico* es prerrequisito del tema *Entidad*) y, además, se ha representado con flechas rojas los prerrequisitos adicionales entre temas (por ejemplo, el tema *Grado* es prerrequisito del tema *Cardinalidad* en el modelo del dominio B).

Es importante resaltar el hecho de que las relaciones *es-prerrequisito* entre temas se van a tener en cuenta a la hora de diseñar el comportamiento de los estudiantes simulados, pero no se tendrán en cuenta a la hora de secuenciar el contenido del sistema mediante el modelo de *Aprendizaje por Refuerzo*, ya que el sistema será capaz de aprender tácticas pedagógicas adecuadas para cada estudiante sin tener este conocimiento previamente definido por el experto.

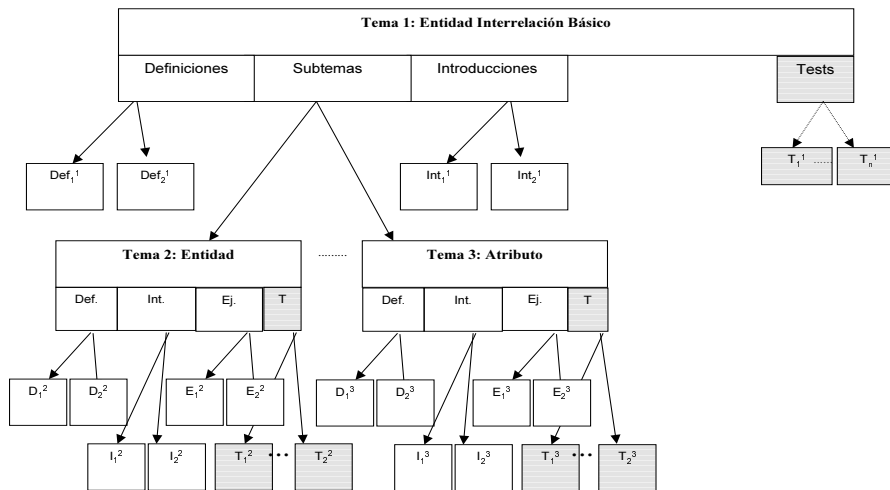


Figura 4.3: Modelo del dominio A

Número de acción	Tema al que pertenece	Tipo de tarea	Formato
1	3	Definición	Texto
2	3	Definición	Imagen
3	3	Introducción	Texto
4	3	Introducción	Imagen
5	3	Ejercicio	Texto
6	3	Ejercicio	Imagen
11	1	Definición	Texto
12	1	Definición	Imagen
13	2	Definición	Texto
14	2	Definición	Imagen
15	2	Introducción	Texto
16	2	Introducción	Imagen
17	2	Ejercicio	Texto
18	2	Ejercicio	Imagen
23	3	Introducción	Texto
24	3	Introducción	Imagen

Tabla 4.1: Acciones en el modelo del dominio A.

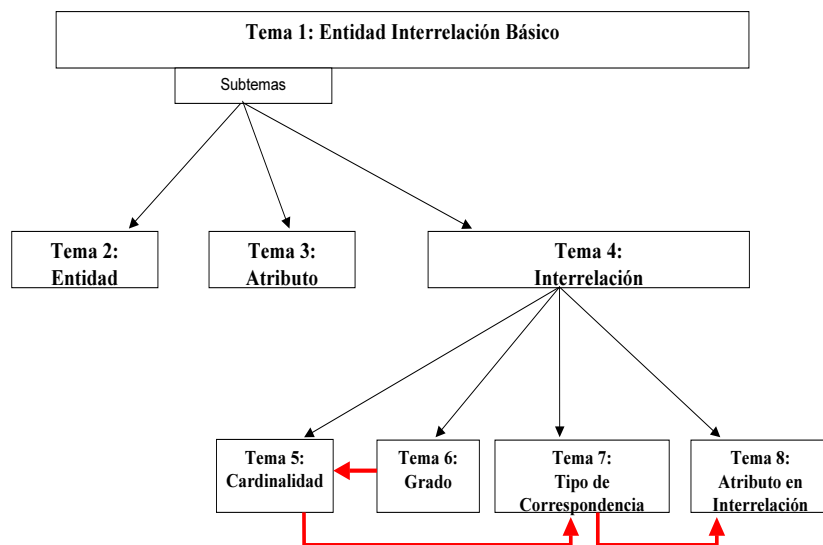


Figura 4.4: Modelo del dominio B

Número de acción	Tema al que pertenece	Tipo de tarea	Formato
1	2	Definición	Imagen
2	2	Definición	Texto
3	2	Ejercicio	Imagen
4	2	Ejercicio	Texto
5	2	Ejercicio	Imagen
6	2	Ejercicio	Texto
7	2	Introducción	Texto
12	1	Definición	Imagen
13	1	Definición	Texto
14	3	Definición	Imagen
15	3	Definición	Texto
16	3	Ejercicio	Imagen
17	3	Ejercicio	Texto
18	3	Ejercicio	Imagen
19	3	Ejercicio	Texto
20	3	Introducción	Texto
25	8	Definición	Imagen
26	8	Definición	Texto
27	8	Ejercicio	Imagen
28	8	Ejercicio	Texto
29	8	Ejercicio	Imagen
30	8	Ejercicio	Texto
31	8	Introducción	Texto
34	6	Definición	Imagen
35	6	Definición	Texto
36	6	Ejercicio	Imagen
37	6	Ejercicio	Texto
38	6	Ejercicio	Imagen
39	6	Ejercicio	Texto
40	6	Introducción	Texto
45	4	Definición	Imagen
46	4	Definición	Texto
47	4	Ejercicio	Imagen
48	4	Ejercicio	Texto
49	4	Ejercicio	Imagen
50	4	Ejercicio	Texto
51	5	Definición	Imagen
52	5	Definición	Texto
53	5	Ejercicio	Imagen
54	5	Ejercicio	Texto
55	5	Ejercicio	Imagen
56	5	Ejercicio	Texto
57	5	Introducción	Texto
62	4	Introducción	Texto
67	7	Definición	Imagen
68	7	Definición	Texto
69	7	Ejercicio	Imagen
70	7	Ejercicio	Texto
71	7	Ejercicio	Imagen
72	7	Ejercicio	Texto
73	7	Introducción	Texto
78	1	Introducción	Texto

Tabla 4.2: Acciones en el modelo del dominio *B*.

Capítulo 5

Experimentación con Estudiantes Simulados

Para poder obtener conclusiones generales en la evaluación de un sistema de enseñanza, se necesitan una gran cantidad de experimentos y, por tanto, se requiere de una gran cantidad de estudiantes para que interactúen con el sistema en periodo de experimentación, aspecto bastante difícil de lograr. En la mayoría de los sistemas de educación se utilizan estudiantes simulados para probar la aplicabilidad del sistema (por ejemplo, el sistema ADVISOR [Beck, 2001]) motivado por circunstancias importantes: primero, es difícil persuadir a una persona para utilizar una aplicación que puede que no esté optimizada, peor aún cuando la aplicación es un sistema de educación que requiere de una gran concentración por parte del estudiante: persuadir a cientos de personas es una tarea impensable; segundo, la experimentación con estudiantes reales tiene un gran coste, ya que éstos necesitan periodos de tiempo grandes en la interacción y algunas veces abandonan por aburrimiento o se dan cuenta de que el sistema no muestra el contenido de la mejor forma.

En este capítulo se presentan los resultados de la experimentación con estudiantes simulados. En esta experimentación se intenta evaluar el número de acciones que el sistema necesita mostrar al estudiante para que éste aprenda el contenido del sistema de enseñanza y el número de estudiantes necesarios para que el sistema aprenda a enseñar a los estudiantes.

En primer lugar se definirá el proceso de la experimentación, motivándola; posteriormente, se detallará el proceso de construcción de los estudiantes simulados; a continuación se definirán los parámetros de evaluación; en la sección siguiente se analizarán los resultados obtenidos, poniendo especial interés en cómo los parámetros de evaluación afectan a la convergencia del sistema; y por último se expondrán las principales conclusiones a este bloque de experimentos.

Es importante resaltar el hecho de que cuando se realizan experimentos con

estudiantes simulados, se están estudiando situaciones reales, ya que los estudiantes simulados interactúan uno a uno con el sistema, suponiendo un modelo de comportamiento simulado. Según los estudiantes simulados interactúan con el sistema, RLATES aprenderá de cada interacción, modificando sus tácticas pedagógicas según sus características de aprendizaje al mismo tiempo que enseña la materia del curso.

5.1. Motivación de la Experimentación

Este trabajo trata de presentar una experimentación exhaustiva sobre el sistema, analizando todos los parámetros que afectan a la convergencia del mismo para poder llegar a conclusiones generales, por ello se ha experimentado en primer lugar con estudiantes simulados.

Los experimentos llevados a cabo en esta sección estudiarán en profundidad varias cuestiones: en primer lugar, se analizará si RLATES es capaz de aprender a enseñar a estudiantes con diferentes características de aprendizaje, es decir, si converge a tácticas pedagógicas adecuadas. Segundo, se analizará la rapidez en la convergencia del sistema (el número de estudiantes necesarios para que el sistema aprenda a enseñar). Tercero, se analizará cómo de buenas son las tácticas pedagógicas aprendidas por el sistema (medido con el número de acciones necesarias que el sistema ha de enseñar para que el estudiante aprenda la materia del curso). Cuarto, se valorarán los parámetros que afectan a la convergencia del sistema, tanto con la estrategia de exploración/explotación *e-greedy* como con la de *Boltzmann*, indicando si alguna de ellas proporciona ventajas frente a la otra. Por último, se tratará de reducir el tamaño de la *fase de entrenamiento* del sistema (ver capítulo 3.2.5), inicializando la información que posee el sistema sobre el modelo del dominio.

El dominio de los sistemas de enseñanza es estocástico, ya que el estudiante puede que apruebe o no un examen de un tema dado dependiendo de infinitas variables. Por ello, los experimentos con estudiantes simulados se han realizado varias veces y posteriormente, a la hora de analizar los resultados, se han utilizado medidas estadísticas como la media y la desviación típica.

5.2. Diseño de la Experimentación

Los estudiantes simulados se han construido teniendo en cuenta únicamente dos variables para definir su comportamiento: la preferencia del formato hipermedia (texto, imagen, video, etc.) y el tipo de contenidos que prefieren (definiciones, introducciones, ejemplos, problemas, etc.).

Además, también se ha tenido en cuenta conocimiento experto sobre los modelos del dominio utilizado, en este caso se ha utilizado el conocimiento de un profesor de “Diseño de Bases de Datos” de la Universidad Carlos III de Madrid. El conocimiento del dominio que se ha tenido en cuenta se refiere a relaciones de *prerrequisitos*, detalladas en el apartado 4.3.

Se han definido tres grupos distintos de estudiantes (ver Tabla 5.1), dependiendo de sus características de aprendizaje. En el *grupo1* de estudiantes se encuentran todos aquellos que se comportan de forma totalmente determinista, y sólo aprenden cuando se les enseña tareas de tipo *Ejercicio* en el formato de *Imagen*. Sin embargo los estudiantes de los grupos *grupo2* y *grupo3* se comportan de forma indeterminista. En el dominio de los sistemas web de educación se da *indeterminismo en el comportamiento del estudiante* cuando a un estudiante (que se encuentra en un determinado estado de conocimiento sobre la materia) se le muestra una determinada página web (el sistema ejecuta una determinada acción o tarea). En algunos casos el estado del estudiante variará tras aprender la información mostrada en la página web (pudiendo variar a diferentes estados de conocimiento), pero también puede ocurrir que el estado de conocimiento del estudiante no varíe tras visitar la página web. Al no conocer con seguridad el siguiente estado conocimiento del estudiante tras visitar una página web (ya que puede variar), se define el dominio como estocástico o indeterminista.

Los estudiantes del *grupo2* requieren tareas del tipo *Definición y Ejercicio*, pero sólo aprenderán con una probabilidad de 0,75 si la tarea es de tipo *Ejercicio* y con una probabilidad de 0,25 si la tarea es de tipo *Definición*. Además, sólo aprenderán si se les presentan páginas web en formato *Imagen*.

Por otro lado, los estudiantes del *grupo3* requieren tanto tareas de tipo *Definición*, como tareas de tipo *Introducción*, como tareas de tipo *Ejercicio*, pero sólo aprenderán con una probabilidad de 0,25 cuando se les muestre tareas de tipo *Definición* o *Introducción*, y con probabilidad de 0,50 cuando se les muestre tareas de tipo *Ejercicio*. Además, aprenderán la información que incluye la página web con una probabilidad de 0,75 si ésta posee formatos de *Imagen*, y con una probabilidad de 0,25 si la página está escrita en formato de *Texto*.

Grupo de Estudiantes	Tipo de Tarea	Formato de Tarea
Grupo1	Ejercicio	Imagen
Grupo2	Definiciones y Ejercicios	Imagen
Grupo3	Definiciones, Introducciones y Ejercicios	Imagen y Texto

Tabla 5.1: Grupos de estudiantes simulados contruidos para los experimentos.

Por otro lado, el sistema de enseñanza contiene las acciones mostradas en la Tabla 4.1 en el modelo del dominio *A* y las acciones mostradas en la Tabla 4.2 en

el modelo del dominio B (ver sección 4.3).

Los modelos predefinidos de los estudiantes (estudiantes simulados) se pueden considerar como *Procesos de Decisión de Markov (MDP)*, donde no importa por qué estados ha pasado anteriormente el estudiante, sino que solamente importa el estado actual del alumno para que éste sea capaz de aprender o no la página web mostrada por el sistema (ver sección 2.4.2). En las figuras 5.1, 5.2 y 5.3 se representa mediante un diagrama la función de transición de cada uno de los *MDP* (grupos de estudiantes) cuando RLATES les enseña el modelo del dominio A . Por otro lado, las figuras 5.4, 5.5 y 5.6 muestran el comportamiento de los estudiantes de cada uno de los grupos cuando el sistema enseña el modelo del dominio B . En la construcción de los modelos simulados de comportamiento de los estudiantes se ha tenido en cuenta en todo momento las relaciones de *pre-requisito* que existen entre los temas de cada modelo del dominio. Además, en las figuras se muestran únicamente las acciones que pueden producir transiciones de estado o aquellas cuya probabilidad es distinta a 1,0. Se supone que cuando un estudiantes se encuentra en un determinado estado, el resto de las acciones que no están representadas para dicho estado harían que el estudiante no cambiara de estado con una probabilidad igual a 1,0.

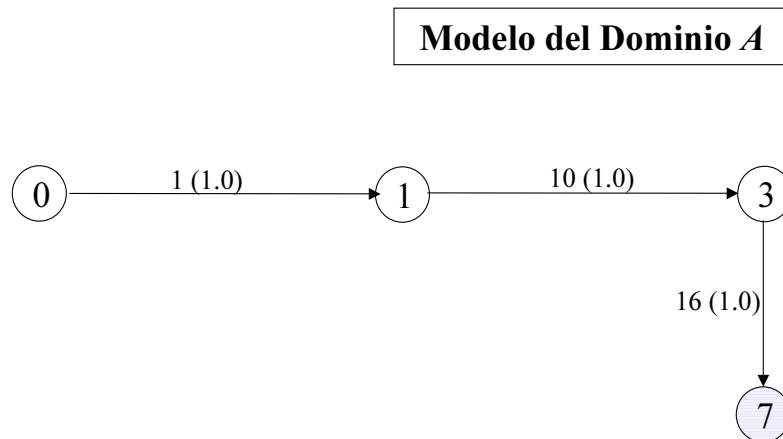


Figura 5.1: Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del *grupo 1* cuando se les enseña el modelo del dominio A

En la Figura 5.2, por ejemplo, se realiza la lectura del MDP de la siguiente forma: cuando el estudiante se encuentra en el estado de conocimiento denominado 0 y se ejecuta la acción 2, el estudiante pasará al estado de conocimiento denominado 1 con una probabilidad de 0,25 y se mantendrá en el estado de conocimiento 0 con una probabilidad de 0,75. Por otro lado, si, encontrándose en el estado 0,

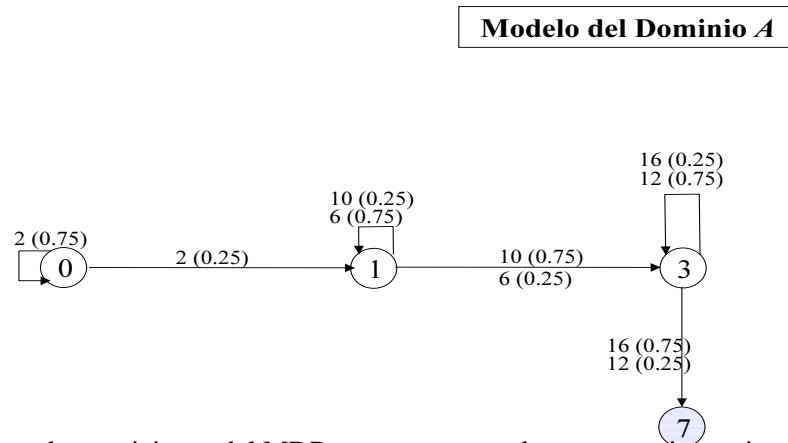


Figura 5.2: Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del *grupo2* cuando se les enseña el modelo del dominio A

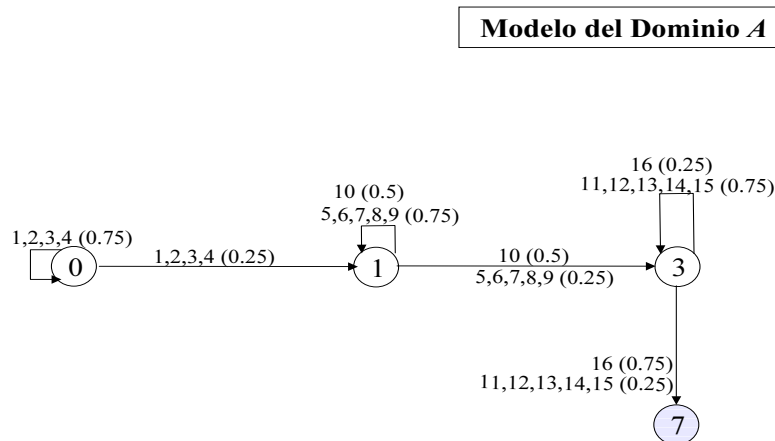


Figura 5.3: Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del *grupo3* cuando se les enseña el modelo del dominio A

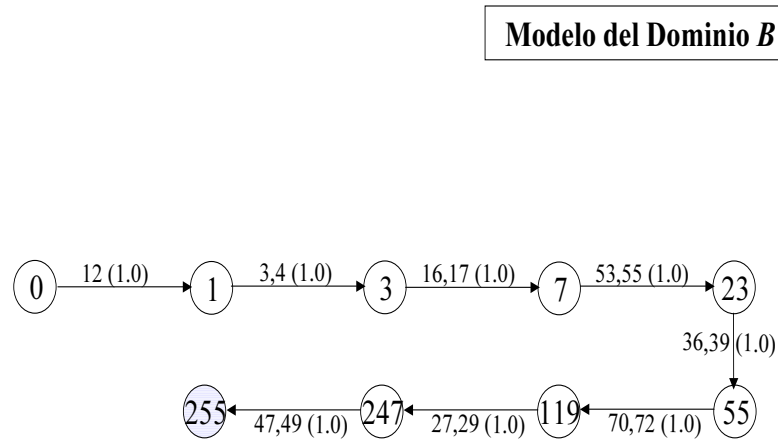


Figura 5.4: Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del *grupo1* cuando se les enseña el modelo del dominio *B*

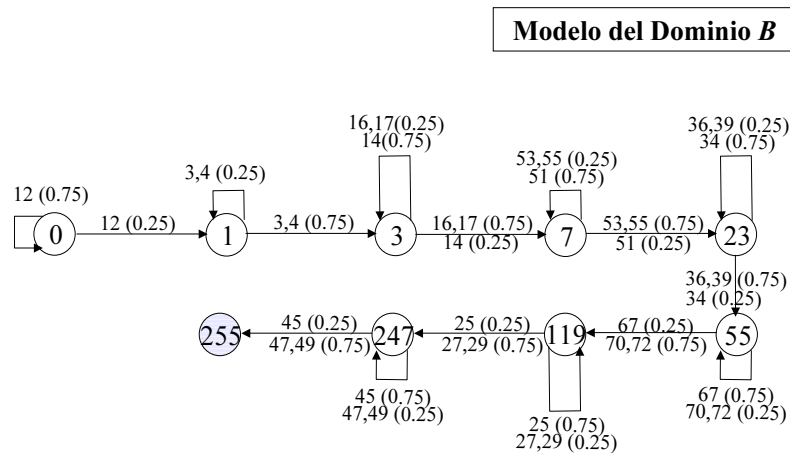


Figura 5.5: Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del *grupo2* cuando se les enseña el modelo del dominio *B*

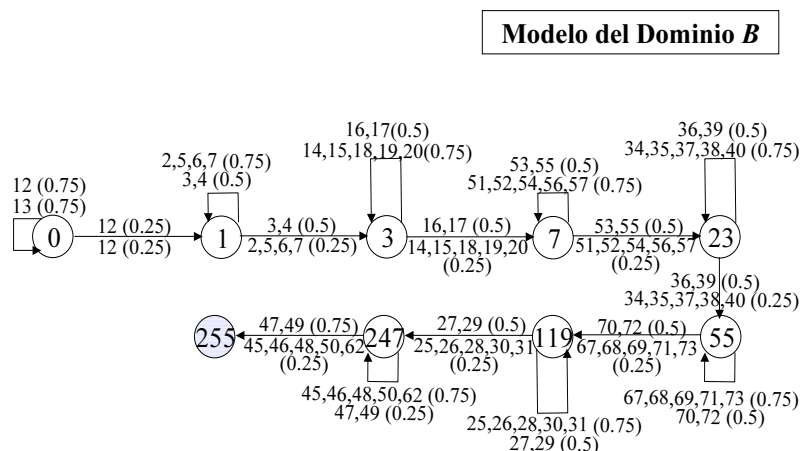


Figura 5.6: Diagrama de transiciones del MDP que representa el comportamiento simulado de los estudiantes del *grupo3* cuando se les enseña el modelo del dominio B

el sistema ejecuta cualquier otra acción del entorno del agente, el estudiante no cambiará de estado con una probabilidad de 1, 0.

Cuando el estudiante se encuentra en el estado de conocimiento 1 y el sistema ejecuta la acción 10, el estudiante cambiará de estado al estado 3 con una probabilidad de 0,75, y se quedará en el mismo estado con una probabilidad de 0,25. El estudiante cambiará de estado al estado 3 cuando el sistema ejecuta la acción 6 con una probabilidad de 0,25 y no cambiará con una probabilidad de 0,75. Por último, si el sistema ejecuta cualquier otra acción, el estudiante no cambiará de estado.

Para finalizar, cuando el estudiante se encuentra en el estado de conocimiento 3 y el sistema ejecuta la acción 12, el estudiante transitará al estado 7 (estado final) con una probabilidad de 0,25 y se mantendrá en el mismo estado con una probabilidad de 0,75. El estudiante también cambiará al estado 7 si el sistema ejecuta la acción 16 con una probabilidad de 0,75, manteniéndose en el mismo estado con una probabilidad de 0,25. Finalmente, si el sistema ejecuta cualquier otra acción del entorno, el estudiante se mantendrá en el mismo estado.

En este punto es necesario recordar que los estados en el sistema se numeran de 0 a 2^{n-1} , siendo n el número de temas del curso. En el ejemplo de la Figura 5.2 los estados se numerarán del 0 a 7 (en una notación decimal) y de 000 a 111 (en notación binaria), pero en el diagrama de transiciones únicamente aparecen los estados posibles en los que se puede encontrar un estudiante simulado, debido, entre otras cosas, a la relación de *prerrequisito* que existe entre los temas del modelo del dominio A. Recordar también que la notación binaria de los estados es más representativa del conocimiento de un estudiante donde, por ejemplo, el

estado 001 indica que el estudiante conoce el tema 1, pero aún no conoce el tema 2 y tres.

Se supone que el sistema se adaptará mejor a los estudiantes del *grupo1*, ya que todos ellos tienen características de aprendizaje similares. Por otro lado, al existir indeterminismo en el comportamiento de los estudiantes del *grupo2* y *grupo3*, el refuerzo que se obtiene al ejecutar una acción cuando el estudiante se encuentra en un estado dado puede ser diferente dependiendo de cuál es el siguiente estado al que transita el estudiante. En una situación real, existirá indeterminismo al ejecutar las acciones, debido, por ejemplo, a las diferentes características de aprendizaje de los estudiantes de un grupo dado, por lo que los estudiantes del *grupo2* y *grupo3* representan más fielmente a los estudiantes reales.

5.3. Parámetros de Evaluación

Es importante, antes de realizar la experimentación, definir los parámetros de evaluación (las variables independientes) del sistema. Se van a utilizar distintos parámetros para evaluar el sistema con estudiantes simulados, donde se pretende comparar la eficiencia del sistema utilizando las distintas técnicas de aprendizaje detalladas en capítulos anteriores:

- **el tamaño del modelo del dominio:** el número de temas y tareas que contiene el modelo del dominio. Para los experimentos con estudiantes simulados se han utilizado los modelos del dominio definidos en la sección 4.3, donde se verá la eficacia y eficiencia del sistema dependiendo del número de temas que se pretende que aprenda el estudiante y tareas que se les mostrarán para el aprendizaje de los temas,
- **el tipo de MDP que representa al grupo de estudiantes:** el grado de indeterminismo al ejecutar las acciones de cada estudiante. Es decir, con qué probabilidad aprenderá un estudiante una determinada información cuando éste posee un conocimiento sobre la materia conocida. Se han creado varios grupos de estudiantes dependiendo de su grado de indeterminismo (ver sección 5.2),
- **la estrategia de exploración/explotación de los algoritmos** al elegir la siguiente acción a tomar. En los experimentos simulados se utilizarán tanto la estrategia de exploración/explotación de *e-greedy*, como la política de Boltzmann descritas en la sección 2.4.7,
- **el ratio de aprendizaje (α).** La velocidad con la que aprende el sistema a partir de las interacciones de otros estudiantes del mismo grupo (ver sección 2.4.6).

Por otro lado, las variables dependientes de las que se sacarán posteriormente resultados son:

- **el número de estudiantes** necesarios para conseguir que el sistema enseñe la materia de forma óptima y
- **el número de páginas que ha de mostrar el sistema (acciones)** para que los estudiantes aprendan.

5.4. Resultados Obtenidos

La experimentación se dividirá en dos grandes grupos, dependiendo de la estrategia de exploración/explotación utilizada en la experimentación (*e-greedy* o *Boltzmann*). En los siguientes apartados, el orden de los experimentos sigue la planificación mostrada en la Figura 5.7. La figura muestra como, en primer lugar, se fija la variable independiente que indica la estrategia de exploración/explotación utilizada. Posteriormente, se fija el modelo del dominio que enseña el sistema. Finalmente se fijan los parámetros de las políticas de exploración/explotación (e en caso de *e-greedy* y τ en el caso de *Boltzmann*), el ratio de aprendizaje (α) y el tipo de estudiante que interactúa con el sistema por este orden. Se han numerado los experimentos con la palabra “**Simul-**” más un número consecutivo para poder destacarlos en el texto.

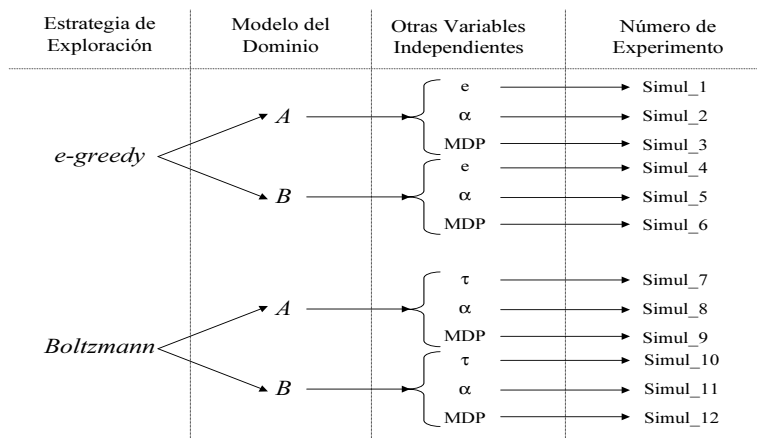


Figura 5.7: Planificación de los experimentos simulados

Por otro lado, también se llevará a cabo un bloque de experimentos para tratar de reducir la *fase de entrenamiento* de RLATES (ver sección 3.2.5).

5.4.1. Estrategia de Exploración/Explotación E-GREEDY

La experimentación con la estrategia de exploración/explotación *e-greedy* se divide en dos grandes grupos, dependiendo del modelo del dominio que se ha utilizado para los experimentos: modelo del dominio *A* o modelo del dominio *B* (ver sección 4.3).

Modelo del Dominio A

EXPERIMENTO SIMUL-1

En primer lugar, se analizará cómo afecta el valor del parámetro e de la estrategia de exploración *e-greedy* en la curva de aprendizaje. Para ello, se fija el ratio de aprendizaje (α) y el tipo de estudiantes con el que interactúa el sistema, dependiendo del indeterminismo de los mismos al ejecutar acciones (ver Tabla 5.1).

En la Figura 5.8 se puede ver las diferentes curvas de aprendizaje de RLATES cuando varía el valor del parámetro e de la estrategia de exploración/explotación *e-greedy*, se fija el valor del ratio de aprendizaje (α) a 0,1 y se interactúa con estudiantes del *grupo2*, cuyo comportamiento no es determinista, sino que incluye algo de indeterminismo.

El eje de abscisas en las gráficas de este apartado muestra el número de estudiantes que interactuó con el sistema en cada experimento, mientras que el eje de ordenadas muestra el número de acciones ejecutadas (número de páginas web visitadas) en cada interacción completa con el sistema.

En esta figura se puede observar cómo, cuanto más alto es el valor del parámetro e de *e-greedy* (más avariciosa es la política), mejor se comporta el sistema, convergiendo a una política óptima en menor tiempo (con un número de estudiantes menor) y mostrando un menor número de páginas web al estudiante (ejecutando un menor número de acciones). Se puede observar que cuando el valor del parámetro e es 0,1 (existe mayor exploración), el sistema no converge, mientras que cuando el valor del parámetro e es mayor (mayor explotación), la convergencia del sistema es más rápida, llegando a valores medios de 5 acciones para mostrar 3 temas (siendo la táctica pedagógica óptima aquella que ejecuta 3 acciones si el estudiante aprendiera siempre lo que se le muestra).

Por otro lado, al observar la gráfica 5.9 que muestra la desviación típica del sistema para el mismo caso, se puede deducir que cuanto mayor es el valor de e ,

menor es la desviación típica (menor diferencia existe entre las interacciones de los estudiantes), y por lo tanto, la convergencia es más estable.

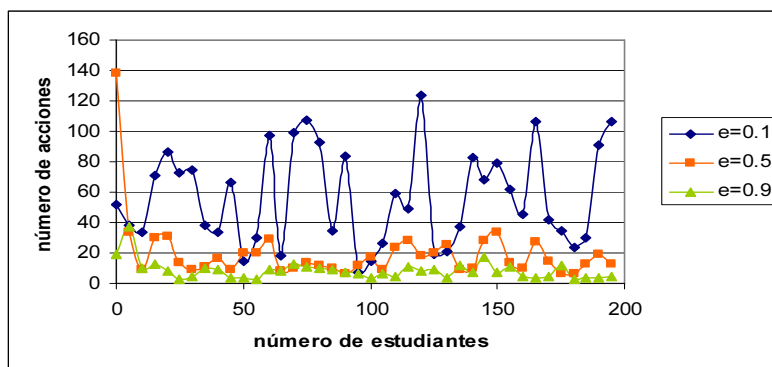


Figura 5.8: SIMUL-1. Curvas de aprendizaje de RLATES variando el parámetro e de la estrategia de exploración e -greedy para estudiantes pertenecientes al *grupo2*, $\alpha=0,1$. Se ha utilizado el modelo del dominio A.

EXPERIMENTO SIMUL-2

En la Figura 5.10, se muestra cómo cambia la convergencia del sistema cuando varía el valor del ratio de aprendizaje (α). Para ello, se ha fijado e a 0,9 (parámetro de la estrategia de exploración/explotación e -greedy), es decir, se adquiere una política muy avariciosa; también se han utilizado, al igual que en el experimento anterior, estudiantes del *grupo2*. En esta gráfica se puede comprobar que cuando el ratio de aprendizaje es pequeño, el sistema tarda un poco más en aprender (por ejemplo, cuando $\alpha=0,1$ el sistema necesita unos 20 alumnos para aprender a enseñar), mientras que si el ratio de aprendizaje es mayor, el sistema aprende más rápido, pero menos estable (se observan picos en la curva que indican que un estudiante dado necesitaría muchas más acciones que el estudiante anterior). Esta diferencia se puede ver mejor en la gráfica 5.11, que muestra la desviación típica de los valores obtenidos.

EXPERIMENTO SIMUL-3

Por último en este grupo de experimentos, se mostrará la diferencia en la convergencia cuando estudiantes de distintos grupos interactúan con RLATES. En la gráfica 5.12 se muestran curvas de aprendizaje de distintos grupos de estudiante cuando el ratio de aprendizaje (α) se ha fijado a 0,5 y el parámetro e de la es-

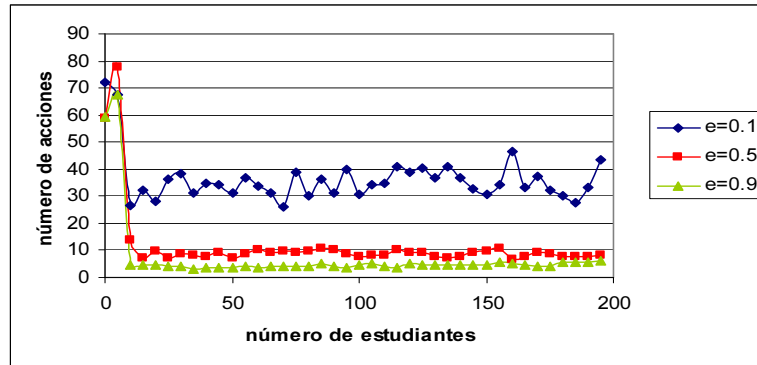


Figura 5.9: SIMUL-1. Desviación típica del sistema variando el parámetro e de la estrategia de exploración e -greedy para estudiantes pertenecientes al *grupo2*, $\alpha=0,1$. Se ha utilizado el modelo del dominio A para estos experimentos.

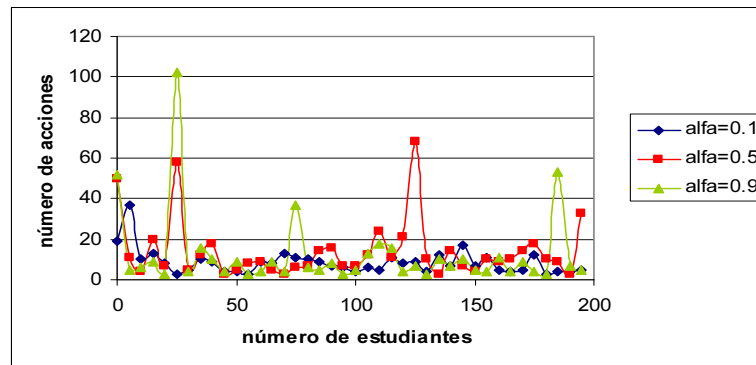


Figura 5.10: SIMUL-2. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al *grupo2* y modelo del dominio A .

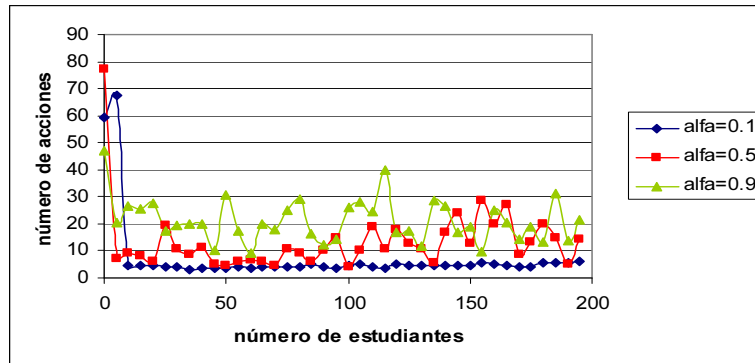


Figura 5.11: SIMUL-2. Desviación típica del sistema variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al *grupo2* y modelo del dominio A.

trategia de exploración/explotación *e-greedy* se ha fijado a 0,9. La gráfica muestra cómo con diferentes tipos de estudiante, el sistema se comporta de forma diferente y, por tanto, su progresión hacia la convergencia de la política óptima es diferente. Se puede concluir gracias a este experimento, que cuando existe indeterminismo al ejecutar las acciones, el sistema aprende más despacio. Posteriormente, en la Figura 5.13 se puede observar la desviación típica del sistema en el mismo caso.

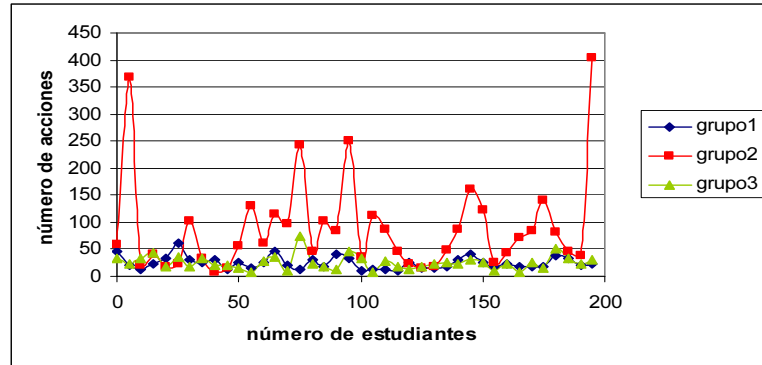


Figura 5.12: SIMUL-3. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, $e=0,9$ y modelo del dominio A.

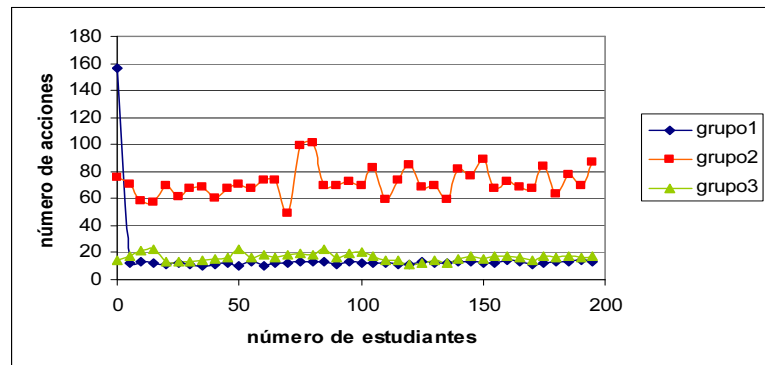


Figura 5.13: SIMUL-3. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, $e=0,9$ y modelo del dominio A.

Modelo del Dominio B

EXPERIMENTO SIMUL-4

Al igual que se hizo con el modelo del dominio *A*, se analizará en primer lugar cómo afecta a la curva de convergencia de RLATES el valor del parámetro e de la estrategia de exploración/explotación de *e-greedy*. Para ello, se fija el ratio de aprendizaje (α) a 0,5 y los estudiantes al tipo *grupo2*.

La Figura 5.14 muestra cómo, cuanto más avariciosa es la estrategia de exploración/explotación, existe más explotación, y el sistema aprende a enseñar más rápido (con menor número de estudiantes) y mejor (con menor número de acciones). Por ejemplo, cuando $e=0,9$, el sistema converge cuando ha interactuado con 15 estudiantes, mostrando una media de 15 páginas web (teniendo en cuenta que la opción óptima es cuando se ejecutan 8 acciones). Sin embargo, cuando el valor de e es menor, por ejemplo, 0,1, RLATES necesitaría el doble de estudiantes (alrededor de 30) para converger, y la táctica a la que converge no es tan buena, necesitando ejecutar una media de 150 acciones para que el estudiante aprenda el contenido del sistema de enseñanza. En la gráfica 5.15, por otro lado, se puede observar la curva de la desviación típica en el mismo experimento, donde cuando más avariciosa es la política, menor diferencia existe entre las interacciones con los estudiantes, y cuando menos avariciosa es la política, más diferencia existe.

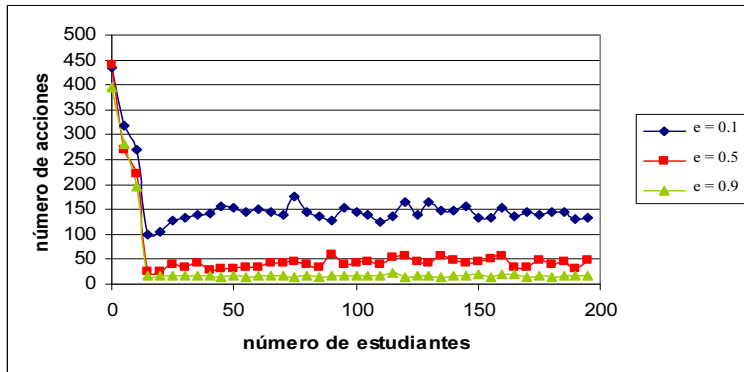


Figura 5.14: SIMUL-4. Curvas de aprendizaje de RLATES variando el parámetro e de la estrategia de exploración *e-greedy* para estudiantes pertenecientes al *grupo2*, $\alpha=0,5$ y modelo del dominio *B*.

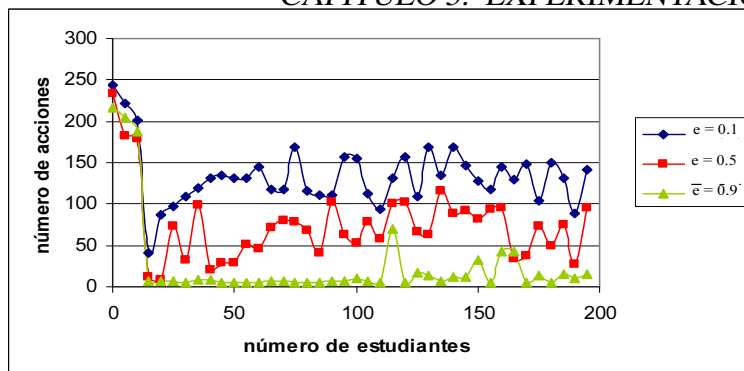


Figura 5.15: SIMUL-4. Desviación Típica de RLATES variando el parámetro e de la estrategia de exploración e -greedy para estudiantes pertenecientes al *grupo2*, $\alpha=0,5$ y modelo del dominio B .

EXPERIMENTO SIMUL-5

A continuación, se analiza el efecto de la variación del ratio de aprendizaje (α) en la convergencia del proceso de aprendizaje de RLATES. Para ello se ha fijado el parámetro e a 0,9 (ya que se ha demostrado en el experimento anterior que es el mejor valor del parámetro para este dominio) y estudiantes del *grupo2* interactúan con el sistema.

En la Figura 5.16 se observa cómo cuanto menor es el valor del ratio de aprendizaje, peor se comporta el sistema. Por ejemplo cuando $\alpha=0,1$, se necesitan alrededor de 65 estudiantes para converger y una media de 230 acciones que ejecutar para que los estudiantes aprendan la materia. Sin embargo, cuando α es mayor, el sistema se comporta mejor, necesitando, cuando $\alpha=0,5$, únicamente 15 estudiantes para converger a una política que necesita solamente una media de 15 acciones para enseñar. Comparando las curvas de aprendizaje de RLATES cuando $\alpha=0,5$ y $\alpha=0,9$, se puede observar que sus resultados son muy parecidos, pero que cuando su valor es 0,9, el sistema converge con únicamente 10 estudiantes, pero su media de número de acciones es 20. Concluyendo, cualquiera de las opciones es buena y, dependiendo de lo que se busque en el sistema (reducir el número de estudiantes para que converja o reducir el número de acciones, optimizando la política) se elegirá uno u otro valor de α .

Por otro lado, en la Figura 5.17 se puede observar cómo cuando $\alpha=0,5$, el sistema se comporta de forma más estable que cuando $\alpha=0,9$, siendo más parecidas las interacciones entre distintos estudiantes.

EXPERIMENTO SIMUL-6

Por último en esta tanda de experimentos, al igual que se hizo con el dominio

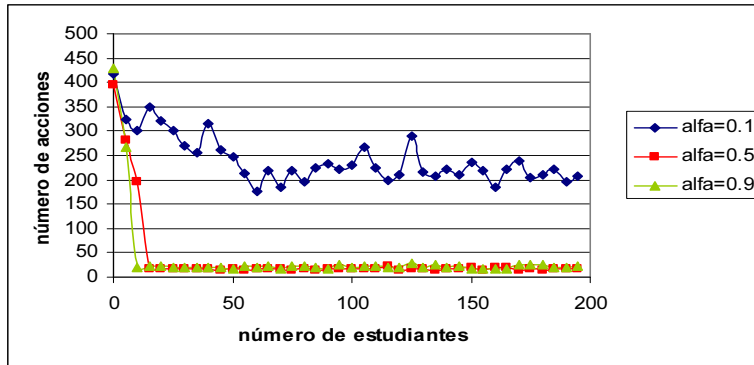


Figura 5.16: SIMUL-5. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al *grupo2* y modelo del dominio *B*.

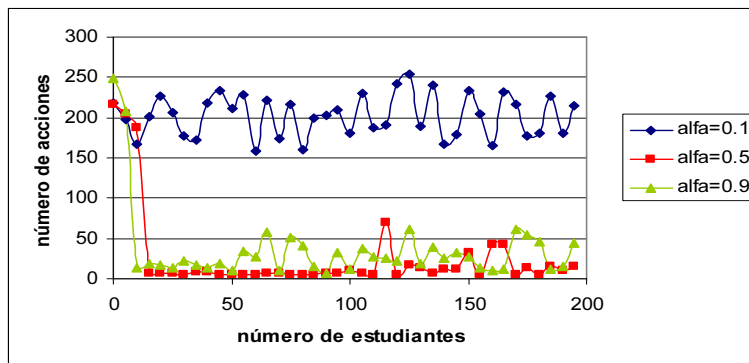


Figura 5.17: SIMUL-5. Desviación típica del sistema variando el ratio de aprendizaje (α) y fijando parámetro $e=0,9$ para estudiantes pertenecientes al *grupo2* y modelo del dominio *B*.

A, se comparará cómo se comporta el sistema cuando distintos tipos de estudiante (con distintos tipos de indeterminismo) interactúan con RLATES. Para ello se ha fijado $\alpha=0,5$ y el parámetro e de la estrategia de exploración/explotación *e-greedy* a 0,9, ya que se ha demostrado con experimentos anteriores que se trata de valores buenos para este dominio de aplicación.

La gráfica 5.18 muestra cómo cuando más indeterminista es el tipo de estudiante que interactúa con RLATES, peor se comporta el sistema. Por ejemplo, cuando estudiantes del *grupo1* (totalmente determinista) interactúan con RLATES, éste converge con aproximadamente 15 estudiantes a una táctica que únicamente necesita ejecutar una media de 8,5 acciones (teniendo en cuenta que la táctica óptima mostraría solo 8 páginas web). Sin embargo, cuando estudiantes muy indeterministas (por ejemplo, los pertenecientes al *grupo3*) interactúan con el sistema, éste converge también cuando alrededor de 15 estudiantes interactúan con RLATES, pero necesita una media de 28 acciones para que el estudiante aprenda la materia.

En la Figura 5.19 se muestra, por otro lado, la desviación típica del experimento donde se puede observar que cuanto más indeterministas son los estudiantes, menos parecidas son sus interacciones, y el sistema se comporta de forma poco estable.

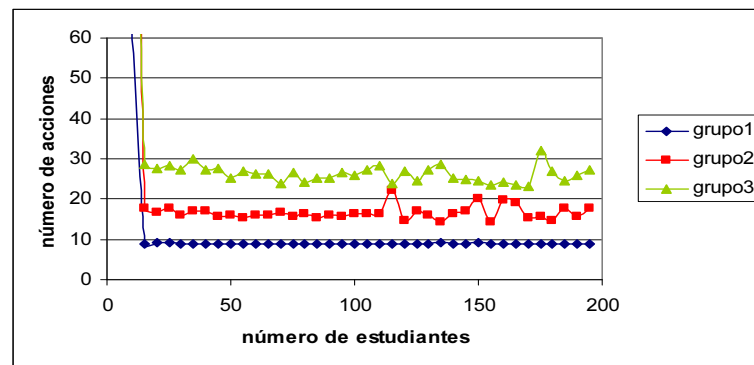


Figura 5.18: SIMUL-6. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0.5, $e=0,9$ y modelo del dominio B .

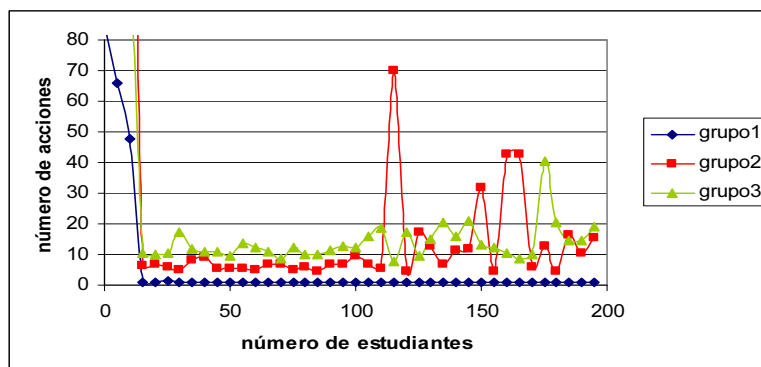


Figura 5.19: SIMUL-6. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, $e=0,9$ y modelo del dominio B .

Conclusiones

Las primeras conclusiones que se han observado son:

- El sistema es capaz de aprender tácticas pedagógicas enseñando a estudiantes con distintas características de aprendizaje (las curvas de aprendizaje convergen).
- El sistema necesita interactuar con un número no muy grande de estudiantes para converger (convergencia rápida). En concreto, el sistema necesita alrededor de quince alumnos cuando interactúa con el grupo de estudiantes más heterogéneo al enseñar el modelo del dominio B .
- El sistema converge a tácticas cercanas a las óptimas (utilizando un número no muy grande de acciones).
- Con ambos modelos del dominio se ha verificado cómo afecta el valor del parámetro e de la estrategia de exploración/explotación de e -greedy, llegando a la conclusión de que cuando más avariciosa es la política, RLATES converge a una táctica pedagógica buena más rápido (utilizando menor número de estudiantes) y mejor (ejecutando menor número de acciones).
- Igualmente, se ha analizado el efecto de la variación del parámetro que ajusta el ratio de aprendizaje del sistema (α). Tanto con el modelo del dominio

A como con el modelo del dominio B se han obtenido resultados similares, indicando que cuando mayor es el valor de α , mejor se comporta el sistema (aprende más rápido a enseñar), obteniendo buenos valores con $\alpha=0,5$ y $\alpha=0,9$. Cuando $\alpha=0,5$ el sistema aprende tácticas mejores, aunque necesita un número ligeramente mayor de estudiantes para converger que cuando $\alpha=0,9$. Además, la desviación típica entre unos estudiantes y otros es menor.

- Finalmente, también se han realizado comparaciones según el tipo de estudiante que interactúa con el sistema en las curvas del proceso de aprendizaje de RLATES. Se ha llegado a la conclusión de que cuanto más heterogéneo sea el grupo de estudiantes, peor aprenderá el sistema y las interacciones entre los estudiantes y el sistema serán diferentes (utilizando distinto número de acciones en las curvas de la desviación típica).

En cuanto a aplicaciones futuras sobre estudiantes reales, se puede concluir que:

- Se puede seguir una política *e-greedy* siempre que se utilicen parámetros de e muy altos.
- Es aconsejable maximizar la homogeneidad del grupo de estudiantes (hacer el MDP lo más determinista posible) mediante agrupación de estudiantes.
- El parámetro α debe ser muy alto (0,9) si se desea minimizar el número de estudiantes en la fase de entrenamiento, pero debería reducirse (0,5) si se quiere obtener una política de comportamiento más cercana a la óptima.

5.4.2. Estrategia de Exploración/Explotación de BOLTZMANN

Al igual que se hizo con la estrategia de exploración/explotación *e-greedy* (ver sección 5.4), la experimentación con la política de *Boltzmann* se divide, a su vez, en dos grupos de experimentos, dependiendo del modelo del dominio que se ha utilizado.

Modelo del Dominio A

EXPERIMENTO SIMUL-7

En primer lugar, se analizará el efecto de la variación del parámetro de la temperatura (τ) en la curva de convergencia del sistema. Para ello se fija el valor del ratio de aprendizaje (α) a 0,5 y se interactúa con estudiantes que son un poco indeterministas, pertenecientes al *grupo2* de estudiantes.

La gráfica 5.20 muestra cómo cuando la temperatura es menor (el comportamiento es más avaricioso), el sistema se comporta mejor (converge con un menor número de estudiantes y ejecutando menos acciones). Por ejemplo, cuando la temperatura es igual a 0.01, el sistema converge a una media de 7,07 acciones (siendo la estrategia óptima la que ejecutaría solamente 3 acciones si los estudiantes aprendieran siempre lo que el sistema les muestra) utilizando únicamente 10 estudiantes para converger.

En la Figura 5.21 se puede comprobar cómo cuando la temperatura es menor, la desviación típica también decrece, de forma que cuando $\tau=0,01$ se obtiene el mejor valor (existe menos diferencia entre las interacciones de los estudiantes).

Por lo tanto, se podría concluir que, para este caso, el valor de $\tau=0,01$ sería la mejor opción.

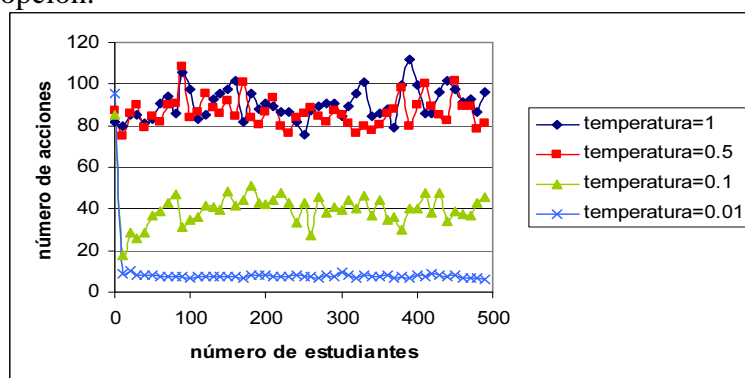


Figura 5.20: SIMUL-7. Curvas de aprendizaje de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es A.

EXPERIMENTO SIMUL-8

A continuación se analizará cómo afecta el parámetro α (ratio de aprendizaje del sistema) en la convergencia de RLATES. Para ello se ha fijado el valor de la temperatura de la estrategia de exploración/explotación de *Boltzmann* (τ) a 0,01 (ya que se demostró en el experimento anterior que éste era el valor óptimo). Se ha trabajado también en este experimento con estudiantes del *grupo2*.

En este caso, tanto en la gráfica 5.22, que muestra la curva de convergencia del sistema, como en la gráfica 5.23, que muestra la desviación típica de RLATES, se puede observar que el ratio de aprendizaje (α) no influye mucho en la convergencia del sistema, por lo que se podrá utilizar cualquier valor en la experimentación,

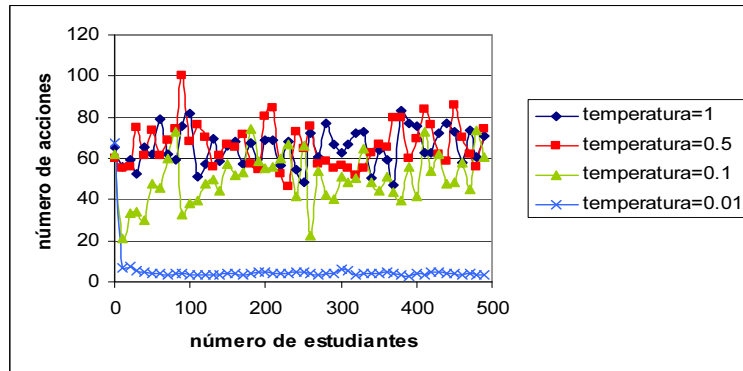


Figura 5.21: SIMUL-7. Desviación típica de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es A.

al contrario que con la estrategia de exploración/explotación *e-greedy*, en la que sí que varía el resultado (el resultado era mejor cuando se elegía un α de 0,5).

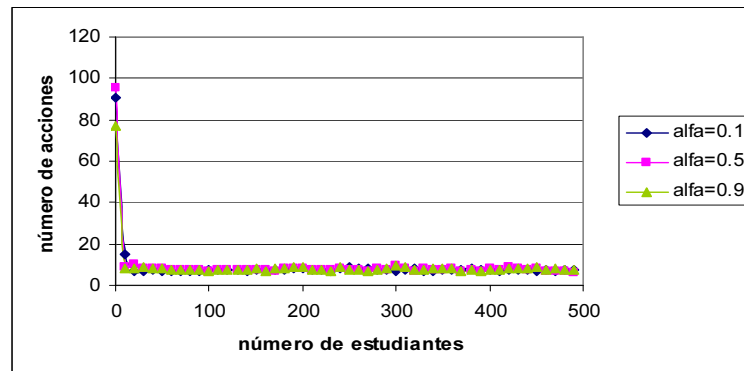


Figura 5.22: SIMUL-8. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es A.

EXPERIMENTO SIMUL-9

Por último, se analiza el efecto en el proceso de convergencia de RLATES que puede suponer la interacción con estudiantes de diferente características de aprendizaje. Para ello se ha fijado τ a 0,01 y α a 0,5.

La gráfica de la Figura 5.24 muestra cómo el grado de indeterminismo en los

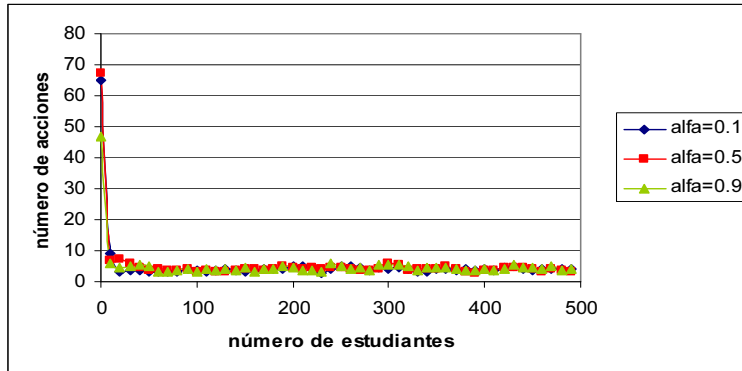


Figura 5.23: SIMUL-8. Desviación típica de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *A*.

estudiantes no afecta mucho a la curva de convergencia de RLATES en este dominio. Por último, la Figura 5.25 muestra que la desviación típica para los estudiantes totalmente deterministas es igual a 0, mientras que para los estudiantes algo indeterministas converge a valores de 5 acciones de diferencia entre interacciones de estudiantes.

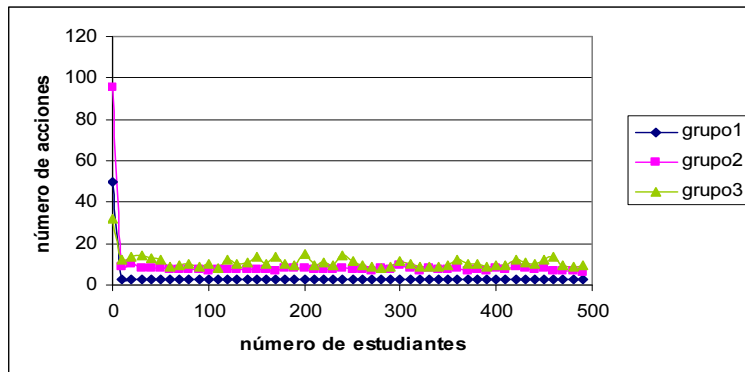


Figura 5.24: SIMUL-9. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es *A*.

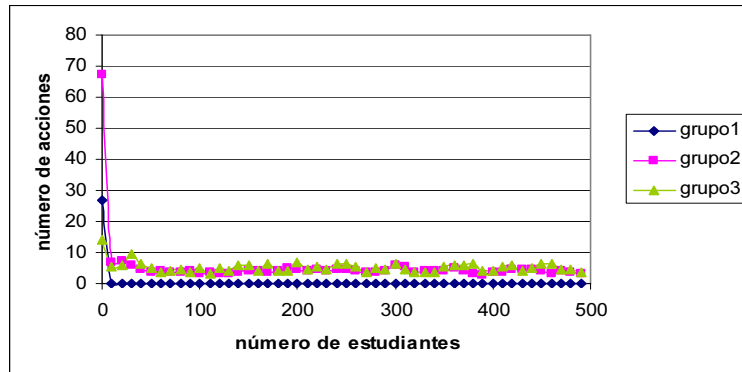


Figura 5.25: SIMUL-9. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es el A.

Modelo del Dominio B

EXPERIMENTO SIMUL-10

En cuanto al modelo del dominio *B* se refiere, como se hizo con el modelo del dominio *A*, se analizará el efecto en la convergencia del sistema cuando varía el parámetro de la temperatura de *Boltzmann* (τ), el ratio de aprendizaje (α), y el indeterminismo en los estudiantes.

Comenzando por analizar cómo afecta la temperatura al fijar α a 0,5 e interactuando con estudiantes de tipo *grupo2*, la gráfica 5.26 muestra cómo cuanto menor es el valor del parámetro τ en la estrategia de exploración/explotación de *Boltzmann*, mejor se comporta el sistema (al igual que ocurría con el modelo del dominio *A*). Igualmente, en la gráfica 5.27 se puede observar cómo el sistema se comporta de forma más estable (con menor diferencia entre interacciones de distintos alumnos) cuanto menor el valor de τ . Por ello, se establece $\tau=0,01$ como un valor apropiado para este dominio.

EXPERIMENTO SIMUL-11

A continuación se analizará el efecto de la variación del ratio de aprendizaje (α) en RLATES. Para ello se ha fijado la temperatura de la estrategia de exploración/explotación de *Boltzmann* (τ) a 0,01 y se trabaja con estudiantes de tipo *grupo2*.

En la gráfica 5.28 se muestra cómo cuando el parámetro α es 0,5 o 0,9, el sistema converge a valores de 15 acciones con sólo 20 estudiantes. Por otro lado, la Figura 5.29 muestra que cuando α toma el valor 0,5 o 0,9, la desviación típica

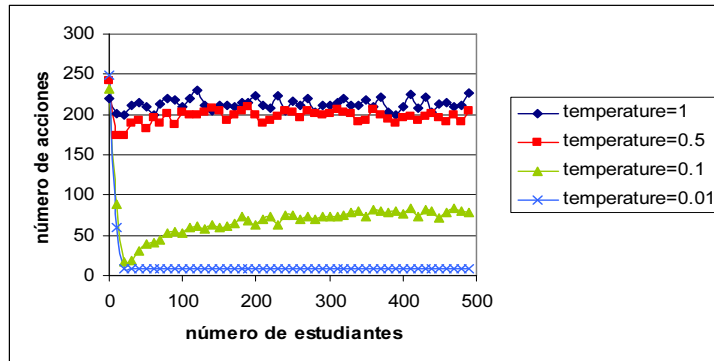


Figura 5.26: SIMUL-10. Curvas de aprendizaje de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*.

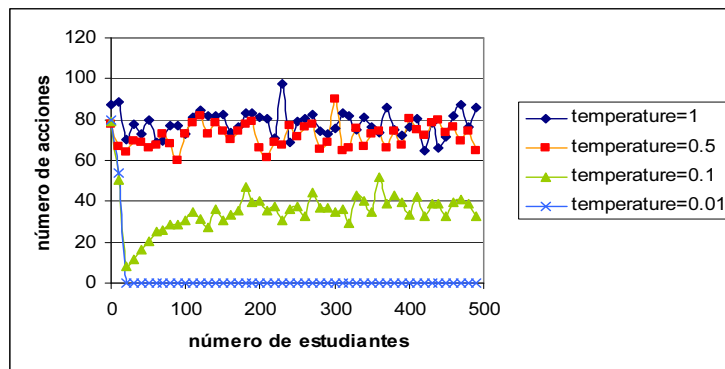


Figura 5.27: SIMUL-10. Desviación típica de RLATES variando el parámetro de la temperatura (τ) en la estrategia de exploración/explotación de *Boltzmann*, fijando el ratio de aprendizaje (α) a 0,5 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*.

alcanza valores de más o menos 5 acciones de diferencia entre interacciones de estudiantes.

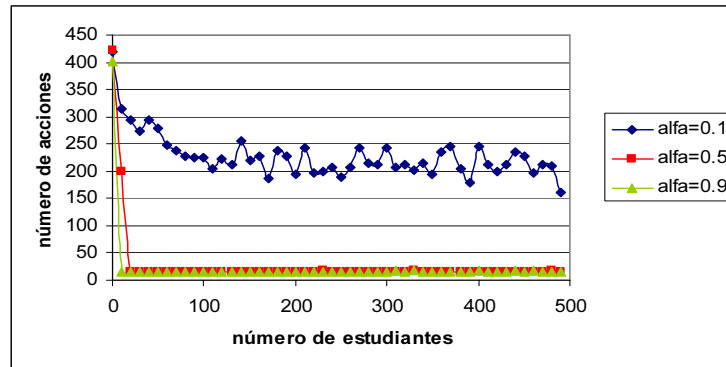


Figura 5.28: SIMUL-11. Curvas de aprendizaje de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*.

EXPERIMENTO SIMUL-12

Por último, se analiza el efecto del indeterminismo en las acciones de los estudiantes. Para ello se fija el valor de τ a 0,01 y el valor de α a 0,5.

En la gráfica 5.30 se puede observar cómo cuanto más indeterminismo existe en las acciones de los estudiantes, peor se comporta el sistema (converge a tácticas menos buenas). Es notorio de resaltar que en el caso en que los estudiantes totalmente deterministas interactúan con el sistema, se alcanza la táctica óptima (ejecución de 8 acciones cuando el estudiante siempre aprende lo que se le presenta) utilizando alrededor de 20 estudiantes para converger.

Para finalizar, la gráfica 5.31 muestra la desviación típica de este experimento, donde se observa que cuanto más determinista es el comportamiento de los estudiantes, menor desviación existe entre las interacciones de distintos estudiantes, destacando que cuando son totalmente deterministas, la desviación es igual a cero.

A continuación, se plantea una cuestión importante: si el sistema de enseñanza es capaz de converger con ambas estrategias de exploración/explotación (*e-greedy* y *Boltzmann*), entonces, ¿cuál será la más adecuada para utilizar cuando se interaccione con estudiantes reales? Para obtener una solución, lo primero que se nos ocurrió es comparar las curvas del proceso de aprendizaje de RLATES. En la gráfica 5.32 se comparan las dos estrategias mostrando la curva de aprendizaje del sis-

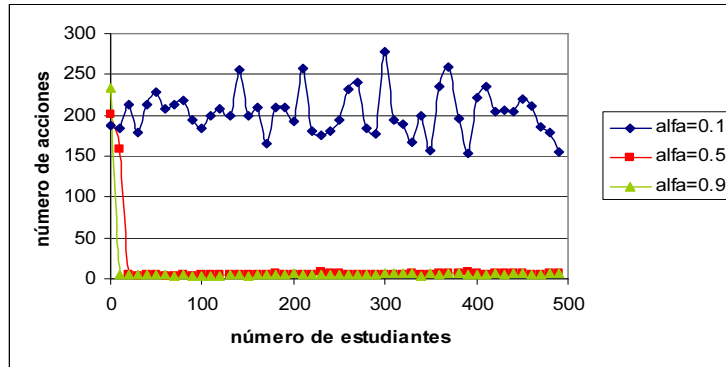


Figura 5.29: SIMUL-11. Desviación típica de RLATES variando el ratio de aprendizaje del sistema (α) y fijando parámetro de la temperatura (τ) a 0,01 e interaccionando con estudiantes de tipo *grupo2*. El modelo del dominio utilizado es *B*.

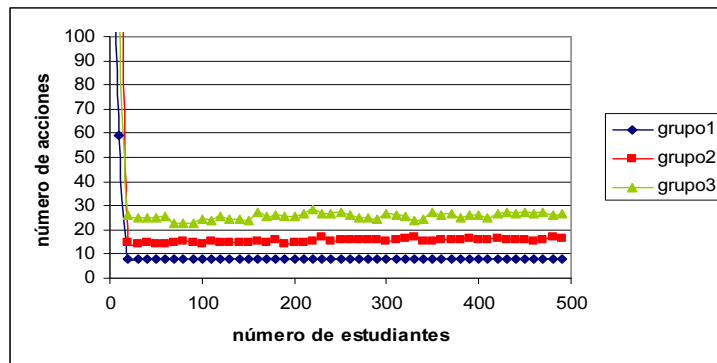


Figura 5.30: SIMUL-12. Curvas de aprendizaje de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es el *B*.

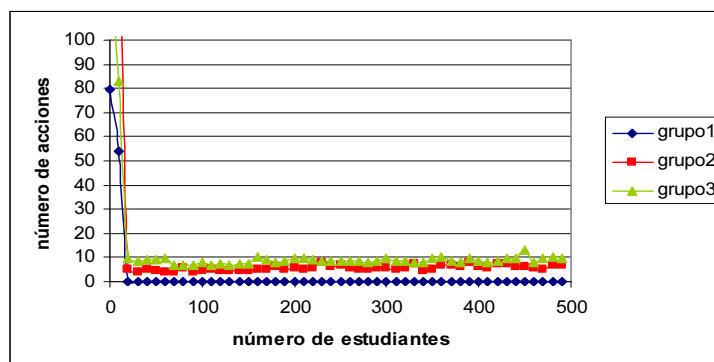


Figura 5.31: SIMUL-12. Desviación típica de RLATES cuando distintos grupos de estudiantes interactúan con el sistema. El ratio de aprendizaje (α) se ha fijado a 0,5, τ a 0,01. El modelo del dominio utilizado es el B .

tema cuando sus parámetros proporcionan mejores resultados en nuestro dominio: $\alpha=0,5$, $e=0,9$ (para la política *e-greedy*), $\tau=0,01$ (para la política de *Boltzmann*), siempre trabajando con estudiantes del *grupo2* (con comportamiento algo indeterminista) y con el modelo del dominio B . En esta gráfica se puede observar cómo existe poca diferencia entre las dos curvas, convergiendo a valores medios de 19 acciones cuando se utiliza la política de *e-greedy* y a valores medios de 24 acciones cuando se utiliza la política de *Boltzmann*. Además, ambos convergen cuando sólo 20 estudiantes interactúan con RLATES.

Por otro lado, observando la gráfica 5.33 se comprueba que el sistema se comporta de forma mucho más estable (las interacciones entre estudiantes son más parecidas) cuando se trabaja con la política de *Boltzmann* que con la política *e-greedy*. Esta diferencia en algunos casos compensa la característica anterior.

Después de analizar las curvas, se plantea otra cuestión para poder decidir entre las dos opciones: ¿alguna de las dos políticas nos proporciona alguna ventaja en cuanto a la interacción con estudiantes reales? La respuesta a esta pregunta es sí, ya que la estrategia de exploración/explotación de *Boltzmann* proporciona una probabilidad normalizada de ejecutar una acción cuando el estudiante se encuentra en un estado de conocimiento de la materia dado, eligiendo la siguiente acción a tomar en base a esa probabilidad. Sin embargo, cuando se utiliza la política *e-greedy* y no se elige la mejor opción, el sistema elige aleatoriamente entre el resto de las acciones (pudiendo llegar a elegir una acción realmente mala).

Esta característica de la política de *Boltzmann* se podría utilizar fácilmente

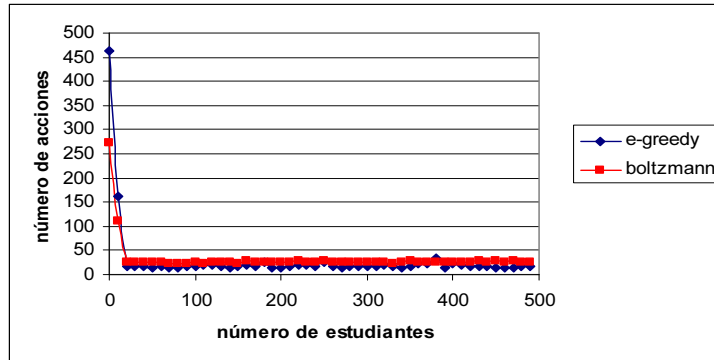


Figura 5.32: Curvas de aprendizaje de RLATES comparando las estrategias de exploración/explotación de *e-greedy* (su parámetro e se ha fijado a 0,9) y *Boltzmann* (su parámetro de temperatura τ se ha fijado a 0,01). El ratio de aprendizaje (α) se ha fijado a 0,5 y se ha trabajado con estudiantes del *grupo2*. El modelo del dominio utilizado es el *B*.

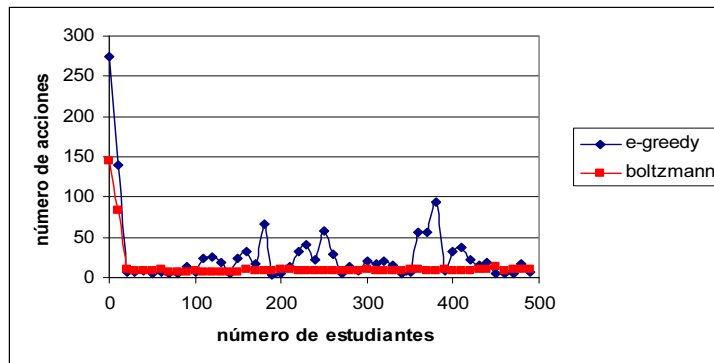


Figura 5.33: Desviación típica de RLATES comparando las estrategias de exploración/explotación de *e-greedy* (su parámetro e se ha fijado a 0,9) y *Boltzmann* (su parámetro de temperatura τ se ha fijado a 0,01). El ratio de aprendizaje (α) se ha fijado a 0,5 y se ha trabajado con estudiantes del *grupo2*. El modelo del dominio utilizado es el *B*.

cuando estudiantes reales interactúan con el sistema, proporcionándoles, en lugar de una sola opción (página web) para continuar aprendiendo, varias, presentando del mismo modo la probabilidad con que el sistema sugiere esa ejecución de la acción, de forma que sea el propio estudiante el que decida si desea seguir el consejo del sistema o continuar aprendiendo a través de otras páginas web. Además, de esta forma, se le proporciona al estudiante sensación de control sobre el sistema, cualidad que se requiere en los sistemas de educación para mantener la atención del estudiante.

Tras esta comparativa, se toma la decisión de implementar la estrategia de exploración/explotación de *Boltzmann* para la interacción con estudiantes reales.

Conclusiones

Como conclusiones a este bloque de experimentos, se observa, que, al igual que ocurría cuando se utilizaba la estrategia de exploración/explotación de *e-greedy*, con *Boltzmann* el sistema también llega a converger a tácticas cercanas a las óptimas (con valores pequeños del número de acciones) y rápidamente (utilizando pocos estudiantes hasta que el sistema converge).

Además, también se han obtenido los siguientes resultados:

- Se han realizado experimentos sobre el efecto de la variación del parámetro τ (temperatura) en la estrategia de exploración/explotación de *Boltzmann*. Se ha concluido finalmente que cuando más avaricioso es el comportamiento del sistema (más explotación y menos exploración; más bajo es el valor de τ), la convergencia será mejor y más rápida. Por lo tanto, se define el valor de $\tau=0,01$ como una buena opción para los sistemas de enseñanza.
- También se han realizado experimentos sobre el efecto de la variación del parámetro α (ratio de aprendizaje de la función de *Q-learning*). Se ha observado cómo cuando este parámetro toma el valor de 0,9, se comporta de la mejor forma.
- Además, se han realizado experimentos sobre el efecto del indeterminismo en las acciones de los estudiantes. Se ha comprobado que cuando más indeterminismo, el sistema converge a peores valores de las tácticas pedagógicas.

En cuanto a aplicaciones futuras sobre estudiantes reales se pueden resaltar los siguientes resultados:

- La estrategia de exploración/explotación de *Boltzmann* proporciona ventajas de aplicación frente a la política *e-greedy* cuando se trata con estudiantes reales, siempre que se tomen valores de τ muy bajos (0,01).

- Se aconseja homogeneizar el grupo de estudiantes, describiendo su MDP lo más determinista posible.
- Se recomienda que el parámetro α tome el valor de 0,5 para obtener un número pequeño de estudiantes en la fase de entrenamiento y obtener una política de comportamiento cercana a la óptima.
- Al utilizar la estrategia de exploración/explotación de *Boltzmann*, el sistema requiere un número mayor de acciones, pero se comporta de forma más estable en cuanto a la diferencia entre las interacciones del sistema respecto a la estrategia de exploración/explotación *e-greedy*

Tras este bloque de experimentación, la conclusión principal es que la estrategia de exploración de *Boltzmann* añade ciertas ventajas cuando se utiliza con estudiantes reales que la política *e-greedy* no podría proporcionar.

5.4.3. Inicialización de las Tácticas Pedagógicas

Hasta el momento se ha demostrado que el sistema de enseñanza es capaz de aprender tácticas pedagógicas cercanas a la óptima, utilizando no muchos estudiantes previamente para converger cuando el sistema parte de una situación inicial sin ninguna información sobre el dominio (es decir, sin inicializar la tabla Q del algoritmo de aprendizaje *Q-learning*). En este apartado nos preguntamos si sería posible reducir ese número de estudiantes necesario para la convergencia, ya que los primeros estudiantes que interactúan con el sistema no aprenden de la mejor forma la materia, ya que cuando ellos interactúan con el sistema, éste todavía no ha aprendido a enseñar.

En esta sección, se experimenta sobre la posibilidad de reducir el periodo de aprendizaje del sistema (el número de alumnos que necesita el sistema hasta que converge a una política pedagógica válida) mediante la inicialización de la tabla Q . Esta inicialización se va a realizar interaccionando con estudiantes simulados con una serie de características de aprendizaje determinadas.

En este sentido, existen dos posibilidades:

- los estudiantes reales siguen el mismo modelo que los estudiantes simulados, por lo que el sistema tendría correctamente inicializada la tabla Q y no necesitaría adaptarse a los nuevos estudiantes.
- no se cumple lo anterior, por lo que el sistema tendría que adaptarse a las nuevas características de los estudiantes.

En esta experimentación nos ponemos en el peor de los casos, donde al principio se inicializa el sistema con estudiantes pertenecientes a un grupo determinado

y, posteriormente, una vez inicializada la tabla Q con tácticas pedagógicas para estudiantes con unas características de aprendizaje determinadas, se cambia el tipo de estudiante, comenzando a interactuar estudiantes con otro tipo de características de aprendizaje.

En la Figura 5.34 se muestra cómo el sistema, que inicialmente parte la ejecución sin previo conocimiento sobre el modelo del dominio, interactúa con 200 estudiantes de tipo *grupo3*, llegando a la convergencia cuando 20 estudiantes han interactuado con RLATES. Una vez que el sistema ya ha convergido a una política pedagógica cercana a la óptima, se cambia el grupo de estudiantes que interactúa con RLATES al *grupo2*. Se comprueba que se requieren, como mucho, 10 estudiantes para llegar a la convergencia.

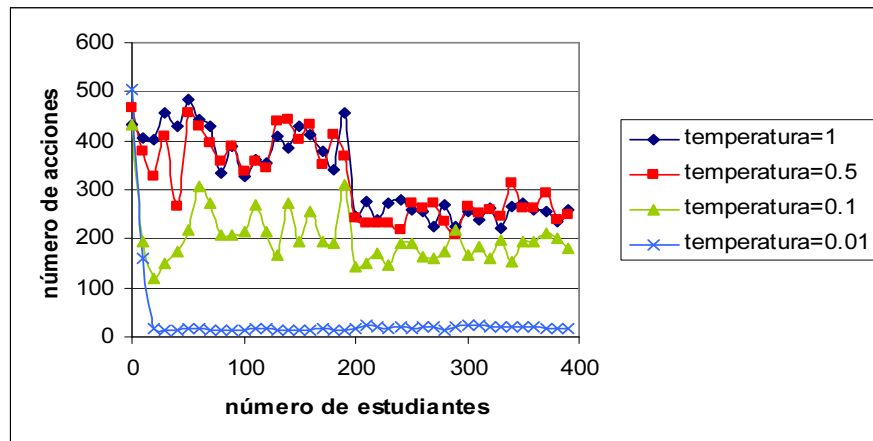


Figura 5.34: Curvas de aprendizaje de RLATES comparando el efecto de la variación en la temperatura de la estrategia de *Boltzmann* y plasmando el efecto de la inicialización del sistema. El ratio de aprendizaje (α) se ha fijado a 0,5 y el modelo del dominio utilizado es el *B*. Se ha trabajado inicialmente con estudiantes del *grupo3* y posteriormente con estudiantes del *grupo2*.

Hay que tener en cuenta que estos experimentos se están realizando en el peor de los casos (cuando las características de aprendizaje de los estudiantes nuevos son muy diferentes a cómo se ha inicializado la tabla Q). En el caso en el que la inicialización coincida con las características de aprendizaje de los estudiantes, el sistema no tendría que volver a aprender la tabla Q , ya que la que utilizaba en primer lugar sería correcta para los estudiantes actuales, por lo que la *fase de*

entrenamiento (ver sección 3.2.5) del sistema se reduciría a cero.

En la Figura 5.35 se puede ver el mismo experimento, pero primero interactuando con estudiantes del *grupo2* y posteriormente interactuando con estudiantes del *grupo3*. De nuevo se puede observar cómo en el mejor de los casos el sistema necesita únicamente 10 estudiantes para converger a la nueva táctica de enseñanza, frente a los 20 que necesitaría si el sistema no estuviera previamente inicializado. Además, el sistema converge mejor (menor número de acciones) cuando está inicializado que cuando no lo está.

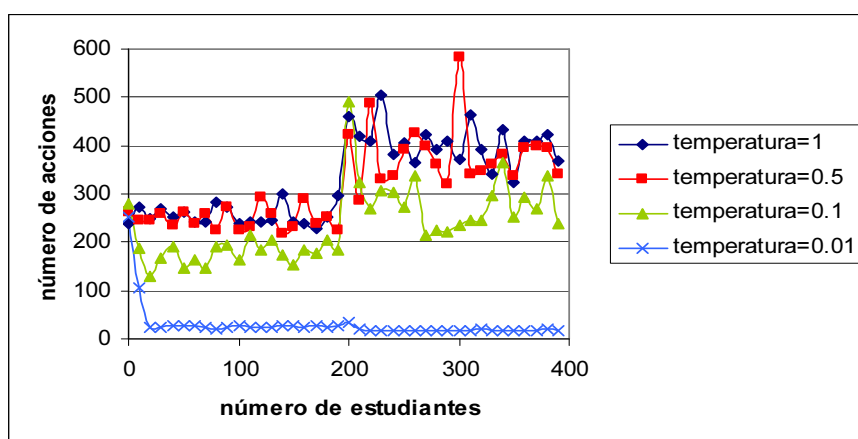


Figura 5.35: Curvas de aprendizaje de RLATES comparando el efecto de la variación en la temperatura de la estrategia de *Boltzmann* y plasmando el efecto de la inicialización del sistema. El ratio de aprendizaje (α) se ha fijado a 0,5 y el modelo del dominio utilizado es el *B*. Se ha trabajado inicialmente con estudiantes del *grupo2* y posteriormente con estudiantes del *grupo3*.

Por último, en la gráfica 5.36 se resumen los dos experimentos anteriores, presentando las curvas de aprendizaje de RLATES cuando el sistema se encuentra inicializado y cuando no, con estudiantes de los grupos *grupo2* y *grupo3* interactúan con RLATES. En esta gráfica se observa claramente cómo el tamaño de la *fase de entrenamiento* del sistema se y cómo, en el caso de los estudiantes del *grupo2*, el sistema aprende una táctica pedagógica mejor (con menor número de acciones). Cuando el sistema interactúa con estudiantes del *grupo3* se observa que, al inicializar el sistema con información pedagógica sobre los estudiantes del *grupo2*, el sistema cae en un mínimo local, necesitando ejecutar un número mayor

de acciones. En este caso es importante tomar la decisión de qué es lo que más interesa para nuestro sistema, aprender las mejores tácticas pedagógicas, o disminuir la *fase de entrenamiento* del sistema. En el caso del sistema de educación a distancia es importante que no muchos estudiantes sufran las consecuencias de la *fase de entrenamiento* del sistema, por lo que creemos que es positivo inicializar el sistema con información pedagógica.

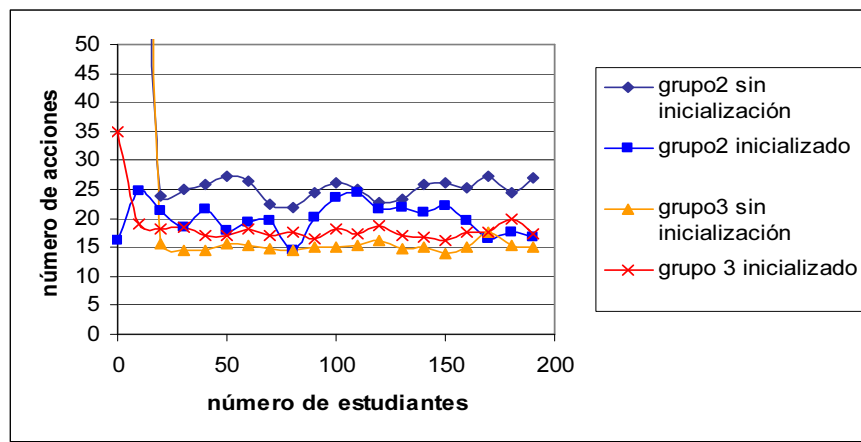


Figura 5.36: Curvas de aprendizaje de RLATES analizando las ventajas de la inicialización.

Conclusiones

Como conclusiones a este grupo de experimentos, se puede decir que la inicialización del sistema RLATES con tácticas pedagógicas es positivo (reduce el tamaño de la *fase de entrenamiento* del sistema y la táctica pedagógica necesita un número menor de acciones a ejecutar), aunque se inicialice con estudiantes que no posean las mismas características de aprendizaje que los actuales.

Se han realizado experimentación en el peor de los casos (cuando las características de aprendizaje de los estudiantes que inicializan el sistema son muy diferentes a los que posteriormente lo utilizarán). Hay que tener en cuenta que si estas características de aprendizaje coincidieran, el tamaño de la *fase de entrenamiento* de RLATES se reduciría a cero.

Tras toda la experimentación realizada en este apartado, se concluye que es

positiva la inicialización del sistema con estudiantes simulados cuando posteriormente estudiantes reales trabajen con RLATES.

5.5. Conclusiones

En esta sección se ha realizado una experimentación exhaustiva con estudiantes simulados, donde se han planteado situaciones parecidas a las reales (que podrían ocurrir con estudiantes humanos interactuando con el sistema). En primer lugar, se analizó si la estrategia de exploración/explotación *e-greedy* proporcionaba un buen aprendizaje al sistema, posteriormente se propuso la utilización de la estrategia de exploración/explotación de *Boltzmann* en lugar de la *e-greedy* y finalmente se realizaron experimentos para tratar de disminuir la *fase de entrenamiento* del sistema. Este apartado resume todas las conclusiones que se han realizado en cada grupo de experimentos:

- El sistema es capaz de aprender tácticas pedagógicas enseñando a estudiantes con distintas características de aprendizaje (las curvas de aprendizaje convergen), tanto con la estrategia de exploración/explotación *e-greedy* como *Boltzmann* en los modelos del dominio utilizados.
- El sistema necesita interactuar con un número no muy grande de estudiantes para converger (convergencia rápida).
- El sistema converge a tácticas cercanas a las óptimas (utilizando un número no muy grande de acciones).
- Cuanto más avariciosa es la política (tanto con *e-greedy* como con *Boltzmann*), RLATES converge a tácticas pedagógicas adecuadas más rápido (utilizando menor número de estudiantes) y mejor (ejecutando menor número de acciones).
- Cuanto mayor es el valor de α , mejor se comporta el sistema (aprende más rápido a enseñar), pero las políticas de comportamiento obtenidas son más inestables.
- Cuanto más indeterminista sea el estudiante, peor aprenderá el sistema y las interacciones entre los estudiantes serán más diferentes (utilizando distinto número de acciones en las curvas de la desviación típica).
- La estrategia de exploración/explotación de *Boltzmann* ofrece ventajas frente a la política *e-greedy*.

- La inicialización del sistema a tácticas pedagógicas válidas para estudiantes con ciertas características de aprendizaje es positiva, incluso cuando los estudiantes que van a utilizar posteriormente RLATES no posean las mismas características de aprendizaje que se han supuesto en la inicialización.

Capítulo 6

Experimentación con Estudiantes Reales

En este capítulo se analizan los resultados al realizar la experimentación con estudiantes reales. En esta experimentación se desea comprobar que la adaptabilidad incorporada en el sistema es útil para los estudiantes. Para ello se realizarán dos tipos de pruebas: experimentos con una versión del sistema que incorpora adaptabilidad a los estudiantes (RLATES) y experimentos con otra versión del sistema que no se adapta a las características de aprendizaje de los estudiantes. También se desea comprobar si los resultados obtenidos en la experimentación con estudiantes simulados puede generalizarse al observar que el comportamiento de RLATES con estudiantes simulados es parecido al comportamiento de RLATES con estudiantes reales. Así mismo, también se desea comprobar si el sistema posee un grado de usabilidad alto.

Con el objetivo de realizar una experimentación adecuada, el primer paso es definir los objetivos de los experimentos se desean realizar. A continuación, se define la fase completa de experimentación, teniendo en cuenta, entre otras cosas, el entorno de las pruebas y la inicialización del sistema. Posteriormente, se detallarán los parámetros de evaluación tenidos en cuenta y finalmente se analizarán los resultados obtenidos y se resumirán en un apartado de conclusiones.

6.1. Motivación de la Experimentación

Los experimentos que se analizan en esta sección parten de la hipótesis de que el sistema se comportará de forma similar a como se comportó con los estudiantes simulados, siempre teniendo en cuenta que el comportamiento de los estudiantes reales es estocástico (indeterminista), dependiendo de muy diversos factores (por ejemplo, diferentes características de aprendizaje, aburrimiento, distracción en el

entorno, motivación de los estudiantes, etc.).

Estos experimentos tratan de responder principalmente a tres cuestiones: primero se comprobará si RLATES es capaz de aprender a enseñar cuando interactúa con estudiantes reales: ¿el sistema será capaz de converger?; se pondrá especial atención en el tamaño de la *fase de entrenamiento* del sistema (número de estudiantes que el sistema necesita para aprender a enseñar correctamente): ¿cuánto tarda en converger el sistema?; por último, también se comprobará lo buena que es la política pedagógica aprendida (el número de acciones que el sistema necesita para enseñar completamente la materia): ¿las tácticas pedagógicas aprendidas son lo suficientemente buenas?. Para responder a estas preguntas, el diseño del experimento se basa en el estudio llevado a cabo con estudiantes simulados, inicializando previamente el sistema con los valores adecuados de los parámetros que afectan la convergencia de RLATES.

Por último, también se desea comprobar la usabilidad del sistema. Para ello, los estudiantes, tras la interacción con el sistema, rellenaron una encuesta que fue diseñada para hallar el grado de usabilidad del sistema.

6.2. Diseño de la experimentación

En este apartado se presenta el diseño de la experimentación realizada, en el que, en primer lugar, se tendrá en cuenta el entorno en el que se han realizado los experimentos, tratando de evitar factores de 'ruido' que puedan corromper los resultados de las interacciones de los estudiantes reales con RLATES. Posteriormente, se tratarán los detalles de inicialización del sistema, basándose en los experimentos realizados con estudiantes simulados. Finalmente, para acabar con el diseño de la experimentación, se mostrarán los principales elementos tenidos en cuenta para diseñar la encuesta de evaluación de la usabilidad del sistema.

6.2.1. Entorno en la Experimentación

Como se comentó en el apartado 4.1, es difícil conseguir una experimentación con estudiantes reales en sistemas de educación adaptativos en la que no influyan las variables dependientes en la experimentación (tales como características propias del estudiante o como las condiciones de entorno del experimento). Las características de aprendizaje del estudiante no se pueden controlar a la hora de realizar los experimentos, pero sí las condiciones de entorno de la experimentación.

Para evitar la influencia de las variables con 'ruido' del entorno en los experimentos, [Chin, 2001] proporcionó una serie de reglas a tener en cuenta:

- Asignar de forma aleatoria suficientes participantes a los grupos.

- Asignar aleatoriamente diferentes horarios a los participantes.
- Comprobar que la habitación no tiene ventanas ni otras distracciones (p.e. posters) y que es tranquila. Los participantes han de estar lo más aislados posible.
- El espacio de que disponen al lado del ordenador debe estar preparado ergonómicamente para los diferentes participantes.
- Si se utiliza una red, evitar que haya tiempos de carga elevados.
- Preparar instrucciones uniformes para los participantes, preferiblemente escritas o en cinta de video o audio.
- Los experimentos deben planearse para minimizar la interacción con el participante.
- Se deberán descartar los datos en los que los participantes requieren hablar con la persona que organiza el experimento durante el experimento.
- Los participantes deberán seguir una serie de reglas locales para comprobar que se lo toman en serio (por ejemplo, firmar un cuestionario después de la interacción).
- Permitir que los participantes estén el tiempo que necesiten. Los experimentos pueden tardar meses en llevarse a cabo.
- Analizar los resultados tras la ejecución de los experimentos.
- Analizar las posibles variables 'ruidosas' que podrían corromper los resultados de los experimentos.

En nuestra experimentación, se han tratado de paliar todos los factores de ruido explicados en el apartado anterior realizando siempre los experimentos en el mismo ordenador del mismo laboratorio, durante varios días (mañana y tarde), de forma que las condiciones de entorno fueran similares para todos los experimentos.

En cuanto a los alumnos que han interactuado con el sistema, se ha trabajado con estudiantes de 2º curso de Ingeniería Técnica en Informática de Gestión (2º ITIG) y de 2º curso de Ingeniería Superior de Informática (2º II). Su conocimiento inicial sobre temas relacionados con el dominio del sistema de educación se conocía gracias a los planes de estudio de las titulaciones. Además, este conocimiento inicial de los estudiantes, fue confirmado por ellos en un cuestionario inicial que rellenaron con el objetivo de realizar una selección de alumnos

adecuada. Los estudiantes de 2º de ITIG seleccionados conocían el *Modelo Relacional de Diseño de Bases de Datos* gracias a la asignatura troncal de segundo curso *Ficheros y Bases de Datos*. Algunos de ellos también conocían el *Diagrama de Clases* propio de las metodologías orientadas a objetos al haber cursado la asignatura optativa de tercer curso *Metodología del Desarrollo del Software*. Por otro lado, los estudiantes de 2º de II seleccionados también conocían el *Modelo Relacional* visto en la asignatura troncal de 2º curso *Ficheros* y el *Diagrama de Clases* visto en la asignatura obligatoria *Metodología de Desarrollo*. Para realizar la experimentación, se han utilizado estudiantes de ambas carreras y de ambos tipos de conocimiento repartidos de forma aleatoria en toda la experimentación, intentando paliar el ruido en las características de aprendizaje de los estudiantes.

A pesar de utilizar alumnos de distintas titulaciones, se ha considerado que pertenecían a un mismo grupo, por lo que no se ha realizado la fase de *agrupación* definida en la sección 3.2.5.

Es interesante comparar las ventajas de utilizar adaptación en el sistema. Para ello, habitualmente se realizan dos tipos de experimentos en el sistema: alumnos que interactúan con una versión del sistema que proporciona adaptación dependiendo de las características del estudiante (RLATES), y otros alumnos que interactúan con una versión del sistema que no proporciona *guía directa* (IGNATES). Algunos investigadores han utilizado *experimentación cruzada* para evaluar sus sistemas, donde el mismo estudiante interactúa en un momento dado con el sistema cuando éste se comporta de forma adaptativa y, en otro momento dado, el mismo estudiante interactúa con el sistema cuando éste no se comporta de forma adaptativa. De esta forma, el alumno podría evaluar las diferencias que existen entre ambas implementaciones. En nuestro sistema no se ha podido utilizar *experimentación cruzada*, ya que el objetivo del sistema es que el sistema aprenda un temario. La primera vez que el estudiante interactúe con el sistema educativo no conocerá el temario, pero la segunda sí, por lo que no trabajará bajo las mismas condiciones iniciales. Además, con el objetivo de que el sistema RLATES pudiera utilizar al máximo la experiencia en interacciones con los estudiantes, modificando sus tácticas pedagógicas, éstos han interactuado con el sistema de forma secuencial. En secciones posteriores se podrán ver los resultados de la comparación de ambas versiones del sistema educativo (ver apartado 6.4).

Por otro lado, con el objetivo de paliar la subjetividad que pudiera añadir el responsable de los experimentos en la interacción de los estudiantes, por ejemplo, indicándoles cómo deberían interactuar con el sistema. Para ello, se ha realizado un experimento *ciego*, donde el responsable no le informaba al estudiantes sobre qué versión del sistema utilizaba (con o sin adaptación).

Tras las interacciones de los estudiantes con el sistema, para garantizar que las diferencias en las variables dependientes se debieran sobre todo a los cambios de valor de las variables independientes, se han utilizado técnicas estadísticas como

la *media* o la *desviación típica* para realizar el análisis del experimento.

6.2.2. Versiones Implementadas del Sistema

Para llevar a cabo la experimentación con estudiantes reales, se han implementado varias versiones del sistema. En concreto, se pensó en comparar las interacciones de los estudiantes cuando trabajaban con la versión adaptativa e inteligente del sistema (versión que se detalló en el apartado 3.4 y a la que se le llamó RLATES) con una versión que no guiaba de forma directa a los estudiantes por el contenido del sistema. Posteriormente se comprobó que la primera versión de RLATES implementada (que denominaremos RLATES1) presentaba algunas características que impedían evaluar correctamente el sistema (por ejemplo, al proporcionar a los estudiantes la corrección de los tests que realizaban, éstos aprendían la forma de aprobar el test, y no el contenido del mismo). Por ello, se creó una segunda versión de RLATES donde se variaron algunas características de su implementación, que denominaremos RLATES2.

En este apartado se pretende detallar el diseño de cada versión del programa implementada, haciendo especial hincapié en los elementos del interfaz con los que puede interactuar el usuario.

Todas las versiones presentaban un interfaz parecido, donde éste se había dividido principalmente en dos marcos. En el marco izquierdo aparecía, entre otra información propia del estudiante actual, el árbol de conocimiento que se pretendía que aprendiera el estudiante. Por otro lado, en el marco derecho aparecía la información que se pretendía que aprendiera el estudiante sobre un tema en concreto. Es en este marco, en el marco derecho, donde varía principalmente la implementación de las distintas versiones del sistema.

Versión del Sistema *IGNATES*

En la figura 6.1 se muestra el interfaz del sistema al que hemos denominado *IGNATES* (del inglés *Indirect Guide Navigation in Adaptive and intelligent Educational Systems*). La principal característica de este sistema es que éste no proporciona al estudiante soporte a la navegación a través del contenido ni presentación del mismo. Es decir, en esta versión es el propio estudiante el que decide qué tema estudiar a continuación, qué tarea (definición, introducción, ejercicio, etc.) dentro de qué tema visitar a continuación, etc. sin ninguna ayuda por parte del sistema.

Para navegar a través de los distintos temas, el estudiante tenía que hacer *click* sobre cualquiera de los nombres de tema que aparecían en el árbol de conocimiento. Si utilizaban los botones de navegación (*Anterior* o *Siguiente*), el sistema les



Figura 6.1: Interfaz de IGNATES.

posicionaba sobre el anterior o siguiente tema respectivamente del árbol de conocimiento, únicamente basándose en la posición de cada tema en el árbol, y no comprobando cuál es el tema que, según las necesidades del estudiante actual, es el más adecuado para visitar a continuación.

Por otro lado, para navegar a través de las distintas tareas de cada tema, el estudiante tenía que pulsar sobre cada pestaña que aparecía en el marco derecho del sistema. En cada una de las pestañas aparecían los diferentes elementos de cada tema (ejercicios, definiciones, ejemplos, etc.). Además, el estudiante decidía cuándo intentar aprobar los tests referentes al tema en cuestión, de forma que si pinchaba en la pestaña denominada *Tests* se le presentaba al estudiante una prueba de tipo *test*.

Versión del Sistema *RLATES1*

En las versiones que se implementaron sobre el sistema RLATES (del inglés *Reinforcement Learning in Adaptive and Intelligent Educational Systems*), RLATES1 y RLATES2, se le prohibió por completo al estudiante pinchar en el árbol de conocimiento, indicándoles que se tenían que dejar guiar por el botón de navegación *Siguiete*. Aunque los estudiantes no podían hacer *click* sobre el árbol

de conocimiento, éstos sí que podían ver su estructura, para que, cuando el sistema les llevaba a un tema concreto, ellos supieran en qué lugar en el árbol de conocimiento se encontraba dicho tema.

Cuando el estudiante pulsaba el botón *Siguiente*, el sistema le ofrecía al estudiante varias opciones para continuar la navegación (ver Figura 6.2) según el algoritmo de *Aprendizaje por Refuerzo* implementado y su política de exploración/explotación de *Boltzmann* (ver apartado 3.3.4). Tras elegir una opción determinada, el sistema le mostraba al estudiante únicamente dos pestañas para visitar (ver Figura 6.3). En la primera pestaña aparecía la tarea determinada que el estudiante había elegido para continuar la navegación a partir de todas las opciones que le había propuesto el sistema. En la segunda pestaña aparecía un test que los estudiantes tenían que contestar para poder continuar con la navegación.

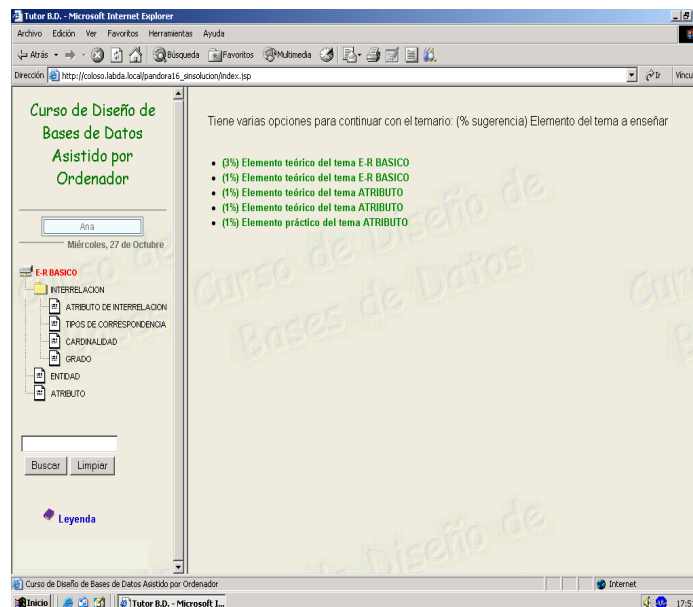


Figura 6.2: Interfaz de RLATES1 y RLATES2. Opciones para continuar la navegación

Versión del Sistema *RLATES2*

La versión de *RLATES2* se diferencia en únicamente tres características con respecto a *RLATES1*. Este cambio en las características del sistema se derivó de los primeros experimentos realizados con estudiantes reales (ver apartado 6.4.1):



Figura 6.3: Interfaz de RLATES1 y RLATES2. Pestañas de contenido

- En RLATES1, el árbol de conocimiento no le indicaba a los estudiantes si el tema lo habían aprobado o no, por lo que éstos a veces se encontraban un poco perdidos en el sistema. En RLATES2 se añaden estos colores indicativos de haber aprobado el tema.
- En RLATES1 las opciones que se presentan a los estudiantes tras pulsar el botón de navegación *Siguiete* se basan estrictamente en la probabilidad de *Boltzmann*, de forma que en algunos casos a los estudiantes sólo se les ofrecía visitar temas que ya habían aprobado. En esta segunda versión del sistema se soluciona este problema, ofreciendo únicamente tareas de temas que el estudiante aún no ha aprobado.
- Al realizar los primeros experimentos con RLATES1, observamos un dato curioso: cuando los estudiantes realizaban tests y no los aprobaban, al proporcionarles la solución a los tests, los alumnos se aprendían la solución a esos test concretos y no aprendían el contenido de los mismos. Para solucionar este problema, en RLATES2 no se proporciona la solución a los tests.

6.2.3. Inicialización del Sistema

El estudio realizado se basa en la experimentación realizada con estudiantes simulados (ver sección 5.4), por lo que la inicialización del sistema tomará los valores de los parámetros que mejoraban la convergencia del sistema, es decir:

- El ratio de aprendizaje de la función de *Q-learning* (α) se fija a 0,9, es decir, el sistema aprende rápidamente a comportarse.
- Se utiliza la estrategia de exploración/explotación de *Boltzmann*, ya que se demostró que poseía ventajas deseables para el sistema que la política *e-greedy* no podía proporcionar (ver secciones 3.2 y 5).
- El parámetro que indica la temperatura en la política de *Boltzmann* (τ) se fija a 0,01, ya que se comprobó en la experimentación con estudiantes simulados que cuanto más avariciosa se comportaba la política, más rápido y mejor aprendía RLATES a enseñar.
- El sistema se inicializa con información del dominio. Esta inicialización se realizará entrenando RLATES con estudiantes simulados que tengan comportamiento parecido a los estudiantes reales con los que posteriormente interactuará el sistema. Como se comentó anteriormente, el comportamiento de los estudiantes reales es estocástico, e imposible de predecir a priori, por lo que escoger un tipo de estudiantes simulados para inicializar el sistema que se adecue perfectamente a las características de aprendizaje de los estudiantes reales es impensable. Por ello, el sistema se inicializará con estudiantes simulados que posean un comportamiento estocástico (único factor que se puede asegurar) y se supondrá que prefieren que se les muestre tareas de tipo *definición* y *ejemplos* en formato de texto e imagen, ya que la experiencia del tutor humano experto en la enseñanza de la materia del curso observó ese comportamiento de forma generalizada en el aula. Por ello, se decidió utilizar los estudiantes simulados de tipo *grupo2* (ver Tabla 5.1), que cumplen todas estas características. La inicialización del sistema con este tipo de estudiantes simulados probablemente no elimine la *fase de entrenamiento* del sistema, pero la hipótesis que se realiza es que reducirá su tamaño, al igual que ocurría en la experimentación realizada con estudiantes simulados.

6.2.4. Evaluación de la Usabilidad del Sistema y Grado de Conocimiento del Estudiante

Es importante, en cualquier sistema hipermedia y, en concreto, en los sistemas hipermedia educativos, evaluar su usabilidad. Al evaluar la usabilidad de un

sistema, es necesario analizar si el sistema es efectivo, eficiente, y seguro; facilita el aprendizaje de los alumnos y que éstos fijen los conocimientos (que los recuerden); y por supuesto, sea útil [Preece *et al.*, 2000].

Para evaluar la usabilidad de un sistema de educación, no sólo hay que evaluar el interfaz de usuario, sino que también hay que evaluar el método didáctico, los materiales instruccionales y los mecanismos de interacción. De esta forma se podrá evaluar si el sistema ayuda o no a los estudiantes a alcanzar sus objetivos [Díaz, 2003].

Realizar una encuesta a los alumnos después de interactuar con RLATES no sólo nos proporcionará una forma de evaluar la usabilidad del sistema, sino que también proporcionará la oportunidad de reunir información valiosa sobre el diseño del sistema (tanto si satisface a los evaluadores, como comprobar que se puede mejorar con sugerencias que nos proporcionen). Por ello, la encuesta (evaluación) ha sido cuidadosamente planeada y preparada, de forma que se puedan recoger datos apropiados y de confianza para trazar conclusiones relevantes.

[Díaz, 2003] proporcionó una serie de criterios para evaluar la utilidad del sistema y la usabilidad del interfaz de usuario. Los criterios que evalúan la utilidad del sistema son:

- la riqueza del sistema, que tiene en cuenta el volumen de información del sistema, diversidad de estilos de presentación e interacción, tipo de tareas y actividades que los estudiantes pueden realizar, etc.,
- completitud: comprobación de que el sistema tiene la información necesaria y el número de mecanismos de interacción suficiente para alcanzar los objetivos educativos de los estudiantes,
- motivación: si el sistema proporciona motivación al estudiante no sólo para utilizar el sistema, sino sobre la materia que se enseña,
- estructura del hipertexto: análisis de las propiedades estructurales del sistema: conectividad, modularidad, etc.,
- autonomía: definiéndolo como el nivel de libertad de navegación o interacción que se le proporciona a los usuarios,
- competencia: capacidad de navegar a través de los contenidos del sistema y alcanzar los objetivos propuestos, entre otras cosas, adaptando el sistema a los diferentes estilos de interacción de los estudiantes y a las preferencias de contenidos,
- flexibilidad: comprobando que el sistema puede ser usado y mantenido.

Por otro lado, los criterios que evalúan la usabilidad del interfaz de usuario son la comprobación de que:

- el interfaz es estético (incluyendo, por ejemplo, información multimedia, que armoniza y es útil para que los usuarios comprendan los conceptos),
- el sistema es consistente (comprobando la utilidad de cada área, botón, enlace, etc. del sistema),
- que el sistema proporcione *autoevidencia* y metáforas (facilidad para que los usuarios adivinen el significado y/o propósito de lo que se le presenta), y
- que el sistema sea previsible (cuando el usuario es capaz de anticiparse a lo que devolverá el sistema).

Los criterios citados anteriormente se han tenido en cuenta a la hora de diseñar la encuesta que rellenan los estudiantes tras la interacción con el sistema educativo, considerando principalmente aquellos que evalúan la usabilidad del sistema en cuanto a la propuesta que se ha realizado en esta tesis doctoral. En la siguiente sección, se proporcionan algunos resultados al cuestionario que rellenaron los estudiantes.

Por otro lado, al finalizar la interacción, cada estudiante, además de rellener un cuestionario para evaluar la usabilidad del sistema, contestaron a un pequeño examen o prueba. El objetivo de este examen final es la de evaluar el nivel de conocimiento del estudiante sobre la materia que se enseñó en el curso. Se tendrá en cuenta esta información a la hora de determinar lo bien o mal que ha enseñado el sistema a cada estudiante, ya que la bondad del proceso de enseñanza que sigue el sistema no sólo depende de lo rápido que aprende a enseñar una secuencia de contenidos válida (medido en número de estudiantes que tarda el sistema en converger) o buena (medido en número de acciones que el sistema necesita mostrar al estudiante para que éste llegue a conocer toda la materia del curso), sino que también interesa conocer cómo de bien ha aprendido el estudiante el conocimiento mostrado (teniendo en cuenta un grado de conocimiento distinto si aprueba o no los exámenes contestados durante el curso).

6.3. Parámetros de Evaluación

Los parámetros utilizados para la experimentación son medidas objetivas dadas por el comportamiento de los estudiantes en la interacción con el sistema. La mayoría de los sistemas de educación adaptativa evalúan su eficacia según el *número de nodos o páginas visitados/as* o el *tiempo* que ha tardado un alumno en

aprender el conocimiento ofrecido por el sistema. Pero existen otros parámetros que no suelen tener en cuenta, por ejemplo *cuánto ha aprendido* el estudiante al finalizar el curso (información que se puede determinar a partir de los resultados de los cuestionarios y/o del nivel de complejidad de cada nodo). Existen otros parámetros que también son importantes, como puede ser la ansiedad, confusión (cuando las páginas cambian), sentimientos, etc., pero estos parámetros son más difíciles de evaluar. Por lo tanto, en nuestro sistema se tendrán en cuenta las siguientes medidas objetivas dadas por el comportamiento del estudiante:

- número de páginas web que ha tenido que visitar el alumno para llegar a su objetivo de aprendizaje (aprender todo el material del curso).
- tiempo total que ha tardado el alumno en alcanzar su meta.
- nivel de conocimiento del estudiante una vez finalizada la interacción con el sistema.

6.4. Resultados Obtenidos

En esta sección se presentan los resultados obtenidos al realizar los experimentos con estudiantes reales. La presentación de los resultados de la experimentación se han dividido en dos apartados, dependiendo de si los resultados se han obtenido de forma objetiva (resultados obtenidos en la interacción de los estudiantes con el sistema RLATES), o subjetivamente (obtenidos a partir del cuestionario de evaluación de usabilidad del sistema).

6.4.1. Resultados de la Interacción Estudiante-RLATES

La experimentación con estudiantes reales se ha dividido en dos, dependiendo del modelo del dominio utilizado en la experimentación (ver sección 4.3). Además, con el modelo del dominio *B* se han realizado dos experimentos, cada uno con una implementación del sistema que se detalla en este apartado. Posteriormente, en cada bloque de experimentos, se distinguirá cuando el sistema funciona con y sin adaptación al estudiante. En la Figura 6.4 se muestra la planificación de la experimentación con estudiantes reales, indicando tanto los parámetros de cada experimento y las variables dependientes (numerados con la palabra “*Real-*” más un número consecutivo).

Es importante destacar el hecho de que en esta experimentación se ha supuesto que todos los estudiantes poseen características de aprendizaje similares (como el conocimiento inicial de la materia, la preferencia en el formato de las páginas web, ya que todos ellos prefieren leer texto acompañado de imágenes en lugar de texto

sin imágenes, etc.). Se ha realizado esta suposición debido a que se ha realizado una selección de alumnos de segundo curso (tanto de la Ingeniería Técnica de Informática de Gestión, como de Ingeniería Superior en Informática), sabiendo de antemano que todos ellos han cursado asignaturas parecidas en cuanto a la materia de *Diseño de Bases de Datos* o similares se refiere, tal y como se introdujo en la sección 6.2. Por lo tanto, se considera que todos los estudiantes que interactúan con RLATES pertenecen al mismo grupo de estudiantes.

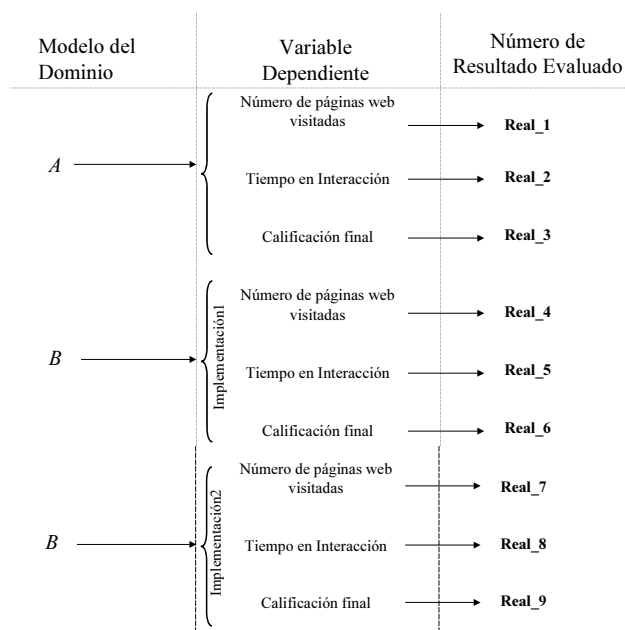


Figura 6.4: Planificación de la experimentación con estudiantes reales.

Modelo del Dominio A

En total, 24 alumnos han trabajado con el sistema cuando éste enseñaba la materia del modelo del dominio A. Doce de ellos han trabajado con el sistema cuando éste se adapta de forma inteligente a las necesidades de cada uno de ellos (*RLATESI*), y otros doce alumnos distintos a los anteriores han trabajado con el sistema sin adaptación (*IGNATES*).

RESULTADO REAL-1

En primer lugar, se desea comprobar si el sistema es capaz de converger interaccionando con un número limitado de estudiantes a tácticas pedagógicas adecuadas para enseñar la materia del curso a los estudiantes que se han seleccionado. En la gráfica 6.5 se muestran dos curvas, una curva de convergencia cuando los estudiantes interactúan con RLATES1 (dibujada en color rojo), y otra curva que muestra el número de páginas web que visitó cada estudiante que interactuó con IGNATES (dibujada en color azul). El eje de abscisas muestra el número de estudiante que interactuó (numerándolos de 1 al número máximo de estudiantes que interactuaron en cada modo), y el eje de ordenadas muestra el número de páginas web que visitó cada estudiante durante la interacción.

Se puede observar cómo la curva de RLATES1 converge a buenas tácticas pedagógicas, llegando a valores de únicamente 3 páginas web cuando el estudiante número 8 interactúa con el sistema (tamaño de la *fase de entrenamiento* del sistema igual a 8). Destacar en este punto que el sistema muestra una única página web por tema (el modelo del dominio A sólo posee tres temas) cuando RLATES1 ha convergido a la política pedagógica óptima. En la curva se observan que los estudiantes 5 y 9 no se comportaron como se esperaba (visitaron más páginas web de las que se suponía que tenían que visitar). Esto se debe a 'ruido' en el sistema derivado posiblemente de infinitas situaciones como, por ejemplo, que quizás esos dos estudiantes no tenían las características de aprendizaje que se esperaba (estaban mal clasificados en este grupo de estudiantes), que no se les diera bien contestar a exámenes de tipo tests, etc.

Por otro lado, la curva de color azul (IGNATES) muestra cómo el número de páginas web que muestra el sistema asciende o desciende dependiendo de las características de aprendizaje y concentración propias de cada estudiante. El número medio de páginas web mostradas a los alumnos que interactúan con el sistema educativo sin adaptación es de 21,41 (mucho mayor que el valor obtenido con RLATES), y la desviación típica corresponde a 13,23, menor diferencia entre las interacciones de los estudiantes, ya que casi todos ellos visitaban casi todas las páginas web del curso antes de finalizar la interacción.

RESULTADO REAL-2

En cuanto al tiempo que han tardado cada uno de los estudiantes en completar una interacción con el sistema, la gráfica 6.6 muestra de nuevo dos curvas: la de color rojo muestra las interacciones con el sistema RLATES1, mientras que la curva de color azul muestra las interacciones con el sistema IGNATES. En el eje de abscisas se muestra el número de estudiante en cada interacción y en el eje de ordenadas el tiempo que tardaron cada uno de los estudiantes en completar la interacción con el sistema.

De nuevo, en la curva de color rojo se puede comprobar cómo el tiempo total que tardaban los estudiantes en completar una interacción (en aprender todo el

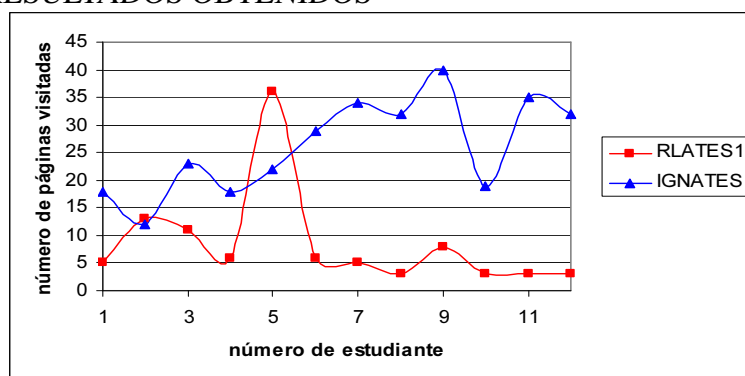


Figura 6.5: REAL-1: Número de páginas web que visitó cada estudiante al completar una interacción con el sistema en cada una de las versiones (RLATES e IGNATES).

contenido del sistema) decrecía según el sistema iban mejorando sus tácticas pedagógicas. Como ocurría con el resultado anterior, esta curva presenta tres picos (en los estudiantes 2, 5 y 9), debido a, como se comentó en el último experimento, errores en la clasificación de los estudiantes o cualquier otro tipo de 'ruido' proveniente del entorno. Pese a estos tres estudiantes que se desmarcan de los estudiantes que se suponía haber seleccionado, la curva de color rojo parece que presenta una convergencia, hasta llegar a valores de tiempo de 5 minutos en finalizar la interacción con el estudiante cuando el sistema ha interactuado con 8 estudiantes.

Sin embargo, al observar la curva de color azul (de IGNATES), el tiempo total que tardan los estudiantes en completar una interacción con el sistema se mantiene en valores medios de 26,16 minutos, con una desviación típica de 8,6 minutos de diferencia.

RESULTADO REAL-3

En este punto se podría pensar que el sistema RLATES1 es mucho más eficiente y eficaz que el sistema educativo IGNATES, pero ¿qué ocurre con el conocimiento final que poseen los estudiantes tras la interacción? ¿cómo aprenden mejor los contenidos del sistema, con o sin adaptación al estudiante? Tras la interacción, cada alumno contestó a un pequeño examen, esta vez de respuesta abierta (no de tipo test) cuya evaluación (realizada por un tutor humano) proporcionaría una medida válida del conocimiento final del estudiante sobre la materia. Tras evaluar cada uno de los exámenes se llegó a la conclusión de que la media de las calificaciones obtenidas por los alumnos que interactuaron con RLATES1 es de 9,58 (puntuando de 0 a 10), mientras que la nota media de los estudiantes que interactuaron con IGNATES es de 9,62 (diferencia poco significativa). En cuando

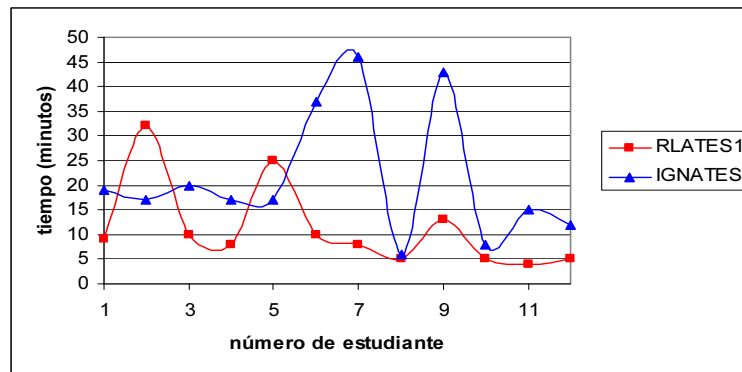


Figura 6.6: REAL-2: Tiempo que tardó cada estudiante en completar una interacción con el sistema.

a la desviación típica de estas calificaciones, es de 0,37 en RLATES1 y 0,35 en IGNATES (igualmente, diferencia poco significativa).

CONCLUSIONES

Gracias a este grupo de experimentos con estudiantes reales y el modelo del dominio A se pueden realizar algunas conclusiones importantes que a continuación se resumen:

- La primera y más importante conclusión es que, tras la experimentación con estudiantes reales, se puede afirmar la hipótesis presentada en la experimentación con estudiantes simulados: el sistema RLATES1 es capaz de converger, modificando sus tácticas pedagógicas aprendiendo a enseñar a los estudiantes en cada interacción.
- Además, se ha comprobado que RLATES1 llega a converger a tácticas pedagógicas óptimas en cuanto al número de páginas web mostradas se refiere, mostrando una única página web por tema para que los estudiantes finalicen su interacción. Si se compara este resultado con el obtenido en la interacción con estudiantes simulados (ver Tabla 6.1), se comprueba que en los estudiantes simulados se proporcionaban valores muy pesimistas que efectivamente, en la realidad se han mejorado. También se ha demostrado que en la interacción con IGNATES se visitan muchas más páginas, habiendo obtenido valores medios de 26.16 páginas web.
- En cuanto al número de interacciones con estudiantes que el sistema necesita para aprender a enseñar, se han obtenido valores de únicamente 8 estudiantes. Al comparar con los valores obtenidos en la simulación de los

estudiantes (ver Tabla 6.1), también se puede comprobar cómo se realizaron predicciones pesimistas en la experimentación con estudiantes simulados.

- También se ha comprobado cómo el tiempo total de interacción con el sistema es mayor cuando los estudiantes interactúan con IGNATES (valores 21,41 minutos) que cuando interactúan con RLATES (valores medios de 11,16 minutos y valores inferiores a 5 minutos cuando el sistema ha convergido a una política pedagógica óptima).
- Por último, se ha comprobado que el grado de conocimiento global que adquieren los estudiantes sobre la materia es muy parecido cuando interactúan con el sistema RLATES y IGNATES.

Gracias a todas estas premisas, se puede concluir que el sistema RLATES1 cuando se adapta a las necesidades de los estudiantes introduce una serie de propiedades muy ventajosas para los estudiantes.

	Estudiantes Simulados	Estudiantes Reales
Número de páginas web	3,2	3
Número de estudiantes necesarios para converger	10	8

Tabla 6.1: Comparación de los resultados obtenidos en la experimentación con estudiantes simulados y reales cuando el sistema utilizaba el modelo del dominio A

Modelo del Dominio B

Con el modelo del dominio *B* se han realizado varios experimentos, con el objetivo de evaluar de forma más objetiva los resultados obtenidos.

En primer lugar, se realizó una primera experimentación con estudiantes que estaban finalizando sus estudios en la materia de *Ficheros y Bases de Datos* o *Ficheros* de las titulaciones de *Ingeniería Técnica en Informática de Gestión* e *Ingeniería en Informática* respectivamente. Además, la mayoría de estos alumnos habían cursado también una asignatura de *Ingeniería del Software* donde se estudiaban algunas nociones del modelo *Entidad-Interrelación (E/R)*. Estos experimentos se realizaron con la versión del sistema RLATES1. En este documento se ha denominado a esta experimentación como *Real-4*, *Real-5* y *Real-6*.

Posteriormente, tras detectar ciertas características en la experimentación anterior que podría incluir cierto “ruido” en los resultados, se repitieron los experimentos, esta vez realizando mejoras en dichas características, modificando algunos parámetros del entorno en RLATES1. A esta nueva versión de RLATES se la denominó RLATES2. En esta ocasión, los experimentos se realizaron con estudiantes que aún no habían comenzado a estudiar las asignaturas de 2º de Ingeniería Técnica en Informática de Gestión, por lo que no tenían conocimientos previos sobre el modelo *Entidad-Interrelación (E/R)*. En los apartados *Real-7*, *Real-8* y *Real-9* se detallan los resultados obtenidos en dicha experimentación.

RESULTADO REAL-4

Como en la experimentación anterior (con el modelo del dominio *A*), en primer lugar se desea comprobar si el sistema es capaz de converger. Para ello, 14 estudiantes interactuaron con RLATES1 y otros 14 estudiantes interactuaron con IGNATES. En la gráfica 6.7 se muestran dos curvas. La curva de color azul muestra el número de páginas web que necesita cada estudiante para concluir una interacción completa con el sistema IGNATES. Por otro lado, la curva de color rojo muestra el número de páginas web que necesitó cada estudiante al interactuar con RLATES1. En esta curva se observa cómo el sistema no parece que vaya a converger a ninguna táctica pedagógica buena para los estudiantes.

En la gráfica se puede observar claramente cómo los estudiantes que interactúan con el sistema RLATES1 visitan un número menor de páginas web para interactuar que los estudiantes que visitan el sistema de educación no adaptativo. En concreto, los estudiantes que interactúan con RLATES1 visitan una media de 34,28 páginas web, mientras que los estudiantes que interactúan con el sistema sin adaptación necesitan una media de 90,85 páginas web para concluir cada interacción. En cuanto a la desviación típica que existe entre las interacciones, las interacciones con RLATES1 poseen una varianza de 16,22, mientras que las interacciones con el sistema sin adaptación posee una varianza de 14,56.

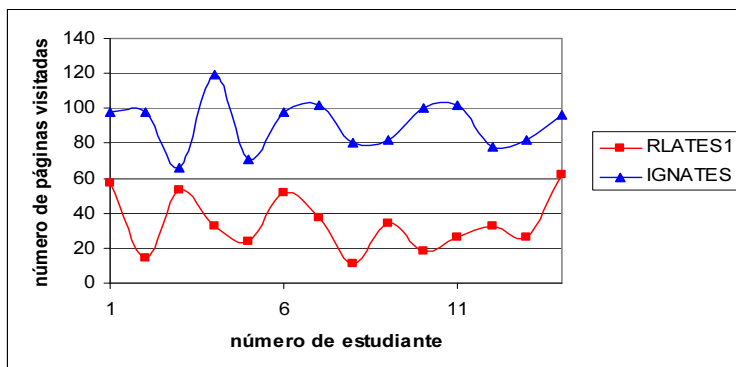


Figura 6.7: REAL-4: Número de páginas web que visitó cada estudiante al completar una interacción con el sistema en cada una de las versiones (RLATES e IGNATES).

RESULTADO REAL-5

En cuanto al tiempo de interacción de cada estudiante con el sistema, la gráfica 6.8 muestra dos curvas. La curva de color azul indica el tiempo (en minutos) que ha tardado cada estudiante cuando interaccionó con el sistema IGNATES. Por otro lado, la curva de color rojo indica el tiempo (en minutos) que ha tardado cada estudiante cuando interaccionó con el sistema RLATES1 (adaptativo e inteligente). Esta curva muestra que el sistema no converge a ninguna táctica pedagógica concreta, ya que no se estabiliza en ningún momento.

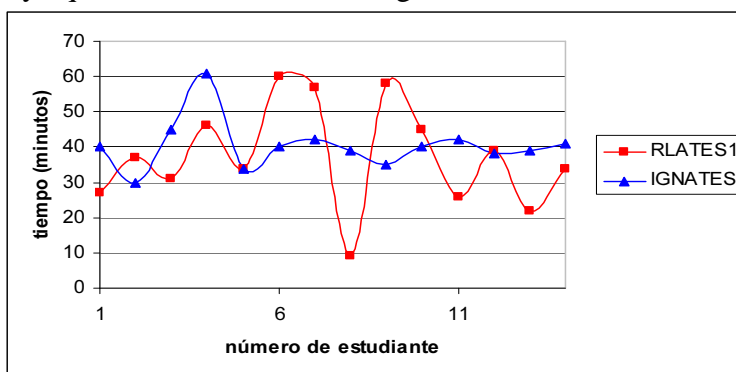


Figura 6.8: REAL-5: Tiempo que tardó cada estudiante en completar una interacción con el sistema.

Al igual que ocurrió con el análisis del número de páginas web que visita cada estudiante en cada interacción, el tiempo medio de los estudiantes que interactúan con RLATES1 es menor (37,5 minutos) que el tiempo medio de los estudiantes

que interactúan con IGNATES (40,43 minutos). En cuanto a la desviación típica, en las interacciones con RLATES1 toma valores de 14,67, mientras que en las interacciones con el sistema IGNATES toma valores menores (7,01).

RESULTADO REAL-6

Para finalizar con el análisis de los primeros experimentos con el árbol de conocimiento *B*, se comprobó, al igual que se hizo en los experimentos con el árbol de conocimiento *A* (ver sección 6.4.1), qué estudiantes aprendían mejor los contenidos del sistema, los que interactuaban con el sistema RLATES1 o IGNATES. Para ello, tras la interacción, cada estudiante contestó a un pequeño examen de respuesta abierta (distinto a los exámenes que habían tenido durante la interacción, ya que eran de tipo test), que posteriormente evaluó un tutor humano.

Tras evaluar el conocimiento final de cada uno de los estudiantes que interactuó con el sistema, se llegó a la conclusión de que no existían diferencias significativas en cuanto al grado de conocimiento final obtenido por cada estudiante. En concreto, los estudiantes que interactuaban con RLATES1 fueron evaluados con un 9,50 en media sobre 10, mientras que los estudiantes que interactuaron con el sistema IGNATES obtenían un 9,57 en media. En cuanto a la desviación típica obtenida, en los resultados de los estudiantes que interactuaron con RLATES era de 0,34, mientras que en los resultados de los estudiantes que interactuaron con el sistema IGNATES era de 0,35.

CONCLUSIONES A LOS RESULTADOS REAL4, REAL5 Y REAL6

Analizando los resultados previamente expuestos en las secciones *Real-4*, *Real-5* y *Real-6*, observamos que esta experimentación presentaba una serie de características que podrían ser las responsables de que RLATES1 no converja:

- **Colores en el Árbol de Conocimiento.** Cuando se realizó este experimento, en el árbol de conocimiento que aparece en el marco izquierdo de la pantalla (ver apartado 3.4), no se distinguían los temas que conocía el estudiante (temas sobre los que el estudiante había aprobado algún test: temas aprobados) de los temas que el estudiante no conocía (temas suspensos). Por ello, en la mayoría de los casos, los estudiantes olvidaban qué temas habían aprobado y qué temas no, por lo que no tenían claro qué tema elegir a continuación para seguir estudiando. Por ello, se modificó la implementación del sistema para distinguir mediante *anotación* (ver apartado 3.4.2) los temas que el estudiante conocía (había aprobado algún test), de los que no. Los temas conocidos se colorean en color verde y los temas no conocidos en color naranja.
- **Solución de Tests.** En la primera versión del sistema, cuando los estudiantes realizaban un test de evaluación de su conocimiento, siempre se les muestra-

ba la solución correcta, suponiendo que esta realimentación podría ser de utilidad para que los estudiantes aprendieran de sus errores. Tras realizar los primeros experimentos se observó que los estudiantes no aprendían más sobre el tema, sino que se aprendían la forma de aprobar el test. Por lo tanto, en la segunda versión del sistema, se decidió no mostrar las soluciones a los tests, pero sí se les indicaba si habían aprobado o no el test.

- **Opciones para seguir estudiando.** Por último, la versión del sistema con adaptación (RLATES1), tras pulsar en la página el botón de navegación “siguiente”, el sistema le ofrecía 5 opciones para elegir, cada una de ellas apuntando a un elemento (definición, introducción, ejercicio, etc.) de un determinado tema del árbol de conocimiento. En la primera versión de RLATES (RLATES1), al estudiante se le ofrecían las cinco primeras opciones según la probabilidad de *Boltzmann* (ver apartado 3.2.3), sin tener en cuenta si el elemento presentado pertenecía a un tema conocido por el usuario o no. En la segunda versión de RLATES (RLATES2) esta característica sí se tiene en cuenta, presentándole al estudiante siempre opciones de elementos a visitar que aún no conoce.

Tras modificar el sistema RLATES con los aspectos previamente expuestos, se realizó otro bloque de experimentos que se detallan en los apartados *Real-7*, *Real-8* y *Real-9*.

RESULTADO REAL-7

De nuevo, tras realizar las modificaciones comentadas en el apartado anterior en el entorno del sistema de educación, se comprueba si RLATES2 es capaz de converger a alguna táctica pedagógica concreta para mostrar el contenido del sistema a los estudiantes. En esta experimentación han trabajado 26 alumnos con el sistema RLATES2 y 26 alumnos con el sistema IGNATES.

En la gráfica 6.9 se muestran dos curvas: la curva de color rojo indica el número de páginas web que ha visitado cada estudiante en su interacción con el sistema RLATES2, mientras que la curva de color azul indica el número de páginas web que visitaron los estudiantes que interactuaron con el sistema IGNATES.

En la gráfica 6.9 se observa cómo RLATES2 converge a una táctica pedagógica buena (mostrando aproximadamente 28 páginas web a cada estudiante) cuando el sistema interaccionó con aproximadamente 13 alumnos. Es decir, el tamaño de la *fase de entrenamiento* del sistema es aproximadamente igual a 13.

Al igual que ocurría en la experimentación con estudiantes reales en el modelo del dominio *A* (ver apartado 6.4.1), en esta experimentación también se ha encontrado algún estudiante que no se comportó como se esperaba (visitaron más páginas web de las que se esperaba). Por ejemplo, el usuario 24 y el usuario 26 parece que no tenían características de aprendizaje similares a la de sus compañeros,

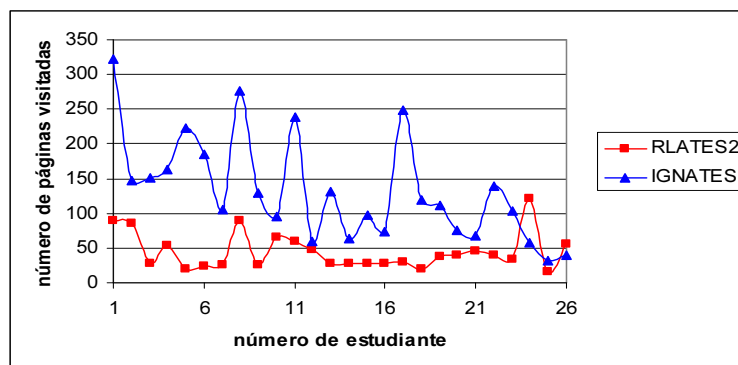


Figura 6.9: REAL-7: Número de páginas web que visitó cada estudiante al completar una interacción con el sistema en cada una de las versiones (RLATES e IGNATES).

aunque también podría derivarse de variables 'ruidosas' en la interacción, como por ejemplo el cansancio, la dificultad de aprobar tests, etc.

Por último, la curva de color azul muestra el número de páginas que visitó cada estudiante al interactuar con el sistema IGNATES. Esta muestra cómo cada estudiante necesitaba un número determinado de páginas según sus necesidades pedagógicas. El número medio de páginas web que visitaron estos estudiantes es 132,88, mientras que el número medio de páginas visitadas por los estudiantes que interactuaron con RLATES2 es bastante inferior: 44,73. Esto se debe a que en RLATES2 el estudiante es guiado a través de las páginas web, mientras que en el sistema IGNATES, el estudiante navega libremente por las páginas web. En cuanto a la desviación típica de las interacciones, se obtuvieron valores de 75,86 páginas web cuando los estudiantes interactuaron con IGNATES, y 26,65 cuando los estudiantes interactuaron con RLATES2.

RESULTADO REAL-8

En este apartado se analiza el tiempo que ha tardado cada estudiante en completar su interacción con el sistema. La gráfica 6.10 muestra dos curvas. La curva de color azul indica el tiempo que ha tardado cada estudiante al interactuar con el sistema IGNATES, y la curva de color rojo indica el tiempo que ha tardado cada estudiante en completar una interacción con RLATES2.

Al analizar la curva de color rojo, se puede observar que no parece que converja el tiempo que tardan los estudiantes en completar la interacción a ningún valor en concreto. Esto se debe a que los estudiantes tienen diferentes características de aprendizaje. En concreto, se observó que algunos estudiantes tardaban más que otros en leer y asimilar el contenido de cada página web. Por ello, aunque en el apartado anterior se ha comprobado que el número de páginas web visitadas

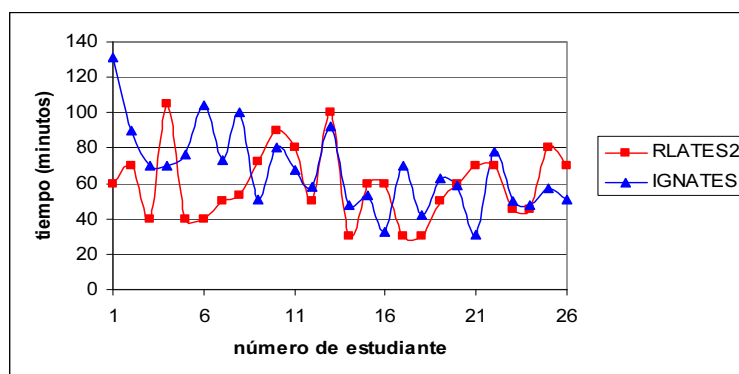


Figura 6.10: REAL-8: Tiempo que tardó cada estudiante en completar una interacción con el sistema.

por los estudiantes que interactuaron con RLATES2 convergía, el tiempo que ha tardado cada estudiante en comprender el contenido del sistema no converge.

Por otro lado, la curva de color azul muestra el tiempo que ha tardado cada estudiante en completar su interacción con IGNATES. El tiempo medio que tardaron los estudiantes en interactuar es de 64,15 minutos, mientras que el tiempo medio que tardaron los estudiantes que interactuaron con RLATES2 es de 59,61 minutos. Estos valores son mejores cuando los estudiantes interactúan con RLATES2 porque el sistema les guía a través de los contenidos del tutor, mientras que los estudiantes que interactúan con el sistema IGNATES navegan libremente a través de las páginas web, por lo que visitan más páginas y, por tanto, necesitan más tiempo para completar la interacción. La desviación típica obtenida cuando los estudiantes interactuaron con el sistema sin adaptación es de 22,97 minutos, mientras que la desviación típica cuando interactuaron con RLATES2 es de 20,55.

RESULTADO REAL-9

Por último, en este bloque de experimentos, se comprobó el grado de conocimiento final de cada estudiante tras interactuar con ambas versiones del sistema. Para ello, un tutor humano les evaluó un ejercicio final de respuesta libre en el que tenían que demostrar sus conocimientos dibujando un *esquema Entidad-Interrelación* completo de una base de datos.

La calificación final media obtenida por los estudiantes que interactuaron con RLATES2 (9,54 sobre 10) no se diferenciaba mucho de la calificación final media obtenida por los estudiantes que interactuaron con IGNATES (9,44). En cuanto a la desviación típica, los estudiantes que interactuaron con RLATES2 presentaron un valor de 0,47, mientras que los estudiantes que interactuaron

con IGNATES presentaban valores de 0,55.

CONCLUSIONES A LOS RESULTADOS REAL7, REAL8 Y REAL9

Como conclusiones a estos experimentos se pueden realizar las siguientes afirmaciones:

- El sistema RLATES2 converge en cuanto al número de páginas web mostradas a los estudiantes con una *fase de entrenamiento* igual a 13 estudiantes. Además, el número medio de páginas visitadas por los estudiantes que interactuaron con RLATES2 es inferior al número medio de páginas visitadas por los estudiantes que interactuaron con el sistema IGNATES.
- En cuanto al tiempo de interacción de los estudiantes con el sistema, se comprobó que algunos estudiantes necesitaban más tiempo que otros en asimilar los contenidos de las páginas web visitadas. Por ello, aunque en RLATES2 el número de páginas web visitadas convergía a un determinado valor, el tiempo que los estudiantes tardaban en completar una interacción variaba. Aún así, el tiempo de interacción de los estudiantes con RLATES2 es significativamente inferior al tiempo de interacción de los estudiantes con IGNATES.
- Por último, se comprobó que el grado de conocimiento final que obtuvieron ambos grupos de estudiantes (los que interactuaron con RLATES2 y los que interactuaron con IGNATES) era bastante bueno, sin diferencias significativas entre ellos.

Por último, se han analizado en profundidad los cambios acontecidos en la política pedagógica que se deriva de la tabla Q aprendida. Es decir, se han analizado las variaciones producidas en las tácticas pedagógicas del sistema RLATES2. Se compararon los valores iniciales de la tabla Q con los valores de la misma tras las interacciones de los 26 estudiantes. Como resumen, se se puede afirmar que, en general, los valores en la tabla Q de las páginas web en formato *texto* (sin imágenes) se mantenían o disminuían, mientras que los valores de las páginas web de elementos de tipo *definición y ejemplo* con formato de *texto e imagen* aumentaban. Además, se mantuvo una secuencia lógica de los contenidos del sistema a nivel de tema (a alto nivel) según los expertos en la materia.

CONCLUSIONES GENERALES EN EL MODELO DEL DOMINIO B

Al analizar este grupo de experimentos en el modelo del dominio B se pueden afirmar algunas conclusiones que exponemos a continuación:

- El sistema es capaz de converger a tácticas de enseñanza adecuadas, afirmando la hipótesis planteada al principio de la experimentación. El sistema

en cada interacción modifica sus tácticas pedagógicas que se encuentran implícitamente representadas en el modelo de *Aprendizaje por Refuerzo*.

- Se ha comprobado que la política pedagógica a la que converge el sistema en cuanto al número de páginas web visitadas por los estudiantes que interactúan con RLATES2 (44,73) es mejor que el resultado obtenido cuando los estudiantes interactuaban con el sistema IGNATES (159,50). Si comparamos los resultados obtenidos con los resultados que se obtuvieron con la experimentación con estudiantes simulados (ver apartado 5.4.2 y tabla 6.2), se puede comprobar cómo en los experimentos simulados se habían realizado supuestos optimistas al suponer que todos los estudiantes se iban a comportar exactamente de la misma forma, por lo que el número de páginas web visitadas al converger la política era menor (24 páginas web visitadas) que los resultados obtenidos en la experimentación con estudiantes reales (44,73).
- Por otro lado, la *fase de entrenamiento del sistema*, es decir, el número de estudiantes con los que necesita interactuar el sistema hasta converger a tácticas pedagógicas adecuadas para ese conjunto de estudiantes, es de únicamente de 13 estudiantes. Al comparar estos resultados con los obtenidos en los experimentos realizados con estudiantes simulados (ver apartado 5.4.3 y tabla 6.2), se puede comprobar cómo en los experimentos simulados se realizaron predicciones pesimistas (se necesitaban 20 estudiantes para converger).
- En cuanto al tiempo total de interacción con el sistema, se ha comprobado que los estudiantes que interactúan con RLATES2 tardan menos tiempo en media en concluir una interacción (59.61) que los estudiantes que interactúan con el sistema IGNATES (74.52).
- Por último, también se analizó el grado de conocimiento global que adquirieron los estudiantes tras interactuar con ambas versiones del sistema (con y sin adaptación), observando que no se encontraban diferencias significativas en este valor.

Por lo tanto, se puede llegar a la conclusión de que el sistema RLATES2 introduce ventajas con respecto a la versión IGNATES del sistema, permitiendo que los estudiantes sean capaces de aprender el contenido del sistema en menor tiempo y visitando un número menor de páginas web.

	Estudiantes Simulados	Estudiantes Reales
Número de páginas web	24	44,73
Número de estudiantes necesarios para converger	20	13

Tabla 6.2: Comparación de los resultados obtenidos en la experimentación con estudiantes simulados y reales cuando el sistema utilizaba el modelo del dominio B

6.4.2. Evaluación de la Usabilidad del Sistema

En este apartado se presentan los resultados obtenidos gracias al cuestionario relleno por los estudiantes tras realizar la interacción con el sistema (tanto con RLATES como con IGNATES). Mediante este cuestionario, son los propios estudiantes los que evalúan la usabilidad del sistema. Hay que tener en cuenta que los resultados que se obtendrán son resultados subjetivos (dependen de la valoración que realice cada estudiante), por lo que posteriormente se utilizarán métodos estadísticos de normalización de resultados, como la media y la desviación típica.

El objetivo principal de esta evaluación es obtener realimentación por parte de los estudiantes en cuanto a la usabilidad del sistema, de forma que en un futuro se puedan tener en cuenta todas las sugerencias realizadas por los usuarios del sistema. En esta tesis doctoral es de especial importancia el conocer si los estudiantes prefieren que se les guíe de forma directa, o prefieren que se les proporcione más autonomía en la navegación.

El cuestionario se ha dividido principalmente en cinco contextos diferentes: el primero de ellos trata del conocimiento inicial que posee el estudiante antes de interactuar con el sistema; el segundo de ellos trata de evaluar la utilidad del sistema; el tercero trata de la interfaz de usuario del sistema; el cuarto trata del contenido del sistema (páginas web); y, por último, el quinto trata de la navegación que los estudiantes realizan para llegar a sus objetivos.

Situación Inicial del Estudiante

En primer lugar, se les preguntó sobre su situación inicial de conocimiento sobre la materia que se enseñaba en el curso u otras materias parecidas. Esta información es necesaria en nuestro sistema para clasificar a los estudiantes según sus características de aprendizaje cuando utilizan el sistema de manera adaptativa. En la Tabla 6.3 se presentan las cuestiones realizadas a los estudiantes en cuanto al conocimiento inicial (antes de interactuar con el sistema) se refiere. En la tabla, en cada una de las casillas, se ha indicado el número exacto de alumnos que respondieron con dicha valoración a la pregunta.

De los 52 estudiantes que interactuaron con la versión inteligente del sistema (RLATES), 29 de ellos no conocían en absoluto el *modelo E-R*, 17 de ellos conocían *Poco* el modelo, 6 de ellos tenían un conocimiento *Neutro*, 9 de ellos conocían *Algo* el modelo y finalmente, 2 de ellos conocían *Mucho* el modelo. Aunque éstas fueron sus respuestas, posteriormente, cuando contestaban a la siguiente pregunta, nos dimos cuenta de que habitualmente confundían estos modelos con el *Modelo Relacional* (perteneciente al *Modelo Lógico* en el material de *Diseño de Bases de Datos*, y no al *Modelo Conceptual*) o con el *Diagrama de Clases* propio de las metodologías de diseño de sistemas orientados a objetos. Por

lo tanto, inicialmente ninguno de ellos habían trabajado con el *Modelo E/R* y se supuso que todos pertenecían al mismo grupo de estudiantes (poseían similares características de aprendizaje). La misma situación inicial pudo ser observada con los estudiantes que interactuaron con el sistema sin adaptación (52 estudiantes en total), donde 16 de ellos respondieron que no conocían en absoluto el modelo, 5 de ellos lo conocían *Poco*, 28 de ellos conocían *Algo* del modelo, 1 de ellos lo conocía *Mucho*, y dos de ellos poseían un conocimiento inicial *Neutro*.

	Mucho	Algo	Neutro	Poco	Nada
¿Conoces el modelo Entidad-Interrelación (E/R)?	3	37	8	20	36
¿Conoces otros modelos que se le parezcan?					
- Modelo:					
- Modelo:					
- Modelo:					
- Modelo:					

Tabla 6.3: Evaluación del sistema: Conocimiento en Modelos de Diseño

También se les preguntó si habían interactuado con sistemas de educación parecidos al sistema educativo que evaluaban en ese instante (ver Tabla 6.4). La gran mayoría de ellos (tanto los que interaccionaron con adaptación como los que interaccionaron sin adaptación del sistema) no habían trabajado con ningún otro sistema de enseñanza parecido (se trataba de un sistema nuevo para ellos). Sin embargo, como se verá posteriormente, no tuvieron ningún problema de interacción con el sistema, sino todo lo contrario, lo encontraron intuitivo y útil. En cuanto al tiempo que trabajaron con el sistema, la mayoría de ellos (56 estudiantes en total) valoraron como *Poco* el tiempo de interacción con el sistema, indicando que no se habían cansado en la interacción con el mismo. Estas valoraciones son subjetivas, pero el sistema durante la interacción recogía los valores objetivos del tiempo exacto que los estudiantes habían estado trabajando con el sistema educativo. En el apartado 6.4.1 se analizan, entre otras cosas, la cantidad de tiempo real que ha tardado cada estudiante en completar su interacción.

Utilidad del Sistema

A continuación, se evaluó la utilidad del sistema preguntándoles, en primer lugar, qué opinión les merecía el sistema con respecto a las características de *Utilidad*, *Complejidad* y *Complejidad*. En la Tabla 6.5 se muestran las valoraciones de los estudiantes, indicando en cada casilla el número exacto de estudiantes que

	Mucho	Algo	Neutro	Poco	Nada
¿Has trabajado con otros sistemas de enseñanza parecidos?			6	12	86
¿Conoces otros modelos que se le parezcan? - Sistema: - Sistema: - Sistema: - Sistema:					
Tiempo que has trabajado con RLATES	12	17	19	56	0

Tabla 6.4: Evaluación del sistema: Conocimiento en Sistemas de Enseñanza

indicó cada valoración. La mayoría de los estudiantes que interaccionaron con el sistema, con ambas versiones (con y sin inteligencia artificial), respondieron que la *Utilidad* era *Buena*, y que la *Complejidad* y *Complejitud* eran *Buenas* o *Neutras*.

	Muy Buena	Buena	Neutra	Mala	Muy Mala
Utilidad	19	73	10	1	1
Complejitud	9	55	35	5	0
Complejidad	11	55	30	6	2

Tabla 6.5: Evaluación de sistema: UTILIDAD. ¿Qué opinión te merece el sistema con respecto a las siguientes características?

También se les preguntó qué mejorarían sobre el sistema. En este apartado, los estudiantes fueron en general muy creativos, proponiendo que, por ejemplo, en las preguntas de test se incorporara un cronómetro para incrementar la presión sobre el evaluado; o que los test se evaluaran de 0 a 10 en lugar de aprobado o suspenso; también echaron de menos más interacción con el sistema a la hora de realizar problemas o ejercicios; además, propusieron que se incorporaran más ejercicios y problemas prácticos, y que éstos fueran más complejos, subiendo el nivel de dificultad según el estudiante que interactúe con el sistema; Otra propuesta es que se les proporcionara una batería de ejercicios o tests que les sirviera para repasar el conocimiento aprendido.

Interfaz de Usuario del Sistema

En cuanto al interfaz de usuario se refiere, en primer lugar se les preguntó su opinión respecto a diferentes características mostradas en la Tabla 6.6. En dicha tabla se muestra exactamente el número de estudiantes que respondió a cada valoración. En general, todos los estudiantes poseían una opinión positiva sobre las características del interfaz de usuario que se consultaban.

	Muy Buena	Buena	Neutra	Mala	Muy Mala
Utilidad	15	78	9	1	1
Rapidez	42	48	10	4	0
Fiabilidad	14	54	27	8	1
Facilidad de Uso	26	49	23	5	1
Consistencia de la interfaz	28	51	21	4	0
Adecuación de la interfaz al tipo de usuario	20	60	17	6	1
Calidad de los iconos	23	60	15	5	1
Legibilidad	37	55	11	1	
Diseño del árbol de conocimiento	18	55	25	5	1
Diseño de los botones de navegación	21	56	21	5	1
Sensación de comodidad con el interfaz	23	68	10	3	0

Tabla 6.6: Evaluación del sistema: INTERFAZ. ¿Qué opinión te merece EL INTERFAZ DEL SISTEMA con respecto a las siguientes características?

Además, se les preguntó su grado de satisfacción global con respecto al sistema (ver Tabla 6.7), respondiendo la mayoría de ellos que se sentían satisfechos con el sistema.

Muy Satisfecho	Satisfecho	Indiferente	Decepcionado	Muy Decepcionado
5	85	12	3	1

Tabla 6.7: Evaluación del sistema: INTERFAZ. Indica tu grado de satisfacción global con respecto al sistema

También se les preguntó si querían hacer algún comentario más sobre la interfaz del sistema. Al contrario que ocurrió en el caso anterior de respuesta libre (cuando se preguntaba sobre mejoras del sistema), la mayoría de los estudiantes no ha propuesto ninguna mejora del interfaz. Esto se debe a que realizaron todas las propuestas (incluidas las del interfaz) en la anterior cuestión de respuesta libre.

Contenido del Sistema

Para evaluar el contenido del sistema, se realizaron cinco preguntas. En la primera de ellas se les pregunta a los estudiantes si creen que lo que se les ha enseñado se ajusta al *Modelo E/R* (ver Tabla 6.8). La mayoría de ellos han contestado que se ajusta *Bastante*, aunque esta opinión no se debería tener muy en cuenta en estos estudiantes, ya que se detectó anteriormente que la mayoría confundían este modelo con otros parecidos.

Completamente	Bastante	Medio	No Mucho	Nada
19	67	16	2	0

Tabla 6.8: Evaluación del sistema: INTERFAZ. ¿Crees que se ajusta al modelo Entidad Interrelación?

A continuación, se les preguntó por el número de páginas web que habían visitado para completar la interacción: si pensaban que eran muchas o pocas (ver Tabla 6.9). En esta cuestión vamos a diferenciar las respuestas de los estudiantes que interaccionaron con RLATES, cuando se adaptaba a sus necesidades, de las respuestas de los estudiantes que interaccionaron con el sistema sin adaptación. Cuando interactuaron con RLATES, la mayoría de los estudiantes respondieron que habían visitado *Pocas* o *Muy Pocas* páginas web, ya que el sistema sólo les enseñaba las que creía oportunas para que aprendieran el temario y fueran capaces de aprobar los tests. Sin embargo, los estudiantes que interaccionaron con la versión no inteligente del sistema respondieron en mayoría que habían visitado *Alguna* o *Muchas* páginas web, ya que el sistema no les guiaba sobre qué páginas web aprender a continuación y visitaban más de las que realmente necesitaban. En la tabla 6.9 se ha representado el número exacto de estudiantes que respondieron cada valoración distinguiendo los estudiantes que interactuaron con la versión adaptativa del sistema (RLATES) de los estudiantes que interactuaron con la versión que no proporcionaba guía directa en la navegación del sistema. Para distinguir ambos valores, se ha introducido un símbolo /"entre ellos. Por ejemplo, 25 alumnos que interactuaron con RLATES respondieron que visitaron *Muy Pocas* páginas web, mientras que ningún alumno que interactuó con IGNATES dio esa respuesta.

Muy Pocas	Pocas	Alguna	Muchas	Demasiadas
25/0	10/17	16/19	1/13	0/3

Tabla 6.9: Evaluación del sistema: CONTENIDO. ¿Piensa que ha visitado más o menos páginas de las necesarias para aprender los contenidos?

Luego se les preguntó si habían visitado páginas que no habían entendido (ver Tabla 6.10). La mayoría de los estudiantes que interactuaron con el sistema en sus dos versiones contestaron que habían visitado *Muy Pocas* páginas web que no habían entendido. En nuestro sistema se supone que probablemente las páginas web que no entendieron los estudiantes son aquellas que explicaban el contenido únicamente con texto, y no con imágenes, ya que en la siguiente pregunta, los estudiantes respondieron que preferían que las páginas web contuvieran imágenes representativas de la información que contenían.

Ninguna	Muy Pocas	Alguna	Muchas	Demasiadas
40	31	26	6	1

Tabla 6.10: Evaluación del sistema: INTERFAZ. ¿Ha visitado páginas que no ha entendido?

También se les preguntó si entendían mejor los contenidos del sistemas cuando éstos iban acompañados de imágenes (ver Tabla 6.11). La respuesta de la mayoría de los alumnos que interaccionaron con las dos versiones del sistema fue que, efectivamente, siempre que se les presentaba los contenidos acompañados de imágenes, entendían mejor la información de la página web.

Nunca	A veces	Regular	Mucho	Siempre
0	4	6	37	57

Tabla 6.11: Evaluación del sistema: CONTENIDO. ¿Ha entendido mejor los contenidos cuando iban acompañados de imágenes?

Por último, también se les preguntó qué tipo de tarea (o elemento) les había resultado más conveniente o útil para comprender los contenidos del sistema (ver Tabla 6.12). Todos ellos coincidían respondiendo que aprendían mejor cuando se les presentaban *Definiciones* o *Ejemplos* y peor cuando se les presentaban *Introducciones*.

	Más Conveniente	Neutro	Menos Conveniente
Introducción	7	24	73
Definición	55	45	4
Ejemplo	65	35	4

Tabla 6.12: Evaluación del sistema: CONTENIDO. ¿Qué tipo de elemento le ha resultado más conveniente para comprender los contenidos?

Navegación en el Sistema

Para finalizar, se les preguntó si les había gustado la forma de navegar por el sistema. En este apartado sí que se realizará una distinción entre los estudiantes que interaccionaron con el sistema RLATES, cuando éste guiaba a los alumnos por el contenido del sistema, y cuando el sistema no ofrecía ningún tipo de adaptación, ya que el cuestionario se ajustó al tipo de navegación que habían realizado.

INTERACCIONES CON RLATES

Recordar en este apartado que los estudiantes en esta versión del sistema navegaban únicamente mediante el botón “*siguiente*” de los botones de navegación. Al pulsar este botón, el sistema les proporcionaba varias opciones de páginas web para continuar el aprendizaje. El estudiante elegía una de estas opciones (basándose en el porcentaje de sugerencia que le proponía el sistema para cada opción) y, tras estudiar el contenido de la página, contestaba a un test para comprobar si había adquirido todo el conocimiento necesario para continuar por otro tema, o necesitaba seguir estudiando con mayor profundidad (más información) de ese tema.

La primera pregunta del cuestionario informa sobre el grado de satisfacción que sentía cada estudiante en la forma de navegar, preguntando si le gustaba que el sistema le guiara a través del botón “*siguiente*”, o prefería que el sistema le proporcionara más autonomía navegando libremente en el árbol de conocimiento. La mayoría de los estudiantes indicaron que preferían *Más* o *Mucha Más* autonomía, aunque aproximadamente un tercio de los estudiantes se encontraban a gusto con la navegación que habían realizado, respondiendo que preferían *Igual* autonomía.

A continuación, se les preguntó a los estudiantes si preferían que, al pulsar el botón “*siguiente*”, el sistema les proporcionara un número mayor de opciones, o preferiría que le mostrara menos o ninguna opción, presentando el contenido directamente (ver Tabla 6.14). 17 estudiantes contestaron que preferían que el sistema le proporcionara alguna opción más y 8 estudiantes preferían *Muchas Más* opciones, aunque hubo 5 estudiantes que preferían que el sistema les llevara a la siguiente página web sin presentar ninguna opción.

Mucha Más (autonomía)	Más	Igual	Menos	Mucha Menos (autonomía)
10	21	16	3	2

Tabla 6.13: Evaluación de RLATES: BOTÓN SIGUIENTE. ¿Le gusta que el sistema le guíe a través del botón de “siguiente” o preferiría que le diera más autonomía (navegar libremente en el árbol de conocimientos)?

Muchas Más (opciones)	Más	Igual	Menos	Ninguna (opción)
8	16	19	4	5

Tabla 6.14: Evaluación de RLATES: BOTÓN SIGUIENTE. Cuando pulsa el botón de “siguiente”, ¿le gusta que el sistema le de varias opciones, o preferiría que le mostrara siempre un contenido directamente?

También se les preguntó si, cuando el sistema les ofrecía las opciones de páginas web a visitar a continuación, siempre elegían la opción que les aparecía en primer lugar (ver Tabla 6.15). La mayoría de ellos elegían esa opción *Casi Siempre*, siguiendo una política de exploración/explotación bastante avariciosa, aunque algún estudiante exploraba a veces por otras opciones.

Siempre	Casi Siempre	A veces	Casi Nunca	Nunca
15	19	14	4	0

Tabla 6.15: Evaluación de RLATES: BOTÓN SIGUIENTE. Al pulsar el botón de “siguiente”, ¿Ha elegido la opción que el sistema le sugería en primer lugar?

Otra cuestión que nos preocupaba es si los estudiantes llegaban a aburrirse en la interacción con el sistema por realizar un test cada vez que aprendían algo nuevo. Por ello se les preguntó en el cuestionario si les había resultado incómodo realizar los tests en el sistema (ver Tabla 6.16). La mayoría de ellos respondieron que no les había resultado incómodo, aunque dos de ellos respondió que le había resultado *Muy* incómodo.

En cuanto al uso de los colores en el árbol de conocimiento que indicaban si el tema había sido consultado por el estudiante, había sido aprobado o suspenso el último test realizado por el estudiante (ver Tabla 6.17), la mayoría de los estudiantes respondieron que les había resultado *Útil* el uso de los colores en el árbol, aunque hubo estudiantes que no lo encontraron nada útil, ya que navegaban úni-

Demasiado (incómodo)	Mucho	Normal	Poco	Nada (incómodo)
2	7	15	17	11

Tabla 6.16: Evaluación de RLATES: TESTS. ¿Le ha resultado incómodo realizar todos los tests (exámenes) del sistema?

camente a través del botón de “siguiente”, sin fijarse prácticamente en el árbol de conocimiento del sistema.

Muy Útil	Útil	Indiferente	Poco Útil	Nada Útil
11	20	14	5	2

Tabla 6.17: Evaluación de RLATES: ÁRBOL DE CONOCIMIENTO. ¿Le ha resultado útil el uso de colores en el árbol de conocimiento?

Por último, se les preguntó si habían echado de menos alguna opción o acción de navegación. A esta pregunta se podía responder de forma libre. Algunas propuestas interesantes son, por ejemplo, que para alguna duda puntual sería interesante la posibilidad de poder encontrar definiciones o ejemplos fácilmente (este problema se podría haber solventado dejando a los estudiantes trabajar con el árbol de conocimiento del sistema); otros estudiantes propusieron realizar otro test cuando han suspendido uno sin necesidad de volver a visitar ninguna página web.

INTERACCIONES CON EL SISTEMA SIN ADAPTACIÓN

En este apartado se recogen los resultados obtenidos en la encuesta sobre la navegación en el sistema cuando éste no se adaptaba a las necesidades del estudiante. Se recuerda que la navegación que realizaron los estudiantes fue únicamente a través del árbol de conocimiento que se encontraba en el marco izquierdo de la interfaz (ver apartado 3.4). Cuando los estudiantes pulsaban los botones de navegación (botón “anterior” y “siguiente”), el sistema les llevaba al anterior o siguiente nodo en el árbol de conocimiento (respectivamente), sin analizar en cada momento qué tema era el más indicado para mostrarle a continuación.

En primer lugar, se les preguntó sobre su grado de satisfacción al acceder al contenido del sistema mediante el árbol de conocimiento (ver Tabla 6.18). La mayoría de ellos se consideraba satisfecho en la interacción con el árbol de conocimiento.

A continuación, se les preguntó si consideraban que el sistema les debería guiar más por el conocimiento, ofreciéndoles sugerencias o pistas sobre qué pági-

Muy Satisfecho	Satisfecho	Indiferente	Decepcionado	Muy Decepcionado
3	41	7	1	0

Tabla 6.18: Evaluación del sistema: ÁRBOL DE CONOCIMIENTO. Indica tu grado de satisfacción sobre el acceso al conocimiento mediante el árbol de conocimiento:

na web es mejor que visite a continuación y en qué formato presentarla (ver Tabla 6.19). La mayoría de ellos sugirieron que el sistema debería proporcionarles menos autonomía. .

Mucha Más (autonomía)	Más	Igual	Menos	Mucha Menos (autonomía)
0	8	14	23	7

Tabla 6.19: Evaluación del sistema: ÁRBOL DE CONOCIMIENTO. ¿Le gusta que el sistema proporcione autonomía (navegar libremente en el árbol de conocimientos) o preferiría que le guiara más por dicho árbol ofreciéndole sugerencias o pistas sobre qué es mejor que aprenda a continuación y en qué formato presentarlo?

A continuación se les preguntó si les había resultado útil el uso de colores en el árbol de conocimiento (ver Tabla 6.20). La mayoría respondieron que les había resultado *Útil* o *Muy Útil* el uso de colores en el árbol, aunque uno de ellos respondió que no le había resultado nada útil.

Muy Útil	Útil	Indiferente	Poco Útil	Nada Útil
15	26	8	2	1

Tabla 6.20: Evaluación del sistema: ÁRBOL DE CONOCIMIENTO. ¿Le ha resultado útil el uso de colores en el árbol de conocimiento?

También se les preguntó si habían echado de menos alguna opción o acción complementaria en la navegación, y respondieron con propuestas interesantes sobre, por ejemplo, tener un desplegable de problemas y/o ejercicios ordenados por dificultad, pudiendo acceder al que se quiera en cualquier momento; o añadir un botón “*Más*” en las definiciones e introducciones que, al pulsar, añada más información al elemento.

Por último se les preguntó si habían tenido problemas al realizar alguna acción, pero todos ellos contestaron que no habían tenido ningún problema al navegar por el sistema.

Conclusiones

En este apartado se resumen las conclusiones obtenidas gracias al cuestionario rellenado por los estudiantes tras su interacción con el sistema en sus dos versiones, tanto adaptativo como no adaptativo.

- Todos los estudiantes partían con el mismo o parecido conocimiento inicial del *Modelo E/R* y de sistemas de enseñanza, por lo que se han agrupado en un único grupo de estudiantes a la hora de interactuar con el sistema en su versión adaptativa (RLATES). Aún así, tras realizar los experimentos, se han detectado algunos estudiantes con características de aprendizaje bastante diferentes a las de sus compañeros.
- La mayoría de los estudiantes creen que el sistema es útil y completo, y la complejidad del sistema es adecuada para ese tipo de estudiantes.
- Los estudiantes poseen una opinión positiva sobre la interfaz de usuario, encontrándose satisfechos globalmente con el sistema el sistema.
- La mayoría de los alumnos opinan que el contenido del sistema se adapta bastante al *Modelo E/R*.
- Los estudiantes que han interactuado con la versión inteligente del sistema (RLATES) creen que han visitado muy pocas páginas web, mientras que los estudiantes que interactuaron con el sistema sin adaptación opinan que han visitado muchas páginas web. Quizás el equilibrio se encontraría cuando los estudiantes interactuaran con la versión completa de RLATES (ver capítulo 3.4).
- La mayoría de los estudiantes contestaron que prácticamente habían entendido toda la información de las páginas web. Las páginas que mejor habían entendido eran aquellas que presentaban imágenes acompañando al texto explicativo.
- Todos ellos preferían que se les mostraran elementos o tareas de tipo *Definición* o *Ejemplo*, indicando que los elementos de tipo *Introducción* no presentaban mucha información sobre el tema al que correspondían.
- Los estudiantes que interactuaron con la versión adaptativa del sistema (RLATES) preferían que el sistema les proporcionara más autonomía, mientras que los estudiantes que interactuaron con el sistema no inteligente, prefieren que el sistema les guíe sobre qué página visitar a continuación. Igualmente, este problema se solucionará cuando los estudiantes puedan interactuar de forma completa con el sistema.

- Los estudiantes que trabajaron con RLATES prefieren que al pulsar el botón de “*siguiente*”, el sistema les muestre las mismas opciones que cuando ellos interaccionaron con el sistema, o quizás alguna más. Y la mayoría de ellos elegían siempre la opción que se les presentaba en primer lugar.
- A los estudiantes no les ha resultado particularmente tedioso realizar los tests en el sistema, no creen que sean demasiados.
- Los estudiantes que tuvieron acceso al árbol de conocimiento se sentían satisfechos con su utilidad y les resultaba en general útil el uso de colores en el árbol.
- Se han realizado algunas propuestas sobre mejoras al sistema, tanto de su interfaz de usuario, como requisitos de navegación y contenidos. Todas estas propuestas se tendrán en cuenta para mejorar el sistema.

Por último, se puede destacar que los problemas encontrados en la navegación con ambas versiones del sistema se pueden solucionar, en principio, fácilmente cuando los estudiantes interactúen con el sistema *RLATES mixto*, sin vetar ninguna opción de navegación.

Capítulo 7

Conclusiones y Líneas Futuras de Investigación

Este capítulo expone las principales conclusiones del trabajo realizado, indicando si se han alcanzado los objetivos planteados al comienzo de la tesis doctoral. Así mismo, se presentan nuevos planteamientos de investigación derivados directamente del trabajo realizado en esta tesis.

7.1. Conclusiones

En capítulos previos se ha descrito la propuesta realizada y la validación de la misma. A continuación se exponen las principales conclusiones que se pueden extraer de este trabajo, indicando los resultados obtenidos y resumiendo las principales aportaciones de la presente tesis doctoral. Finalmente, se describirá la difusión que se ha dado al trabajo realizado, indicando el contenido de cada publicación.

7.1.1. Resultados Obtenidos

Los resultados obtenidos se pueden dividir en tres grandes bloques. En primer lugar se realizó la propuesta, definiendo el problema de *Secuenciar y Presentar* el contenido del curso en los sistemas formación a distancia como un problema de *Aprendizaje por Refuerzo*. Posteriormente se validó la propuesta realizada, comprobando su utilidad tanto de forma teórica (simulando el comportamiento de los estudiantes) como práctica (valorando la utilidad del sistema educativo en las aulas, con estudiantes reales).

Descripción de la Propuesta

En esta tesis doctoral se ha definido el problema de los *Sistemas de Educación Adaptativos e Inteligentes en Web de Secuenciar el contenido y Presentar el contenido* en el formato adecuado como un problema de *Aprendizaje por Refuerzo*. Al aplicar el modelo de *Aprendizaje por Refuerzo* en el módulo pedagógico del sistema de educación se evita el costoso proceso de analizar y predefinir cada estrategia pedagógica que se tendría que tomar en cada instante de la interacción con cada estudiante, definiendo las tácticas pedagógicas del sistema de forma implícita, derivadas de los valores de la función de *valor-acción* implementada en el sistema.

El sistema de educación es capaz de aprender a enseñar al mismo tiempo que el estudiante aprende la materia, basándose únicamente en la experiencia adquirida con otros estudiantes de características pedagógicas similares. Este proceso de aprendizaje se realiza mediante prueba y error, al igual que los tutores humanos aprenden a enseñar a los alumnos en las aulas.

El sistema adapta su enseñanza a cada estudiante individualmente, mostrándoles el contenido de forma cercana a la óptima. De este modo, el estudiante mejora su aprendizaje obteniendo mejores resultados en un tiempo menor de enseñanza que otros sistemas educativos no adaptativos.

Se ha representado el sistema mediante el modelo de *Aprendizaje por Refuerzo*, comprobando que cada componente del modelo tiene una aplicación directa en el módulo pedagógico del sistema de educación. Se ha definido el estado del modelo como el estado de conocimiento del estudiante; las acciones a ejecutar como páginas web a mostrar al estudiante con el contenido del curso; la función de percepción del estado actual como los exámenes y/o tests de evaluación de conocimiento del estudiante; la señal de refuerzo derivada directamente del estado de conocimiento del alumno sobre la materia; y la función de *valor-acción* como parte de la táctica pedagógica del sistema representada de forma implícita. La definición del sistema como un problema de *Aprendizaje por Refuerzo* se ha realizado de forma general e independiente del dominio de enseñanza del sistema educativo.

Se ha definido el sistema de forma que es capaz de *Secuenciar* el contenido del curso no sólo a alto nivel (como previos trabajos hacían [Beck, 2001]), sino también a bajo nivel en el árbol de conocimiento del modelo del dominio, siendo capaz de decidir qué tarea mostrar a continuación (*definición, introducción, ejercicio, etc.*) y en qué formato de presentación (*video, imagen, texto, etc.*).

Validación con Estudiantes Simulados

Posteriormente se ha validado esta propuesta desde el punto de vista teórico, demostrando que el sistema es capaz de aprender tácticas pedagógicas buenas para cada estudiante con el que interactúa en cada momento. Para ello se ha modelado de forma predefinida el comportamiento de los estudiantes, asociando a cada grupo de estudiantes un *Proceso de Decisión de Markov*. Mediante el MDP de cada tipo de estudiantes se ha modelado la conducta del alumno al interactuar con el sistema. El MDP del grupo de estudiantes se creó en base a información del dominio (relaciones de *pre-requisito* entre los temas) y en base a una cierta probabilidad de aprender un determinado concepto según el tipo de elemento presentado *definición, introducción, ejercicio, etc.* y el formato de presentación del concepto *video, imagen, texto, etc.* Con el objetivo de evaluar de forma exhaustiva la propuesta, se han definido distintos grados de heterogeneidad en el comportamiento de los estudiantes, y se han agrupado según sus características de aprendizaje a la hora de interactuar con el sistema. A la hora de validar el sistema con estudiantes simulados se han planteado situaciones reales, donde los estudiantes simulados interactuaban uno a uno con el sistema, asimilando en algunos casos la información proporcionada en las páginas web que se le presentaban y en otros no, dependiendo de su MDP.

Se ha demostrado que el sistema de educación es capaz de aprender a enseñar mediante prueba y error, convergiendo a tácticas pedagógicas buenas (ejecutando un número de acciones no muy elevado) con estudiantes de distintas características de aprendizaje. También se ha comprobado que el sistema es capaz de converger, utilizando dominios de varias dimensiones (número de temas y conceptos a enseñar dentro de cada tema). Además, el tamaño de la *fase de entrenamiento* del sistema no es muy grande, ya que el sistema no necesita muchos alumnos para aprender de sus interacciones con el objetivo de alcanzar una táctica pedagógica cercana a la óptima.

Se ha estudiado el efecto de los parámetros del sistema en el proceso de aprendizaje, evaluando las necesidades pedagógicas del sistema de educación. Se ha demostrado que cuanto más avariciosa es la política de exploración/explotación, RLATES converge a una táctica pedagógica buena más rápido (utilizando menor número de estudiantes) y mejor (ejecutando menor número de acciones). Además, el sistema se comporta mejor con valores altos de α , el sistema aprende más rápido las tácticas de enseñanza (interaccionando un menor número de estudiantes). También se realizaron experimentos para comprobar cómo afecta la heterogeneidad en el comportamiento del estudiante, concluyendo que cuanto más estocástico es el grupo de estudiantes, peor aprende el sistema.

Se ha realizado también una evaluación en profundidad del tipo de estrategias de exploración/explotación que se pueden aplicar a los problemas derivados de

los sistemas de formación a distancia, valorando las ventajas y desventajas de la estrategia de exploración/explotación *e-greedy* y la estrategia de *Boltzmann*. Con ambas políticas el sistema de educación es capaz de aprender a enseñar mediante un proceso de prueba y error, convergiendo a tácticas pedagógicas buenas (mostrando un número no muy grande de páginas web a los estudiantes) interactuando en su *fase de entrenamiento* con un número no muy grande de estudiantes.

Para finalizar con la evaluación del sistema con estudiantes simulados, se ha estudiado la posibilidad de disminuir la *fase de entrenamiento* del sistema, donde éste aún no ha aprendido la mejor forma de mostrar los contenidos a los estudiantes. Para ello se ha demostrado que realizar una inicialización del sistema con información sobre tácticas pedagógicas válidas para los estudiantes con ciertas características de aprendizaje es positivo, incluso cuando los estudiantes que utilizaban posteriormente RLATES no posean las mismas características de aprendizaje.

Validación con Estudiantes Reales

Una vez validado el sistema desde el punto de vista teórico (con estudiantes simulados), se ha evaluado la propuesta desde el punto de vista práctico, comprobando que el sistema puede implantarse en sustitución de las aulas físicas. Se han realizado experimentos con 104 estudiantes reales, demostrando que el sistema es capaz de converger a buenas tácticas pedagógicas, modificando su secuencia y presentación de contenidos (desde el punto de vista de la elección del *formato* de presentación) según el estudiante con el que trabaja en un determinado momento, interactuando con un número no muy grande de estudiantes. Se ha demostrado que las tácticas pedagógicas a las que convergen son adecuadas en cuando el número de páginas web mostradas al estudiante. Además, se ha demostrado que la *fase de entrenamiento* del sistema no es muy grande, necesitando pocos estudiantes para aprender a enseñar de la forma adecuada.

Se realizaron experimentos comparando la eficiencia y eficacia del sistema con su versión adaptativa (aplicando *Aprendizaje por Refuerzo* al módulo pedagógico del sistema) y su versión no adaptativa. Se ha demostrado que el tiempo total de interacción con el sistema es menor cuando los estudiantes interactúan con la versión adaptativa (RLATES). Cuando el sistema se adapta a los estudiantes éstos aprenden más rápido el contenido del sistema. Además, se ha comprobado que el grado de conocimiento global que adquieren los estudiantes sobre la materia es muy parecido cuando interactúan con el sistema con y sin adaptación.

De esta forma, se ha comprobado que el sistema se comporta de forma *inteligente*, sin necesidad de realizar el proceso de analizar y predefinir cada estrategia pedagógica necesaria para adaptar la enseñanza a cada estudiante en cada instante de la interacción. El sistema es capaz de aprender por sí mismo tácticas pedagógi-

cas para cada estudiante en cuanto a la navegación a través de los contenidos del tutor y en cuanto al formato de presentación de dichos contenidos basándose únicamente en la experiencia derivada de la interacción con estudiantes de las mismas características de aprendizaje.

Además, se ha demostrado que la estrategia de exploración/explotación de *Boltzmann* ofrece ventajas frente a la estrategia *e-greedy*, ya que ésta permite definir la probabilidad de la bondad de una acción de forma normalizada frente al resto de las acciones, permitiendo al sistema ofrecer al estudiante esta información de forma que sea él mismo el que elija el siguiente conocimiento a aprender. De esta forma se le proporciona al estudiante sensación de control de la interacción con el sistema, característica deseable en los sistemas de interacción hombre-máquina.

Para finalizar con la experimentación con estudiantes reales, se ha realizado una comprobación de la usabilidad del sistema, llegando a la conclusión de que es buena. Esta conclusión se ha derivado de los cuestionarios rellenados por los estudiantes tras su interacción con el sistema en sus versiones, RLATES y IGNATES. La mayoría de los estudiantes piensan que el sistema es útil, completo y su complejidad es adecuada. Además, todos ellos poseen una opinión positiva sobre el interfaz de usuario, encontrándose satisfechos globalmente con el mismo, tanto en su versión adaptativa, como en su versión no adaptativa.

Los estudiantes que han interactuado con RLATES1 y RLATES2 creen que han visitado muy pocas páginas web, mientras que los estudiantes que interactuaron con el sistema IGNATES opinan que han visitado muchas páginas web. Quizás el equilibrio se encontraría cuando los estudiantes interactuaran con la versión completa de RLATES, planteada como trabajo futuro de esta tesis doctoral. En cuanto al contenido del sistema, la mayoría de los estudiantes contestaron que prácticamente habían entendido toda la información de las páginas web y que las páginas que mejor habían entendido eran aquellas que presentaban imágenes acompañando al texto explicativo, prefiriendo que se le mostraran elementos de tipo *Definición* o *Ejemplo*, indicando que los elementos de tipo *Introducción* no presentaban mucha información sobre el tema al que correspondían.

En cuanto a la autonomía del sistema, los estudiantes que interactuaron con la versión de RLATES adaptativa preferían que el sistema les proporcionara más autonomía, mientras que los estudiantes que interactuaron con el sistema no inteligente, prefieren que el sistema les guíe sobre qué página visitar a continuación. De la misma forma, este problema se solucionará seguramente cuando los estudiantes interactúan de forma completa con el sistema.

En cuanto a la estrategia de exploración/explotación de *Boltzmann*, los estudiantes que trabajaron con la versión inteligente de RLATES prefieren que al pulsar el botón de *“siguiente”*, el sistema les muestre las mismas opciones que cuando ellos interactuaron con el sistema, o quizás alguna más. Además, casi todos ellos han elegido siempre la opción que se les presentaba en primer lugar.

Por otro lado, en cuanto a la evaluación que realiza el sistema, a ningún estudiante le ha resultado tedioso realizar los test en el sistema, no creen que sean demasiados. Los estudiantes que tuvieron acceso al árbol de conocimiento se sentían satisfechos con su utilidad y les resultaba en general útil el uso de colores en el árbol y se han realizado propuestas sobre mejoras a RLATES.

Como conclusión, se puede destacar que los problemas encontrados en la navegación con ambas versiones del sistema se pueden solucionar fácilmente cuando los estudiantes interactúan con el sistema *RLATES mixto*, sin vetar ninguna opción de navegación.

7.1.2. Principales Aportaciones

En este capítulo se resumen las principales aportaciones del trabajo realizado en la presente tesis doctoral, centradas en las ventajas que ofrece la aplicación del modelo de *Aprendizaje por Refuerzo* en el módulo pedagógico de un *Sistema de Educación Adaptativo e Inteligente en Web*, planteando diferencias significativas con trabajos previos de *Secuenciación* y Elección de Formatos de Presentación de contenidos.

Aplicación de *Aprendizaje por Refuerzo*

En primer lugar, la definición de los problemas de *Secuenciación* y *Presentación* del contenido (desde el punto de vista de la elección del *formato* de presentación del contenido) del curso en un sistema de formación a distancia como un problema de *Aprendizaje por Refuerzo* permite que el sistema de educación adapte de forma automática sus tácticas pedagógicas según el estudiante que interactúa en cada momento con el sistema.

- **Evitar predefinir tácticas pedagógicas.** Esta adaptación se realiza dinámicamente basándose únicamente en la experiencia adquirida previamente al interactuar con otros estudiantes de características de aprendizaje similares, solventando uno de los mayores problemas de los sistemas de educación: la predefinición de las estrategias pedagógicas a seguir en cada momento de la interacción con los estudiantes. Hasta el momento, la mayoría de los sistemas de educación se construían a partir de sistemas expertos basados en reglas. La codificación de dichas reglas es costosa, debido a la gran cantidad de reglas pedagógicas que existen para incorporar una mayor adaptación al usuario, y debido a la dificultad de incorporar el conocimiento del experto en el sistema.
- **Flexible ante ruido.** Otra ventaja derivada de la aplicación de *Aprendizaje por Refuerzo* en el módulo pedagógico es la flexibilidad que poseerá el

sistema ante pequeñas instancias de ruido. Por ejemplo, al evaluar el conocimiento del estudiante mediante un test, el sistema puede percibir el estado actual de conocimiento del estudiante de forma errónea. El sistema será capaz de recuperarse ante estos datos ruidosos al aplicar esta técnica de inteligencia artificial, ya que se trata de una característica inherente al modelo de aprendizaje por refuerzo.

- **Aplicación de RL en el módulo pedagógico.** En previos trabajos [Beck, 2001] aplicó el modelo de *Aprendizaje por Refuerzo* al módulo del usuario, modelando a los estudiantes al mismo tiempo que aprendía estrategias pedagógicas. Esta propuesta presenta principalmente el problema de la dimensión del número de estados del sistema. Al representar como parte del estado todas las características de aprendizaje del estudiante a tener en cuenta (que son muchas y muy variadas), el número de estados del sistema crece exponencialmente. El sistema aprende a enseñar muy despacio, ya que su aprendizaje se basa únicamente en interacciones con otros estudiantes de exactamente las mismas características de aprendizaje, siendo difícil encontrar estudiantes de las mismas características y por tanto, realizando una escasa generalización. Además, cuanto mayor sea el tamaño del estado del sistema, mayor será el espacio de estados, y su convergencia a tácticas pedagógicas buenas es más lenta, ya que el espacio de exploración crece exponencialmente con el número de características de los estados. Para solventar este problema, en esta tesis se ha propuesto aplicar el modelo de *Aprendizaje por Refuerzo* únicamente al módulo pedagógico del sistema, y sus *estados* se han definido como el conocimiento que posee el estudiante sobre los temas y tareas del dominio (conocimiento a bajo nivel en el árbol del dominio), sin incluir ninguna otra característica de aprendizaje del estudiante. Se ha propuesto dividir el problema planteado por Beck en tres fases. En la primera fase se realiza una *Agrupación* de los estudiantes. Como se comentó en el apartado 3.2.5, se recomienda agrupar a los estudiantes según sus características de aprendizaje, de forma que se mantenga distintas tácticas pedagógicas según el grupo de estudiantes, aunque no es obligatoria. En la fase de *Entrenamiento* del sistema, éste explora nuevas alternativas pedagógicas, hasta conseguir una buena táctica de educación para el estudiante actual. Finalmente, en la fase de *Uso* del sistema se explota el conocimiento previamente adquirido, ya que el sistema será capaz de secuenciar y presentar el conocimiento de una forma cercana a la óptima, adaptándose a las necesidades de los estudiantes. Esta tesis doctoral se centra en la segunda y tercera fase, definiendo el problema de agrupación de estudiantes como un problema adicional al planteado en este trabajo.

- **Aplicación de RL a bajo nivel en el árbol de conocimiento.** Por otro lado, Beck, con el objetivo de reducir su espacio de estados, aplicaba el modelo de *Aprendizaje por Refuerzo* únicamente a los temas del árbol de conocimiento, aprendiendo secuencias de contenidos de temas, pero no de tareas a ejecutar. En esta tesis doctoral se ha propuesto aplicar el modelo a bajo nivel (a nivel de tareas en el árbol de conocimiento del modelo del dominio). De esta forma RLATES es capaz de no sólo secuenciar el conocimiento a nivel de tema, sino también a nivel de tarea (*definiciones, introducciones, ejercicios, etc.*), pudiendo también decidir en qué formato se presenta este contenido cuando en el sistema se almacenan las tareas en distintos formatos.

Validación del Sistema

Una vez definido el sistema como un problema de *Aprendizaje por Refuerzo*, se ha validado el sistema desde el punto de vista teórico, mediante la construcción de modelos de comportamiento simulados de estudiantes, y posteriormente se ha validado el sistema también desde el punto de vista práctico, interactuando estudiantes reales con RLATES y evaluando su usabilidad.

- **Estudio de las variables de aprendizaje.** Se ha estudiado en profundidad el efecto de las variables que afectan al aprendizaje del sistema, estudiando varias estrategias de exploración/explotación en el sistema.
- **Estudio de estrategias de exploración/explotación.** Se han valorado finalmente las ventajas que ofrece la política de exploración/explotación de *Boltzmann* frente a la política *e-greedy* en sistemas de formación a distancia.
- **Inicialización de tácticas pedagógicas** Además, se ha demostrado que inicializar el sistema con tácticas pedagógicas es positivo, ya que RLATES es capaz de aprender más rápido (interactuando con un número menor de estudiantes) y mejor (necesitando mostrar un número menor de páginas web) si el sistema se ha inicializado previamente, aunque esta inicialización no sea la más adecuada para las características pedagógicas de los estudiantes con los que interactúa. En este trabajo se ha realizado la inicialización del sistema interactuando con estudiantes simulados.

Destacar que se ha definido el problema de forma general, independientemente del dominio de aplicación del sistema de educación. Además, al utilizar tecnologías derivadas de Internet, cualquier estudiante con un ordenador conectado a Internet podría conectarse al servidor de RLATES, independientemente de la plataforma en la que se encuentre, sin necesidad de desplazarse a las aulas físicas.

7.1.3. Difusión de Resultados

Durante el desarrollo de la tesis doctoral se han realizado una serie de publicaciones que marcan la evolución del trabajo, reflejando en cada una de ellas los progresos realizados en la investigación en cada una de sus fases. A continuación se exponen las publicaciones realizadas, así como su motivación.

1. **Estudio del estado del arte.** Se ha recopilado información sobre el estado de la cuestión en sistemas informáticos de enseñanza, poniendo especial hincapié en aquellos que se comportan de forma inteligente, adaptándose en cada momento a las necesidades del estudiante con el que interactúan en cada instante. Se ha estudiado con especial interés la aplicación de técnicas de inteligencia artificial utilizadas y las distintas formas de adaptación al estudiante. Una primer estudio de estos sistemas se publicaron como parte del estado del arte en el artículo en revista:

- *EDU-EX: A Tool for Auto-Regulated Intelligent Tutoring Systems Development Based on Models.* P. Domingo, A. García-Crespo, B. Ruiz y A. Iglesias. Artificial Intelligence Review (JCR). Volume 18, Pp. 15-32, 2002.

2. **Aplicación de inteligencia artificial sobre sistemas de enseñanza.**

En este estado de la cuestión encontramos un punto en el que podíamos realizar aportaciones novedosas y que coincidía con nuestros intereses en los sistemas de enseñanza y las técnicas de inteligencia artificial. A partir de aquí, se profundizó en el *modelo de Aprendizaje por Refuerzo (Reinforcement Learning)* y se propuso un modelo teórico de cómo aplicarlo en el entorno de los programas de enseñanza, y en particular, sobre los *Tutores Inteligentes (Intelligent Tutoring Systems)*. Para ello fue necesario definir la arquitectura del sistema, describiendo los detalles de cada uno de sus módulos. Se definió una estructura jerárquica en el modelo del dominio, sin necesidad de incorporar conocimiento del experto sobre relaciones entre temas o conceptos (como *prerrequisitos, está-relacionado, etc.*). En el módulo del estudiante se utilizó la metodología *overlay*. En el módulo pedagógico se aplicó el modelo de *Aprendizaje por Refuerzo* y por último, el módulo del interfaz simplemente mostraba el contenido en la secuencia y formato que le proporcionaba el módulo pedagógico. Este trabajo se publicó posteriormente como artículo en congreso:

- *Applying Reinforcement Learning in Intelligent Tutoring Systems.* A. Iglesias, P. Martínez y F. Fernández. 4th International Conference on New Educational Environments. Pp. 11-14, 2002.

3. **Estudio de la incidencia de la aplicación de *Aprendizaje por Refuerzo* en el módulo pedagógico.** A continuación se planteó la necesidad de completar el sistema de educación mediante la utilización de técnicas ya aplicadas anteriormente en otros sistemas de educación en el resto de los módulos del sistema: el módulo del dominio, el módulo del interfaz y el módulo del estudiante. Al sistema completo se le ha denominado *RLATES: Reinforcement Learning Adaptive and inTelligent Educational System*. En el módulo del estudiante se aplicó la técnica *overlay*; sobre el módulo del dominio se realizaron varias publicaciones que se detallan a continuación; y sobre el módulo del interfaz se analizó la incidencia de la aplicación de *Aprendizaje por Refuerzo* en el módulo pedagógico. Las publicaciones centradas en el módulo del interfaz también se detallan a continuación.
 - a) ***Dominio del Sistema de Enseñanza.*** Como se ha comentado anteriormente, la aplicación del modelo de *Aprendizaje por Refuerzo* se puede aplicar en cualquier sistema de enseñanza independientemente del dominio. Se estudió la posibilidad de su aplicación en un dominio conocido, para facilitar la tarea de construcción del modelo del dominio. Las siguientes publicaciones surgieron a partir de la aplicación del modelo teórico para un dominio concreto, el *Diseño de Bases de Datos*:
 - *Diseño de Bases de Datos: Problemas Resueltos.* A. Miguel, P. Martínez, E. Castro, J.M. Cavero, D. Cuadra, A.M. Iglesias y C. Nieto. Ed. Ra-ma, 2000. Se trata de un libro de ejercicios de la materia de *Diseño de Bases de Datos* que sirvieron posteriormente para la creación de los materiales del curso del sistema RLATES.
 - *Learning to Teach Database Design by Trial and Error* . A. Iglesias, P. Martínez, D. Cuadra, E. Castro y F. Fernández. 4th International Conference on Enterprise Information Systems, Pp. 500-505, 2002. Este artículo presenta ejemplos del modelo del dominio del sistema y de cómo aplicar el modelo de aprendizaje por refuerzo para el dominio en particular.
 - b) ***Adaptación del Interfaz de Usuario.*** Posteriormente se estudió en profundidad el tipo de adaptación a usuario en el interfaz que se podría ofrecer fácilmente derivado directamente de la aplicación del modelo de *Aprendizaje por Refuerzo* en el modelo pedagógico del sistema de enseñanza. Este estudio y la implementación en el sistema RLATES supuso la publicación de dos artículos más:
 - *Navigating through the RLATES Interface: a Web-based Adapti-*

ve and Intelligent Educational System. A. Iglesias, P. Martínez, F. Fernández. Proceedings of the Workshop on Human Computer Interface for Semantic Web and Web Applications (HCI-SWWA2003). Part of the International Federated Conferences (OTM2003). Lecture Notes on Computer Science (Springer Verlag). Pp. 175 - 184. 2003. Este artículo trata de las distintas formas de adaptar la navegación y el contenido de la materia del sistema de enseñanza a cada estudiante en cada instante y su implementación en RLATES.

- *Construyendo un Sistema de Educación Adaptativo e Inteligente en Web para aprendizaje de Diseño de Bases de Datos*. A. Iglesias, A. Iglesias, P. Martínez, E. Castro. VIII Jornadas de Ingeniería del Software y Bases de Datos. Pp. 757-762. 2003. En este artículo se describe la implementación del sistema aplicada al dominio del *Diseño de Bases de Datos*, poniendo especial hincapié en el desarrollo de la interfaz.

4. **Integración dentro del proyecto PANDORA.** La concepción del Sistema de Educación e Inteligente en *Web RLATES* surgió por la necesidad de un sistema de apoyo a la enseñanza a una herramienta CASE, PANDORA (acrónimo de *Plataforma CASE para el aprendizaje y desarrollo de Bases de Datos y su enseñanza vía Internet*. Proyecto de investigación CICYT (TIC99-0215)). El objetivo de Pandora es desarrollar un entorno CASE para desarrollo de BD en todas sus fases (modelado conceptual, diseño lógico e implementación) y para aprendizaje de los conceptos necesarios para realizar esta labor vía Internet. Por lo tanto, es necesario incluir un componente didáctico que facilite el aprendizaje del diseño de BD. Para ello se pensó en la integración de nuestro sistema, RLATES, dentro de la herramienta PANDORA. El siguiente artículo presenta una aproximación teórica de cómo se podría integrar RLATES como un módulo más del proyecto de investigación PANDORA:

- *Integrating Intelligent Methodological and Tutoring assistance in a CASE platform: the PANDORA experience*. E. Castro, D. Cuadra, P. Martínez y A. Iglesias. Informing Science and IT Education Conference. Pp. 11-14, 2002.

5. **Validación del sistema.** Como se describió en el capítulo 4 de este documento, la validación de RLATES se ha realizado en dos fases. La primera fase consta de la evaluación del sistema desde el punto de vista teórico (con estudiantes simulados), donde se tratará de elegir los mejores parámetros

para que el modelo de aprendizaje por refuerzo aplicado al sistema de educación sea más eficiente a la hora de enseñar. La segunda fase trata de evaluar el sistema desde un punto de vista práctico, por lo que se ha realizado experimentación con estudiantes reales. En este caso son los alumnos los que evalúan el sistema, indicando si se sienten cómodos aprendiendo con el sistema y notan mejoría frente al mismo sistema sin adaptación.

La experimentación realizada con estudiantes simulados se ha publicado en los siguientes artículos:

- *An experience applying Reinforcement Learning in a Web-based Adaptive and Intelligent Educational System.* A. Iglesias, P. Martínez, F. Fernández. *Informatics in Education International Journal*, Volumen 2. Páginas 1-18. 2003. En este artículo se demuestra que el sistema de enseñanza es capaz de converger a una táctica de comportamiento casi óptima, donde el número de páginas que se ha de mostrar a cada estudiante es pequeño y que el sistema no necesita interactuar con muchos estudiantes del mismo tipo si se utilizan los parámetros apropiados.
- *Analysing the Advantages of Using Exploration and Exploitation Strategies in an Adaptive and Intelligent Educational System.* A. Iglesias, P. Martínez, R. Aler, F. Fernández. *Second International Conference on Multimedia and Information and Communication Technologies in Education (m-ICTE 2003)*. Vol. 1. Pp. 489-493. 2003. Este artículo muestra la importancia de elegir una buena estrategia de selección de la siguiente página a mostrar al estudiante (comparando explotación y exploración), concluyendo que la estrategia de Boltzmann es mucho más adecuada para el sistema que la estrategia e-greedy utilizada por Joseph Beck en [Beck, 2001].
- *Learning Content Sequencing in an Educational Environment According to Student Needs.* A. Iglesias, P. Martínez, R. Aler, F. Fernández. *15th International Conference on Algorithmic Learning Theory. Lecture Notes in Computer Science (Springer Verlag)*. Pp. 454 - 463. Padova (Italy). 2004. El artículo muestra cómo el sistema converge mejor (mostrándole al estudiante un número menos de páginas web) y más rápido (interactuando con un número menor de estudiantes para llegar a la convergencia) cuando su táctica pedagógica ha sido previamente inicializada con conocimiento de las interacciones con otros estudiantes, aunque éstos no posean las mismas características de aprendizaje.

En cuando a la experimentación con estudiantes reales, actualmente se está escribiendo un artículo que resume la validación realizada con dichos estudiantes.

7.2. Trabajos Futuros

La tesis doctoral que se presenta en este documento evalúa un método de adaptación de un programa de enseñanza en web según las necesidades del estudiante. Para ello se ha aplicado el modelo de *Aprendizaje por Refuerzo* al módulo pedagógico del sistema de educación, siendo ésta una de las principales aportaciones del trabajo. Esta tesis da lugar a nuevas líneas de investigación. En este capítulo se esbozan aquellas que se relacionan directamente con nuestro trabajo, algunas de ellas muy relacionadas entre sí.

Para implementar el sistema RLATES se han utilizado diversos métodos y técnicas en cada uno de sus módulos que han sido utilizados para realizar la evaluación del trabajo de investigación. En la literatura se pueden encontrar mejoras en los módulos del sistema que no son objeto del trabajo realizado en la presente tesis doctoral. Esta sección no se ocupa de identificar los posibles trabajos futuros en estos módulos, sino que se centra en futuras líneas de investigación sobre el trabajo llevado a cabo.

7.2.1. Modelado y Clasificación de Estudiantes

Es importante para cualquier programa que se adapte al usuario, realizar un buen modelo de estudiante con todas las características del mismo que se han de tener en cuenta en la adaptación del sistema.

En la implementación de RLATES no se ha realizado un modelo exhaustivo del estudiante, sino que se han tenido en cuenta solamente el conocimiento que tiene el estudiante actual sobre la materia (temas) del curso, aplicando la técnica *overlay* para modelar este conocimiento.

Al tener en cuenta únicamente el conocimiento actual del estudiante, se ha demostrado que el sistema es capaz de adaptarse al conjunto de todos los estudiantes que interactúan con RLATES, adaptando su comportamiento cada vez que un estudiante interactuaba con el sistema. Hay que tener en cuenta que RLATES se adaptará al conjunto de estudiantes que se le presenta. Si se realizara un trabajo previo de modelado y clasificación de estudiantes dependiendo de las características de aprendizaje de cada uno de ellos (teniendo en cuenta todas las características necesarias para la adaptación deseada), el sistema sería capaz de adaptarse a cada clase de estudiantes.

Esta clasificación permitiría que RLATES se adaptara mejor a las características de aprendizaje de cada grupo de estudiantes, sin necesidad de que RLATES supiera cuáles son las características de aprendizaje de la clase de estudiantes. Esta característica de RLATES permite que la clasificación pueda realizarse con cualquier técnica de clasificación, incluyendo métodos subsimbólicos, que no devuelven las causas de la clasificación (por ejemplo, redes neuronales).

Otra línea de investigación interesante sería mantener la clasificación de los estudiantes durante la interacción con el sistema, por si el estudiante pudiera cambiar de clase (características de aprendizaje), o que, simplemente, la clasificación no fuera correcta a priori. El mantenimiento del modelo de estudiante y clasificación se podría llevar a cabo recogiendo datos de la interacción con RLATES.

7.2.2. Mejorar la Evaluación del Conocimiento del Estudiante

Otro tema a tener en cuenta es mejorar la evaluación del estudiante en RLATES. Los exámenes siempre han sido una forma sencilla de evaluar a todos los alumnos de forma objetiva, pero, en general, a los estudiantes no les gusta sentirse observados para ser evaluados. Por ello, sería un gran paso para que el estudiante se sintiera a gusto con la herramienta el eliminar el concepto de examen que, aunque en la experimentación con estudiantes reales respondieron que no les importaba, habitualmente los alumnos no se encuentran cómodos con evaluaciones tan directas. Una solución a este problema sería evaluar a los estudiantes a lo largo de toda la interacción, proponiéndoles pequeños ejercicios capaces de evaluar su conocimiento en cada instante.

Hasta el momento, RLATES solamente realiza exámenes de tipo test. Sería conveniente el poder incluir otro tipo de exámenes de respuestas libres, o, incluso utilizar herramientas que les permitiera contestar y solucionar ejercicios y problemas del curso que se enseña en el sistema.

Por otro lado, hasta el momento RLATES evalúa el conocimiento del estudiante únicamente con dos valores: 0 si el estudiante no conoce el tema y 1 si lo conoce. Hasta el momento sólo se han tenido en cuenta estos valores con el objetivo de simplificar los experimentos al evaluar si el sistema es capaz de aprender nuevas tácticas pedagógicas. Pero para mejorar la adaptación del sistema al estudiante es conveniente que éste aumente el conocimiento que posee sobre el estudiante, añadiendo más información sobre el grado de conocimiento de los estudiantes sobre los temas del dominio (por ejemplo, añadiendo porcentajes de conocimiento o valores continuos para denotar este conocimiento).

7.2.3. Reducción del Número de Estados y Acciones

Esta mejora tiene que ver con la implementación del modelo de aprendizaje por refuerzo. En RLATES se ha definido el estado del sistema como el conocimiento actual que tiene el estudiante sobre cada tema del curso. Se ha de tener en cuenta, por tanto, que el número de estados crece exponencialmente con respecto al número de temas que tiene el curso y el grado de conocimiento que se desea almacenar sobre cada tema (muy relacionado con el apartado anterior). Cuando

el número de estados es muy grande, la necesidad de cómputo del sistema crece exponencialmente, por lo que se recomienda reducir el número de estados de conocimiento en los que un estudiante se puede encontrar. Existen varias técnicas de reducción del número de estados utilizadas en problemas similares a éste previamente. Una posible solución es utilizar métodos de generalización de estados [Santamaría *et al.*, 1998], como se planteó en el apartado 3.2.2. Otra manera de reducir el número de estados es plantearse algoritmos jerárquicos que traten la jerarquía de temas del modelo del dominio, como se comentó en el apartado 2.4.6.

Por otro lado, al definir las acciones que puede ejecutar el sistema a bajo nivel en el árbol de conocimiento, el espacio de acciones se amplía. Esto podría suponer un problema si el número de acciones es muy elevado, ya que el sistema tendría que realizar más cálculos para poder aprender a enseñar, ralentizando su proceso de aprendizaje al tener que explorar un número mayor de acciones. Este problema se podría solventar mediante generalización del número de acciones [Santamaría *et al.*, 1998].

7.2.4. Inicialización de Tácticas Pedagógicas con Conocimiento de un Experto

Como se ha comentado en apartados anteriores, RLATES es capaz de aprender un buen comportamiento (mostrando el contenido del tutor adaptándose a las necesidades del estudiante) sin tener que definir previamente ninguna estrategia pedagógica.

También se demostró que el sistema converge más rápido y mejor cuando se inicializan el sistema con algún tipo de conocimiento sobre las tácticas pedagógicas, aunque éstas no sean las más adecuadas.

En RLATES se inicializaron las tácticas a partir de interacciones del sistema con estudiantes simulados 5.4, tal y como hizo Beck en [Beck, 2001]. Gracias a este tipo de inicialización se soluciona uno de los procesos más costosos de cualquier sistema de enseñanza: el introducir todo el conocimiento del experto en el programa.

Sin embargo, una posible línea de investigación futura podría ser incluir conocimiento del experto en el modelo del dominio, como por ejemplo, interrelaciones entre temas y elementos de temas *es-prerrequisito*, *están-relacionados*, dificultad de los elementos y de los temas, etc. para posteriormente poder aplicar alguna técnica que aprenda un comportamiento no aleatorio del sistema de enseñanza (por ejemplo, planificación [Vassileva, 1995]).

7.2.5. Desarrollo y Evaluación del Sistema *RLATES mixto*

Como se ha comentado anteriormente, en la experimentación realizada con estudiantes reales se han comparado los resultados obtenidos cuando los estudiantes interactuaban con una versión adaptativa del sistema (*RLATES*) y cuando el sistema no se adaptaba a sus necesidades mediante el modelo de *Aprendizaje por Refuerzo* (*IGNATES*).

Se demostró que la aplicación del modelo de *Aprendizaje por Refuerzo* en el módulo pedagógico del sistema de educación era útil para los estudiantes. Se plantea, por tanto, como futuras líneas de trabajo el desarrollar y evaluar el sistema *RLATES mixto*, donde se incluirán todos los componentes detallados en la sección 3.3.4 y se evaluará la utilidad del sistema.

En la Universidad Carlos III de Madrid se planea la utilización de este sistema de educación a distancia como apoyo a los alumnos matriculados en las asignaturas que tratan el tema de *diseño de bases de datos*.

7.2.6. Aplicación del Modelo de *Aprendizaje por Refuerzo* a la Elección de las Estrategias Pedagógicas

Otra línea de trabajo futuro que se plantea en esta tesis doctoral es que, tras comprobar la utilidad de la aplicación del módulo de *Aprendizaje por Refuerzo* en el módulo pedagógico del sistema a nivel de tarea y tema en el árbol de conocimiento, aplicar este mismo modelo para secuenciar otro tipo de elementos de las políticas pedagógicas.

Por ejemplo, se podrían predefinir ciertas estrategias pedagógicas a adoptar cuando un determinado estudiante se encuentra en un determinado estado. Posteriormente se podría aplicar el modelo de *Aprendizaje por Refuerzo* para identificar qué, cómo y cuándo aplicar cada estrategia pedagógica predefinida de antemano.

Una mejora a esta opción, que implicaría la predefinición de las estrategias pedagógicas sería, a partir del sistema *RLATES* (que secuenciar el conocimiento a nivel de tarea y tema), aplicar mecanismos de generalización que permitan crear estrategias pedagógicas. Y posteriormente, cuando se consigan obtener varias estrategias pedagógicas, aplicar de nuevo RL para decidir qué estrategia pedagógica tomar a continuación.

Bibliografía

- [Albrecht *et al.*, 1998] D. W. Albrecht, I. Zukerman, and A. Nicholson. Bayesian models for keyhole plan recognition in an adventure game. *User Modeling and User-Adapted Interaction*, 8(1-2):5–47, 1998.
- [Albus, 1981] J.S. Albus. *Brain, Behaviour, and Robotics*. Byte Books, Peterborough, NH, 1981.
- [Aleven and Koedinger, 2000] V. Aleven and K.R. Koedinger. The need for tutorial dialog to support self-explanation. In *Building Dialogue Systems for Tutorial Applications, Papers of the 2000 AAAI Fall Symposium*, pages 65–73, 2000.
- [Anderson and Reiser, 1985] J. Anderson and B. Reiser. The lisp tutor. In *Byte*, volume 10:4, pages 159–175, 1985.
- [André *et al.*, 1993] E. André, W. Finkler, W. Graf, A. Schauder, and W. Wahister. *Intelligent Multimedia Presentations*, chapter WIP: The Automatic Synthesis of Multimodal Presentations. Mark T. Maybury (ed.), MIT Press, 1993.
- [Anjaneyulu, 1997] K. Anjaneyulu. Concept level modelling on the www. In *Proc. of Workshop "Intelligent Educational Systems on the World Wide Web." at AI-ED'97, 8th World Conference on Artificial Intelligence in Education, Kobe, Japan, ISIR*, pages 26–29, 1997.
- [Asnicar and Tasso, 1997] F. A. Asnicar and C. Tasso. ifweb: A prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In P. Brusilovsky, J. Fink, and J. Kay, editors, *Proceedings of Workshop "Adaptive Systems and User modeling on the World Wide Web." at 6th International Conference on User Modeling, UM97*, pages 3–11, 1997.
- [Baffes and Mooney, 1996] P. Baffes and R. Mooney. Refinement-based student modeling and automated bug library construction. *Journal of Artificial Intelligence in Education*, 7(1):75–116, 1996.

- [Bart *et al.*, 1976] A. Bart, M. Beard, and R. Atkinson. The computer as tutorial laboratory: the stanford bip project. *International Journal of Man-Machine Studies*, 8:567–595, 1976.
- [Barto *et al.*, 1995] A. G. Barto, S.J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 1(72), 1995.
- [Beaumont, 1998] I. Beaumont. User modeling in the interactive anatomy tutoring system anatom-tutor. In P. Brusilovsky and A Kobsand J. Vassileva, editors, *Adaptive Hypertext and Hypermedia*, volume 4, pages 91–116. Kluwer Academic Publisher, The Netherlands, 1998.
- [Beck, 2001] J. Beck. *ADVISOR: A machine learning architecture for intelligent tutor construction*. PhD thesis, University of Massachusetts Amherst, 2001.
- [Bellman, 1957] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ., 1957.
- [Bodner *et al.*, 1997] R. Bodner, M. Chignell, and J. Tam. Website authoring using dynamic hypertext. In *Proceedings of Webnet*, pages 104–109, 1997.
- [Boticario *et al.*, 2001] J.G. Boticario, E. Gaudio, and C. Catalina. Towards personalised learning communities on the web. In Anneke Eurolings Kai Hakkarainen Pierre Dillenbourg, editor, *European Perspectives on Computer-Supported Collaborative Learning. Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, pages 115–122, 2001.
- [Boyle and Encarnación, 1998] C. Boyle and A.O. Encarnación. Metadoc: an adaptive hypertext reading system. In P. Brusilovsky and A Kobsand J. Vassileva, editors, *Adaptive Hypertext and Hypermedia*, volume 3, pages 71–89. Kluwer Academic Publisher, The Netherlands, 1998.
- [Bra and Calvi, 1998] P. De Bra and L. Calvi. Aha! an open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia*, 4:115–139, 1998.
- [Brusilovsky and Cooper, 1999] P. Brusilovsky and D. Cooper. Adaptive hypermedia for a web-based performance support system. In *Proc. of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*, 1999.
- [Brusilovsky and Pesin, 1994] P. Brusilovsky and L. Pesin. An intelligent learning environment for cds/isis users. In J. J. Levonen and M. T. Tukianinen, editors, *Proc. of The interdisciplinary workshop on complex learning in computer environments (CLCE94), Joensuu, Finland, EIC*, pages 29–33, 1994.

- [Brusilovsky *et al.*, 1996] Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber. Elm-art: An intelligent tutoring system on world wide web. In Claude Frasson, Gilles Gauthier, and Alan Lesgold, editors, *Intelligent Tutoring Systems*, volume 1086, pages 261–269. Springer, 1996.
- [Brusilovsky *et al.*, 1997] P. Brusilovsky, S. Ritter, and E. Schwarz. Distributed intelligent tutoring on the web. *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, pages 482–489, 1997.
- [Brusilovsky *et al.*, 1998] P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems*, 30:291–300, 1998.
- [Brusilovsky, 1998] P. Brusilovsky. Adaptive educational systems on the worldwide-web: A review of available technologies. In *Proceedings of Workshop "WWW-based tutoring."* 4th International Conference on Intelligent Tutoring Systems, San Antonio, TX, 1998.
- [Brusilovsky, 1999] P. Brusilovsky. Adaptive and intelligent technologies for web-based education. *Kunstliche Intelligenz. Special Issue on Intelligent Tutoring Systems and Teleteaching*, 4, 1999.
- [Burns and Capps, 1988] Hugh Burns and Charles Capps. Foundations of intelligent tutoring systems: An introduction. In Lawrence Erlbaum Associates, editor, *Foundations of Intelligent Tutoring Systems*, pages 1–19. Hillsdale, N.J., 1988.
- [Burns *et al.*, 1991] H. L. Burns, J. W. Parlett, and C. L. Redfield. *Intelligent Tutoring Systems: Evolutions in Design*. Lawrence Erlbaum Associates, NH Hillsdale, 1991.
- [Burton, 1982] R. Burton. Diagnosis bugs in a simple procedural skill. In D. Sleeman and L. Brown, editors, *Intelligent Tutoring Systems*. London: Academic Press, 1982.
- [Carolis *et al.*, 1998] B. De Carolis, F. de Rosis, C. Andreoli, V. Cavallo, and M.L. de Cicco. The dynamic generation of hypertext presentations of medical guidelines. *The New Review of Hypermedia and Multimedia*, 4:67–88, 1998.
- [Carr and Goldstein, 1977] B. Carr and I. Goldstein. Overlays: A teory of modeling for computer aided instruction. Technical report ai memo 406, AI Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1977.

- [Carro *et al.*, 1999] R.M. Carro, E. Pulido, and R. Rodríguez. Tangow: Task-based adaptive learner guidance on the www. In *Proc. of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*, 1999.
- [Carro *et al.*, 2001] R.M. Carro, E. Pulido, and P. Rodríguez. Tangow: a model for internet based learning. *International Journal of Continuing Engineering Education and Life-Long Learning, IJCEELL*, 11(1-2), 2001.
- [Castro *et al.*, 2002] E. Castro, D. Cuadra, P. Martínez, and A. Iglesias. Integrating intelligent methodological and tutoring assistance in a case platform: the pandora experience. In *Proceeding of the Informing Science and IT Education Conference*, pages 11–14, 2002.
- [Chen, 1976] P. Chen. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 1976.
- [Chin, 2001] D. Chin. Empirical evaluations of user models and user-adapted system. *User Modelling and User Adapted Interaction*, 11(1):181–194, 2001.
- [Chiu and Webb, 1998] B. C. Chiu and I. G. Webb. Using decision trees for agent modeling: Improving prediction performance. *User Modelling and User Adapted Interaction*, 8(1-2):131–152, 1998.
- [Cho *et al.*, 1999] Byung-In Cho, Joel Michael, Allen Rovick, and Martha Evens. Curriculum planning model for an intelligent tutoring system. In *Proceedings of 12th International Florida Artificial Intelligence Research Symposium. Orlando, FL.*, pages 197–201, 1999.
- [Cho, 2000] Byung-In Cho. *Dynamic Planning Models to Support Curriculum Planning and Multiple Tutoring Protocols in Intelligent Tutoring Systems*. Phd thesis, Illinois Institute of Technology, 2000.
- [Clancey, 1987] William Clancey. *Methodology for Building Intelligent Tutoring Systems*, chapter Artificial Intelligence and Instruction Applications and Methods Reading, pages 193–227. Greg P. Kearsley. MA: Addison-Wesley, 1987.
- [Corbett and Anderson, 1992] A. T. Corbett and J. R. Anderson. Student modeling and mastery learning in a computer-based programming tutor. In C. Frason, G. Gauthier, and G. I. McCalla, editors, *In Proceedings of the Second International Conference on Intelligent Tutoring Systems*, pages 413–420. Springer, 1992.
- [Corbett *et al.*, 1993] A. T. Corbett, J. R. Anderson, and O'Brien A. T. The predictive validity of student modeling in the act programming tutor. In P. Brna,

- S. Ohlsson, and H. Pain, editors, *In Proceedings of the Artificial Intelligence and Education 93 Conference. AI-ED93*. 1993.
- [Díaz, 2003] P. Díaz. Usability of hypermedia educational e-books. *D-Lib Magazine*, 9(3), March 2003.
- [Dietterich, 2000] Tomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of Artificial Intelligence Research*, 13(227-303), 2000.
- [Dillenbourg, 1999] P. Dillenbourg. *Collaborative-Learning: Cognitive and Computational Approaches*. Elsevier Science, p. dillenbourg edition, 1999.
- [Eliot *et al.*, 1997] C. Eliot, D.Ñeiman, and M. Lamar. Medtec: A web-based intelligent tutor for basic anatomy. In S. Lobodzinski and I. Tomek, editors, *Proc. of WebNet'97, World Conference of the WWW, Internet and Intranet*, pages 161–165, 1997.
- [Fernández-Castro, 1989] I. Fernández-Castro. *Estrategias de Enseñanza en un Sistema Inteligente de Enseñanza Asistida por Ordenador*. Phd tesis, Universidad del País Vasco, 1989.
- [Fernández, 1999] Fernando Fernández. Vqql: Un modelo de aprendizaje or refuerzo para dominios continuos e indeterministas. Proyecto fin de carrera. ingeniería en informática, Universidad Carlos III de Madrid. Escuela Politécnica Superior, 1999.
- [Fink *et al.*, 1998] J. Fink, A. Kobsa, and A.Ñill. Adaptable and adaptive information provision for all users, including disabled and elderly people. *New Review of Hypermedia and Multimedia*, 4:163–188, 1998.
- [Freedman, 1999] Reva Freedman. Atlas: A plan manager for mixed-initiative, multimodel dialogue. In *AAAI'99 Workshop on Mixed-Initiative Intelligence, Orlando*, pages 107–114, 1999.
- [Gaudioso, 2002] E. Gaudioso. *Contribuciones al Modelado del Usuario en Entornos Adaptativos de Aprendizaje y Colaboración a través de Internet mediante técnicas de Aprendizaje Automático*. Phd tesis, Universidad Nacional de Educación a Distancia, 2002.
- [Green and Carberry, 1999] N. Green and S. Carberry. A computational mechanism for initiative in answer generation. *User Modeling and User-Adapted Interaction*, 9(1-2):93–132, 1999.

- [Heck *et al.*, 2000] R.H. Heck, L.K. Johnsrud, and V.J. Rosser. Administrative effectiveness in higher education: Improving assessment procedures. *Research in Higher Education*, 41(6):663–684, 2000.
- [Henze and Nejd, 1999] N. Henze and W. Nejd. Adaptivity in the kbs hyperbook system. In *Proc. of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW*, 1999.
- [Höök, 1997] K. Höök. Evaluating the utility and usability of an adaptive hypermedia system. *User Modeling and User-Adapted Interaction. Special Issue on Adaptive Hypertext and Hypermedia*, 6(2-3):131–156, 1997.
- [Hohl *et al.*, 1996] H. Hohl, H.D. Böcker, and R. Gunzenhäuser. Hypadapter: An adaptive hypertext system for exploratory learning and programming. *User Modeling and User-Adapted Interaction* 6, 2-3:131–156, 1996.
- [Hoppe, 1994] U. Hoppe. Deductive error diagnosis and inductive error generalization for intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 5:27–49, 1994.
- [Ito *et al.*, 1998] J. Ito, Y. Watanabe, K. Kondo, and M. Okamoto. Pen based user interface for an its on www client. In *Proceedings of ICCE'98*, pages 324–327, Beijing, China, 1998.
- [Kaelbling *et al.*, 1993] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Hierarchical learning in stochastic domains: Preliminary results. In *Tenth International Conference on Machine Learning, Amherst, MA*, 1993.
- [Kaelbling *et al.*, 1996] Leslie P. Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
- [Kay and Kummerfeld, 1997] J. Kay and B. Kummerfeld. User models for customized hypertext. In C. Nicholas and J. Mayfield, editors, *Intelligent hypertext: Advanced techniques for the World Wide Web*, volume 1326. Springer-Verlag. Lecture Notes in Computer Science, 1997.
- [Kayama and Okamoto, 2001] M. Kayama and T. Okamoto. N3. nn navigation support system-knowledge-navigation in hyperspace. the sub-symbolic approach. *Journal of Educational Multimedia and Hypermedia*, 10(1):85–103, 2001.
- [Khan and Yip, 1996] T. Khan and Y.J. Yip. Pedagogical principles of case-based cal. *Journal of Computer Assisted Learning*, 1(12):172–192, 1996.

- [Knott *et al.*, 1996] A. Knott, C. Mellish, J. Oberlander, and M. O'Donnell. Sources of flexibility in dynamic hypertext generation. In *Proceedings of Eighth International Natural Language Generation Workshop (Brighton, UK, 1996)*.
- [Kono *et al.*, 1994] Y. Kono, M. Ikeda, and R. Mizoguchi. Themis: A nonmonotonic inductive student modeling system. *Journal of Artificial Intelligence in Education*, 5(3):371–413, 1994.
- [Krause *et al.*, 1993] J. Krause, A. Hirschmann, and E. Mittermaier. The intelligent help system comfohelp: Towards a solution of the practicability problem for user modeling and adaptive systems. *User Modeling and User-Adapted Interaction*, 3(3), 1993.
- [Kushniruk and Wang, 1994] A. Kushniruk and H. Wang. A hypermedia-based educational system with knowledge-based guidance. In *Proceedings of EDMEDIA'94 - Word Conference on Educational Multimedia and Hypermedia. Vancouver, Canada, 1994*, pages 335–340, 1994.
- [Lai *et al.*, 1995] M.C. Lai, B.H. Chen, and S.M. Yuan. Toward a new educational environment. In *Proc. of 4th International World Wide Web Conference*, 1995.
- [Landon, 2000] B. Landon. *Online Educational Delivery Applications: A Web-tool for Comparative Analysis*. UNL Libraries Internet Resources Catalog Record, 2000.
- [Langley and Ohlsson, 1984] P. Langley and S. Ohlsson. Automated cognitive modeling. In *Proceedings of the Second National Conference on Artificial Intelligence*, 1984.
- [Lebowitz, 1987] M. Lebowitz. Experiments with incremental concept formation: Unimen. *Machine Learning*, 2:103–138, 1987.
- [Leigh *et al.*, 1997] J. Leigh, A.E. Johnson, and T.A. DeFanti. Issues in the design of a flexible distributed architecture for supporting persistence and interoperability in collaborative virtual environments. In *Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM). Conference on High Performance Networking and Computing, San Jose, CA*, pages 1–14, 1997.
- [M. Debevc and Sveciko, 1996] D. Donlagic M. Debevc, B. Meyer and R. Sveciko. Design and evaluation of an adaptive icon toolbar. *User Modeling and User-Adapted Interaction*, 6(1), 1996.
- [Martin and VanLehn, 1993] J. Martin and K. VanLehn. Olae: progress toward a multi-activity, bayesian student modeler. In *Proceedings of the International Conference on Artificial Intelligence in Education*, 1993.

- [Mitchell, 1997] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [Murray, 1999a] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *J. of Artificial Intelligence in Education*, 10:98–129, 1999.
- [Murray, 1999b] W.R. Murray. Control for intelligent tutoring systems: a comparison of blackboard architectures and discourse management networks. *Intelligent Tutoring Systems*, 3(1):107–124, 1999.
- [Nakabayashi *et al.*, 1997] K. Nakabayashi, M. Maruyama, Y. Koike, H. Touhei, and Y. Fukuhara. Architecture of an intelligent tutoring system on the www. In *Proceedings of the 8th World Conference of the AIED Society*, pages 18–22, 1997.
- [Nakabayashi *et al.*, 1998] K. Nakabayashi, M. Maruyama, Y. Kato, H. Touhei, and Y. Fukuhara. Architecture of an intelligent tutoring system on the www. *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, pages 39–46, 1998.
- [Nomoto *et al.*, 1997] T. Nomoto, N. Matsuda, T. Hirashima, and J. I. Toyoda. Toward learning from surfing. In P. Brusilovsky, K. Nakabayashi, and S. Ritter, editors, *Proceedings of Workshop "Intelligent Educational Systems on the World Wide Web."* at the 8th World Conference on Artificial Intelligence in Education: AI-ED97, pages 40–46, 1997.
- [O'Donnell, 1997] M. O'Donnell. Variable length on-line document generation. In *Proceedings of Flexible Hypertext Workshop, help at the Eighth ACM International Hypertext Conference (Hypertext '97)*, 1997.
- [Preece *et al.*, 2000] J. Preece, Y. Rogers, and H. Sharp. *Interaction Design: beyond human-computer interaction*. John Wiley and Sons, New York, 2000.
- [Prentzas *et al.*, 2002] J. Prentzas, I. Hatzilygeroudis, and J. Garofalakis. A web-based intelligent tutoring system using hybrid rules as its representational basis. In Stefano A. Cerri, Guy Gouarderes, and Fábio Paraguaçu, editors, *Proceedings of the 6th International Conference, ITS 2002*, volume 1, pages 119–128. Lecture Notes in Computer Science. Springer Verlag, 2002.
- [Rea *et al.*, 2000] A. Rea, D. White, R. McHaney, and C. Sánchez. *Pedagogical Methodology in Virtual Courses. Web-Based Learning and Teaching Technologies: Opportunities and Challenges*. Anil Aggarwal. Idea Group Publishing (Ed), 2000.

- [Reinsmith, 1992] W.A. Reinsmith. *Archetypal forms in teaching: A continuum*. Greenwood Press, Westport, US, 1992.
- [Rich, 1979] E. Rich. User modelling via stereotypes. *Cognitive Science*, 3(4):329–354, 1979.
- [Rios *et al.*, 1999] A. Rios, E. Millán, M. Trella, J.L. Pérez de-la Cruz, and R. Conejo. Internet based evaluation system. In S.P. Laojie and M. Vivet, editors, *Artificial Intelligence in Education: Open Learning Environments*, pages 387–394, IOS Press, Amsterdam, 1999.
- [Santamaría *et al.*, 1998] Juan C. Santamaría, Richard S. Sutton, and Ashwin Ram. Experiments with reinforcement learning in problems with continuous state and action spaces. *Adaptive Behavior*, 6(2):163–218, 1998.
- [Schöch *et al.*, 1999] V. Schöch, M. Specht, and G. Weber. An empirical evaluation of a tutorial agent. In Ottmann T. and Tomek I., editors, *Proc. of ED-MEDIA/ED-TELECOM'98 - 10th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications, Freiburg, Germany, AACE*, pages 1242–1247, IOS, Amsterdam, 1999.
- [Shah, 1997] Farhana Shah. *Recognizing and Responding to Student Plans in an Intelligent Tutoring System: CIRCSIM-Tutor*. PhD thesis, Illinois Institute of Technology, 1997.
- [Signore *et al.*, 1997] O. Signore, R. Bartoli, and G. Fresta. Tailoring web pages to users' needs. In P. Brusilovsky, J. Fink, and J. Kay, editors, *Proceedings of Workshop Adaptive Systems and User modeling on the World Wide Web. 6th International Conference on User Modeling, UM97*, pages 85–90, 1997.
- [Singh, 1992] Santinder Pal Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning*, 8(3):323–340, 1992.
- [Singh, 1993] Santinder Pal Singh. *Learning to Solve Markovian Decision Processes*. Cmpsci technical report, Department of Computer Science, University of Massachusetts, 1993.
- [Sison and Shimura, 1998] R. Sison and M. Shimura. Student modelling and machine learning. *International Journal of Artificial Intelligence in Education*, 9:128–158, 1998.
- [Sison *et al.*, 1998] R. Sison, M. Numao, and M. Shimura. Discovering error classes from discrepancies in novice behaviors via multistrategy conceptual clustering. *User Modeling and User-Adapted Interaction*, 8(1-2):103–129, 1998.

- [Sison *et al.*, 2000] R. Sison, M. Numao, and M. Shimura. Multistrategy discovery and detection of novice programmer errors. *Machine Learning*, 38:157–180, 2000.
- [Sison, 2001] R. Sison. Framework for intelligent internet-based learning support. In *Proceedings of the Ninth International Conference on Computers in Education*, pages 926–931, 2001.
- [Sleeman and Brown, 1982] D. Sleeman and S. Brown. *Intelligent Tutoring Systems. Computers and People Series*. Academic Press, London, 1982.
- [Sleeman, 1977] D. Sleeman. A system which allows student to explore algorithms. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 780–78, 1977.
- [Specht and Oppermann, 1998] M. Specht and R. Oppermann. Ace-adaptive courseware environment. *The New Review of Hypermedia and Multimedia*, 4:141–161, 1998.
- [Specht *et al.*, 1997] M. Specht, G. Weber, S. Heitmeyer, and V. Schöch. Ast: Adaptive www-courseware for statistics. In P. Brusilovsky, J. Fink, and J. Kay, editors, *Proc. of Workshop Adaptive Systems and User Modeling on the World Wide Web. 6th International Conference on User Modeling, UM97, Chia Laguna, Sardinia, Italy, Carnegie Mellon Online*, pages 91–95, 1997.
- [Stern and Woolf, 1998] M.K. Stern and B.P. Woolf. Curriculum sequencing in a web-based tutor. In B.P. Goettl, H.M. Half, C.L. Redfield, and V.J. Shute, editors, *Intelligent Tutoring Systems*, pages 574–583. Lecture Notes in Computer Science, Springer Verlag, 1998.
- [Stern, 2001] Mia Keryn Stern. *Using Adaptive Hypermedia and Machine Learning to Create Intelligent Web-based Courses*. Phd, University of Massachusetts, Amherst, 2001. Directed by: Professor Beverly Park Woolf.
- [Suthers and Jones, 1997] D. Suthers and D. Jones. An architecture for intelligent collaborative educational systems. *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, pages 55–62, 1997.
- [Sutton and Barto, 1998] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [Sutton, 1988] R. S. Sutton. Learning to predict by the method of temporal difference. *Machine Learning*, 3:9–44, 1988.

- [Sutton, 1990] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Seventh International Conference on Machine Learning*, 1990.
- [Sutton, 1991] R. S. Sutton. Planning by incremental dynamic programming. In *Eighth International Workshop on Machine Learning*, pages 353–357, 1991.
- [Teorey *et al.*, 1986] T. Teorey, D. Yang, and J. Fry. A logical design methodology for relation databases using the extended entity-relationship model. *Computer Surveys*, 18(2), 1986.
- [Tong and Angelides, 1999] Amelia K.Y. Tong and Marios C. Angelides. Formalising tutoring strategy selection in multimedia tutoring systems. *Information and software technology*, 1(41):67–90, 1999.
- [VanLehn and Zhendong, 2001] K. VanLehn and N. Zhendong. Bayesian student modelling, user interfaces and feedback: a sensitivity analysis. *International Journal of Artificial Intelligent in Education*, 2:155–184, 2001.
- [VanLehn *et al.*, 1996] K. VanLehn, S. Ohlsson, and R.Ñason. Applications of simulated students: An exploration. *Journal of Artificial Intelligent in Education*, 2(5):135–175, 1996.
- [VanLehn, 1988] Kurt VanLehn. *Student Modeling*, pages 55–78. M. Polson. Foundations of Intelligent Tutoring Systems, Hillsdale, N.J, lawrence erlbaum associates edition, 1988.
- [Vassileva, 1995] J. Vassileva. Reactive instructional planning to support interacting teaching strategies. In VA AACE: Charlottesville, editor, *In Proceedings of the 7-th World Conference on AI and Education AIED'95*, pages 334–342, Washington, 1995.
- [Vassileva, 1997] J. Vassileva. Dynamic course generation on the www. In Boulay B. and Mizoguchi R., editors, *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*, pages 498–505, IOS, Amsterdam, 1997.
- [Verdejo *et al.*, 2002] M.F. Verdejo, B. Barros, T. Read, and M. Rodríguez-Artacho. A system for the specification and development of an environment for distributed cscl environments. In Stefano A. Cerri, Guy Gouarderes, and Fábio Paraguaçu, editors, *Proceeding of ITS 2002*, volume 2363, pages 139–148. Springer Verlag, 2002.
- [Villano, 1992] M. Villano. Probabilistic student models: Bayesian belief networks and knowledge space theory. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, 1992.

- [Virvou and du Bulay, 1999] M. Virvou and B. du Bulay. Human plausible reasoning for intelligent help. *User Modelling and User Adapted Interaction*, 9(4):323–377, 1999.
- [Watkins and Dayan, 1992] C. J. C. H. Watkins and P. Dayan. Q-Learning. *Machine Learning*, 8:279–292, 1992.
- [Watkins, 1989] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King’s College, Cambridge, UK, 1989.
- [Webb *et al.*, 2001] G. I. Webb, M. J. Pazzani, and D. Billsus. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):19–29, 2001.
- [Weber and Specht, 1997] G. Weber and M. Specht. User modeling and adaptive navigation support in www-based tutoring systems. In A. Jameson, C. Paris, and C. Tasso, editors, *User Modeling*, pages 289–300. Springer Verlag, 1997.
- [White and Frederiksen, 1990] B. Y. White and J. R. Frederiksen. Causal model progressions as a foundation for intelligent learning environments. *Artificial Intelligence*, 42:99–157, 1990.
- [Woolf, 1987] B.P. Woolf. Representing complex knowledge in an intelligent machine tutor. *Computational Intelligence*, 3:45–55, 1987.
- [Zukerman and Albrecht, 2001] I. Zukerman and D. W. Albrecht. Predictive statistical user models for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):5–18, 2001.

Apéndice A

Ejemplos de Contenidos del Sistema de Educación




En este anexo se presentan algunas páginas web que muestran el contenido del sistema RLATES. En concreto, se mostrará los tipos de elemento *Definición*, *Introducción* y *Ejercicio* con formato de *Imagen* y con formato de *Texto*. También se presentan dos ejemplos de páginas web que contienen tests que han de contestar los estudiantes.

Las páginas que se muestran a continuación muestran, en concreto, definiciones, introducciones, ejercicios y tests del tema *Entidad* que pertenece tanto al árbol de conocimiento *A* como al árbol de conocimiento *B* utilizados en los experimentos y detallados en la sección 4.3.

The screenshot displays a web-based interface for the 'Entidad' (Entity) topic. At the top, the word 'ENTIDAD' is written in large, bold, red capital letters. Below the title, there are three navigation icons: a grey square with a left arrow labeled 'Anterior', a black square with a green upward arrow labeled 'Salir', and a black square with a green right arrow labeled 'Siguiente'. Below these icons is a blue header bar with two tabs: 'Introducción' (selected) and 'Tests'. The main content area is a light blue box containing an icon of an open book and the heading 'Introducción:'. The text in this box explains that in the real world, objects have specific characteristics, such as a car with license plate M-2548-HM being red, and another with V-5427-AB being blue. It states that this leads to the concept of an 'entidad' (entity), which is defined in the E/R model. The text also mentions that the goal is to ensure learning of this concept through examples and tests to evaluate student comprehension. At the bottom of the interface, there are three more navigation icons: 'Anterior', 'Salir', and 'Siguiente'. A footer at the very bottom reads '© UNIVERSIDAD · CARLOS III · DE · MADRID'.


Figura A.1: Contenido de RLATES. Un elemento de tipo *Introducción* en formato *Texto* del tema Entidad

ENTIDAD

 Anterior
  Salir
  Siguiete

Definición

Tests

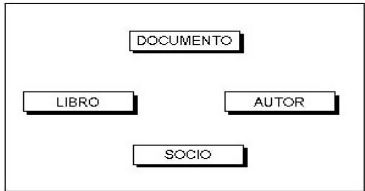


Definición:




Entidad es "una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa" ANSI (1977). Es cualquier objeto, que existe en la realidad y acerca del cual queremos almacenar información en la base de datos. También se le denomina **tipo de entidad**.

En general, se puede afirmar que un tipo de entidad tiene que tener existencia propia.

La representación gráfica de un tipo de entidad es un rectángulo etiquetado con el nombre del tipo de entidad, como se muestra a continuación:



Representación de cuatro tipos de entidad.

 Anterior
  Salir
  Siguiete

© UNIVERSIDAD · CARLOS III · DE · MADRID

Figura A.2: Contenido de RLATES. Un elemento de tipo *Definición* en formato *Imagen* del tema Entidad

The screenshot displays a web-based interface for the topic "ENTIDAD". At the top, the word "ENTIDAD" is written in large, bold, red capital letters. Below it are three navigation icons: a left arrow labeled "Anterior", a house icon labeled "Salir", and a right arrow labeled "Siguiete". A blue tabbed interface shows two tabs: "Definición" (selected) and "Tests". The "Definición" tab contains a light blue box with a book icon and the heading "Definición:". The text inside the box reads: "Entidad es *'una persona, lugar, cosa, concepto o suceso, real o abstracto, de interés para la empresa'*" ANSI (1977). Es cualquier objeto, que existe en la realidad y acerca del cual queremos almacenar información en la base de datos. También se le denomina **tipo de entidad**.

En general, se puede afirmar que un tipo de entidad tiene que tener existencia propia.

La representación gráfica de un tipo de entidad es un rectángulo etiquetado con el nombre del tipo de entidad en el interior del mismo.

At the bottom of the page, there are three more navigation icons: "Anterior", "Salir", and "Siguiete". A footer bar at the very bottom contains the text "© UNIVERSIDAD · CARLOS III · DE · MADRID".

Figura A.3: Contenido de RLATES. Un elemento de tipo *Definición* en formato *Texto* del tema Entidad

ENTIDAD

Anterior **Salir** Siguiente

Ejercicio Tests

Ejercicio:

ENUNCIADO:

Una empresa desea conocer de todo candidato a ocupar un puesto de trabajo, sus datos personales, la formación y los conocimientos adquiridos, la experiencia profesional (en caso de existir), las características del puesto deseado y los datos relativos a la movilidad.

Se pide, representar gráficamente los tipos de entidad encontrados.

SOLUCIÓN:

Candidato es un objeto del mundo real, sobre el cual se desea almacenar cierta información en la base de datos de la empresa. Por tanto, constituye un tipo de entidad que se representa tal y como se muestra a continuación:

CANDIDATO

Representación gráfica del tipo de entidad CANDIDATO.

Anterior **Salir** Siguiente

© UNIVERSIDAD CARLOS III DE MADRID

Figura A.4: Contenido de RATES. Un elemento de tipo *Ejercicio* en formato *Imagen* del tema Entidad

The screenshot displays a web interface for the 'Entidad' topic. At the top, the word 'ENTIDAD' is written in large red letters. Below it are three navigation buttons: 'Anterior' (left arrow), 'Salir' (up arrow), and 'Siguiente' (right arrow). A blue header bar contains two tabs: 'Ejercicio' (selected) and 'Tests'. The main content area is light blue and contains a small cartoon character next to the heading 'Ejercicio:'. Below this, the text is organized into sections: 'ENUNCIADO:' followed by a paragraph describing a company's need for candidate information, and 'SOLUCIÓN:' followed by a paragraph defining 'Candidato' as a real-world entity. At the bottom, there are the same three navigation buttons as at the top, and a footer with the text '© UNIVERSIDAD · CARLOS III · DE · MADRID'.

ENTIDAD

Anterior Salir Siguiente

Ejercicio Tests

Ejercicio:

ENUNCIADO:

Una empresa desea conocer de todo candidato a ocupar un puesto de trabajo, sus datos personales, la formación y los conocimientos adquiridos, la experiencia profesional (en caso de existir), las características del puesto deseado y los datos relativos a la movilidad.

Se pide, describir los tipos de entidad encontrados.

SOLUCIÓN:




Candidato es un objeto del mundo real, sobre el cual se desea almacenar cierta información en la base de datos de la empresa. Por tanto, constituye un tipo de entidad que se representa dibujando un rectángulo, en cuyo interior se escribe la palabra CANDIDATO.

Anterior Salir Siguiente

© UNIVERSIDAD · CARLOS III · DE · MADRID

Figura A.5: Contenido de RLATES. Un elemento de tipo *Ejercicio* en formato *Texto* del tema Entidad

ENTIDAD

 Anterior
  **Salir**
 Siguiente

Definición
Tests




Test

Test de ENTIDAD:

PREGUNTA 1:

Marque las afirmaciones que considere correctas:




- Todo tipo de entidad, recibe como nombre, un sustantivo singular. Dicho nombre, identifica de forma única al tipo de entidad.
- Un tipo de entidad es la estructura genérica que permite modelar un conjunto de objetos del mundo real, que tienen las mismas propiedades. Así, por ejemplo, seta es un tipo de entidad, porque nos permite referirnos al conjunto de todas las setas que encontramos en el mundo real, y, además, todas ellas tienen una serie de características comunes, como tener nombre, color, sabor, etc.
- Un tipo de entidad es una estructura que permite modelar un objeto del mundo real. Así, por ejemplo, la seta que tiene por nombre Amanita Verda, color blanco y sabor agrio, es un tipo de entidad.
- En el esquema E/R, un tipo de entidad se representa mediante un rombo, en cuyo interior se escribe el nombre que dicho tipo de entidad recibe, que es un sustantivo singular en mayúsculas.

 Anterior
  **Salir**
 Siguiente

© UNIVERSIDAD · CARLOS III · DE · MADRID

Figura A.6: Contenido de RLATES. Ejemplo 1 de Test del tema Entidad

ENTIDAD

 Anterior
  Salir
  Siguiente

Ejercicio Tests

Test




Test de ENTIDAD:

PREGUNTA 1:

Dado el siguiente enunciado en cursiva, determinar cuáles son los tipos de entidad encontrados:

"Se sabe que todo alumno se matricula de una o varias asignaturas, y, en una asignatura se matriculan varios alumnos. De un alumno se desea recoger su DNI, Nombre, Apellidos, Dirección, Código postal, Ciudad y Teléfono. De una asignatura se quiere saber su Código, Nombre y Responsable."

DNI, Responsable y Teléfono son tipos de entidad.
 Alumno es un tipo de entidad.
 Se matricula es un tipo de entidad.
 Asignatura es un tipo de entidad.

 Anterior
  Salir
  Siguiente

© UNIVERSIDAD · CARLOS III · DE · MADRID

Figura A.7: Contenido de RLATES. Ejemplo 2 de Test del tema Entidad