

DISEÑO Y EVALUACION DE UN SISTEMA TUTOR INTELIGENTE PARA APRENDIZAJE EN SALUD



Monografía para optar al título de
Ingeniero de Sistemas

DIANA CRISTINA MUÑOZ MENESES
JEIMY ALEXANDRA ORTIZ BUITRÓN

Director:
PhD. Carolina González Serrano

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Ingeniería de Sistemas
Grupo IDIS – Investigación y Desarrollo en Ingeniería de Software
Línea de Investigación Ingeniería del Software
Popayán, Mayo de 2010



AGRADECIMIENTOS

Nuestros agradecimientos en primer lugar a nuestras familias, por su cariño y apoyo incondicional durante todo nuestro proceso de formación.

A nuestra directora, PhD Carolina González Serrano por su colaboración, consejos, apoyo y paciencia sin lo cual no hubiese sido posible la realización de éste proyecto.

A nuestros compañeros, amigos y profesores por el apoyo brindado en los momentos más oportunos.

A los departamentos de Pediatría y Enfermería de la Universidad del Cauca, por su apoyo en el desarrollo de este proyecto. Al licenciado Wilson Espinosa por su apoyo en el área pedagógica.

Finalmente a la Universidad del Cauca, institución que nos brindó la oportunidad a través del programa de Ingeniería de Sistemas, para realizar nuestros estudios de pregrado.



TABLA DE CONTENIDO

AGRADECIMIENTOS	2
TABLA DE CONTENIDO	3
LISTA DE TABLAS	6
LISTA DE FIGURAS	7
INTRODUCCIÓN	1
1. ATENCIÓN INTEGRADA A LAS ENFERMEDADES PREVALENTES DE LA INFANCIA	4
1.1 ESTRATEGIA AIEPI	4
1.2 COMPONENTES	5
2. SISTEMAS TUTORES INTELIGENTES	7
2.1 DEFINICIÓN	7
2.2 PERSPECTIVA HISTÓRICA	8
2.2.1 EVOLUCIÓN DE LOS SISTEMAS DE ENSEÑANZA ASISTIDA POR COMPUTADOR	8
2.2.2 AMBIENTES VIRTUALES DE APRENDIZAJE	9
2.3 ARQUITECTURA DE LOS SISTEMAS TUTORES INTELIGENTES	10
2.3.1 MÓDULO EXPERTO	10
2.3.2 MÓDULO DEL ESTUDIANTE	12
2.3.3 MÓDULO DEL TUTOR	13
2.3.4 MÓDULO DEL ENTORNO	14
2.4 SISTEMAS TUTORES INTELIGENTES DESARROLLADOS PARA EL DOMINIO DE SALUD	15
3. AGENTES INTELIGENTES	19
3.1 DEFINICIÓN DE AGENTE	19
3.2 CARACTERÍSTICAS DE LOS AGENTES	19
3.3 CLASIFICACIÓN DE LOS AGENTES	20
3.4 SISTEMAS MULTI-AGENTE – SMA	21
3.4.1 INTERACCIÓN ENTRE AGENTES	21
3.4.2 COORDINACIÓN ENTRE AGENTES	22
3.4.3 COMUNICACIÓN ENTRE AGENTES	22
3.4.4 PROTOCOLOS DE COMUNICACIÓN	23
3.4.5 NEGOCIACIÓN ENTRE AGENTES	24
<hr/>	
DISEÑO Y EVALUACIÓN DE UN SISTEMA TUTOR INTELIGENTE PARA APRENDIZAJE EN SALUD	



3.5 SISTEMAS TUTORES INTELIGENTES DESARROLLADOS CON TECNOLOGÍA DE AGENTES	25
3.6 METODOLOGÍA INGENIERAS PARA EL DESARROLLO DE SISTEMAS MULTI-AGENTE	27
4. ARQUITECTURA DEL SISTEMA	32
<hr/>	
4.1 MÓDULO DEL ESTUDIANTE	33
4.1.1 EL MODELO DEL ESTUDIANTE	33
4.1.2 INICIALIZACIÓN DEL MODELO DEL ESTUDIANTE	35
4.1.3 PROCESO DE DIAGNÓSTICO.	37
4.2 MÓDULO EXPERTO	39
4.3 MÓDULO DEL TUTOR	41
4.3.1 EL AGENTE TUTOR.	42
4.3.2 EL AGENTE DE CONCEPTO.	46
4.4 MÓDULO DEL ENTORNO	48
5. PROTOTIPO DE SISTEMA TUTOR INTELIGENTE PARA APRENDIZAJE EN SALUD	50
<hr/>	
5.1. ESPECIFICACIÓN DE REQUISITOS	51
5.1.1 REQUISITOS FUNCIONALES	51
5.1.2 REQUISITOS NO FUNCIONALES	51
5.2. ANÁLISIS	52
5.2.1 MODELO DE INTERACCIÓN ENTRE AGENTES	55
5.2.2 DIAGRAMAS DE COLABORACIÓN	56
5.2.3 MODELO DE LA ORGANIZACIÓN	58
5.2.4 MODELO DEL ENTORNO	59
5.3 DISEÑO	59
5.3.1 MODELO DE AGENTES	59
5.3.2 MODELO DE TAREAS Y OBJETIVOS	61
5.3.3 DIAGRAMA DE CLASES DEL SISTEMA	63
5.3.4 PATRONES DE DISEÑO	64
5.4 IMPLEMENTACIÓN	65
5.4.1 DIAGRAMA DE COMPONENTES	65
5.5 PRUEBAS DE SOFTWARE	68
5.6 MANUAL DE USUARIO Y DE INSTALACIÓN	68
6. EVALUACIÓN DEL SISTEMA TUTOR PROPUESTO FRENTE A UN SISTEMA DE ENSEÑANZA TRADICIONAL	69
<hr/>	
6.1 DEFINICIÓN DE INDICADORES DE EVALUACIÓN	69
6.2 DISEÑO DEL CURSO “ESTRATEGIA AIEPI: MÓDULO RECIÉN NACIDO”	71
6.2.1 ESPECIFICACIÓN DE LA ESTRUCTURA DEL CURSO EN LA PLATAFORMA MOODLE.	71
6.2.2 ESPECIFICACIÓN DE LA ESTRUCTURA DEL CURSO EN EL SISTEMA TUTOR INTELIGENTE SIAS.	72
6.3 EVALUACIÓN DE LA PLATAFORMA MOODLE EN LA ENSEÑANZA DE UN CURSO DE CAPACITACIÓN EN AIEPI	76
6.3.1 EVALUACIÓN NIVEL DE CONOCIMIENTO	76

DISEÑO Y EVALUACIÓN DE UN SISTEMA TUTOR INTELIGENTE PARA APRENDIZAJE EN SALUD



6.3.2 EVALUACIÓN DE LAS ESTRATEGIAS DE ENSEÑANZA	78
6.4 EVALUACIÓN DEL SISTEMA TUTOR INTELIGENTE SIAS EN LA ENSEÑANZA DE UN CURSO DE CAPACITACIÓN EN AIEPI	80
6.4.1 EVALUACIÓN NIVEL DE CONOCIMIENTO	81
6.4.2 EVALUACIÓN VELOCIDAD DE APRENDIZAJE	83
6.4.3 EVALUACIÓN DE LAS ESTRATEGIAS DE ENSEÑANZA	84
6.5 ANÁLISIS DE LOS RESULTADOS DE LA EVALUACIÓN DEL SISTEMA SIAS FRENTE A MOODLE	85
6.5.1 EVALUACIÓN NIVEL DE CONOCIMIENTO	85
6.5.2 EVALUACIÓN VELOCIDAD DE APRENDIZAJE	86
6.5.3 EVALUACIÓN ESTRATEGIAS DE ENSEÑANZA	87
6.5.4 EVALUACIÓN DE USABILIDAD	88
7. RESULTADOS OBTENIDOS	90
8. CONCLUSIONES Y RECOMENDACIONES	92
8.1 CONCLUSIONES	92
8.2 RECOMENDACIONES Y TRABAJO FUTURO	93
BIBLIOGRAFIA	94



LISTA DE TABLAS

Tabla 1. Asociaciones entre los Elementos del UP y Entidades del sistema [97]¡Error!	Marcador no definido.
Tabla 2. Estilos y estrategias de enseñanza	41
Tabla 3. Relación Categoría-Indicador	71
Tabla 4. Estructura del Tutorial.....	75
Tabla 5. Resultados discriminados por indicador señalando la diferencia porcentual	87
Tabla 6. Resultados discriminados por indicador señalando la diferencia porcentual	89



LISTA DE FIGURAS

Figura 1. Arquitectura Tradicional de un ITS.....	11
Figura 2. Visión Esquemática de un Agente Inteligente (tomado de [29]).....	20
Figura 3. Modelo de Sistemas Multi-agente según INGENIAS.....	27
Figura 4. Arquitectura del prototipo de STI.	32
Figura 5. Modelo de la Red Bayesiana usada.....	33
Figura 6. Proceso de propagación de probabilidades en la Red Bayesiana.	38
Figura 7. Estructura del tutorial.....	40
Figura 8. Casos de uso correspondientes al actor Estudiante.....	52
Figura 9. Diagrama de Casos de uso correspondiente al actor Profesor.	54
Figura 10. Diagrama de Casos de uso correspondiente al actor Administrador.....	55
Figura 11. Modelo de interacciones: Identificar características del Estudiante.....	56
Figura 12. Modelo de interacciones: Actualizar Modelo del Estudiante.	56
Figura 13. Diagrama de Colaboración: Identificar Perfil del Estudiante.....	57
Figura 14. Diagrama de Colaboración: Actualizar Modelo del Estudiante.....	57
Figura 15. Modelo de la Organización.....	58
Figura 16. Modelo del entorno.....	59
Figura 17. Agente Tutor.....	60
Figura 18. Agente Concepto.....	61
Figura 19. Modelo de Tareas y Objetivos para el objetivo: Identificar las características del Estudiante.....	61
Figura 20. Modelo de Tareas y Objetivos para el objetivo: Orientar al Estudiante.....	62
Figura 21. Modelo de Tareas y Objetivos para el objetivo: Evaluar Progreso Estudiante.....	62
Figura 22. Modelo de Tareas y Objetivos para el objetivo: Actualizar Modelo del Estudiante.....	63
Figura 23. Diagrama de Clases del Sistema.....	64
Figura 24. Diagrama de Componentes.....	66
Figura 25. Estudiantes de noveno semestre del programa de Enfermería.....	72
Figura 26. Estructura del curso virtual: Estrategia AIEPI: Módulo Recién Nacido.....	73
Figura 27. Estudiantes de décimo semestre del programa de Medicina.....	73
Figura 28. Interfaz para el Profesor.....	76
Figura 29. Resultados Test Conocimiento Inicial.....	77
Figura 30. Resultados Test Conocimiento Final.....	77
Figura 31. Consolidado Calificaciones Test Conocimiento Final.....	78
Figura 32. Resultados categoría Calidad Pedagógica discriminados por indicador.....	79
Figura 33. Resultados categoría Usabilidad discriminados por indicador.....	79
Figura 34. Estilo de Aprendizaje.....	80
Figura 35. Resultados Test Conocimiento Inicial.....	82
Figura 36. Resultados Test Conocimiento Final.....	82
Figura 37. Consolidado Calificaciones Test Conocimiento Final.....	82
Figura 38. Tiempo empleado y tiempo estimado para el aprendizaje de los conceptos.....	83
Figura 39. Resultados para cada indicador de la categoría Calidad pedagógica.....	84
Figura 40. Resultados para cada indicador de la categoría Usabilidad.....	85
Figura 41. Resultados generales del nivel de conocimiento en Moodle y SIAS.....	86
Figura 42. Resultados calidad pedagógica en SIAS y Moodle discriminados por indicador.....	88
Figura 43. Resultados usabilidad en SIAS y Moodle discriminados por indicador.....	89



INTRODUCCIÓN

La Organización Mundial de la Salud (OMS) [1], en 1995, finalizó unas guías de reconocimiento y manejo de las enfermedades que más afectan a los niños en los países en desarrollo, las cuales llamó AIEPI (Atención Integrada a las Enfermedades Prevalentes de la Infancia; en inglés, IMCI) [2]. La implementación de la estrategia AIEPI contempla la participación tanto de los servicios de salud como de la comunidad y la familia, y se lleva a cabo por medio de dos componentes: el AIEPI Clínico y el AIEPI Comunitario. El AIEPI Clínico está dirigido a mejorar: (i) el desempeño del personal de salud para la prevención y el tratamiento de enfermedades en la niñez y (ii) el funcionamiento de la organización y los servicios de salud para brindar atención de calidad apropiada. El AIEPI Comunitario está dirigido a mejorar las prácticas familiares y comunitarias de cuidado y atención de la niñez.

El éxito de la estrategia AIEPI depende en gran medida del buen entrenamiento del personal de salud (pediatras, médicos generales, enfermeras, auxiliares de enfermería, médicos practicantes, promotores de salud, etc), además de la participación activa de las familias y la comunidad (padres de familia, profesores de guarderías y jardines infantiles, madres comunitarias, etc). Lo anterior, con el objetivo de detectar y tratar las enfermedades más prevalentes en la infancia y cómo prevenirlas, contribuyendo así a la mejora general de la atención en salud materno-perinatal y disminuyendo la mortalidad.

En la actualidad el uso de plataformas e-learning se ha incrementado en el proceso formativo de la comunidad estudiantil de diferentes disciplinas, incluyendo a los estudiantes de ciencias de la salud. A pesar de existir diferentes experiencias exitosas, se han encontrado muchos problemas. Por ejemplo, el estudio realizado por [3][4] ha identificado que los principales problemas de la educación virtual para estudiantes y profesionales de la salud están relacionados con la organización del material educativo, la usabilidad de los sistemas y la calidad pedagógica. Además, existe una resistencia a cambiar los métodos de enseñanza, por ejemplo, los estudiantes necesitan que los contenidos se presenten con diferentes estrategias de enseñanza acordes a su estilo de aprendizaje.

Para brindar enseñanza virtual efectiva en la estrategia AIEPI, es necesario que los contenidos estén debidamente organizados y se adapten fácilmente a las necesidades de los estudiantes, proporcionar material educativo en diferentes formatos y guiar a los estudiantes durante todo el proceso de aprendizaje. Sin embargo, los sistemas actuales de enseñanza virtual para el dominio de salud, no proporcionan un nivel adecuado de enseñanza personalizada y adaptable a las características de aprendizaje de los estudiantes.

En este sentido, los Sistemas Tutores Inteligentes-STI aprovechan las ventajas de los sistemas tradicionales de aprendizaje asistido por computador y se caracterizan por la inclusión de experiencia adicional relacionada con el entorno de aprendizaje del estudiante. Estos sistemas facilitan el proceso de aprendizaje por su capacidad de adaptación a las necesidades de los estudiantes y porque brindan instrucción adecuada en el momento apropiado según la evolución del estudiante.



Para lograr el desarrollo de STI más robustos, de forma más rápida y con menores costos, se hace uso de arquitecturas multi-agente [5]. El desarrollo del sistema propuesto bajo el paradigma de agentes, permite mayor flexibilidad en el tratamiento de cada uno de los componentes del sistema tutor. Cada componente se puede modelar con un agente o una división de cada módulo/componente en un conjunto de agentes que, funcionan como entidades, semi o completamente autónomas, las cuales se comunican entre sí y actúan racionalmente de acuerdo a sus percepciones del exterior y el estado de su conocimiento. En este sentido, el STI brinda soluciones de mayor calidad y permite a los estudiantes interactuar de una forma más eficiente.

En el marco del proyecto para el fortalecimiento de la estrategia AIEPI financiado por la Universidad Politécnica de Madrid desarrollado entre el año 2006-2008 [6], se trabajó en la implementación de talleres presenciales sobre la estrategia AIEPI comunitaria y clínica en los municipios de Silvia y Jambaló en el Departamento del Cauca. Este proyecto también consideraba la orientación de estos mismos cursos de forma virtual usando un sistema de gestión de contenidos denominado EVA (Entorno Virtual de Aprendizaje) [7], el cual es una adaptación de la plataforma .LRN. Por restricciones de tiempo y recursos, solamente algunos cursos fueron implementados en EVA. En el proceso de diseño de los cursos en forma electrónica se encontraron muchas dificultades debido a la carencia de personalización de los contenidos existentes, dado que todos los usuarios poseen sus propios perfiles y estilos de aprendizaje. Para contribuir a la solución de la problemática planteada anteriormente, se formula la siguiente pregunta de investigación: **¿es posible mejorar el proceso actual de enseñanza-aprendizaje en la estrategia AIEPI mediante el uso de Sistemas Tutores Inteligentes?**

El principal objetivo de este trabajo, es el diseño y evaluación de un prototipo de Sistema Tutor Inteligente, el cual tiene en cuenta los diferentes perfiles de los estudiantes de ciencias de la salud, ofreciéndoles alternativas pedagógicas altamente compatibles con sus características cognitivas y su evolución durante el curso. Con el fin de incrementar sus conocimientos y habilidades para la correcta aplicación de la estrategia AIEPI Clínico, en cada sesión de tutoría se proporciona información relevante sobre los problemas de salud y enfermedades que con más frecuencia afectan a la infancia. Es importante mencionar que el prototipo desarrollado puede ser extensible a otros dominios.

Para cumplir con el objetivo mencionado anteriormente, se han planteado los siguientes objetivos específicos:

- a. Diseñar modelos de agentes para la posterior implementación del prototipo de Sistema Tutor Inteligente propuesto.
- b. Desarrollar un prototipo de Sistema Tutor Inteligente que, usado para orientar cursos en el dominio de salud, contribuya al mejoramiento del proceso de enseñanza-aprendizaje.
- c. Definir y aplicar un esquema de evaluación basado en indicadores para determinar el aporte obtenido con el uso de Sistemas Tutores Inteligentes, frente a sistemas de enseñanza tradicional.

El trabajo que se presenta es de investigación aplicada, por lo tanto la metodología seleccionada debe estar apoyada sobre la base de métodos de investigación de ingeniería [8]. Para llevar a cabo la investigación se requiere seguir las siguientes fases:



Fase de información. Para el análisis del estado del arte se recurrirá a la investigación documental, se revisarán en detalle las iniciativas más importantes de desarrollo de Sistemas Tutores Inteligentes, metodologías y tecnologías asociadas. Para ello se llevará a cabo una búsqueda documental en libros y revistas especializadas, repositorios de centros de investigación accesibles por Internet, o contacto con expertos.

Fase Preposicional y de Análisis. Estudio y análisis previo del funcionamiento de algunos sistemas de enseñanza tradicionales y su desempeño. Formulación, análisis e implementación del Sistema Tutor Inteligente para el dominio de Salud. En esta fase se incluye el diseño de los componentes básicos del sistema, se integran metodologías de desarrollo y técnicas de Inteligencia Artificial.

Fase de evaluación. Evaluación de la aproximación por medio del método científico de la observación y la experimentación. El sistema desarrollado se evaluará en dos contextos: el primero, mediante la orientación de un curso virtual de capacitación usando la plataforma EVA; el segundo corresponde a la implantación y orientación del mismo curso usando el Sistema Tutor Inteligente desarrollado. En ambos escenarios se identificará un conjunto de indicadores para evaluar el nivel de conocimiento, velocidad de aprendizaje, estrategias de enseñanza utilizadas y la usabilidad de ambos sistemas.

En el presente documento se describen los conceptos teóricos y el trabajo realizado durante el desarrollo del proyecto. El documento se encuentra organizado en 9 capítulos como se explica a continuación:

En el primer capítulo se presenta el problema, se justifica la solución propuesta, se definen los objetivos y metodología.

En los capítulos 2, 3 y 4, se presenta la teoría, tecnologías y sistemas desarrollados relacionados con los objetivos del proyecto.

En los capítulos 5 y 6, se presenta el diseño de los componentes básicos del sistema, se describe la arquitectura del sistema y se presentan los artefactos obtenidos durante el proceso de desarrollo.

En el capítulo 7 se presenta la evaluación del sistema tutor inteligente desarrollado por medio del método científico de la observación y la experimentación. La evaluación se realiza en dos contextos: el primero, mediante la orientación de un curso virtual de capacitación usando la plataforma Moodle. El segundo corresponde a la implantación y orientación del mismo curso usando el Sistema Tutor Inteligente desarrollado SIAS. En ambos escenarios se identifican un conjunto de indicadores. Con los resultados obtenidos se determinan las ventajas y desventajas de cada una de las aproximaciones evaluadas. Por último se encuentran los resultados obtenidos y las conclusiones y trabajo futuro, en los capítulos 8 y 9 respectivamente.



1. ATENCIÓN INTEGRADA A LAS ENFERMEDADES PREVALENTES DE LA INFANCIA

1.1 ESTRATEGIA AIEPI

En los últimos años, los niños y niñas se han convertido en víctimas mortales de enfermedades prevenibles y curables como: enfermedades agudas de las vías respiratorias, enfermedades diarreicas, paludismo, sarampión, desnutrición, entre otras. Para ayudar a resolver este problema. La OMS, en 1995 finalizó unas guías de reconocimiento y manejo de las enfermedades que más afectan a los niños en los países en desarrollo, que llamó AIEPI (Atención Integrada a las Enfermedades Prevalentes de la Infancia; en inglés, IMCI) [2].

La estrategia AIEPI enfoca la atención en el estado de salud de los menores más que en las enfermedades que ocasionalmente puedan afectarlos; de este modo se incrementan las oportunidades de detección precoz y tratamiento de enfermedades que pueden pasar inadvertidas para los padres y para el personal de salud. Esta estrategia es considerada un instrumento útil para la detección precoz y tratamiento efectivo de las principales enfermedades que afectan la salud de los menores de 5 años [2]. Con la ejecución de esta estrategia por parte de los profesionales de salud y comunidad en general se puede reducir la gravedad de las enfermedades que afectan a los niños menores de 5 años y una disminución significativa en la cantidad de muertes de este grupo infantil.

La implementación de esta estrategia contribuye con la reducción del uso innecesario de tecnologías de diagnóstico y tratamiento, mejora la cobertura de las medidas de prevención, y promueve mejores prácticas de cuidado y atención en el hogar [2].

La estrategia AIEPI permite a las familias y comunidad en general, proceder adecuadamente en la atención de niños y niñas menores de 5 años, proporciona información para administrar correctamente el tratamiento indicado y promover la aplicación de medidas preventivas dentro del hogar.

Los objetivos que persigue la estrategia AIEPI principalmente son:

- Reducir la mortalidad en los niños menores de 5 años, especialmente la ocasionada por neumonía, diarrea, malaria, tuberculosis, dengue, meningitis, trastornos nutricionales y enfermedades prevenibles por vacunas, así como sus complicaciones.
- Reducir la incidencia y/o gravedad de los episodios de enfermedades infecciosas que afectan a los niños, especialmente neumonía, diarrea, parasitosis intestinales, meningitis, tuberculosis, malaria, y trastornos nutricionales.
- Garantizar una adecuada calidad de atención de los niños menores de 5 años tanto en los servicios de salud como en el hogar y en la comunidad.
- Fortalecer aspectos de promoción de la salud y prevención de la enfermedad de la niñez en la rutina de la atención de los servicios de salud y en la dinámica familiar y comunitaria.



- Apoyar los esfuerzos locales dirigidos a fortalecer actitudes, habilidades y destrezas en la familia y la comunidad para el cuidado de los niños menores de 5 años y la protección del ambiente.

1.2 COMPONENTES

La implementación de la estrategia AIEPI se lleva a cabo por medio de dos componentes: el AIEPI Clínico y el AIEPI Comunitario.

AIEPI Clínico

El AIEPI Clínico está dirigido a mejorar: (i) el desempeño del personal de salud para la prevención y el tratamiento de enfermedades en la niñez y (ii) el funcionamiento de la organización y los servicios de salud para que brinden atención de calidad apropiada.

El proceso de AIEPI puede ser utilizado por diferentes profesiones de la salud y describe cómo tratar de forma integrada las afecciones presentadas por niños entre 0 y 5 años; ofrece una guía para evaluar a un niño sistemáticamente por signos generales, proporciona el tratamiento y actividades para la prevención de enfermedades y promoción de estilos de vida saludables.

Los elementos del proceso AIEPI para tratar a los niños de 0 a 5 años incluye [2]:

- Evaluación: lo primero que se deben detectar son los signos de peligro, para lo cual se deben hacer preguntas comunes, examinar al niño, verificar la nutrición y estado de vacunación.
- Clasificación: Las enfermedades se clasifican por colores. Rojo, si requiere tratamiento y referencia urgente; Amarillo, si requiere tratamiento médico específico y consejería; verde, si requiere consejería sobre el tratamiento en la casa.
- Identificar Tratamiento: Después de clasificar la enfermedad se debe identificar el tratamiento adecuado. Si requiere referencia urgente, antes de transferir al niño se debe administrar un tratamiento esencial. Si requiere tratamiento en casa, se debe elaborar un plan integrado de tratamiento y suministrar la primera dosis de los medicamentos. En ambos casos se debe solicitar a la persona a cargo que regrese en una fecha específica para hacer seguimiento de la enfermedad.
- Tratamiento: Proporcionar instrucciones prácticas para suministrar el tratamiento, que incluyen cómo administrar medicamentos orales, alimentos y líquidos durante la enfermedad y tratamiento de infecciones locales en la casa.
- Evaluar alimentación y ofrecer consejería.
- Proporcionar atención de seguimiento: en la consulta de seguimiento se debe determinar si existen nuevos problemas.

El proceso de manejo de casos se presenta en dos series diferentes de Cuadros de Procedimientos. El primero dirigido al tratamiento de niños entre 0 y 2 meses de edad, el segundo dirigido al tratamiento de niños entre 2 meses y 5 años.

Cuadro de Procedimientos Niños de 0 a 2 meses de edad

Para el manejo de los niños menores de dos meses se describen los siguientes cuadros de procedimiento:

- Evaluar y determinar riesgo durante el embarazo que afecta el bienestar fetal



- Primera evaluación del recién nacido
- Evaluar y clasificar al niño de 0 a 2 meses de edad, los pasos a seguir son: (i) determinar si hay enfermedad muy grave o infección local, (ii) determinar si tiene diarrea, (iii) evaluar alimentación, (iv) evaluar otros problemas, (v) evaluar problemas de desarrollo y (vi) verificar estado de vacunación.
- Determinar el tratamiento
- Tratar al niño
- Aconsejar a la madre

Cuadro de Procedimientos Niños de 2 meses a 5 años de edad

Para el manejo de los niños entre dos meses y 5 años de edad se describen los siguientes cuadros de procedimiento:

- Evaluar y clasificar al niño de 2 meses a 5 años de edad, los pasos a seguir son:
 - (i) Verificar si hay signos generales de peligro,
 - (ii) Evaluar tos o dificultad respiratoria,
 - (iii) Determinar si se presenta diarrea, fiebre, dolor de oído, problemas de garganta, problemas de maltrato, desnutrición y anemia,
 - (iv) Evaluar otros problemas,
 - (v) Evaluar problema de desarrollo y
 - (vi) Verificar estado de vacunación
- Determinar el tratamiento
- Tratar al niño
- Aconsejar a la madre

AIEPI Comunitario

El AIEPI Comunitario está dirigido a mejorar las prácticas familiares y comunitarias de cuidado y atención de la niñez.

El manejo de casos solo es eficaz en la medida en que las familias llevan a sus hijos a un profesional de salud capacitado para ofrecer atención de manera oportuna e integral. Si una familia espera para traer a un niño a un consultorio hasta que esté sumamente enfermo, o concurren a un profesional de salud no capacitado, el niño tiene más probabilidades de morir por la enfermedad. Por consiguiente, enseñar a las familias cuándo buscar atención para un niño enfermo es una parte importante del proceso de manejo integrado de casos [2].



2. SISTEMAS TUTORES INTELIGENTES

2.1 DEFINICIÓN

VanLehn [9] define los sistemas tutores inteligentes como *“un sistema de software que utiliza técnicas de inteligencia artificial-IA para representar el conocimiento e interactúa con los estudiantes para enseñárselo”*.

Wolf [10] define los STI como: *“sistemas que modelan la enseñanza, el aprendizaje, la comunicación y el dominio del conocimiento del especialista y el entendimiento del estudiante sobre ese dominio”*. Giraffa [11] los delimita como: *“un sistema que incorpora técnicas de Inteligencia Artificial - IA a fin de crear un ambiente que considere los diversos estilos cognitivos de los alumnos que utilizan el programa”*.

Los STI son ambientes flexibles, interactivos y adaptativos para el aprendizaje. Flexibles porque abren abanicos de posibilidades para las navegaciones de los estudiantes. Interactivos porque los canales para la comunicación pueden permitir cruce de ideas. Adaptativos porque sobre la marcha el sistema puede cambiar las estrategias de enseñanza, variando los ejemplos, rompiendo las secuencias, demostrando imposibilidades y evidenciado equivocaciones a partir de casos concretos [12].

Un STI es dinámico, proactivo e inteligente similar a un docente en plena acción dentro de un salón de clase, a diferencia del software educativo tradicional que es pasivo como un libro, donde no existe una adaptación a las características del estudiante, no se varían los ejemplos y la secuencia es la misma. Si el estudiante no entiende, lo más conveniente sería que busque otro medio para cumplir su propósito. En cambio, un docente en su actuar debe alterar sus planes de forma inmediata cuando sus estudiantes no avanzan. Las funciones realizadas por el docente es lo que intenta simular un STI.

El software educativo tradicional no es adaptativo. Pese a que sus productos se basan en la multimedia, su esquema es pasivo y lineal. Por lo general se presenta una secuencia de lecciones y sus propuestas de evaluación no van más allá de presentar un consolidado de preguntas acertadas y equivocadas, dándoles una apreciación final de aprobación y desaprobación.

Los productos de los Sistemas Tutoriales Inteligentes son programas que se expresan también a través de la multimedia pero su diseño exige que se apliquen técnicas de inteligencia artificial. Su arquitectura les permite secuencias no lineales, los provee de la posibilidad de múltiples metodologías de enseñanza y los soporta con bases de conocimientos para la reelaboración de ejemplos cuando las circunstancias lo exigen; además de la planificación de diversas estrategias. Su evaluación es constructiva; por ejemplo, puede explicar a un estudiante por qué se equivocó, puede dar la oportunidad de repetir el ejercicio, pero con otros datos o en otro contexto; puede hacer construcciones a partir de las respuestas para demostrar que se está contradiciendo o negando un concepto [12].



2.2 PERSPECTIVA HISTÓRICA

Las aplicaciones educativas surgieron desde los primeros años de la década de los 60. Estas aplicaciones han incluido programación de cursos, instrucción asistida por computador, realización de pruebas, simulación de modelos y procesos, así como desarrollo de tutoriales mediante el uso de lenguajes de programación o lenguajes y sistemas de autoría [13]. Los primeros desarrollos en la rama de la Instrucción Asistida por Computador-IAC [14] han presentado el material de estudio, intentando motivar y controlar el aprendizaje de una materia mediante el desarrollo de programas de instrucción. Desarrollos posteriores de la IAC han incorporado el concepto de individualización de la instrucción, en el que el material y la ejercitación de los cursos se adaptan a las necesidades, intereses y nivel de conocimiento del estudiante.

2.2.1 Evolución de los sistemas de enseñanza asistida por computador

Programación Lineal

La programación lineal aparece alrededor de los años 50. Estos sistemas están basados en la teoría conductista y se caracterizan por presentar el conocimiento en forma lineal, donde ningún factor puede alterar el orden de enseñanza establecido por el programador. Se encuentran conformados por la salida del programa, entrada del estudiante y la reacción del sistema. Estos sistemas no proporcionan enseñanza individualizada, siendo esta su principal limitación [15].

Programas Ramificados

Los programas ramificados a diferencia de los programas lineales, presentan los temas de acuerdo a las necesidades de los estudiantes, utilizan la técnica de correspondencia de patrones, para tratar las respuestas de los estudiantes como aceptables o parcialmente aceptables. A pesar de las mejoras presentadas, no ofrecen enseñanza individualizada, el sistema no depende del estudiante, sino de las respuestas [13].

Sistemas Generativos

A los primeros usos de las técnicas de Inteligencia Artificial en IAC se les llamó IAC GENERATIVA, pues los sistemas desarrollados tenían capacidad para generar problemas desde una gran base de datos que contenía la materia de estudio, construir soluciones, diagnosticar las respuestas del estudiante y controlar la complejidad de los problemas presentados. Sin embargo, estos sistemas construyen una única solución para un problema que puede tener N soluciones [16].

Las desventajas presentadas por los sistemas de instrucción asistida por computador son:

- La interacción entre el tutor y el estudiante es restringida
- Son estáticos y no ofrecen flexibilidad.
- La respuesta del sistema no tiene en cuenta las características específicas de cada estudiante.
- Después de construidos, el conocimiento no se modifica con el paso del tiempo.

Sistemas Tutores Inteligentes

En los años 70 se empieza a hablar y a desarrollar Sistemas Tutoriales Inteligentes - STI y es Carbonell [17] quien define este segundo tipo de Instrucción Asistida por Computador que es conocido como IAC basada en el conocimiento o IAC Inteligente. Pero los STI



señalados por Carbonell, eran más que simples generadores de problemas, debían ofrecer lo que Brown [18] llama entorno de aprendizaje reactivo, en el cual el programa de instrucción se encarga del entrenamiento del estudiante y el diálogo tutorial orienta la instrucción según los conocimientos, intereses y conceptos erróneos del estudiante. Koffman y Blount [19] y otros investigadores señalan además que un STI debe ser capaz de deducir hipótesis basadas en la historia de errores del estudiante para encontrar cuál es la causa real de sus dificultades.

Para el desarrollo de los STI, los investigadores han empleado técnicas de Inteligencia Artificial sobre la comprensión del lenguaje natural, representación del conocimiento, métodos de inferencia y técnicas de aprendizaje, además de algunas aplicaciones específicas como simplificación algebraica, integración simbólica, diagnóstico médico, solución de problemas y prueba de teoremas. Nuevos desarrollos de los STI, se caracterizan por la inclusión de experiencia adicional relacionada con el entorno de aprendizaje del estudiante y métodos y técnicas de enseñanza. Con estas nuevas características se han desarrollado sistemas más flexibles, adaptados a los intereses del estudiante y con métodos pedagógicos que facilitan el proceso de aprendizaje [16].

2.2.2 Ambientes virtuales de aprendizaje

Los entornos virtuales de aprendizaje, son herramientas utilizadas con el objetivo de facilitar y mejorar los procesos de aprendizaje en entornos organizacionales como las universidades [20].

Los ambientes virtuales proporcionan métodos de enseñanza flexibles, sin restricciones de tiempo y de espacio, donde el estudiante es responsable de su aprendizaje. También incorporan nuevas formas de interacción entre el docente y sus estudiantes.

Según López, Escalera y Ledesma [21] los elementos que conforman un ambiente virtual de aprendizaje son: usuarios, contenido a enseñar, especialistas y, un sistema de administración, que permite hacer seguimiento al aprendizaje del estudiante.

Generalmente, los ambientes virtuales proporcionan: plantillas para generar contenidos, servicios de información, comunicación y servicios para control de acceso y seguimiento de grupos.

Actualmente existen diferentes ambientes virtuales, a continuación se listan algunos ejemplos:

- Moodle: sistema de gestión de cursos, sirve como herramienta de soporte para los docentes para crear comunidades de aprendizaje en línea. Se caracteriza por tener una curva de aprendizaje elemental. Sin embargo, uno de los principales inconvenientes es el tamaño de su API [22].
- Entorno Virtual de Aprendizaje - EVA: sistema de gestión de contenidos que sirve de soporte al desarrollo de actividades de enseñanza-aprendizaje en modalidades presencial, semi-presencial, y a distancia. Proporciona diferentes servicios de información, comunicación y seguimiento. Se caracteriza por su robustez y escalabilidad, pero presenta curvas de aprendizaje lentas [7].



- Entorno Virtual de Aprendizaje Interactivo – EVAI: conjunto heterogéneo de recursos que comparten el soporte digital y la tecnología de Internet de modo sinérgico para posibilitar un nuevo modo de interacción humana orientado a la exploración y el aprendizaje [23].

Además de los ambientes mencionados están: Claroline [24], eCollege [25], Blackboard [26], Dokeos [27], entre otros.

2.3 ARQUITECTURA DE LOS SISTEMAS TUTORES INTELIGENTES

Para definir la especificación básica de un STI es necesario considerar ciertas preguntas clave: ¿qué enseñar?, ¿cómo enseñar? y ¿a quién enseñar? Generalmente, cada una de estas áreas de conocimiento se almacena y mantiene en módulos diferentes, dotando los STI de modularidad para posteriores reutilizaciones en otros STI.

La arquitectura básica de un STI consiste de un **módulo experto**, un **módulo del estudiante** y un **módulo tutor**, que operan de forma interactiva y se comunican a través de un módulo central que se suele denominar **módulo de interfaz** o del entorno [28].

El módulo experto contiene el conocimiento específico y detallado de acerca de la materia que se pretende enseñar obtenido de los expertos humanos, el módulo del estudiante guarda toda la información relativa al estudiante que se genera durante la interacción con el sistema y el módulo pedagógico controla los planes y decisiones pedagógicas. Finalmente, el módulo de interfaz gestiona la interacción de las otras componentes del sistema y controla el interfaz hombre-máquina. La Figura 1 representa la arquitectura básica de los STI.

2.3.1 Módulo Experto

El módulo experto provee los conocimientos del dominio. En primer lugar, se encuentra la presentación de la materia de una manera adecuada, es decir, la generación de explicaciones, preguntas, tareas y respuestas que permitan que el estudiante adquiriera las habilidades y conceptos. En segundo lugar, el módulo debe ser capaz de encontrar la solución de los problemas generados, realizar correcciones a soluciones erradas que se presenten, y explicar su razonamiento en un lenguaje comprensible para el estudiante.

En el momento de diseñar un módulo experto es necesario considerar el tipo de conocimiento que se está modelando. El conocimiento se puede dividir en tres tipos: El conocimiento procedural, el conocimiento declarativo y el conocimiento cualitativo [29].

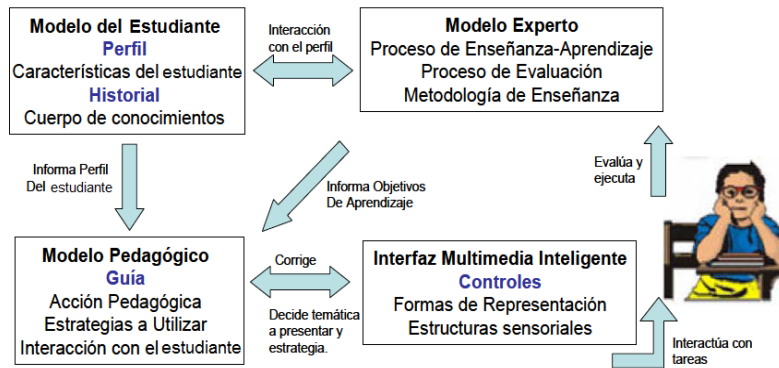


Figura 1. Arquitectura Tradicional de un ITS

El conocimiento procedural o de procedimientos (“*saber cómo*”) es conocimiento acerca de cómo llevar a cabo cierta tarea y por tanto suele ser específico de cada dominio en particular.

El conocimiento declarativo (“*saber qué*”) es un conjunto de hechos que se organizan de forma adecuada para razonar sobre ellos. Un ejemplo de dominio de conocimiento declarativo es el sistema presentado por Carbonell, en el los nodos representan hechos y los enlaces representan relaciones jerárquicas [17].

El tercer tipo, el conocimiento cualitativo, es quizás el más difícil de modelar. Se usa para modelar relaciones espaciales y procesos dinámicos. Dentro de este tipo se encuentra el razonamiento causal que resulta de gran importancia en los sistemas de diagnóstico de averías. El razonamiento sobre la estructura causal de un dispositivo se usa para determinar potenciales problemas. Las Redes Bayesianas parecen la estructura más adecuada para modelar este tipo de conocimiento [30].

En el trabajo presentado por Anderson [31] se agrupan los modelos expertos en tres categorías: (i) los modelos de caja negra, (ii) los modelos de caja de cristal y (iii) los modelos cognitivos. Los modelos expertos de caja negra son capaces de resolver problemas sobre el dominio. Las soluciones a dichos problemas se usan como ejemplo para los estudiantes y para determinar si las soluciones presentadas por éstos son o no correctas [31]. El modelo de caja negra es, como su nombre indica, totalmente opaco para el estudiante.

Un modelo más transparente al usuario es el llamado modelo de caja de cristal. En este modelo, cada paso en el razonamiento puede ser revisado e interpretado. Para construir un modelo de caja de cristal, se debe utilizar la misma metodología que la usada en un sistema experto. El experto humano en el dominio y el ingeniero de conocimiento trabajan juntos para definir el espacio, identificar y formalizar los conceptos claves, diseñar un sistema en el cual implementar el conocimiento y probar y refinar este sistema. La lección más importante que podemos sacar del desarrollo de este sistema es que al construir el módulo experto no sólo debemos pensar en el conocimiento de la materia, sino también en la forma en la que lo vamos a representar.

Por último, tenemos los llamados modelos cognitivos que simulan cómo usa un humano el conocimiento que queremos enseñar [32]. El objetivo de este modelo es descomponer el conocimiento en componentes plenas de significado, y usar este conocimiento de una forma similar a la humana. Este tipo de modelo experto puede comunicarse con un



estudiante de una forma mucho más extensa que los modelos anteriores. Sin embargo, la construcción de modelos cognitivos es un proceso muy complicado y que consume mucho tiempo. Además se plantea la necesidad de determinar qué componentes psicológicas son esenciales para modelar el aprendizaje y cuáles pueden ser sacrificadas a cambio de una menor complejidad computacional.

2.3.2 Módulo del Estudiante

El módulo del estudiante en un STI representa el conocimiento que tiene el estudiante del dominio a enseñarle. Las acciones del estudiante son interpretadas en un intento de reconstruir el estado de conocimiento que le llevó a realizar esas acciones. Esta información puede entonces compararse con la contenida en el módulo experto, y ser usada para tomar decisiones didácticas que guiarán al estudiante y organizarán sus actividades. El modelo del estudiante es una representación cualitativa aproximada, posiblemente parcial, del conocimiento del estudiante sobre cierto dominio. Este modelo puede explicar total o parcialmente aspectos específicos del comportamiento del estudiante. Decir que el modelo del estudiante es una representación cualitativa significa que no es ni numérica ni física, sino que describe los objetos y los procesos en términos de relaciones espaciales, temporales y causales. Decir que el modelo del estudiante es aproximado y posiblemente parcial significa que nos interesa más la utilidad computacional que la fidelidad cognitiva [33].

El modelado del estudiante es el elemento central en el diseño y desarrollo de un STI, ya que constituye la inteligencia del sistema tutor. Las cuestiones fundamentales en el modelado del estudiante son [34]: (i) selección de la estructura, (ii) inicialización y (iii) diagnóstico.

- **Selección de la estructura:** que se usará para representar el modelo del estudiante. Esta información puede almacenarse de muchas formas distintas: en un vector, en una red semántica, en una red bayesiana, en forma de afirmaciones, etc.

- **Inicialización del modelo del estudiante:** la estructura elegida para representar el conocimiento del estudiante debe inicializarse cuando la interacción con el sistema comienza. Para ello disponemos de varias opciones: (i) utilizar la información disponible acerca del estudiante, (ii) pedirle que se clasifique en clases o estereotipos de estudiantes previamente definidos o (iii) realizar tests previos, etc.

En la actualidad hay varias aproximaciones para inicializar el modelo del estudiante [35]. Partiendo de que cuando un estudiante utiliza por primera vez el ambiente de aprendizaje, su modelo asociado está vacío, algunos sistemas asumen que el nuevo estudiante no tiene ningún conocimiento acerca del dominio a aprender [36]. Aunque este método es la forma más fácil de inicializar el modelo del estudiante, presenta bajo rendimiento en los estudiantes debido a las diferencias de conocimiento y a las características personalizadas. Otra forma de inicialización, involucra el diseño de una prueba previa para evaluar el conocimiento del nuevo estudiante. En cierta forma, el resultado de la prueba puede proporcionar una referencia acerca de los conocimientos del estudiante, pero carece de conocimientos concretos en la parte de evaluación e ignora los efectos de las características de los estudiantes sobre el aprendizaje.

- **Diagnóstico:** una vez que el modelo del estudiante se ha inicializado comienza la interacción con el sistema. El procedimiento de diagnóstico elegido actualizará el modelo



del estudiante tras sus interacciones con el sistema. Esta actualización utiliza dos fuentes de información: (i) el modelo del estudiante actual y (ii) su comportamiento en el proceso interactivo de enseñanza. El comportamiento del estudiante puede medirse en función de distintas variables que es preciso definir previamente: (i) soluciones a problemas, (ii) respuestas a preguntas y (iii) tiempo empleado en lectura de pantallas.

El modelo del estudiante puede ser utilizado para diferentes propósitos: (i) Para determinar si el estudiante está preparado para continuar con el siguiente tema del currículum, y para elegir este tema. (ii) Para generar explicaciones que el estudiante pueda entender (al nivel de detalle adecuado a sus conocimientos actuales). (iii) Para ofrecer consejos y ayudas sin que el estudiante lo solicite. En este sentido, es importante que el tutor no interrumpa a los estudiantes con demasiada frecuencia, y que les permita aprender de sus errores. (iv) Para generar problemas al nivel adecuado. La generación dinámica de problemas es otra área que se apoya fuertemente en el modelo del estudiante. (v) Para seleccionar la estrategia de enseñanza más apropiada dado el nivel de conocimiento actual.

Una vez identificados los puntos débiles del estudiante, se genera un problema que el módulo experto resuelve paralelamente para ser capaz de diagnosticar la solución del estudiante. Así, a cada estudiante que interactúe con el sistema se le presentará una colección diferente de problemas, adecuada a su nivel de conocimiento. Básicamente, los tipos de modelo del estudiante que se han utilizado son [40]:

- *Modelo de superposición (overlay model).*
- *Modelo diferencial (differential model).*
- *Modelo de perturbación (perturbation model).*
- *Modelo basado en restricciones.* Este modelo es una modificación del modelo de superposición propuesto en [41] e implementado con éxito en el tutor de SQL de Mitrovic [42].

2.3.3 Módulo del Tutor

El módulo tutor (denominado también planeador instruccional) controla las actividades e interacción instructora de forma similar a un tutor humano. Diferentes niveles de control determinan diferentes estrategias [29]: en el nivel de control máximo (*monitoring*) el sistema adapta las acciones a las necesidades del estudiante, llevando siempre el control. En el nivel intermedio (*mixedinitiative dialogue*) el estudiante y el sistema comparten el control mediante el intercambio de preguntas y respuestas. En el otro extremo tenemos las actividades autorizadas (*guided-discovery learning*), en las que la intervención del sistema se reduce a modificar el entorno.

Al elegir entre estas estrategias de control cada dominio y cada estudiante deben ser evaluados de forma independiente. De este modo, las estrategias de control pueden guardarse en el módulo de instrucción y ser seleccionadas de forma que el tipo de instrucción que reciba el estudiante sea individualizada. Los dos tutores más frecuentes son los tutores *expositivos* y los tutores de *procedimientos*. Cada uno de ellos se asocia con el tipo de dominio que se pretende enseñar. En los tutores expositivos se transmite conocimiento sobre hechos, y las inferencias que se realizan van encaminadas a mantener el enfoque y la coherencia.

Los tutores de procedimientos enseñan habilidades y procedimientos, y tienen la tarea adicional de ordenar las habilidades que componen la habilidad principal que se desea que el estudiante adquiera y de elegir los ejercicios y ejemplos.



La evolución de las técnicas de planeación instruccional ha sido paralela a los avances de las técnicas de resolución de problemas en la AI [43]. Después de plantearse varias aproximaciones, Giraffa [44] convierte el planeador instruccional en una herramienta flexible con capacidad de adaptar los conocimientos con un determinado grado de abstracción dependiendo de los conocimientos del estudiante [45]. Los sistemas tutores desarrollados bajo este modelo, cimientan su modelo instruccional en los paradigmas conductistas y cognitivistas. Este tipo de ambiente trata de convertir al estudiante en un experto del dominio que se está tratando. Para lograrlo realizan un plan que es constantemente modificado de acuerdo a los avances logrados o a las dificultades presentadas. Su propósito es el de tratar de simular un docente en su labor de enseñanza resaltando la capacidad de planear los objetivos a lograr en cada etapa o sesión de aprendizaje, detectar vacíos de conocimiento en el estudiante y utilizar estrategias de enseñanza eficaces oportunas.

El estudiante juega un papel activo debido a que está en continua interacción con el sistema tratando de aclarar dudas y reforzando conocimientos. Este tipo de aprendizaje es lo que explica el conductismo, es decir, el aprendizaje de procedimientos, donde se proponen unos objetivos de adquisición del conocimiento en forma secuencial para que el estudiante logre en determinadas sesiones de aprendizaje o clases [46].

Cuando un estudiante explora el ambiente de un sistema tutor va adquiriendo unos conocimientos de un dominio específico los cuales modifican el conocimiento que ya tenían almacenado en sus estructuras mentales. Los STI son herramientas que motivan al estudiante para que continúe su búsqueda de más conocimiento, con el propósito de que los adquiera, los organice y los adhiera. Este avance se da de acuerdo a su ritmo de aprendizaje. De esta forma los estudiantes se convierten en constructores de su propio conocimiento, consiguiendo un aprendizaje más efectivo, porque se convierten en partícipes activos en su proceso formativo [47].

Elorriaga y Fernández [48], proponen un modelo instruccional que integra cinco niveles:

- Unidades básicas de aprendizaje (*Basic UnSTI of Learnin*-BULs): Corresponde a la secuencia que representa el conjunto de elementos dentro de los cuáles el conocimiento del dominio es organizado.
- Objetivos instruccionales (*Instructional Objectives*-IOs): Corresponden a las habilidades cognitivas, y al conjunto de habilidades que el tutor planea para que sean adquiridas por el estudiante durante la sesión de aprendizaje.
- Procesos cognitivos: El conjunto de actividades mentales que suceden en el estudiante.
- Eventos instruccionales: Condiciones externas que deben ocurrir para que el aprendizaje se lleva a cabo.
- Acciones instruccionales: Corresponden a las acciones que el estudiante o el tutor llevan a cabo durante las sesiones de aprendizaje.

2.3.4 Módulo del Entorno

El módulo del entorno especifica y da soporte a las actividades del estudiante y a los métodos que se usan para realizar dichas actividades. Los entornos deben ser intuitivos y atractivos, de forma que el estudiante centre toda su atención en el proceso de aprendizaje de la materia y no en aprender a utilizar el entorno. Burton define [49] los seis aspectos clave en el diseño del entorno:

- a) *aspectos del dominio* que se desean representar;
- b) *nivel de abstracción* de la representación;



- c) *fidelidad* de la representación;
- d) *orden en la presentación* de contenidos;
- e) *herramientas* de corrección y ayuda y
- f) *nivel de control* que ejercerá la herramienta.

Aún cuando actualmente se cuenta con herramientas mucho más potentes para el diseño de interfaz, estas seis características siguen considerándose básicas a la hora de planificar su desarrollo.

Los STI se caracterizan por representar separadamente los componentes más importantes en módulos. Por tanto, al igual que cualquier software que se comunica con usuarios, la interfaz de comunicación corresponde con un módulo bien planificado, de fácil manipulación, y que favorece el proceso de comunicación tutor-estudiante.

Artículos de Freedman y Rosenking [50] y Kearsley y Seidel [51] han concluido que la nueva generación de STI debe incorporar conceptos nuevos sobre adquisición, representación y manejo del conocimiento del experto, resolución de problemas de diferentes formas, explicación de su razonamiento y adquisición de conocimiento en cuanto a estrategias de enseñanza a partir de la interacción estudiante-sistema.

2.4 SISTEMAS TUTORES INTELIGENTES DESARROLLADOS PARA EL DOMINIO DE SALUD

En los últimos años se han desarrollado sistemas tutoriales como complemento para la educación en el área de la salud. Estos sistemas han sido planteados desde diferentes enfoques pedagógicos. A continuación se presenta una descripción de los diferentes STI desarrollados en el dominio de salud:

CIRCSIM-Tutor [52], desarrollado para apoyar la enseñanza a los estudiantes de medicina. En la versión 2, asiste a los estudiantes en el proceso de aprendizaje y les ayuda en la resolución de problemas relacionados con aspectos del dominio médico, específicamente problemas cardiovasculares. El sistema utiliza diálogos para liderar el aprendizaje de los estudiantes, proceso de diagnóstico, actualización del modelo del estudiante y decidir cada paso a seguir en el proceso de aprendizaje

En la versión 3, los objetivos del sistema son: (i) que el estudiante, utilizando este sistema, adquiera de forma cualitativa un modelo causal del sistema cardiovascular y (ii) que los estudiantes aprendan métodos de resolución de problemas que les permita dar solución a cualquier problema en el dominio [53].

La principal desventaja de CIRCSIM es que tiene predefinida una única solución para cada problema presentado. A diferencia de CIRCSIM-Tutor, el STI propuesto se basa en una arquitectura multi-agente proporcionando ventajas sobre la arquitectura tradicional de los STI tales como, el desarrollo de sistemas tutores más robustos, de forma más rápida y con menores costos.

COMET (Collaborative Medical Tutor) [54], Sistema Tutor Inteligente para aprendizaje médico basado en la estrategia de resolución de problemas (Medical problem-based learning - PBL). El objetivo de este sistema, es lograr que los estudiantes resuelvan los problemas de forma colaborativa. La Interfaz de usuario incluye una interfaz multi-modal,



utiliza redes bayesianas para modelar el conocimiento individual del estudiante y de un grupo pequeño de estudiantes. También, aplica algoritmos genéricos y genera tutoriales de alerta para guiar en el proceso de resolución de problemas. COMET ha sido probado empíricamente comparando los tutoriales de alerta dados por el sistema y los dados por un tutor humano. Además, se comparó la efectividad del sistema en la enseñanza de conocimiento médico y habilidades de razonamiento clínico con un tutor humano. Sin embargo, no se han hecho evaluaciones formales sobre la interfaz con el estudiante.

A diferencia de COMET, el sistema tutor propuesto presenta los contenidos de una forma adecuada de acuerdo al nivel de conocimiento y estilo de aprendizaje de cada estudiante, logrando enseñanza individualizada y adaptada a las habilidades de los estudiantes.

Sistema Tutor para apoyar jóvenes con Diabetes Mellitus tipo 1 a convivir con este disturbio metabólico [56], sistema que pretende difundir conocimientos a las personas para que puedan mantener o mejorar su calidad de vida. La investigación que enmarca el desarrollo de éste sistema, intenta establecer métodos de diseño para STI, que adapten la presentación del contenido según las características de los usuarios del sistema. Un aspecto importante en el desarrollo del sistema, es la adaptación cultural al ambiente en que será usado.

En éste trabajo el contenido o módulo del experto se divide en unidades, tópicos y nodos. Dentro del modelo del estudiante se consideran características como: estilos de aprendizaje, tipos de inteligencia, etapas de desarrollo, nivel socio-económico, identificación socio-cultural, tiempo de convivencia con la diabetes, entre otras. Sin embargo, todavía se está investigando la forma adecuada para determinar las características de los estudiantes y cómo mejorar la capacidad de representar la información y conocimientos dentro del sistema. Para implementar el módulo del tutor se propone usar redes neuronales de tipo Interactive Activation and Competition - IAC, utilizar un Sistema Especialista - SE para analizar los cuestionarios y actividades de comunicación con los estudiantes.

A diferencia de este sistema, el STI propuesto está definido para capacitación en la estrategia AIEPI. El sistema propuesto hace uso de tests previos para determinar el nivel de conocimiento inicial del estudiante. Con la información obtenida en el test, el sistema procede a inicializar el modelo del estudiante y presentar el contenido adecuado de acuerdo a su perfil.

Sistema tutorial para el estudio de la fisiología del aparato respiratorio del cuerpo humano [57], éste sistema constituye un medio de enseñanza interactivo donde los estudiantes podrán ampliar sus conocimientos ya sea de forma individual o en forma colectiva. El sistema posee una base de datos, la cual ha sido elaborada con base al conocimiento de expertos en el tema objeto de estudio. Además, para comprobar la calidad del sistema, se han realizado dos tipos de pruebas. La primera, una prueba de expertos realizada con los profesores de los departamentos de Computación y Fisiología y, la segunda, una prueba experimental aplicada a una muestra de 16 estudiantes de cuarto semestre de la carrera de Medicina en el laboratorio de Computación. Los resultados obtenidos en ambos escenarios fueron satisfactorios y los estudiantes concluyeron que el sistema cuenta con las características necesarias para facilitar el alcance de los objetivos de la materia.

La diferencia entre el sistema anteriormente mencionado y el sistema tutor propuesto radica en la implementación de cada uno de los módulos. El sistema propuesto, utiliza



agentes inteligentes para implementar el módulo del tutor y utiliza redes bayesianas [58] para representar el modelo del estudiante. Otra diferencia se encuentra en la forma de evaluar el sistema. La evaluación del sistema propuesto, se basa en indicadores y la comparación con sistemas de enseñanza tradicional como moodle. Contrario a esto, el sistema tutor para fisiología, ha sido utilizado por usuarios con diferente nivel de conocimiento, pero no se han realizado evaluaciones formales donde se demuestren sus beneficios comparados con otros sistemas de aprendizaje.

Experto en Anatomía [16], tutor que tiene como área de estudio el sistema cardiovascular. Éste sistema utiliza las redes semánticas para estructurar el conocimiento y los métodos de enseñanza. El módulo del tutor se compone de sesiones y cada sesión consta de cuatro etapas: inicio, pre-diagnóstico, enseñanza y terminación. El módulo interfaz es el encargado de las siguientes tareas: presentación del sistema, captura de datos, presentación de pre diagnóstico y evaluación, presentación del material de enseñanza y, finalmente, se encarga de la interpretación de las respuestas del estudiante y de interpretar las preguntas que se hagan al sistema durante la sesión de tutoría.

Los principales inconvenientes que presenta este tutor son: No hay suficientes criterios para evaluar al estudiante y no se definen métodos específicos para presentar información de acuerdo al tipo de estudiante. También, presenta dificultades en el análisis sintáctico y semántico de cada frase que compone las preguntas de los estudiantes. A diferencia de este sistema, en el sistema tutor propuesto se han identificado cuatro estilos de aprendizaje. Para cada estilo de aprendizaje se han seleccionado estrategias diferentes de enseñanza de acuerdo a las características propias de cada estilo. Lo anterior con el objetivo de presentar los contenidos con la estrategia de enseñanza adecuada.

Sistema Tutor Inteligente basado en lenguaje natural para enseñanza en patología [59], este sistema se enfoca en la generación de reportes de diagnóstico, para lo cual los autores proponen implementar una interfaz basada en lenguaje natural (Natural Language Interfaz- NLI). En éste trabajo se intenta probar la efectividad de la realimentación en el tiempo justo dentro del sistema. Para lo cual, se desarrollaron dos tipos de interfaz: (i) interfaz inmediata que proporciona a los estudiantes realimentación positiva y negativa después de cada acción, (ii) interfaz retardada, en la cual el estudiante recibe realimentación solo cuando ha terminado el reporte.

Los reportes escritos por los estudiantes fueron calificados por personas externas y fueron comparados con el criterio dado por un experto en el dominio (cáncer de piel). Después de la calificación, se realizaron encuestas a los estudiantes para establecer el nivel de satisfacción de usuario y así determinar cuál interfaz preferían los estudiantes.

La diferencia entre el sistema anteriormente mencionado y el sistema propuesto radica en la forma de evaluación. Para el sistema propuesto al finalizar el tutorial los estudiantes podrán evaluar la calidad pedagógica y la usabilidad del sistema. Además, es importante mencionar que el sistema propuesto presenta una interfaz web, monitorea el desempeño del estudiante en el desarrollo de las actividades y le proporciona diferentes consejos.

TELEOS (Computer learning systems in orthopedic surgery) [60], sistema implementado con una arquitectura multi-agente, donde cada agente se construye según las restricciones de conocimiento, por lo anterior, se desarrollaron diferentes tipos de agentes relacionados con los tipos de conocimiento. El sistema cuenta con: un simulador para preparar al estudiante cuando se enfrente a una cirugía real, un agente caso clínico



para proveer retro-alimentación pedagógica durante la cirugía y un módulo web semántico que permite re-direccionar al estudiante a las partes relevantes del curso.

Los agentes se modelan con redes bayesianas. El modelo de diagnóstico, se centra en las acciones y eventos de los usuarios. El objetivo de los agentes asociados al modelo de retro-alimentación didáctica es proponer la mejor decisión en términos del proceso de aprendizaje.

Al igual que este sistema el sistema propuesto hace uso de una arquitectura multi-agente, y realiza el proceso de diagnóstico de conocimiento del estudiante usando redes bayesianas. Sin embargo, el sistema propuesto cambia las estrategias de enseñanza según la evolución del estudiante en el tutorial.

Direne y otros autores [61] presentan los resultados obtenidos de un estudio empírico realizado con estudiantes de radiología, para identificar las características presentes en una sesión de tutoría persona-persona. En dicho estudio se logran identificar quince (15) indicadores que son analizados y parte de ellos implementados en el sistema **RUI (an Intelligent Tutoring System for multiple domains of radiological expertise)**. Actualmente, la interfaz de RUI no permite que los estudiantes realicen preguntas. Sin embargo, esta deficiencia se compensa con la incorporación de técnicas de diagnóstico diferencial.

El sistema RUI está definido específicamente para el estudio de radiología. En este sentido, no se pueden incluir las características identificadas en el estudio realizado por Direne, para implementar el sistema propuesto porque el área objeto de estudio es diferente. Además, los indicadores identificados se encuentran estrechamente ligados al caso de estudio y para nuestra propuesta, es necesario definir indicadores que permitan extender el prototipo desarrollado a diferentes cursos del dominio médico.

LAHYSTOTRAIN (intelligent training system for laparoscopy and hysteroscopy) [62], sistema de entrenamiento que combina técnicas de realidad virtual, tecnología multimedia y STI, que se encuentra dividido en dos escenarios:

El sistema de entrenamiento básico que consiste en un sistema de entrenamiento basado en Web, dirigido a estudiantes novatos y se enfoca en la teoría y aspectos conceptuales de laparoscopia e histeroscopia. La arquitectura del tutor inteligente, detecta errores superficiales e infiere su causa.

El STI incluye: el *Asistente*, el *tutor* y cuenta con tres agentes inteligentes que representan a un cirujano auxiliar, una enfermera y un anestesiólogo. El primer prototipo, ha sido validado por miembros de la Unión Europea en diferentes entornos de salud.

A diferencia del sistema anteriormente mencionado, el sistema tutor propuesto utiliza estrategias de enseñanzas acordes a las características de cada estudiante.



3. AGENTES INTELIGENTES

3.1 DEFINICIÓN DE AGENTE

El concepto *agente* ha sido definido por varios investigadores, pero en la actualidad, no existe una definición exacta que sea aceptada por toda la comunidad científica. A continuación, se presentan algunas de éstas definiciones:

“Los agentes realizan tres funciones: percepción de las condiciones dinámicas de un entorno, acción que afecta a dichas condiciones y razonamiento (para interpretar percepciones y determinar acciones a ejecutar) [63].

“Un agente es un sistema computacional que habita en un entorno dinámico complejo. El agente puede sentir y actuar sobre el entorno, y tiene un conjunto de objetivos que intenta alcanzar a través de estas acciones” [64].

“Un agente es cualquier entidad que percibe su entorno a través de sensores y actúa sobre ese entorno mediante efectores. Un agente es racional cuando realiza la mejor acción posible a partir de los datos percibidos” [65].

La definición más aceptada por la comunidad científica es la propuesta por Wooldridge y Jennings: “Cualquier proceso computacional dirigido por el objetivo capaz de interactuar con su entorno de forma flexible y robusta” [5]. En esta definición la flexibilidad se considera en términos de tres características: reactividad, pro-actividad y sociabilidad.

3.2 CARACTERÍSTICAS DE LOS AGENTES

Todos los sistemas software no son agentes; por lo tanto, se deben establecer propiedades específicas que permitan distinguir un agente de un software convencional. A continuación, se describen las características particulares de los agentes inteligentes:

- **Autonomía:** se refiere a la capacidad que tiene el agente para actuar sin intervención humana directa o de otros agentes, controlar sus acciones y su estado interno.
- **Sociabilidad:** se refiere a la capacidad de interacción con otras entidades (agentes, usuarios), utilizando algún mecanismo de comunicación.
- **Reactividad:** se refiere a la capacidad de un agente para percibir estímulos del entorno que habita y reacciona en un tiempo determinado.
- **Iniciativa ó Pro-actividad:** se refiere a la capacidad que tiene un agente para actuar guiado por los objetivos que debe cumplir.
- **Movilidad:** se refiere a la habilidad que tiene un agente para trasladarse en una red de comunicación informática.
- **Veracidad:** el agente no comunica información falsa intencionadamente.
- **Benevolencia:** el agente no tiene objetivos contradictorios y siempre intenta realizar la tarea que se le solicita.

- **Racionalidad:** el agente tiene unos objetivos específicos y siempre intenta llevarlos a cabo.

Según Russell [66], la racionalidad de un agente depende de cuatro factores:

1. Medida con la que se evalúa el grado de éxito.
2. Secuencia de percepciones
3. Conocimiento que el agente tiene del medio
4. Acciones que el agente puede llevar a cabo

Considerando los factores anteriores, un *agente racional ideal* deberá emprender todas aquellas acciones que favorezcan obtener el máximo de su medida de rendimiento, basándose en las evidencias aportadas por la secuencia de percepciones y en todo conocimiento incorporado en tal agente. En la definición anterior, se considera un elemento adicional, el conocimiento integrado del agente. Si el agente actúa basado solamente en su conocimiento integrado, se dice que éste no es autónomo. En la Figura 2 se presenta el esquema de un agente inteligente.

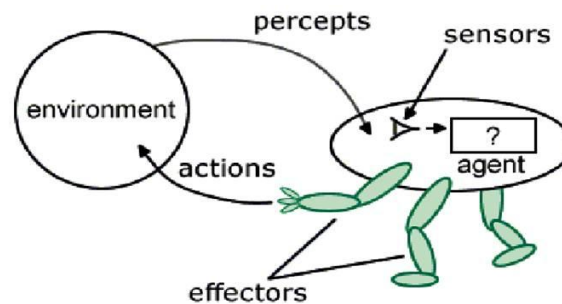


Figura 2. Visión Esquemática de un Agente Inteligente (tomado de [29])

Según Wooldridge [5], de todas las características mencionadas anteriormente, se consideran como las más relevantes, la autonomía, sociabilidad, reactividad e iniciativa.

3.3 CLASIFICACIÓN DE LOS AGENTES

Los agentes inteligentes se pueden dividir en:

- Agentes Biológicos
- Agentes Hardware
- Agentes Software

Según Hong [67], los agentes software se pueden clasificar en cuatro dimensiones: cuantitativa, movilidad, sensibilidad y comportamiento.

En la dimensión cuantitativa se encuentran los agentes individuales y sistemas multi-agente. Los sistemas multi-agente se comunican con el usuario, con recursos y con otros agentes, para resolver problemas de manera más eficiente.

En la dimensión movilidad se encuentran los agentes estáticos y los agentes móviles. Los agentes móviles tienen la capacidad de migrar en su red local o acceder a redes remotas.



En la dimensión sensibilidad, la clasificación se hace de acuerdo a la capacidad para resolver problemas. En este grupo se encuentran los agentes deliberativos y reactivos. Los agentes deliberativos contienen un modelo simbólico del mundo, representado internamente de forma explícita y toman decisiones mediante razonamiento lógico. Los agentes reactivos responden a un evento o estímulo producido dentro del entorno.

En la dimensión comportamiento, los agentes se pueden clasificar con base a la autonomía, aprendizaje y colaboración. En este grupo se encuentran los agentes colaborativos y de interfaz. Los agentes colaborativos, son útiles para desarrollar tareas de resolución de problemas. El objetivo de los agentes de interfaz es satisfacer las necesidades del usuario con base a sus hábitos y comportamientos.

Otras clasificaciones presentan los siguientes tipos de agentes [68]:

- Agentes de información: agentes encargados de administrar, recolectar y clasificar información de diferentes fuentes que se encuentran distribuidas, estos agentes pueden ser estáticos o móviles, pueden aprender o no del entorno, pero no son sociales.
- Agentes Híbridos: agentes que surgen de la combinación de agentes reactivos y agentes deliberativos.

3.4 SISTEMAS MULTI-AGENTE – SMA

Un sistema *multi-agente* es un sistema conformado por un conjunto de agentes autónomos que trabajan conjuntamente para resolver un problema [69].

Un *sistema multi-agente* es una red poco acoplada de entidades capaces de solucionar problemas que están más allá de la capacidad y el conocimiento individual de cada entidad [70].

Según Chávez [71], los SMA se caracterizan por lo siguiente:

- Cada agente tiene información incompleta para solucionar un problema.
- No hay un sistema global de control, el control está distribuido
- Los datos se encuentran descentralizados
- La computación es asíncrona

En un SMA están activos diferentes agentes autónomos independientes [72]. En este sentido, los SMA se convierten en una solución a la restricción que presenta un agente individual para resolver un problema complejo. Usar un sólo agente para resolver ésta clase de problemas no es adecuado, porque el agente puede encontrar una solución que no es útil ó incluso nunca encontrarla. En cambio, en un SMA los agentes trabajan conjuntamente para encontrar la solución adecuada al problema.

3.4.1 Interacción entre Agentes

En un SMA es necesario establecer mecanismos de interacción entre agentes. Estos mecanismos sirven para que un agente pueda prever las acciones de otros agentes en las tareas propias de planificación y, en cómo puede influir en las acciones de otros agentes



para el beneficio de sus objetivos. Existen tres tipos de interacción: Indiferencia, Cooperación y Antagonismo.

El proceso de interacción entre agentes puede llevar a dos situaciones. La primera denomina cooperación entre agentes (Cooperative Multiagent Systems-CMAS), se da cuando un agente no puede encontrar la solución a un problema y necesita de otros agentes para obtener la solución. La segunda denominada Competición, se da cuando se tienen dos agentes con objetivos diferentes que quieren lograr sus metas satisfactoriamente (Self Interested Multiagent Systems-SMAS), cuando se presenta ésta situación es necesario definir estrategias de negociación.

Para que se lleve a cabo el proceso de interacción debe existir un patrón de comunicación. El protocolo de interacción entre agentes – AIP [73], describe el patrón de comunicación como una secuencia permitida de mensajes entre agentes y las restricciones en el contenido de dichos mensajes.

3.4.2 Coordinación entre Agentes

Un SMA puede estar conformado por cualquier cantidad de agentes. Cuando el número de agentes del sistema crece, aparece la necesidad de ayuda para localizar otros agentes que tengan la información o que nos puedan ofrecer los servicios que se requieren [74]. Para mantener la coordinación dentro de un SMA se pueden clasificar los agentes en tres categorías: agentes proveedores ó P-agentes, agentes requeridores ó R-agentes y agentes mediadores ó intermediarios. Los agentes intermediarios, son los encargados de proporcionar determinados servicios y asistencia a los agentes del sistema.

3.4.3 Comunicación entre Agentes

Un agente necesita conocer el medio que los rodea, para lo cual, debe interactuar con otros agentes y con su ambiente. Para lograr la interacción se deben establecer reglas y un canal de comunicación común.

Los requerimientos de comunicación se pueden agrupar en los siguientes componentes:

- Componentes de representación: se encargan de resolver problemas de entendimiento mutuo. Se incluyen las ontologías y la base del conocimiento.
- Componentes de comunicación: comprende el protocolo de transporte, lenguaje de comunicación y el protocolo de interacción. El protocolo de transporte establece el mecanismo de transmisión de mensajes entre emisor y receptor. El lenguaje de comunicación permite la interoperabilidad entre agentes y le da un significado común al mensaje intercambiado. El protocolo de interacción describe la secuencia de intercambio de mensajes entre dos agentes.

La forma cómo se comunican los agentes depende de la forma en que se organizan. A continuación se presentan los diferentes métodos de comunicación:

- **Sistema de Pizarra:** La pizarra es una estructura de datos que es usada como mecanismo general de comunicación entre las múltiples fuentes de conocimiento y es gestionada y arbitrada por un controlador [63]. La pizarra es un espacio compartido por todos los agentes, donde los agentes pueden poner información que estará disponible a todos los agentes del sistema. Este método no utiliza comunicación directa entre agentes, si un agente necesita información debe consultar obligatoriamente la pizarra. En investigaciones posteriores, se incluye un nuevo



elemento, el *moderador*, que se encarga de supervisar el estado del problema, los problemas que se deben resolver después y trata de organizar el trabajo entre y hacia los agentes.

- **Sistemas de Mensaje/Dialogo:** En éste método la comunicación se da entre un agente emisor y un agente receptor en forma directa. Generalmente, el mensaje está dirigido a un único receptor, sin embargo se puede usar broadcasting para enviar el mismo mensaje a diferentes agentes. Para establecer comunicación entre agentes y definir mecanismos de cooperación, es necesario contar con un protocolo de interacción bien definido.

3.4.4 Protocolos de Comunicación

Los protocolos de comunicación definen las reglas que permiten la comunicación entre agentes y están definidos por niveles:

- Nivel inferior: especifica el método de interconexión
- Nivel Medio: especifica el formato o sintaxis de la información
- Nivel Superior: especifica el significado o semántica de la información.

Especificaciones FIPA

La fundación para agentes Foundation for Intelligent Physical Agents-FIPA [75] ha definido diferentes especificaciones que sirven como soporte para el desarrollo sistemas con tecnología de agentes.

Las especificaciones FIPA se pueden organizar de acuerdo al dominio de agentes al que se dirige:

- Aplicaciones
- Arquitecturas Abstractas
- Comunicación entre agentes
- Gestión de agentes
- Transporte de mensajes a los agentes

Las especificaciones para comunicación entre agentes están organizadas en protocolos de interacción, actos comunicativos y lenguajes de contenido. El protocolo de interacción se utiliza para el intercambio de mensajes, los actos comunicativos se basan en la teoría de los actos del habla y para representar el contenido del mensaje en diferentes lenguajes, FIPA propone especificaciones como: Knowledge Interchange Format – KIF [76], Resource Description Framework –RDF, entre otros.

Especificaciones Knowledge Query Manipulation Language-KQML

La fundación para agentes Knowledge Sharing Effort- KSE [77], ha propuesto el lenguaje de comunicación Knowledge Query and Manipulation Language – KQML [78], que se encuentra dividido en:

1. Nivel de Mensaje: en este nivel se determina los tipos de interacciones que se puedan presentar en una comunicación.

2. Nivel de Contenido: en este nivel KSE propone usar un formato de intercambio de conocimiento. KIF es una notación del cálculo de predicados de primer orden extendido para aumentar su expresividad. Adicionalmente al formato, es necesario usar una ontología para permitir que el contenido de un mensaje se pueda interpretar sin ambigüedad e independientemente del contexto.



3. Nivel de Comunicación: contiene información sobre el transporte de los datos y se establecen parámetros a bajo nivel como quién es el emisor, el receptor, el protocolo de transmisión, etc.

Las características más relevantes de KQML son:

- Los mensajes KQML no solo comunican oraciones en un lenguaje, también comunican una actitud acerca del contenido.
- Las performativas indican las acciones permitidas que los agentes pueden usar cuando se comunican
- El entorno de comunicación puede mejorarse con los agentes facilitadores o intermediarios.

Toda implementación de KQML debe tener en cuenta las performativas estándar, las cuales están divididas en tres grupos:

- Performativas de discurso: se utilizan en el intercambio de información y conocimiento entre agentes
- Performativas de intervención y mecánica de la conversación: intervienen en el curso normal de una conversación.
- Performativas de red y de facilitación: permiten a los agentes encontrar otros agentes capaces de procesar sus mensajes, pero no son actos del habla.

3.4.5 Negociación entre Agentes

En un SMA se debe tener en cuenta un aspecto muy importante denominado *Negociación*. La negociación se considera como un proceso de comunicación, enlazada con el objetivo de cumplir un contrato aceptado por un grupo de agentes determinado [79].

El proceso de negociación debe cumplir con las siguientes características:

- Eficiencia: para llegar a un acuerdo los agentes no deben gastar recursos.
- Estabilidad: no deben existir incentivos para que los agentes se aparten de las estrategias de acuerdo.
- Simplicidad: los mecanismos no deben imponer altas demandas computacionales y de ancho de banda a los agentes.
- Distribución: el mecanismo no requiere de un sistema de distribución central.
- Simetría: el mecanismo no debe tener ningún prejuicio sobre un agente.

Dentro de un SMA se puede presentar negociación estricta ó negociación generalizada. En la negociación estricta, la negociación gira alrededor de un plan común y la ejecución estricta del plan. En la negociación generalizada, se presenta un entrelazamiento entre negociación y ejecución.

Las cuatro posibles formas de actuar de un agente en el proceso de son:

- Cooperación simétrica: La negociación favorece al agente con mejores resultados que los que hubiera obtenido de forma individual.
- Compromiso simétrico: Los agentes logran el objetivo de manera independiente y llegan a un compromiso, en el cual la calidad de los resultados disminuye.
- Cooperación/Compromiso no simétrico: Un agente obtiene mejor resultado que el otro agente.
- Conflicto: No se puede llegar a un acuerdo razonable.



3.5 SISTEMAS TUTORES INTELIGENTES DESARROLLADOS CON TECNOLOGÍA DE AGENTES

Los principios de sistemas multi-agentes se han mostrado como una solución en el desarrollo de sistemas de enseñanza, porque el problema de enseñanza-aprendizaje se puede resolver más fácil de forma cooperativa. Los ambientes de enseñanza basados en arquitecturas multi-agente posibilitan el desarrollo de sistemas tutores más robustos, de forma más rápida y con menores costos[29]. Los módulos que conforman un STI pueden estar distribuidos y subdivididos en partes más pequeñas, funcionando como entidades, semi o completamente autónomas, las cuales se comunican entre sí y actúan racionalmente de acuerdo a sus percepciones del exterior y el estado de su conocimiento.

Numerosos sistemas han sido planteados y desarrollados basados en tecnologías de agentes. A continuación se hace una breve reseña de los mismos:

AME-A [80] Sistema que cuenta con un agente que modela las características afectivas y de conocimiento de los estudiantes, para proporcionar un aprendizaje estático. En este sistema el modelo del estudiante es validado para establecer un conjunto de estrategias pedagógicas.

MASCARET [81], modelo que propone organizar las interacciones entre agentes, dotar a los agentes con habilidades reactivas, cognitivas y sociales para simular un entorno físico y social. El entorno social es simulado por agentes que ejecutan tareas de forma colaborativa. El modelo organizacional se fundamenta en los conceptos de organización, roles, características de comportamiento y agentes. Además, incluye agentes pedagógicos representados en un avatar para desempeñar un rol y obtener un comportamiento pedagógico.

I-Help [82] El sistema está basado en una arquitectura multi-agente, integrada por agentes personales (usuarios, humanos) y agentes de aplicación. Estos agentes usan una ontología y un lenguaje de comunicación común. Todos los agentes son autónomos. Cada agente posee un modelo de usuario de otros agentes y se comunican por medio de combinadores para encontrar recursos apropiados para sus usuarios, dependiendo del tipo de ayuda requerida.

STI-TB [83], sistema para educación en tuberculosis. El objetivo principal del sistema es mejorar el proceso de toma de decisiones y está dirigido a los estudiantes de pre-grado y profesionales en medicina. El STI está basado en arquitectura multi-agentes y adopta el razonamiento basado en casos como paradigma de aprendizaje. Uno de los inconvenientes que presenta este sistema es que la validación se hizo con estudiantes simulados, nunca se probó en un entorno real con usuarios reales.

Electrotutor [84], aproximación que implementa un ambiente distribuido de enseñanza aprendizaje inteligente (*Intelligent Learning Environment, ILE*) basado en una arquitectura multi-agente, compuesta por siete agentes. Con el objetivo de actuar sobre el ambiente, cada uno de los agentes posee una representación interna parcial del mundo que los rodea. Para esto se emplea una metáfora de estados mentales para modelar la base de conocimientos que representa los estados del ambiente donde el agente está inmerso.

Grunspenkis [85], ha presentado una aproximación para el desarrollo de un *framework* conceptual para un sistema tutor inteligente. El sistema tutor está basado en el paradigma



de agentes. Este *framework* ha sido implementado únicamente para los prototipos del estudiante y modelos expertos. Se requiere mayor trabajo para la implementación de más modelos y agentes en el sistema y su evaluación en la práctica.

STIM-Tutor [86], sistema inteligente para educación médica, desarrollado bajo el enfoque multi-agente. Los módulos del STI son representados con agentes autónomos y cada uno implementa una fase del ciclo de razonamiento basado en casos. Los objetivos que pretende alcanzar éste sistema son: ayudar al estudiante a encontrar ejercicios apropiados según su nivel de conocimiento, combinar el aprendizaje únicamente textual con casos prácticos y asistir al estudiante durante el proceso de resolución de problemas y toma de decisiones.

González [87], ha propuesto una integración de arquitecturas entre sistemas de salud que usan HL7 y sistemas tutores inteligentes, con el objetivo de compartir los datos clínicos de un paciente. En éste caso se ha integrado el sistema SINCO-TB con el sistema STIM-Tutor.

Antonio y otros autores [88], han propuesto una arquitectura para el desarrollo de ambientes virtuales basada en una colección de agentes software cooperativos. Los agentes que conforman la arquitectura son: agente planificador, agente tutor, agente planificador de camino, agente experto, agente mundo, agente percepción, agente simulación, agente comunicación y el agente estudiante. Actualmente se está aplicando una versión mejorado a dispositivos de percepción táctil.

Linquin, Tao, Yining [89], han propuesto un *framework* basado en una arquitectura multi-agente, que combina sistemas de entrenamiento virtual y sistemas tutores inteligentes; los agentes cooperan con otros agentes utilizando el mecanismo de paso de mensajes. El objetivo del sistema de entrenamiento virtual inteligente es entrenar a los estudiantes de emergencias en un entorno optativo para mejorar su capacidad para tomar decisiones. Éste sistema cuenta con seis agentes colaborativos: agente tutor, agente simulación, agente experto, agente estudiante, agente administrador de recursos y agente interfaz inteligente.

WITS [90], STI basado en Web, desarrollado con tecnología de agentes. El sistema se compone de dos entornos de aprendizaje: entorno colaborativo y entorno de aprendizaje personalizado. El entorno de aprendizaje colaborativo se compone de cinco partes: aprendizaje temático, chat, correo electrónico, FAQ y foro de discusión. El entorno de aprendizaje personalizado es la parte más importante del sistema, está conformado de cuatro agentes inteligentes. Los cuatro agentes son: agente administrador curso, agente interfaz, agente consultor, agente retroalimentación. El prototipo del sistema ha sido usado para dictar el curso de introducción a la informática en la universidad Southwest en China y su efectividad se ha validado practicando encuestas a los estudiantes del curso.

SITUA [91], sistema de tutorización basado en agentes que permite la edición de tutoriales vía web. Este sistema utiliza redes bayesianas para representar el modelo del estudiante, lo que facilita el proceso de diagnóstico sobre la evolución del estudiante en el tutorial. El modulo del tutor se implementa como un sistema multi-agente que permite monitorizar los progresos del estudiante y evaluar los conocimientos que el estudiante va adquiriendo en cada parte del tutorial. Los tutoriales son creados por expertos en el dominio y se estructuran en documentos XML.

SITUA ha sido desarrollado con la idea de ser el punto de partida para la evolución de lo que puede llegar a ser un STI completo, siguiendo la estructura básica de un STI y utilizando tecnologías en continuo proceso de investigación. Teniendo en cuenta que este sistema no cumple con la totalidad de las características de un STI, se deben mejorar aspectos relacionados con la adaptabilidad del módulo de diagnóstico del estudiante, implementando mecanismos que determinen las experiencias previas en el dominio del tutorial para construir y adaptar la estructura de la red bayesiana. Además, de identificar el estilo de aprendizaje de cada estudiante para presentar el contenido de acuerdo a las preferencias de los estudiantes.

3.6 METODOLOGÍA INGENIAS PARA EL DESARROLLO DE SISTEMAS MULTI-AGENTE

El desarrollo de sistemas complejos distribuidos basados en tecnologías multi-agente requiere el uso de alguna metodología durante las fases de análisis y diseño [92]. La principal ventaja de su utilización es que el conocimiento adquirido al desarrollar aplicaciones se podrá reutilizar o adaptar a nuevos proyectos. A través del tiempo se han realizado diversas investigaciones y se han planteado diferentes metodologías para el desarrollo de Sistemas Multi-agente-SMA, encontrando así muchas dificultades y deficiencias en las mismas. En el ANEXO A se describen algunas metodologías de desarrollo de SMA.

La metodología INGENIAS concibe el SMA como la representación computacional de un conjunto de modelos. Cada uno de estos modelos muestra una visión parcial del sistema: los agentes que lo componen, las interacciones que existen entre ellos, cómo se organizan para proporcionar la funcionalidad del sistema, qué información es relevante en el dominio y cómo es el entorno en el que se ubica el sistema a desarrollar. Para especificar cómo tienen que interactuar los modelos mencionados, se definen meta-modelos. Un meta-modelo es una representación de los tipos de entidades que pueden existir en un modelo, sus relaciones y restricciones de aplicación. Los meta-modelos que se describen son una evolución del trabajo realizado en MESSAGE [93]. El resultado obtenido son cinco meta-modelos que giran alrededor de dos entidades la *organización* y el *agente* y que son presentados en la Figura 3.

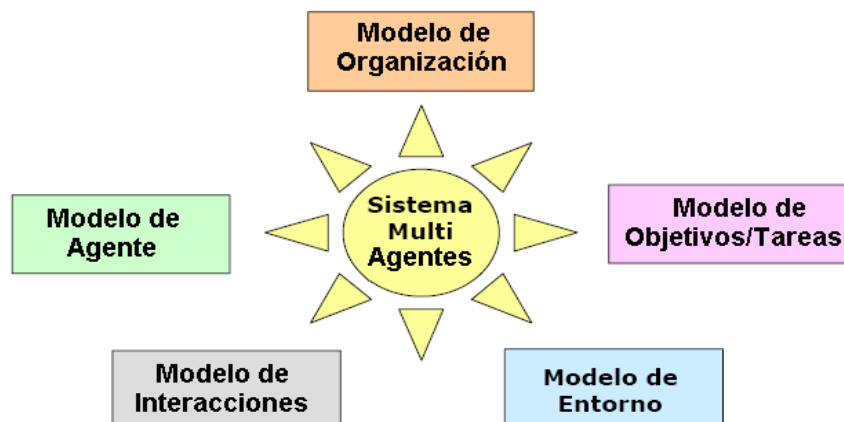


Figura 3. Modelo de Sistemas Multi-agente según INGENIAS



Modelo de agente: Describe agentes particulares y los estados mentales en que se encontrarán a lo largo de su vida. El meta-modelo de agente se usa para describir agentes particulares excluyendo las interacciones con otros agentes. Este meta-modelo se centra en la funcionalidad del agente y en el diseño de su control.

Modelo de tareas y objetivos: Se usa para asociar el estado mental del agente con las tareas que ejecuta. Los objetivos son los estados que un agente quiere lograr o evitar, las tareas son las transiciones de estados y conducen al logro de los objetivos.

Modelo de la organización: Define cómo se agrupan los agentes, la funcionalidad del sistema y qué restricciones hay que imponer sobre el comportamiento de los agentes. El modelo de organización es el equivalente a la arquitectura del sistema en un SMA.

Modelo de interacción: Detalla cómo se coordinan y comunican los agentes. Las interacciones determinan el comportamiento de los agentes mostrando cuál es su reacción cuando actúan sobre ellos. Y cómo el comportamiento va a ser función de los objetivos de los agentes y las tareas a ejecutar, se puede concluir que existe un importante vínculo entre interacciones, objetivos y tareas.

Modelo de entorno: Define qué existe alrededor del nuevo sistema y cómo lo percibe cada agente.

En la tabla 1 se describen las fases de INGENIAS. Es importante mencionar que para el desarrollo de este proyecto se incluyó la fase de transición y los flujos de trabajo: implementación, pruebas y despliegue. Esto fue necesario para desarrollar la totalidad del prototipo software.

FASES				
	Inicio	Elaboración	Construcción	Transición
Análisis	Generar Casos de uso Generar modelo de organización Generar modelo del entorno	Refinar Casos de uso Generar modelos de agentes Refinar modelo de organización Generar modelo de tareas y objetivos Refinar modelo del entorno	Refinar casos de uso	
Diseño	Generar un prototipo	Centrar el modelo de la organización en flujos de trabajo Llevar restricciones al modelo de tareas y objetivos Expresar la ejecución de tareas dentro de modelos de interacción Generar modelos de	Generar nuevos modelos de agente o refinar los existentes. Depurar la organización centrando el desarrollo en las relaciones sociales	



		agente para detallar patrones de estado mental		
Implementación		Realizar diagrama de componentes	Se desarrolla el sistema a partir de la especificación del diseño. Se refina el diagrama de componentes.	Se elaboran el manual de usuario. Se elabora el manual de instalación.
Pruebas		Se diseña el plan de pruebas	Se realizan las pruebas del sistema con el fin de garantizar la corrección del desarrollo.	
Despliegue				Se completa y refina el manual de usuario e instalación. Se realiza la implantación en el servidor de la aplicación.

Tabla 1. Fases del proceso de desarrollo.

En INGENIAS, los problemas presentados con la metodología MASCommonKADS se evitan, utilizando meta-modelos como mecanismo de especificación. El desarrollo de meta-modelos está soportado por herramientas que permite el procesamiento automático de los modelos generados. Los meta-modelos en algunos casos profundizan más en el detalle que MAS-CommonKADS. Tal es el caso del meta-modelo de organización, el de tareas y objetivos o el de agentes.

Los principios de la metodología BDI (Beliefs, Desires, Intentions) se hallan presentes en INGENIAS de diferentes formas. Los planes, entendidos como secuenciación de tareas, aparecen como flujos de trabajo. Los objetivos han sido introducidos explícitamente, incluyendo su refinamiento en subobjetivos, el establecimiento de dependencias entre objetivos [94] y dependencias con tareas. Su uso se ha extendido a otros modelos como el de organización y el de interacciones, vertebrando la ejecución de tareas e iniciación de interacciones a lo largo de toda la metodología. Por lo tanto, el grado de integración del modelo BDI en INGENIAS es superior al logrado en la propuesta de [95] y todo ello dentro de un ciclo de vida de software complejo, como es el Proceso Unificado (*Unified Process, UP*) [96][97]. Además, se ha logrado la independencia total de implementación dotando a la metodología de un procedimiento para parametrizar con los modelos generados armazones software escritos en cualquier lenguaje.

El proceso de desarrollo seguido por INGENIAS es similar al seguido por MaSe pero mediante otras herramientas, como Rational Rose, Together o Paradigm+. En estas herramientas, el usuario tiene libertad para elaborar diagramas incompletos o generar sus propias vistas del sistema, tomando elementos de diferentes modelos e incorporándolos a un nuevo diagrama. También se intenta separar de UML en el sentido de no repetir las mismas soluciones. La forma en que se tratan las interacciones en INGENIAS es un buen ejemplo de ello. Las interacciones buscan generalizar diferentes alternativas existentes,



diagramas de secuencia o colaboración [96] y diagramas de protocolo AUML [98]. Aparte de la generalización, las interacciones se integran completamente dentro de otros modelos como un elemento más. De hecho, la iniciación de interacciones se representa como el producto de ejecutar una tarea.

Otro aspecto tenido en cuenta en INGENIAS es el proceso de generación de los modelos, esto es, qué actividades están involucradas en su producción. El resultado es un conjunto estructurado de actividades detalladas que guiarán a futuros usuarios de la metodología en su utilización. Las actividades se enmarcan en diferentes flujos de trabajo para cada modelo, con lo que se facilita una futura integración de la metodología en entornos automatizados de gestión de proyectos software.

Frente al cuestionamiento sobre si al desarrollar sistemas se necesita tener una buena herramienta o una buena metodología, la postura de INGENIAS es que las herramientas de soporte no tienen que condicionar la metodología y que de hecho han de ser independientes. La independencia se consigue usando meta-modelos como elemento de construcción. Ello facilita el portar la metodología a diferentes herramientas, por lo tanto cualquier herramienta que soporte metamodelado podría servir como herramienta soporte de desarrollo.

Como solución genérica, para resolver los problemas de complejidad de implementación usando GAIA, la metodología INGENIAS proporciona un procedimiento con el que se facilita el salto a la implementación del sistema. La implementación se plantea como la parametrización de un armazón software utilizando los modelos que componen la especificación.

En INGENIAS para evitar obviar las distintas dependencias entre los modelos propuestos como ocurre con GAIA y MaSE, las dependencias entre los distintos meta-modelos se revisan independientemente. La mayoría se refiere a que al introducir ciertas entidades, hay que definir aspectos adicionales en otros modelos. Por ejemplo, al crear un objetivo, siempre hay que asociar una tarea o tareas que permiten alcanzarlo. Si el objetivo se identifica dentro de un modelo de agente, entonces, se necesita que en un modelo de objetivos y tareas se indique qué tarea o tareas deben ejecutarse.

A diferencia de GAIA, en INGENIAS el meta-modelo de organización se ve respecto del SMA como el equivalente a la arquitectura del sistema de un sistema convencional. Sirve para definir a alto nivel cómo se organizan los elementos del sistema para hacer posible los objetivos comunes a los agentes que participan en la organización.

INGENIAS es una evolución del trabajo de MESSAGE por lo que se plantea el desarrollo a lo largo del ciclo de vida del software de los modelos generados. El paso de una etapa a otra está marcado por el nivel de detalle alcanzado en cada modelo. Así, las interacciones inicialmente pueden detallarse con diagramas de colaboración para luego concretarse en el diseño con otros tipos de diagramas que alcancen más detalle.

Para resolver los problemas de complejidad de implementación usando GAIA, INGENIAS plantea la implementación como la parametrización de un armazón software utilizando los modelos que componen la especificación. Además, para evitar obviar las distintas dependencias entre los modelos propuestos como ocurre con GAIA y MaSE, las dependencias entre los distintos metamodelos se revisan independientemente. Finalmente, a diferencia de MESSAGE donde no se tenía en cuenta lo que rodeaba la aplicación, INGENIAS incluye un meta-modelo de entorno donde la percepción de los agentes se expresa en función de estos elementos.



Para elegir una metodología, es necesario tener en cuenta las áreas en las que se especializa cada metodología. Si se está acostumbrado a trabajar con sistemas basados en conocimiento, la mejor opción sería la metodología MASCommonKADS. Si la experiencia del usuario está en el área de los objetos, la recomendación sería de MaSE. Si por el contrario está interesado en un enfoque más orientado a agentes, puede seleccionar ZEUS, INGENIAS, BDI o GAIA. Y si lo que se quiere es tener un soporte de herramientas: ZEUS, MaSE, INGENIAS. En los casos en que se requiera un proceso de desarrollo robusto, detallado y ensayado en desarrollos reales, la recomendación sería MASCommonKADS o INGENIAS.

En el presente proyecto se plantea trabajar con la metodología INGENIAS [99], la cual fue utilizada en el proyecto “Contribuciones al diseño de sistemas tutores inteligentes usando razonamiento basado en casos” [29] con resultado exitoso. Las principales razones por las que se elige esta metodología son: (i) se integra fácilmente al proceso de desarrollo UP, (ii) los meta-modelos permiten representar todos los aspectos de un SMA y cada meta-modelo se encarga de representar aspectos específicos del sistema, (iii) permite describir agentes particulares y considera el estado mental de cada agente como algo dinámico que evoluciona con el tiempo, por lo tanto se puede incluir el aprendizaje entre las capacidades del agente, (iv) usar meta-modelos hace que la metodología sea independiente de las herramientas de soporte, lo cual facilita el portar la metodología en diferentes herramientas y (v) la metodología INGENIAS ha mejorado e integrado muchos de los resultados de investigaciones anteriores.

La descripción de la metodología INGENIAS y otras metodologías para el desarrollo de sistemas multi-agente se puede ver en detalle en el ANEXO A.

4. ARQUITECTURA DEL SISTEMA

Este capítulo presenta la descripción detallada de la arquitectura propuesta. Esta arquitectura modela los cuatro (4) componentes fundamentales que integran los Sistemas Tutores Inteligentes (descritos en el Capítulo 3), y provee características de adaptación a los sistemas desarrollados bajo la arquitectura presentada.

Los sistemas tutoriales inteligentes se dividen en cuatro partes o módulos fundamentales:

- Módulo del Estudiante
- Módulo del Experto
- Módulo del Tutor
- Módulo del Entorno

En la Figura 4 se muestra gráficamente la distribución de cada uno de los módulos en la arquitectura del prototipo de STI y la manera en que estos interactúan entre sí.

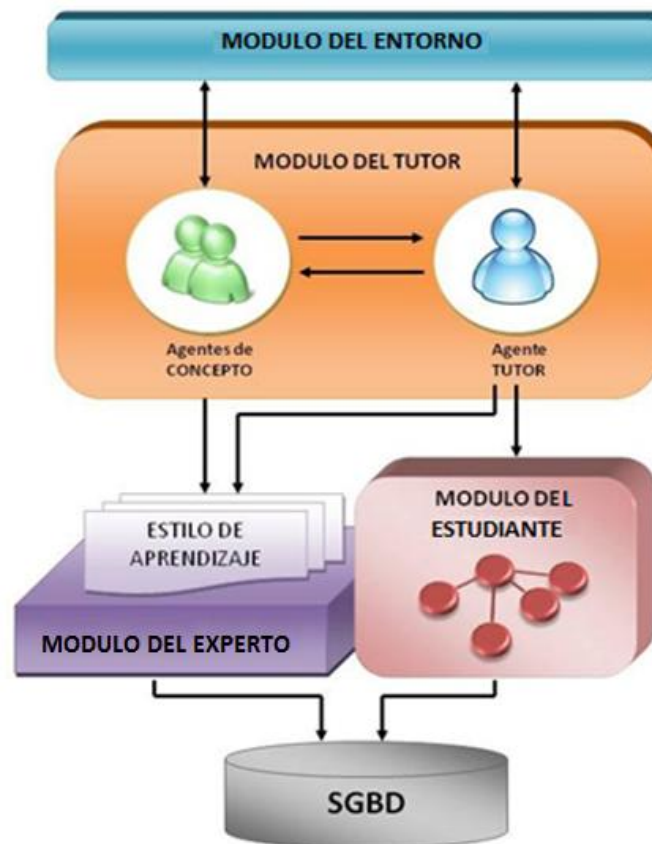


Figura 4. Arquitectura del prototipo de STI.

A continuación se describen en detalle cada uno de los cuatro módulos que componen la arquitectura del sistema:

4.1 MÓDULO DEL ESTUDIANTE

El módulo del Estudiante permite almacenar la información del conocimiento que tiene el estudiante sobre los contenidos del tutorial antes y durante su proceso de aprendizaje. El módulo del Estudiante va a facilitar el manejo de esta información, facilitar su actualización cuando sea necesario y realizar procesos de diagnóstico sobre ella para permitir, por ejemplo, aconsejar al estudiante en determinados aspectos del tutorial. El módulo del Estudiante está formado por el modelo del estudiante, inicialización del modelo del estudiante y el proceso de diagnóstico.

4.1.1 El Modelo del Estudiante

El uso de Redes Bayesianas para modelar el estudiante, ha sido utilizado satisfactoriamente en el desarrollo de STI como Bits [100] y en la aproximación propuesta por Gamboa y Fred [101]. Las características típicas de las redes bayesianas (ver ANEXO A) se adaptan perfectamente a la división del conocimiento en conceptos con relaciones de dependencia, realizada en estos sistemas, permitiendo representar fácilmente la jerarquía de conceptos en la red. Al mismo tiempo, las redes bayesianas proporcionan una serie de mecanismos de inferencia que se pueden utilizar para el proceso de diagnóstico sobre el modelo del estudiante.

En la Figura 5 describe el modelo de la red bayesiana que se utiliza para representar el modelo del estudiante. En la estructura de la red bayesiana, se pueden distinguir fácilmente los niveles jerárquicos de agregación que se corresponden con los elementos en los que se divide la estructura del tutorial. Cada nodo de la red bayesiana va a representar a cada uno de los elementos de la estructura de contenidos del tutorial. A continuación se describe en detalle la representación de cada uno de los nodos de la red.

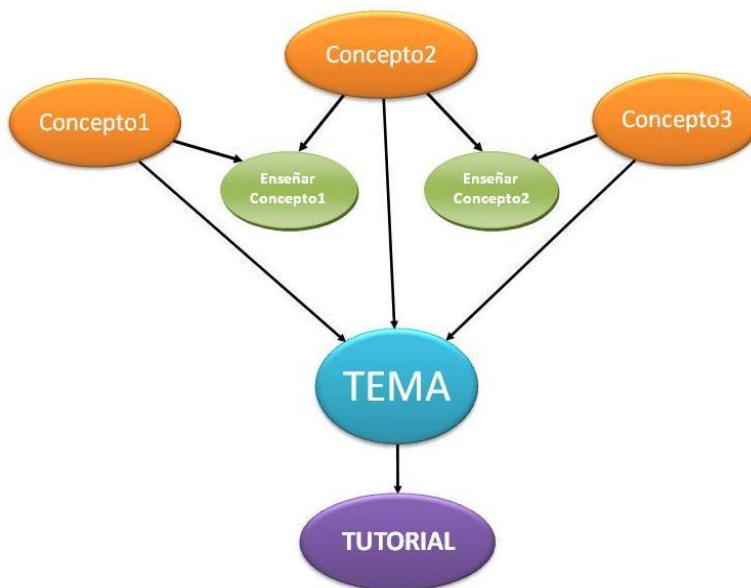


Figura 5. Modelo de la Red Bayesiana usada



Nodos de Concepto

Los nodos que representan a los conceptos del tutorial, se encuentran representados en color naranja en la Figura 5. Estos nodos van a representar el nivel de conocimiento que el estudiante ha adquirido en cada uno de los conceptos del tutorial, cada nodo va a estar asociado al concepto. Cada nodo de tipo concepto va a tener asociados dos estados, el estado *Conocido* y el estado *Desconocido*. En el estado *Conocido* se almacena el porcentaje de conocimiento adquirido por el estudiante en el concepto que representa el nodo, que representa al mismo tiempo la probabilidad que tiene el estudiante de conocer dicho concepto. Como la suma de los valores de los dos estados debe ser uno, en el estado *Desconocido* se va a almacenar el valor $(1-\text{Conocido})$, y va a representar la probabilidad de que el estudiante no conozca el concepto. Para asignar los valores iniciales al estado *Conocido* en los nodos de la red, se realiza un test de conocimiento inicial que permite determinar el nivel de conocimiento que tiene el estudiante en los conceptos asociados a cada nodo. El valor del estado *Conocido* en cada nodo irá aumentando a medida que el estudiante adquiere conocimientos en los conceptos, hasta llegar a un máximo de 1 que indica que el estudiante conoce la totalidad de los contenidos del concepto.

Nodos “Enseñar Concepto”

Los nodos enseñar concepto que se representan con color verde en la Figura 5, se utilizan para indicar el grado de conveniencia o viabilidad de que el estudiante comience a estudiar un concepto determinado del tutorial. Cada nodo concepto que aparece en la red, va a tener asociado mediante un arco dirigido un nodo enseñar concepto. El nodo enseñar concepto tendrá dos estados asociados. El estado *Viable* indica la probabilidad de que sea viable que el estudiante comience a estudiar el concepto representado por el nodo concepto al que está asociado. El estado *Inviable* indica la probabilidad contraria, es decir, la probabilidad de que no sea viable que el estudiante estudie el concepto. Inicialmente, cuando el valor del estado *Conocido* del nodo concepto es cero, el valor del estado *Viable* del nodo enseñar contendrá un valor cercano a uno ya que como el estudiante no conoce nada sobre el concepto, es viable que el estudiante comience a estudiarlo. A medida que el estudiante adquiere conocimientos en el concepto, el valor del estado *Conocido* del nodo aumenta, con lo que la viabilidad de estudiar el concepto disminuye paulatinamente.

Mediante las restricciones introducidas en la red bayesiana con los nodos enseñar concepto, no sólo se va a conseguir determinar la viabilidad de estudio de determinados conceptos por parte del estudiante sino que se va a establecer un itinerario de estudio a seguir por los estudiantes que comiencen a estudiar el tutorial.

A la hora de construir la red bayesiana asociada a un tutorial hay que crear convenientemente la tabla de *probabilidades condicionadas* de los nodos *enseñar concepto* para obtener el tipo de comportamiento deseado de la red a la hora de calcular el grado de conveniencia de estudio de cada nodo y definir el itinerario de estudio que seguirán los estudiantes en el tutorial.

Nodos Tema

Los nodos de tema que se representan en la Figura 5 con color azul, se utilizan para almacenar el grado de conocimiento que adquiere el estudiante en un tema determinado del tutorial. El nivel de conocimiento del tema por parte del estudiante que representan este tipo de nodos, va a ser un nivel de conocimientos global respecto al conocimiento adquirido por el estudiante en cada uno de los conceptos del tema, es decir, que el grado



de conocimiento indicado por el nodo tema va a depender directamente del grado de conocimiento indicado por los nodos concepto para cada uno de los conceptos que pertenecen al tema.

Del mismo modo que ocurre con los nodos de concepto, cada nodo tema va a tener dos estados asociados, el estado *Conocido* y el estado *Desconocido*. En el estado *Conocido* se almacena el porcentaje de conocimiento que el estudiante va adquiriendo a medida que aumenta el porcentaje de conocimiento almacenado en los nodos concepto que representan a los conceptos del tema. El estado *Conocido* representa por lo tanto, la probabilidad de que el estudiante conozca el tema representado por el nodo. En el estado *Desconocido* se va a almacenar el valor $(1-\text{Conocido})$, y va a representar la probabilidad de que el estudiante no conozca el tema representado por el nodo. Cuando el estado *Conocido* alcanza el valor 1 significa que el estudiante conoce la totalidad de los conceptos contenidos en el tema. El aumento del valor de conocimiento en los nodos de tema, se va a producir cuando se realice la propagación de probabilidades en la red en el proceso de diagnóstico del estudiante como se verá más adelante. Este aumento en el valor de conocimiento del tema, dependerá del peso que tenga el concepto en el que el estudiante ha aumentado su conocimiento dentro del tema. Estos pesos se utilizan para crear las tablas de probabilidades condicionadas de los nodos de tema, con respecto a los nodos concepto en el proceso de creación de la red bayesiana y son los mismos pesos que el experto especifica en el proceso de creación de la estructura del tutorial.

Nodos Tutorial

El nodo tutorial representado en la Figura 5 con color morado, se utiliza para almacenar el grado de conocimiento adquirido por el estudiante en la totalidad del tutorial. Del mismo modo que ocurre con los nodos concepto y los nodos tema, cada nodo tutorial va a tener dos estados asociados, el estado *Conocido* y el estado *Desconocido*. En el estado *Conocido* se almacena el porcentaje de conocimiento que el estudiante va adquiriendo a medida que aumenta el porcentaje de conocimiento almacenado en los nodos tema que representan a los temas del tutorial (conocimiento que se aumenta a su vez gracias a los nodos de concepto). El estado *Conocido* representa por lo tanto, la probabilidad de que el estudiante conozca el tutorial representado por el nodo. En el estado *Desconocido* se va a almacenar el valor $(1-\text{Conocido})$, y va a representar la probabilidad de que el estudiante no conozca el tutorial representado por el nodo. Cuando el estado *Conocido* alcanza el valor 1 significa que el estudiante conoce la totalidad de los temas y conceptos contenidos en el tutorial.

El nivel de conocimiento del tutorial adquirido por el estudiante y representado en este nodo, va a ser por lo tanto, un nivel de conocimiento dependiente de lo que el estudiante haya aprendido en los temas del tutorial, y en consecuencia en los conceptos del tutorial. Del mismo modo que ocurre con los nodos tema respecto a los nodos concepto, los nodos tutorial aumentan su valor en su estado conocido cuando se realiza una propagación de las probabilidades en la red y los valores de conocimiento adquiridos por el estudiante en los conceptos del tutorial, se propagan hasta llegar al nodo tutorial.

4.1.2 Inicialización del modelo del estudiante

Considerando que los estudiantes poseen diferentes estilos o formas de aprendizaje, es importante que un STI sea capaz de ofrecer instrucción individualizada, es decir, debe ser capaz de adaptar sus instrucciones de acuerdo a cada estudiante. Este es el principio general que rige la concepción de un STI [102]. Para lograr esto, se contemplan dos aspectos importantes que permiten determinar el perfil con el cual el estudiante inicia su proceso de aprendizaje, estos son: el *estilo de aprendizaje* y el *nivel de conocimiento*



inicial. Por tanto, para inicializar la estructura que representa el conocimiento, se decidió realizar un test de conocimiento inicial y un test para identificar el estilo de aprendizaje de cada estudiante.

Estilo de Aprendizaje

Una de las características fundamentales de la educación incide en la atención a las diferencias individuales, potenciando aquellos aspectos más interesantes de cada persona frente a otros quizás no tan destacables. Cada persona aprende a su propio ritmo, y por ello debemos pensar en una situación de enseñanza que permita a los estudiantes avanzar según sus capacidades e intereses. Sin embargo, desde las propuestas de la Escuela Nueva [103], la adaptación del proceso de enseñanza-aprendizaje al estudiante ha sido muy claro en la teoría pero no ha sido muy fácil llevarlo a la práctica.

Rajadell [103] afirma que la edad cronológica, la inteligencia, el estilo cognitivo, los intereses o el contexto son criterios insuficientes para organizar y homogenizar a un grupo de personas. Así mismo sostiene que el educador será capaz de ajustar la metodología o los recursos si conoce el estilo de aprendizaje de sus estudiantes. Los estilos de aprendizaje son los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables, de cómo los estudiantes perciben, interaccionan y responden a sus ambientes de aprendizaje [104]. Los estilos propuestos son cuatro:

- **Activo:** caracterizado por ser animador, improvisador, descubridor, arriesgado y espontáneo.
- **Reflexivo:** generalmente se trata de una persona ponderada, concienzuda, receptiva, analítica y exhaustiva.
- **Teórico:** normalmente es metódico, lógico, objetivo, crítico y estructurado.
- **Pragmático:** acostumbra a ser experimentador, práctico, directo, eficaz y realista.

El estilo de aprendizaje es la forma en que la información es procesada y se centra en las fortalezas y no en las debilidades, por lo que no existe un estilo de aprendizaje correcto o incorrecto, sino que está dado de acuerdo a cada persona. Un estudio realizado a los estudiantes de ciencias de la Salud en la Universidad de Cádiz demostró que el estilo predominante es el Reflexivo seguido por el Pragmático, Teórico y finalmente por el Activo [105]. Esto nos indica que los estilos se presentan en mayor o menor medida en un grupo de estudiantes y así mismo los estilos predominan en mayor o menor grado de acuerdo a cada estudiante en particular, lo cual nos sugiere que se deberían aplicar diferentes estrategias para cada estudiante de acuerdo a su estilo predominante. Esto representa un problema para la educación actual donde existe un docente enseñando a un grupo de personas con diferentes estilos de aprendizaje.

Con el objetivo de solventar en cierta medida los problemas de personalización de la educación, se ha definido como parte del proceso de diagnóstico del STI la realización, al inicio del tutorial, del **Cuestionario Honey-Alonso de Estilos de Aprendizaje - CHAEA** el cual consta de 80 preguntas que se responden de manera afirmativa o negativa dando como resultado el estilo que predomina en cada estudiante [106]. Se ha seleccionado el cuestionario de CHAEA, por ser uno de los cuestionarios más reconocidos en el medio presencial y virtual, porque su fiabilidad ha sido demostrada en diferentes investigaciones [107]. Asimismo, es el cuestionario más utilizado en el área de salud para determinar los estilos de aprendizaje predominantes de los estudiantes, entre los trabajos se puede mencionar el estudio llevado a cabo por la Universidad de Cádiz [105], el estudio realizado por la Universidad de Madrid [108], entre otros.



Los resultados del test son puntuaciones para cada uno de los estilos de donde se puede deducir que el estilo que mayor puntuación haya entregado será el que predomine en el estudiante y por lo tanto será el que se use principalmente en el transcurso del tutorial, los demás estilos están presentes en el estudiante pero en menor medida. No obstante, si las estrategias correspondientes al estilo principal fallan, se tomará como opción el estilo que haya tenido la segunda puntuación más alta en los resultados del test.

En el ANEXO B se presenta el cuestionario de CHAEA y aspectos específicos sobre los estilos de aprendizaje.

Nivel de Conocimiento Inicial.

Teniendo en cuenta que los estudiantes a través de su experiencia adquieren diferentes conocimientos y competencias, no se puede afirmar que al iniciar un curso todos los estudiantes tienen el mismo nivel de conocimiento por ejemplo, un grupo de estudiantes de pregrado que han cursado las mismas materias y se encuentran en el mismo semestre de su carrera. Como lo expresa Montserrat [109], las variables personales tenderán siempre a personalizar la aplicación de las estrategias de enseñanza, dado que nunca los conocimientos y experiencias previas de los estudiantes ante el aprendizaje de una materia o de un tema, serán las mismas. En este sentido, puede darse el caso que algunos estudiantes conozcan, por alguna razón, una temática específica que sus compañeros de clase no o viceversa y dicha temática forme parte del curso que está por empezar. Lo más seguro es que los estudiantes que conozcan la temática no deseen volver a verla y puedan llegar a sentir cierta desmotivación, lo cual ya representaría un problema en el proceso de enseñanza-aprendizaje. Por tanto, no sería muy apropiado iniciar un curso para un grupo de estudiantes, sin tener en cuenta los conocimientos previos que posee cada uno.

Para dar respuesta en cierta medida a este problema, se propone realizar un **Test de Conocimiento Inicial** a los estudiantes en el momento en que inician su proceso de aprendizaje en el STI. Esto con el fin de determinar los temas o conceptos que el estudiante ya conoce y de esta manera configurar un itinerario de enseñanza adecuado, donde se le permita al estudiante avanzar y aprender conceptos o temas que realmente no conozca. Dicho test se compone de preguntas de selección específicas sobre los diferentes temas que componen el tutorial.

4.1.3 Proceso de diagnóstico.

Para realizar el proceso de diagnóstico se tomará el resultado del *test de conocimiento inicial* como entrada a la red bayesiana y posteriormente, se realizará el proceso de propagación de las probabilidades contenidas en los nodos de la red. Este proceso se repite a medida que el estudiante va adquiriendo más conocimiento en cada uno de los conceptos del tutorial. A continuación se describirá la forma en que se realiza la propagación de probabilidades en la red:

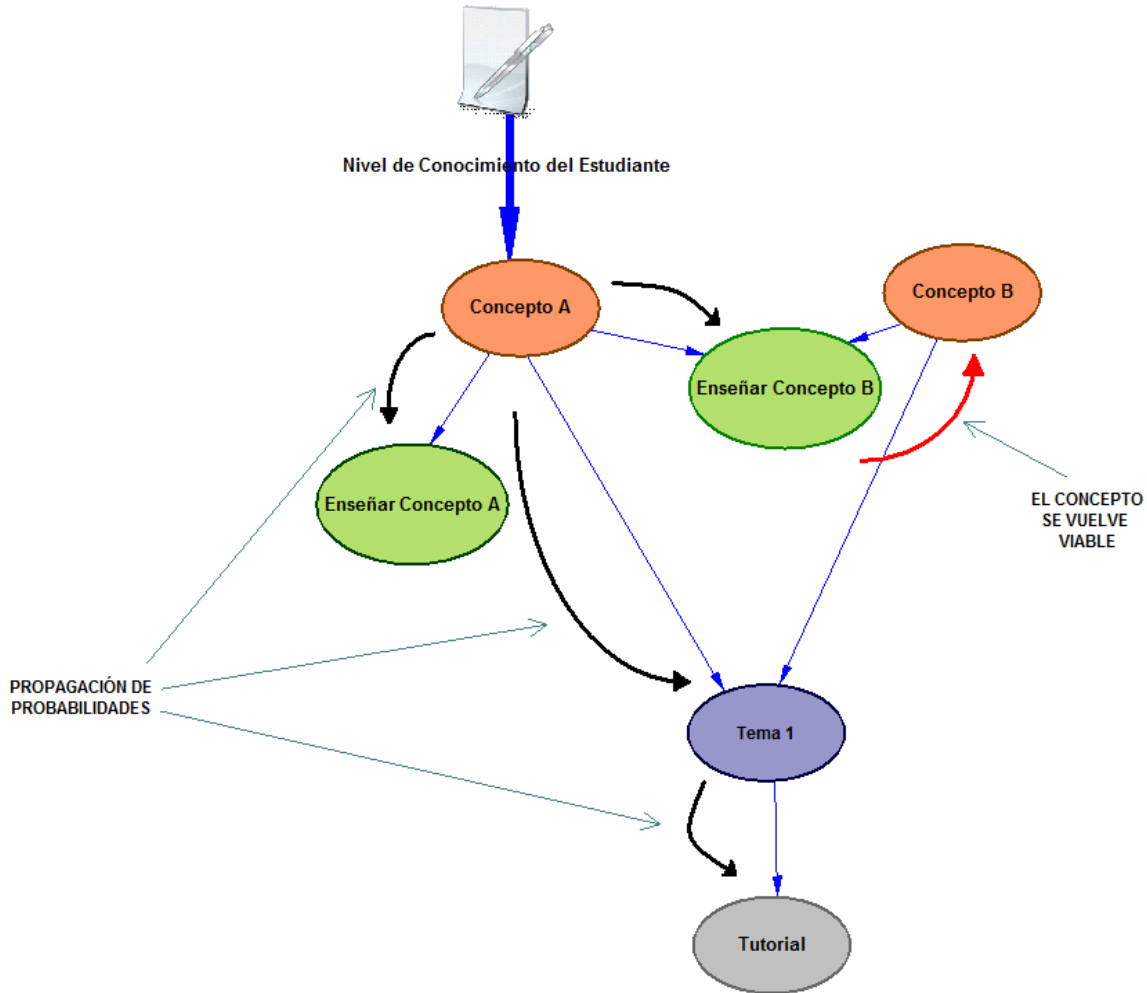


Figura 6. Proceso de propagación de probabilidades en la Red Bayesiana.

Cuando el estudiante aumenta su nivel de conocimiento en un concepto, el valor sobre el nivel de conocimiento adquirido por el estudiante en ese concepto se introduce en el estado *Conocido* del nodo concepto asociado. Una vez actualizado el estado del nodo, se realiza la propagación de las probabilidades de la red, lo que implicará una actualización automática de los nodos directa o indirectamente dependientes del nodo concepto que se ha actualizado.

Esta propagación de probabilidades va a implicar que se actualice el nivel de conocimiento adquirido por el estudiante en el nodo *tema* que depende del nodo *concepto* actualizado y a su vez se va a actualizar el nivel de conocimiento adquirido por el estudiante en el nodo *tutorial*. La propagación de probabilidades también afecta a los valores de los estados de los nodos *enseñar concepto* que dependan del nodo *concepto* actualizado. De esta manera puede suceder que después de realizar una propagación determinados conceptos del tutorial sean accesibles cuando antes no lo eran, debido a que sus nodos *enseñar concepto* han actualizado sus valores de estado tras la propagación, aumentando el grado de viabilidad de estudio por parte del estudiante de los conceptos a los que representan. En la Figura 6 se describe gráficamente el proceso de propagación de probabilidades en una red bayesiana de ejemplo.



El encargado de actualizar los valores de conocimiento en los estados de los nodos concepto así como de realizar la propagación de probabilidades en los nodos de la red, es el **agente tutor** que está a cargo del estudiante en el tutorial. Cuando un concepto previamente visto no es aprobado por el estudiante, se debe realizar un cambio en las estrategias de enseñanza y esto se logra cambiando a su vez, el estilo de aprendizaje. El responsable de realizar este cambio de estrategias es el **agente Concepto**.

Como se explicó anteriormente, no hay un estilo de aprendizaje único para cada persona sino un estilo predominante. Teniendo en cuenta esto, se ha definido en el STI realizar un cambio de estilo por el que más se acerque al estilo predominante y de esta manera presentar el mismo concepto reprobado, pero esta vez con diferentes estrategias de enseñanza.

4.2 MÓDULO EXPERTO

El módulo experto incluye los elementos que representan el conocimiento del dominio. En la base de conocimiento del sistema se almacenan la totalidad de los conocimientos sobre el dominio que van a ser adquiridos por el estudiante, es decir, todos los contenidos de los tutoriales que se van a alojar en el sistema para ser vistos por los estudiantes. Los contenidos de los tutoriales son creados por expertos en el dominio de salud.

Tanto para el proceso de creación de los contenidos por un experto, como para el proceso de aprendizaje de dichos contenidos por los estudiantes, es necesaria la existencia de una estructura adecuada que sirva de soporte para dichos contenidos.

Para representar la estructura del tutorial se ha optado por seguir la división de los conocimientos del dominio en base a conceptos, utilizada eficazmente en sistemas como Bits [100], de tal manera que la adquisición de conocimientos en determinados conceptos influye en los conocimientos que se adquieren en conceptos dependientes. De esta manera el estudiante va adquiriendo conocimientos en el dominio, a medida que adquiere conocimientos en cada uno de los conceptos.

Por tanto, en la estructura del tutorial se distinguen los siguientes elementos (ver Figura 7):

Tutorial: en un nivel superior se encuentra el elemento tutorial que engloba todo el conocimiento sobre un dominio determinado.

Temas: en un primer nivel, cada tutorial se divide en un número determinado de temas. Cada tema contiene un número de conceptos relacionados con el conocimiento que se pretende enseñar en el tema y un examen que debe ser superado por el estudiante.

Conceptos: los conceptos representan las unidades de conocimiento que deben ser aprendidas por el estudiante a lo largo del tutorial. Cada concepto está formado por una serie de contenidos que explican el conocimiento que representa el concepto.

Preguntas de selección: al finalizar los contenidos de cada concepto se encuentran una serie de preguntas de selección que tienen como objetivo evaluar el conocimiento que el estudiante adquiere sobre el concepto a medida que examina las explicaciones.

Preguntas de examen: son las preguntas contenidas en los exámenes de los temas. Se trata de ejercicios de desarrollo que el estudiante debe resolver y que serán evaluados y corregidos por un experto.

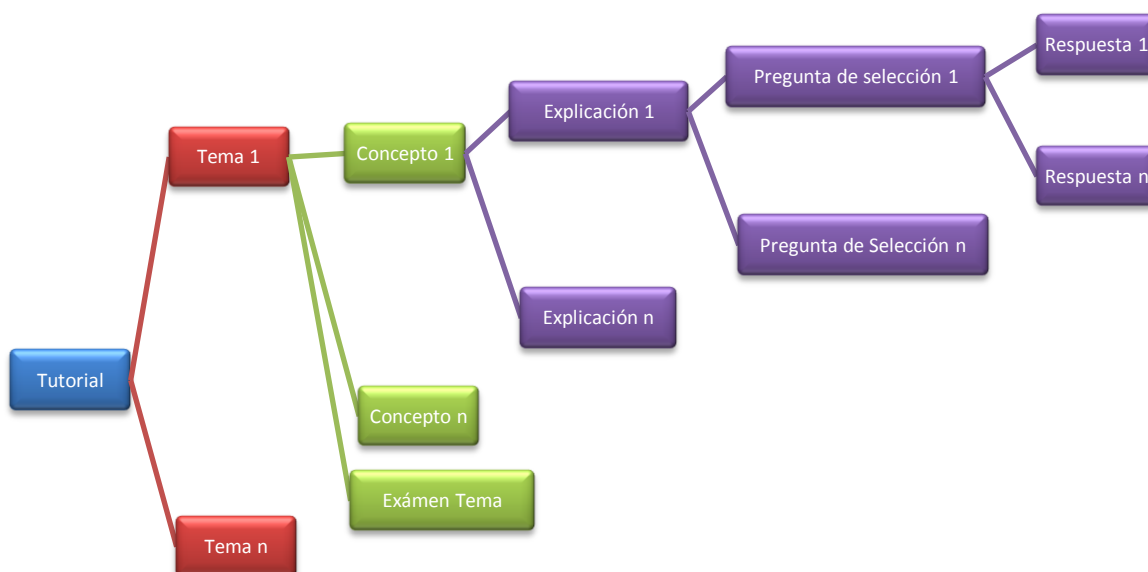


Figura 7. Estructura del tutorial.

La forma de clasificar los contenidos del tutorial estará dada de acuerdo al estilo de Aprendizaje visto anteriormente en el proceso de diagnóstico. Es decir que en la base de conocimiento existirán cuatro versiones de un mismo tutorial, una por cada estilo de aprendizaje. Esto implica que cada concepto estará construido y almacenado en la base de conocimiento utilizando diferentes estrategias de enseñanza.

Las estrategias de enseñanza según Martínez y Bonachea [110] son las acciones que realiza el docente, con el objetivo consciente que el estudiante aprenda de la manera más eficaz. Estas acciones se ven reflejadas en el STI de acuerdo a cada estilo de aprendizaje, es decir, se han seleccionado tres estrategias diferentes para cada uno de los cuatro estilos de aprendizaje (activo, reflexivo, teórico y pragmático).

Cada estilo de aprendizaje plantea características específicas del estudiante que se ubica en determinado estilo [111] por ejemplo, un estudiante cuyo estilo de aprendizaje es *activo*, se caracteriza por ser animador, improvisador, descubridor, arriesgado y espontáneo. Teniendo en cuenta estas características, se realizó un proceso para definir las estrategias que se consideraron más adecuadas para cada estilo.

Díaz y. Martins [112] presentan una lista con diferentes estrategias de acuerdo a cuatro características presentes en los estudiantes:

- Capacidad de *Observar*.
- Capacidad de *Analizar*.
- Capacidad de *Teorizar*.
- Capacidad de *Sintetizar*.
- Capacidad de *Aplicar* y transferir lo aprendido.



Para ver los detalles de las estrategias de acuerdo a las capacidades antes mencionadas, referirse al ANEXO B. Es importante aclarar que las listas de estrategias que plantean los autores mencionados, son solamente una descripción de sugerencias a las cuales un docente puede agregar las que cree su imaginación. Para la construcción de la base de conocimiento del STI se retomaron los datos de las listas para tener una base de estrategias y poder realizar la selección más apropiada.

El siguiente paso en el proceso de selección fue agrupar las estrategias y asociarlas a un estilo de aprendizaje determinado. Espinosa de la Pava [113] plantea que las estrategias no son excluyentes y que por tanto, es posible encontrar estrategias que se pueden utilizar para varios estilos de aprendizaje. Así mismo, el autor expresa que se deben seleccionar mínimo tres estrategias por cada estilo. En consecuencia se eligió un número reducido de estrategias por cada característica adaptándolas de acuerdo a las necesidades propias del STI y se asociaron a los cuatro estilos de aprendizaje como se muestra en la Tabla 2.

ESTILO DE APRENDIZAJE	ESTRATEGIA
<i>Activo</i>	Estudio de casos con solución
	Entrevistas de personas
	Consulta de biografías
	Análisis de la estructura de organizaciones
<i>Teórico</i>	Investigación bibliográfica, lectura de informes de investigación y textos
	Clases expositivas
	Reflexión individual
	Entrevistas de personas
	Comparación de teorías
<i>Reflexivo</i>	Comparación de objetos y fenómenos (fotos y videos)
	Entrevistas de personas
	Consultas bibliográficas
	Clases expositivas
	Diagnostico de situaciones
	Diagramas, esquemas, gráficos
<i>Pragmático</i>	Consulta de biografías
	Estudio de casos
	Análisis de la estructura de organizaciones
	Diagnóstico de situaciones

Tabla 2. Estilos y estrategias de enseñanza

Finalmente, será el experto en el dominio de salud el encargado de construir los contenidos de acuerdo a las diferentes estrategias de enseñanza para su posterior almacenamiento en la base de conocimiento.

4.3 MÓDULO DEL TUTOR

Las tareas del modulo tutor van a ser desempeñadas por un sistema multi-agente. Las propias características que brindan los agentes, la versatilidad a la hora de combinar e



integrar los agentes en sistemas de razonamiento y aprendizaje, y sobre todo la posibilidad de utilizar diferentes tipos de agentes para realizar las distintas tareas que desempeña el módulo instructor, son las principales razones que han influenciado en la decisión de elegir como módulo instructor un sistema multi-agente.

Las tareas fundamentales del sistema multi-agente son las siguientes:

- *Dar soporte al proceso de diagnóstico* del estudiante, actualizando en los nodos de la red, los valores sobre el nivel de conocimiento adquirido por el estudiante en los conceptos del tutorial.
- *Monitorizar* el comportamiento del estudiante durante el proceso de aprendizaje contabilizando el tiempo empleado por el estudiante en estudiar los contenidos del tutorial, para ser tomado en cuenta a la hora de aconsejar al estudiante.
- *Evaluar* y calcular el grado de conocimiento adquirido por el estudiante en cada uno de los conceptos del tutorial.
- *Aconsejar* al estudiante sobre las decisiones que éste toma a lo largo del proceso de aprendizaje del tutorial.

Estas tareas son llevadas a cabo por los dos tipos de agentes existentes en el sistema multi-agente. Estos tipos de agentes son el **agente tutor** y el **agente de concepto**. Teniendo en cuenta la clasificación de Hong [67], éstos agentes se clasifican como: agentes estáticos en la dimensión movilidad, agentes reactivos en la dimensión sensibilidad y agentes de interfaz en la dimensión comportamiento.

En este sentido, cuando un estudiante se encuentre estudiando uno de los tutoriales, tendrá a su servicio un agente tutor que va a seguir su proceso de aprendizaje a lo largo del tutorial y tendrá también a su servicio un número determinado de agentes de concepto que se encargarán de enseñarle los contenidos de los conceptos del tutorial, a la vez que evalúan sus progresos en dichos conceptos. A continuación se describen en detalle las funciones que desempeñan estos dos tipos de agentes.

4.3.1 El agente tutor.

En el momento en el que un estudiante accede a uno de los tutoriales del sistema, el sistema multi-agente va a crear un agente tutor que le va a ser asignado a ese estudiante en el tutorial al que ha accedido. En el caso de que no sea la primera vez que el estudiante accede al tutorial, ya habrá un agente tutor para ese estudiante y para ese tutorial creado en el sistema multi-agente, por lo que se va a reutilizar. Por lo tanto se puede decir que el sistema multi-agente podrá contener tantos agentes tutores como estudiantes se encuentren estudiando un tutorial en un momento determinado.

Una vez que el agente tutor es creado y asignado a un estudiante en un tutorial determinado, será el encargado de llevar a cabo las siguientes tareas:

- Realizar el proceso de diagnóstico del estudiante.
- Proporcionar contenidos del tutorial.
- Comprobar viabilidad de conceptos.
- Proporcionar exámenes al estudiante.
- Proporcionar exámenes al profesor.
- Proporcionar informes.
- Aconsejar al estudiante.



Realizar el proceso de diagnóstico del estudiante. Existen dos situaciones en las que se realiza el proceso de diagnóstico del estudiante: (i) al inicio cuando se obtienen los resultados del *Test de Conocimiento Inicial* junto con el cuestionario CHAEA [106] y (ii) cuando después de estudiar los contenidos de un concepto del tutorial, el estudiante aumenta su nivel de conocimiento en dicho concepto. Para ambos casos, el agente tutor va a ser el encargado del proceso de diagnóstico actualizando el modelo del estudiante.

En el primer caso, los valores sobre el nivel de conocimiento adquirido por el estudiante son el resultado de realizar el test de conocimiento inicial y el estilo de aprendizaje estará determinado por los resultados que se obtengan del cuestionario CHAEA. Para el segundo caso, estos valores sobre el nivel de conocimiento del estudiante, son proporcionados al agente tutor por el agente de concepto adecuado. Después de actualizar el nodo de concepto en la red, el agente tutor desencadena el proceso de propagación de probabilidades para actualizar los valores de conocimiento en los distintos nodos de la red.

Una vez realizado el proceso de propagación de probabilidades, el agente tutor puede consultar y analizar la nueva información contenida en los nodos de la red sobre los niveles de conocimiento adquiridos por el estudiante en los distintos elementos del tutorial, o la información sobre las restricciones de viabilidad de estudio de los conceptos del tutorial. Esta información será utilizada por el agente tutor entre otras tareas, para realizar el cambio de estilo de aprendizaje cuando un concepto ya visto no ha sido aprobado por el estudiante.

Proporcionar contenidos del tutorial. El agente tutor es el encargado de proporcionar al estudiante los contenidos de todos los elementos del tutorial, excepto los contenidos de los conceptos (que serán mostrados al estudiante por el agente de concepto). Cuando el estudiante selecciona en la interfaz de usuario el enlace que representa un tema del tutorial, el agente tutor se encarga de mostrar al estudiante la portada del tema que contiene la descripción del tema seleccionado.

Comprobar viabilidad de conceptos. El agente tutor es el encargado de comprobar si el estudiante puede iniciar el estudio de conceptos del tutorial. Para ello utiliza la información que le brinda la red bayesiana obtenida, después de realizar el proceso de diagnóstico del estudiante. Cuando un estudiante selecciona un concepto del tutorial para comenzar su estudio, lo primero que hace el agente tutor es verificar si el concepto es *viable* para el estudiante.

Si lo es, el agente tutor comprueba mediante el servicio de directorio proporcionado por la plataforma de agentes del sistema multi-agente, si existe un agente de concepto creado anteriormente para enseñar el concepto al estudiante. Si existe un agente de concepto previo lo va a reutilizar, pero si no existe creará uno nuevo. Una vez obtenido el agente de concepto para el concepto seleccionado por el estudiante, el agente tutor delega en él la responsabilidad de enseñar los contenidos del concepto al estudiante. A partir de ese momento el estudiante pasa a interactuar directamente con el agente de concepto, y el agente tutor se dedicará a obtener los datos enviados por el agente de concepto o a atender las peticiones del profesor si estuviera en ese momento en el sistema.

Si el concepto no es viable, el agente tutor notificará al estudiante mediante un mensaje que no es posible que comience a estudiar ese concepto en ese momento. Adicionalmente intentará determinar cuál es la causa por la que el estudiante no puede acceder a ese concepto, es decir, que conceptos necesita el estudiante conocer primero



para poder acceder al concepto que ha seleccionado. Para determinar estos conceptos “clave” va a utilizar la información que le brinda la red a través del nodo enseñar concepto del concepto seleccionado. Comprobando el nivel de conocimiento de los nodos de concepto de los que depende dicho nodo enseñar y las posibles relaciones de esos nodos concepto con otros nodos, el agente de concepto notifica al estudiante en un mensaje, qué conceptos del tutorial debe conocer primero antes de poder comenzar a estudiar el concepto seleccionado.

Si un concepto no es viable, el estudiante nunca va a poder acceder a sus contenidos mientras este concepto permanezca inviable para él. Este proceso de comprobación de viabilidad realizado por el agente tutor va a repetirse siempre que el estudiante seleccione el concepto al que no puede acceder.

Proporcionar exámenes al estudiante. Cuando el estudiante accede a la portada de un tema del tutorial, independientemente de la acción que el estudiante vaya a realizar: (i) el examen del tema o (ii) estudiar un concepto; el agente tutor realizará una serie de comprobaciones. La primera de ellas consiste en verificar si alguno de los conceptos son accesibles (viables) para el estudiante. Si el estudiante no puede comenzar a estudiar ninguno de los conceptos del tema, el agente le notificará de esta situación mediante un mensaje.

En la segunda comprobación se verifica si el estudiante ha realizado anteriormente el examen del tema que ha seleccionado.

- Si el estudiante conoce todos los conceptos del tema y todavía no ha realizado el examen del tema, va a aconsejar al estudiante que lo haga cuando éste selecciona el tema.
- Si el estudiante ya ha realizado el examen del tema y éste ya ha sido corregido por el profesor a cargo del estudiante, va a comprobar los resultados obtenidos por el estudiante en el examen, notificándole en cada caso si lo ha superado o no lo ha superado y aconsejándole que repase los conceptos del tema en caso de no haber superado el examen.
- Si el estudiante ya ha realizado el examen pero el profesor aun no lo ha corregido, el agente tutor va a enviar un mensaje al estudiante diciéndole que debe esperar a que el profesor corrija el examen para ver los resultados.

Una vez que el estudiante se encuentra en la portada del tema, puede solicitar en cualquier momento realizar el examen del tema. En ese momento el agente tutor va a comprobar si el estudiante conoce todos los conceptos del tema. Si es así, el estudiante puede realizar el examen del tema, con lo que le proporciona los enunciados de las preguntas del examen para que el estudiante las responda.

Si el estudiante no conoce alguno de los conceptos del tema, no podrá realizar el examen, y el agente tutor se lo notificará con el mensaje apropiado.

Cuando el estudiante termina de responder a las preguntas del examen, las envía de vuelta al sistema.

Éstas son recogidas por el agente tutor que las va a almacenar hasta que sean solicitadas por el profesor para ser corregidas. En el momento en el que el estudiante envía las respuestas del examen, el agente tutor va a enviar una notificación al profesor para indicarle que tiene exámenes pendientes por corregir para ese estudiante.

Proporcionar exámenes al profesor. El agente tutor, va a ser el encargado de proporcionar los exámenes de los estudiantes a los profesores para que los corrijan. Cuando un



profesor accede al sistema, va a poder comprobar si tiene exámenes de alguno de sus estudiantes pendientes de corregir, gracias a las notificaciones enviadas por el agente tutor. En el momento que el profesor solicita corregir el examen de un estudiante, va a ser el agente tutor del propio estudiante, el que le proporcione las respuestas elaboradas por el estudiante en el examen para que las corrija.

Una vez que el profesor ha corregido el examen y ha evaluado convenientemente las respuestas del estudiante, envía las correcciones y evaluaciones de vuelta al sistema. Estas correcciones son recogidas por el agente tutor del estudiante, que procederá a almacenarlas para que el estudiante las visualice cuando lo crea conveniente. Al mismo tiempo, el agente tutor va a notificar al estudiante que el examen ya ha sido corregido por el profesor. Esta notificación será mostrada al estudiante la próxima vez que acceda al tutorial.

Proporcionar informes. Tanto el estudiante cuando se encuentra realizando el tutorial, como el profesor cuando se encuentra corrigiendo exámenes del estudiante, pueden solicitar informes sobre el nivel de conocimientos adquiridos por el estudiante en determinados elementos del tutorial.

El encargado de proporcionar dichos informes va a ser el agente tutor que está a cargo del estudiante en el tutorial. El agente tutor determina el tipo de informe que tiene que mostrar al estudiante o al profesor en base al elemento seleccionado por ellos cuando solicitan el informe.

Aconsejar al Estudiante. Cuando el estudiante se encuentra realizando el tutorial, puede en cualquier momento solicitar al sistema un consejo sobre cuál es el siguiente paso que debería realizar en el tutorial. El encargado de aconsejar al estudiante, va a ser el agente tutor, que basándose en los progresos realizados por el estudiante en el tutorial hasta ese momento y en la información que es capaz de obtener de la red bayesiana del modelo del estudiante, intenta determinar cuál es el mejor paso que debe realizar el estudiante en el tutorial.

Para obtener el consejo adecuado, el agente tutor va a proceder de la siguiente manera:

- En primer lugar va a dar prioridad a realizar los exámenes de los temas del tutorial. Va a comprobar si el estudiante puede realizar algún examen de los temas del tutorial, es decir, si el estudiante conoce todos los conceptos en alguno de los temas del tutorial. Si se da esta situación, el agente tutor va a aconsejar al estudiante que realice el examen del tema.
- En segundo lugar, va a dar prioridad a aquellos conceptos que el estudiante ya ha comenzado a estudiar. Es decir, si el estudiante ha comenzado a estudiar algún concepto, el agente va a aconsejarle que continúe estudiando aquel concepto cuyo nivel de conocimiento está más cercano al valor que el agente considera válido para darlo por conocido.

En el caso de que haya más de un concepto que el estudiante ya ha comenzado a estudiar y que estos conceptos tengan el mismo valor respecto al nivel de conocimientos adquiridos por el estudiante en cada uno de ellos, el agente tutor va a realizar un proceso de selección entre estos conceptos. Este proceso determinará cuál de ellos le resulta más beneficioso comenzar a estudiar (el proceso de selección de concepto se explica más adelante en este apartado). El agente tutor aconsejará al estudiante que comience a estudiar el concepto que ha determinado como más adecuado para él.

- En tercer lugar va a dar prioridad a aquellos conceptos que son viables para que el estudiante comience a estudiar en el tutorial. El agente tutor va a consultar en la red



bayesiana, los valores de los estados *Viable* de los nodos *enseñar concepto*, para determinar que conceptos son viables para el estudiante. Entre todos los conceptos que son viables, el agente tutor se va a quedar con aquel nodo que tiene un mayor valor en el estado *Viable* de su nodo *enseñar concepto*. Este será el concepto que el agente tutor va a aconsejar al estudiante para que comience a estudiarlo. En el caso hipotético de que coincidan varios conceptos que tengan el mayor valor de viabilidad, el agente tutor va a intentar determinar cuál es el más adecuado para que el estudiante comience a estudiarlo realizando una selección entre los conceptos. El proceso de selección se describe a continuación.

Selección de conceptos. Cuando el agente de concepto obtiene varios conceptos en alguno de los pasos descritos anteriormente, va a realizar un proceso de selección para determinar cuál es el concepto que puede ser más adecuado para que el estudiante comience a estudiarlo.

Para llevar a cabo la selección, va a realizar *predicciones* sobre la red bayesiana que representa el modelo del estudiante, es decir, va a comprobar que es lo que ocurriría si el estudiante conociera alguno de los conceptos que están incluidos en el proceso de selección.

Para cada uno de estos conceptos va a realizar los siguientes pasos:

1. Asigna el valor **1** en el estado *Conocido* del nodo concepto, que representa al concepto a desempatar, es decir, se considera el concepto como conocido por el estudiante.
2. Realiza una propagación de probabilidades para determinar el valor del porcentaje de conocimiento que se obtiene en el nodo tutorial con esta situación.
3. Almacena el porcentaje de conocimiento obtenido en el nodo tutorial y repite los tres primeros pasos para los demás nodos de los conceptos que hay que desempatar.
4. De los porcentajes de conocimiento del nodo tutorial, obtenidos en los pasos anteriores para todos los conceptos empatados, se va a quedar con el mayor valor.
5. El agente tutor va a aconsejar al estudiante, el concepto cuyo nodo ha obtenido el mayor valor en la predicción.
6. Si se da el caso aislado de que haya varios nodos que obtengan el mismo valor en la predicción, el agente tutor hará una selección entre los conceptos empatados por estricto orden de aparición en el tutorial (Se supone que los conceptos que aparecen antes en el tutorial, tienen mayor prioridad).

4.3.2 El agente de concepto.

Los agentes de concepto son los encargados de enseñar los contenidos de los conceptos del tutorial al estudiante cuando se encuentra realizando un tutorial determinado. Como se ha visto en el apartado anterior, cuando un estudiante selecciona un concepto del tutorial con la intención de comenzar su estudio, el agente tutor va a comprobar la viabilidad de que el estudiante inicie el estudio del concepto seleccionado. Si el concepto es viable, será creado un agente de concepto en el sistema multi-agente para que sea el encargado, a partir de ese momento, de enseñar los contenidos del concepto al estudiante y evaluar los conocimientos adquiridos por el estudiante, con base a los resultados obtenidos en las preguntas cortas de las explicaciones del concepto.

Las funciones y tareas principales de un agente de concepto cuando se encuentra enseñando los contenidos de un concepto al estudiante son:

- Proporcionar los contenidos de las explicaciones del concepto cuando el estudiante las solicita.



- Monitorizar el tiempo que el estudiante emplea en estudiar cada una de las explicaciones del concepto y calcular el tiempo total empleado en el estudio de los contenidos del concepto por parte del estudiante.
- Proponer al estudiante, preguntas cortas de selección cuando éste termina el estudio de una explicación del concepto.
- Realizar la corrección de las respuestas proporcionadas por el estudiante a las preguntas de selección de las explicaciones y mostrar al estudiante los resultados obtenidos en las preguntas. El agente le va a mostrar al estudiante las preguntas, indicando si la respuesta elegida para cada pregunta es correcta o no.
- Calcular el nivel de conocimiento obtenido por el estudiante en la explicación que acaba de estudiar, en base a los resultados obtenidos en la evaluación de las preguntas de la explicación.
- Proporcionar consejos al estudiante, si éste los solicita cuando se encuentra estudiando el concepto.
- Realizar cambio de estilo de aprendizaje. Cuando un estudiante ve por primera vez un concepto, éste le es presentado por el agente de Concepto con ciertas estrategias de enseñanza previamente definidas de acuerdo a su estilo de aprendizaje identificado en el proceso de diagnóstico. Si el estudiante no aprueba dicho concepto deberá verlo nuevamente. Por tanto, el agente concepto ante esta situación, realizará un cambio del estilo de aprendizaje del estudiante por el siguiente estilo que mayor puntuación haya obtenido en los resultados del cuestionario CHAEA, esto significa que se elegirá el segundo estilo que predomine sobre los demás. De esta forma el agente concepto, presentará los contenidos con las estrategias asociadas al nuevo estilo de aprendizaje.
- Proporcionar en todo momento, información al agente tutor sobre los progresos que realiza el estudiante en el estudio del concepto. El agente de concepto le va a proporcionar al agente tutor, los resultados obtenidos por el estudiante en las preguntas de cada explicación junto con el valor de conocimiento adquirido por el estudiante en dicha explicación. También le va a proporcionar el tiempo que el estudiante emplea en el estudio de los contenidos del concepto. Todos estos valores serán almacenados debidamente por el agente tutor en la base de datos del sistema para ser utilizados con posterioridad.
- El agente de concepto también le va a proporcionar al agente tutor, el valor global de conocimiento adquirido por el estudiante en el concepto, para que el agente tutor realice la actualización pertinente del nodo de concepto en la red bayesiana del modelo del estudiante y para que realice el proceso de diagnóstico requerido.

Una de las funciones del agente de concepto vistas en los puntos anteriores, es proporcionar un consejo al estudiante sobre cómo se desarrolla el estudio de los contenidos del concepto. El agente de concepto va a intentar aconsejar al estudiante sobre cuáles son las acciones que tiene que realizar para mejorar su nivel de conocimiento en el concepto que se encuentra estudiando. Para ello se va a servir de determinados elementos contenidos en la estructura del tutorial y de determinados valores que se van analizando y almacenando a medida que el estudiante realiza el tutorial.

Por un lado va a utilizar el tiempo de estudio que el alumno emplea en estudiar cada explicación del concepto, para aconsejar al estudiante que repase determinadas explicaciones si no se les ha dedicado el tiempo suficiente de estudio. Para poder determinar si no se le ha dedicado el tiempo suficiente a cada explicación, va a comparar el tiempo estimado para cada explicación que se encuentra almacenado en la estructura del tutorial y ha sido definido por un experto en el dominio, con el tiempo dedicado por el



alumno en el estudio y que ha sido contabilizado por el propio agente durante el estudio de la explicación.

Por otro lado, el agente de concepto va a aconsejar al estudiante que repase aquellas explicaciones del concepto que considera que todavía no conoce, es decir, en aquellas explicaciones en las que el estudiante no ha acertado al menos la mitad de las preguntas cortas de selección que se le han realizado.

El agente de concepto también va a tener en cuenta las **referencias** que tienen o pueden tener las preguntas cortas de las explicaciones, a otras explicaciones del mismo concepto o de conceptos diferentes. Para analizar estas referencias, el agente tutor va proceder de la siguiente manera:

- Primero obtiene aquellas preguntas que ha fallado, de entre todas las preguntas cortas que se le han realizado al estudiante cuando termina de estudiar la explicación.
- Entre las preguntas que ha fallado, comprueba si alguna de ellas tiene referencias a otras explicaciones.
- Si alguna de esas preguntas falladas contiene referencias a otras explicaciones, va a comprobar si el estudiante conoce esas explicaciones a las que hace referencia la pregunta.
- Finalmente, el agente de concepto va a aconsejar al estudiante que repase aquellas explicaciones que son referenciadas por las preguntas que ha fallado y que no conoce completamente, es decir, que el nivel de conocimiento adquirido por el estudiante en dichas explicaciones no alcanza el nivel máximo (100%).

4.4 MÓDULO DEL ENTORNO

La interfaz de usuario permite proporcionar la funcionalidad básica del entorno de instrucción de un STI, es decir, proporcionar las herramientas y facilitar las actividades para ayudar en el proceso de aprendizaje del estudiante. La interfaz de usuario se presenta de acuerdo a los tres tipos de usuario del sistema (estudiante, profesor y administrador).

Interfaz del estudiante. Esta interfaz dispone de una estructura y distribución adecuadas para facilitar el proceso de aprendizaje del estudiante en el tutorial. Por medio de esta interfaz se mostrará información acerca de: los contenidos del tutorial que se encuentre aprendiendo el estudiante en páginas Web con el formato HTML o de texto plano, las preguntas de selección de las explicaciones, los ejercicios de los exámenes de cada tema y un índice del tutorial. La interfaz también cuenta con un sistema de mensajes que sirve para permitir que los agentes del sistema multi-agente envíen notificaciones y consejos al estudiante cuando se encuentra realizando el tutorial.

Interfaz del profesor. Esta interfaz cuenta con un esquema similar a la del estudiante, carece de ciertos elementos de interacción con el sistema multi-agente debido a que este no necesita ser evaluado por el sistema, pero incluye una parte destinada a la corrección de exámenes. La principal información que se muestra en esta interfaz es acerca de: los contenidos del tutorial con el índice ya que el profesor puede requerir examinar los contenidos del tutorial, información acerca de los estudiantes que tiene a su cargo donde se le notifica si tiene exámenes pendientes por corregir a alguno de ellos y los formularios adecuados para corregir y evaluar las respuestas de los exámenes de tema.



Interfaz del administrador. Esta interfaz le va a permitir al administrador, realizar las tareas relacionadas con la gestión de los tutoriales y las tareas relacionadas con la gestión de los usuarios del sistema. Las principales funcionalidades permitidas son: introducir los ficheros XML con los contenidos de los tutoriales que desea importar en el sistema, exportación de los contenidos de los tutoriales existentes en el sistema, a ficheros XML externos. Así mismo, ofrece información acerca de los profesores y estudiantes registrados en cualquiera de los tutoriales del sistema y donde se le va a permitir registrar nuevos profesores en el tutorial. Adicionalmente, presenta información acerca de los estudiantes que tiene asignados cada uno de los profesores en un tutorial determinado y permite al administrador asignar nuevos estudiantes a cada uno de estos profesores.



5. PROTOTIPO DE SISTEMA TUTOR INTELIGENTE PARA APRENDIZAJE EN SALUD

En este capítulo se describe el desarrollo del prototipo de sistema tutorial inteligente orientado a la instrucción de la estrategia AIEPI clínico, específicamente el módulo recién nacido, llamado *SIAS - Sistema Tutor Inteligente para Aprendizaje en Salud*.

El sistema SIAS está basado en el sistema SITUA[91] desarrollado en el Departamento de Informática de la Universidad de Vigo (ver apartado 3.6).

SIAS cuenta con cuatro módulos: del experto, del estudiante, del tutor y del entorno, característica principal que permite ubicar a este sistema en el contexto de los sistemas tutoriales inteligentes.

El sistema desarrollado aprovecha parte de la estructura existente en SITUA pero incorpora características de adaptabilidad al perfil de cada estudiante. El sistema SIAS hace particular énfasis en el módulo del estudiante, del experto y del tutor. En el módulo del estudiante se modificó la parte de inicialización del modelo del estudiante en donde se tiene en cuenta dos aspectos importantes: *el estilo de aprendizaje y el nivel de conocimiento con el cual el estudiante inicia el proceso de enseñanza-aprendizaje* (ver apartado 4.1.1). Estos aspectos son importantes para identificar características propias de cada estudiante que permiten al sistema adaptar su proceder en cada proceso de enseñanza. Por otra parte, en el módulo experto se realizó un proceso de identificación y clasificación de los contenidos. Este proceso se realizó teniendo en cuenta las diferentes estrategias de enseñanza de acuerdo a los cuatro estilos de aprendizaje: *activo, teórico, pragmático y reflexivo* (ver apartado 4.1.2).

A diferencia de SITUA, el sistema SIAS incluye características que permiten obtener información de gran importancia sobre el estudiante que inicia el proceso de aprendizaje. Esta información es lo que permite al módulo del tutor elegir las estrategias posiblemente más adecuadas para enseñar determinada temática, variando los ejemplos y rompiendo las secuencias a medida que el estudiante avanza en el tutorial. Así mismo, el sistema hace uso de una red bayesiana para establecer el itinerario de estudio que cada estudiante está en capacidad de seguir, esto debido a que los conocimientos y experiencias previas de los estudiantes ante el aprendizaje de una temática, no son las mismas.

Otro aspecto importante es que el desarrollo del prototipo de sistema tutor inteligente se basa en la Metodología INGENIAS[99], la cuál ha sido seleccionada previamente por los beneficios que aporta en el desarrollo de sistemas basados en tecnologías multi-agente. La metodología propone un proceso el cual inicia con la identificación de las funcionalidades del sistema y los principales agentes. Posteriormente, se identifican los casos de uso en los cuales se produce alguna interacción entre agentes, se define la forma como se organizan los agentes y los objetivos y tareas que cada agente debe realizar. Así mismo, INGENIAS genera especificaciones que incrementalmente cubren las fases de análisis, diseño e implementación.



5.1. ESPECIFICACIÓN DE REQUISITOS

El sistema tutor inteligente pretende capacitar en la estrategia AIEPI clínico, teniendo en cuenta las características propias de cada estudiante que inicie el curso. Por esta razón, el desarrollo del STI tiene ciertas capacidades inteligentes que permiten adaptar los contenidos a cada estudiante, permitiéndoles aprender de la manera más adecuada.

A través de la interacción con el tutor, el estudiante podrá acceder a los diferentes temas y avanzar en los contenidos que componen la estructura de conocimiento del sistema. Así mismo, el tutor evalúa el progreso del estudiante y le provee consejos para guiarlo por el mejor camino de tal forma que alcance un nivel de conocimiento adecuado.

La interacción entre el profesor y el tutor le permitirá al profesor obtener la información del progreso de cada estudiante en los contenidos del tutor. Además, el profesor tendrá la responsabilidad de calificar la evaluación de algunos componentes del tutorial, que el sistema no hace de forma automática.

A continuación se hace una breve descripción de los requisitos funcionales y no funcionales asociados al prototipo de sistema tutor inteligente. En el ANEXO C, se describe en más detalle los requisitos presentados en este apartado.

5.1.1 Requisitos funcionales

El prototipo del sistema ofrece las siguientes funcionalidades:

- Identificación de las características de los estudiantes.
- Presentación de contenidos e índice de acuerdo a las características de cada estudiante.
- Evaluar conceptos.
- Presentación de exámenes a los estudiantes.
- Calificación de los exámenes por parte del docente.
- Actualización del nivel de conocimiento de los estudiantes.
- Envío de notificaciones y consejos.
- Presentación de Informes.
- Gestión de tutoriales.
- Gestión de usuarios.

5.1.2 Requisitos No funcionales

- **Fiabilidad:** El sistema será tolerante a fallos. Sin embargo, la probabilidad de que ocurran fallos siempre va a estar presente. Por lo tanto, el usuario va a poder recuperarse ante diferentes situaciones de error pero no se asegura la ausencia de estos.
- **Disponibilidad:** La disponibilidad de una aplicación Web depende de la confiabilidad del servidor. De acuerdo a lo anteriormente dicho, no es posible definir con exactitud el grado de confiabilidad que va a tener la aplicación.
- **Seguridad;** la seguridad se maneja a nivel de autenticación de usuarios.
- **Consistencia:** La consistencia en el diseño de una página Web es muy importante, debido a que esta nos ayuda a crear un ambiente de tranquilidad y seguridad para que los visitantes de la página se sientan cómodos revisando el servicio. En este

sentido, la aplicación garantizara consistencia de actualización y de interfaz de usuario.

- Portabilidad: La portabilidad en la aplicación se verá reflejada a nivel del usuario final el cual podrá tener acceso a ella desde cualquier navegador web estándar.

5.2. ANÁLISIS

A continuación, se presenta el comportamiento del sistema tutor con base en las relaciones con los usuarios: Estudiante, Profesor y Administrador. En la Figura 8 se ilustra el diagrama de los casos de uso que corresponden al actor Estudiante.

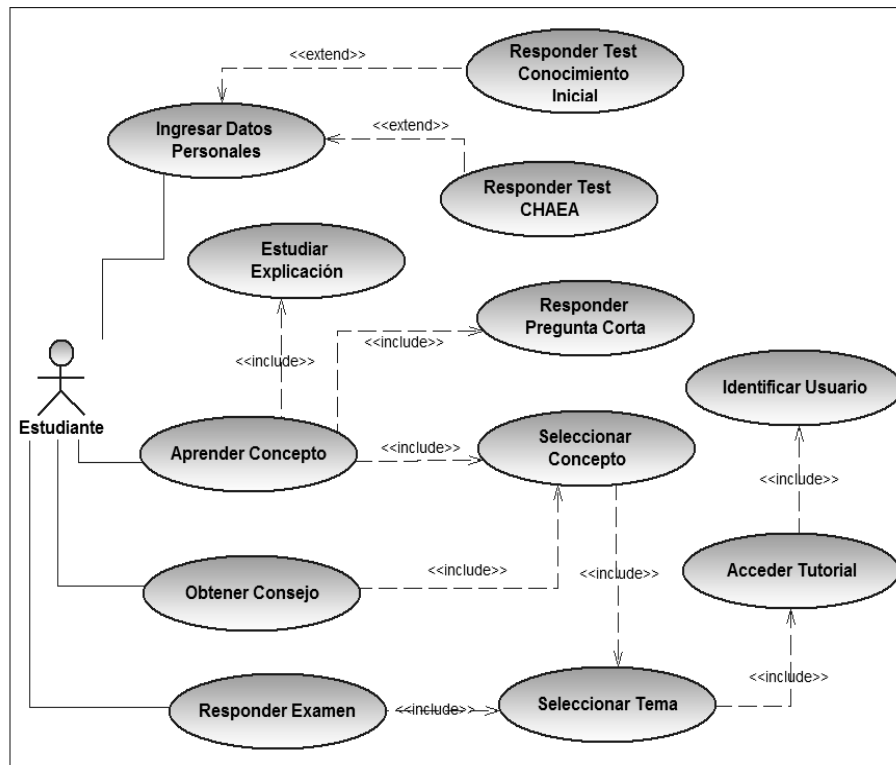


Figura 8. Casos de uso correspondientes al actor *Estudiante*.

El proceso comienza cuando un Estudiante desea iniciar el tutorial, para esto es necesario que realice un proceso de registro. Dicho proceso involucra en primer lugar, ingresar una serie de datos básicos que son requeridos. Luego, resolver un test de conocimiento inicial donde se evalúan el nivel de conocimiento con el que se encuentra el estudiante antes de iniciar el proceso de aprendizaje. Por último, el estudiante debe contestar el Cuestionario Honey - Alonso de Estilos de Aprendizaje – CHAEA [106], para identificar su estilo de Aprendizaje.

Posteriormente, el estudiante que se encuentre previamente identificado por el sistema y que ha accedido al tutorial, selecciona aquel tema que desea aprender. Cuando el estudiante selecciona un tema del tutorial, el sistema determinará si es viable que comience el aprendizaje de los conceptos de ese tema. Si no es viable propondrá una alternativa de estudio en otros conceptos del tutorial. Si el tema es viable, el estudiante podrá seleccionar un concepto del tema para comenzar su aprendizaje.



Así mismo, se analizará nuevamente el grado de conocimiento necesario para poder llevar a cabo con éxito el aprendizaje del concepto actual y determinará si es viable que el estudiante inicie su aprendizaje con dicho concepto. De ser viable, se mostrará los contenidos con los cuales puede iniciar su aprendizaje. De lo contrario, el sistema mostrará un consejo al estudiante indicándole que no es recomendable que inicie con ese concepto sin antes haber aprendido otros conceptos previos necesarios. El estudiante tendrá entonces la oportunidad de seleccionar otro concepto diferente repitiéndose el proceso.

Una vez se conoce el concepto viable que quiere aprender el estudiante, el sistema recupera y prepara los contenidos didácticos según la estructura de dicho concepto y se lo enseña. Dichos contenidos serán tanto explicaciones textuales teóricas como contenidos prácticos en diferentes formas de acuerdo a la estrategia más adecuada. Además, contendrán preguntas cortas donde el estudiante debe seleccionar una respuesta entre varias opciones. El sistema mide los avances del estudiante después de que ha estudiado las explicaciones en base al número de respuestas correctas respondidas por el estudiante. Al mismo tiempo el sistema tomará en cuenta el tiempo dedicado por el estudiante a la hora de estudiar los contenidos del concepto. Una vez que el sistema tiene el resultado final que proporciona un porcentaje sobre el grado de conocimientos adquiridos por el estudiante en ese concepto, se realiza el proceso de actualización del estado del estudiante en el tutorial con los nuevos conocimientos adquiridos para realizar las comprobaciones sobre la viabilidad de estudiar nuevos conceptos.

El estado en el cual se encuentra el estudiante dentro del tutorial, será guardado de manera persistente cuando el estudiante abandona el mismo, para poder ser recuperado de nuevo y permitir al estudiante continuar donde lo dejó cuando acceda de nuevo al tutorial. De la misma manera, se guardan de manera persistente las respuestas del estudiante a las preguntas que se le han realizado en el tutorial.

En la Figura 9, se describe el diagrama de casos de uso correspondiente al actor Profesor.

El proceso de realizar un Examen de algún tema comienza cuando un Estudiante selecciona uno de los temas del tutorial para realizar el examen de prueba sobre los contenidos de ese tema. El sistema comprueba si el estudiante ha aprendido todos los conceptos indispensables para realizar el examen de ese tema. En el caso de que el estudiante no haya aprendido todos los conceptos del tema, el sistema aconsejará al estudiante indicándole que todavía no tiene los conocimientos necesarios para examinarse de ese tema. El estudiante en este punto podrá seleccionar otro tema para realizar el examen o aprender los conceptos necesarios para realizar dicho examen. En caso de que el estudiante haya aprendido todos los conceptos del tema, el sistema planteará las preguntas del examen del tema al estudiante. Una vez que el estudiante responde a todas las preguntas del examen, el sistema se encargará de guardar las respuestas y notificar al profesor a cargo del estudiante que hay preguntas pendientes de corregir para ese estudiante. De esta manera cuando el profesor acceda al sistema, podrá comprobar cuales de sus estudiantes están pendientes de una corrección de examen.

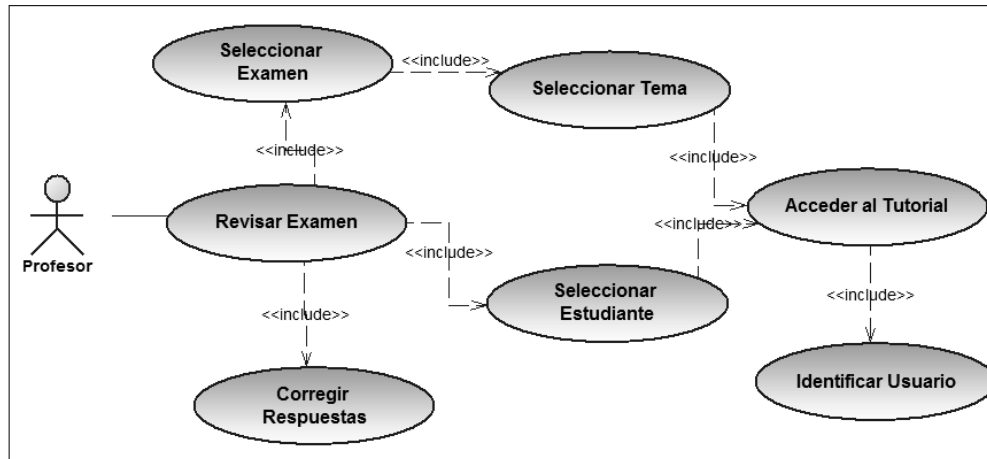


Figura 9. Diagrama de Casos de uso correspondiente al actor *Profesor*.

Cuando el profesor selecciona el estudiante al que quiere corregir un examen, el sistema recupera el estado del tutorial para ese estudiante. El profesor selecciona el tema en el que se encuentra el examen que quiere corregir, y el sistema comprueba si el tema tiene un examen pendiente de corregir. Si el examen está pendiente de corregir, el sistema notifica al profesor y proporciona las respuestas del estudiante al profesor para que estas puedan ser corregidas y evaluadas. Una vez obtenida la calificación final por parte del profesor, el sistema notifica al estudiante de que su examen ha sido corregido para que el estudiante pueda revisarlo. Al mismo tiempo, el sistema actualizará el nivel de conocimiento del estudiante en el tutorial para dar por completo el tema si la evaluación del profesor es satisfactoria.

En la Figura 10, se describe el diagrama de casos de uso correspondiente al actor Administrador. Este usuario tiene asignadas dos tareas importantes: Gestionar los tutoriales y Gestionar los usuarios. Para realizar la correcta gestión de los tutoriales, el sistema permite al administrador introducir nuevos tutoriales. Los contenidos de los tutoriales a introducir en el sistema deben ser incluidos en ficheros XML con una estructura fija y bien definida, de la misma manera se puede hacer el proceso de exportación. Así mismo, el administrador tiene la posibilidad de eliminar los tutoriales en cualquier momento, al eliminar el tutorial se eliminarán los contenidos y toda la información relacionada con el mismo.

El administrador es el encargado de eliminar o realizar el registro de profesores en los tutoriales. Igualmente, el administrador asignará a cada profesor un grupo de estudiantes en el tutorial en que ha sido registrado dicho profesor. Es importante mencionar que si un estudiante se encuentra sin profesor, esto no representa un impedimento para que el estudiante inicie su proceso de aprendizaje, solamente quedarán los exámenes sin calificar hasta que se asigne un profesor.

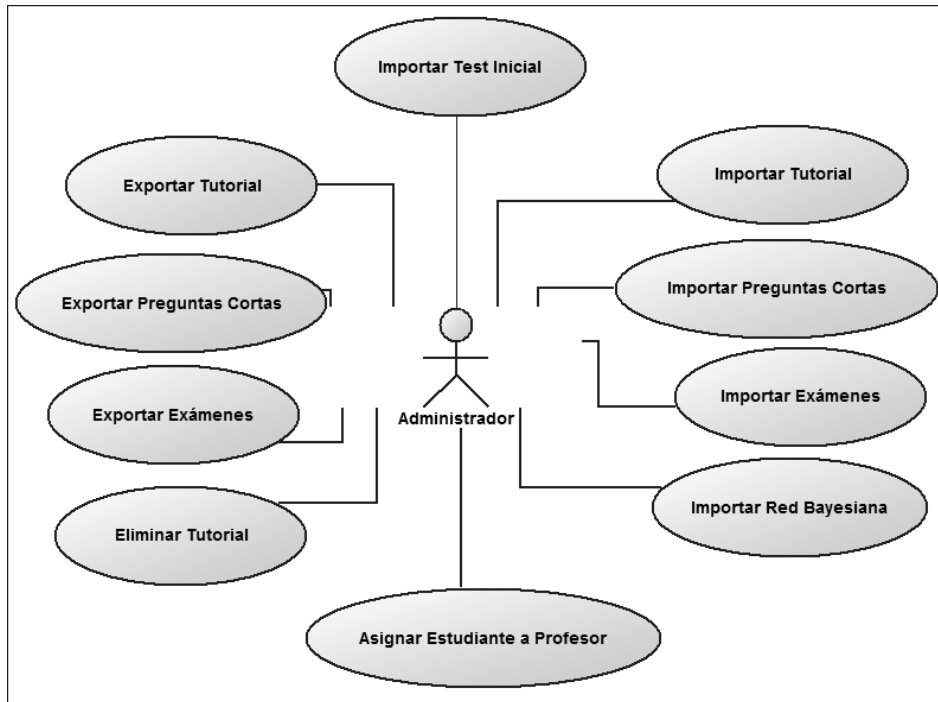


Figura 10. Diagrama de Casos de uso correspondiente al actor *Administrador*.

En el ANEXO C se presentan los anteriores casos de uso en formato expandido.

5.2.1 Modelo de Interacción entre Agentes

El modelo de interacciones de INGENIAS representa las interacciones entre agentes y el objetivo que persigue cada interacción. Este modelo permite observar el comportamiento de un agente cuando recibe instrucciones de otro agente o cuando solicita ayuda para realizar una tarea.

Para generar el modelo de interacciones, primero se analizan los casos de uso en los cuales se produce una interacción. Con los resultados del análisis anterior, se procede a identificar las interacciones más importantes y para cada interacción se determina los agentes o roles participantes y el objetivo que se desea cumplir.

Cuando un estudiante ingresa por primera vez al tutorial, el sistema determina el nivel de conocimiento inicial y el estilo de aprendizaje del nuevo estudiante, con estos resultados obtiene un modelo inicial del estudiante. El agente *tutor* obtiene del modelo inicial, el estilo de aprendizaje y clasifica al estudiante. Después, el agente *tutor* proporciona la información del estilo de aprendizaje del estudiante al agente *concepto*, encargado de enseñar el contenido, para presentarlo de la forma más adecuada.

Una vez el estudiante interactúe con el sistema y estudia el contenido de un concepto, el agente *tutor* se comunica con el agente *concepto* adecuado, le solicita el nivel de conocimiento adquirido por el estudiante y con este dato actualiza el modelo del estudiante. En la Figura 11 se presenta el modelo de interacción para identificar las características del estudiante y la Figura 12 describe el modelo de interacción para actualizar el modelo del estudiante.

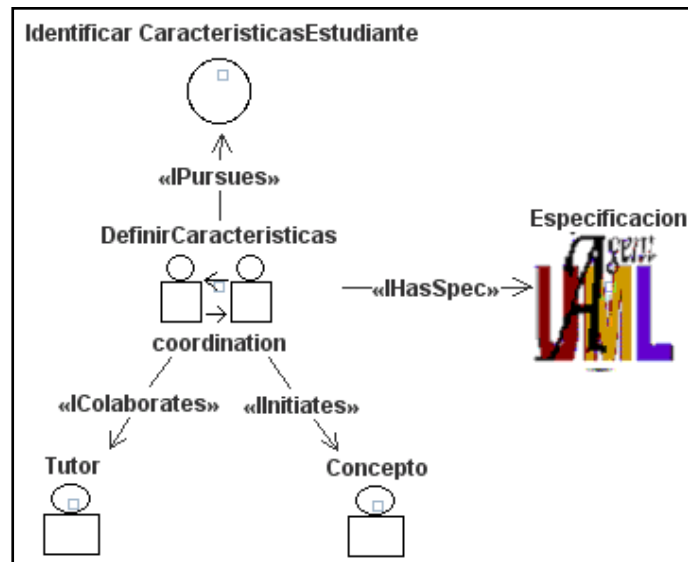


Figura 11. Modelo de interacciones: Identificar características del Estudiante.

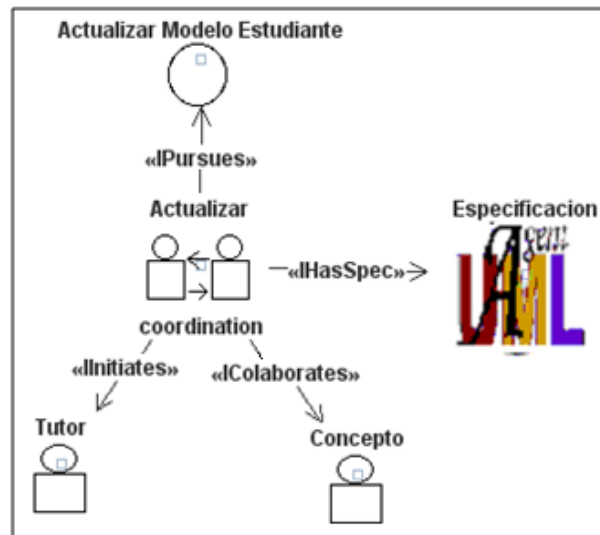


Figura 12. Modelo de interacciones: Actualizar Modelo del Estudiante.

5.2.2 Diagramas de Colaboración

Para cada interacción presente en el modelo anterior se debe generar el diagrama de colaboración, donde se define el protocolo de comunicación utilizado para comunicar a los agentes. A continuación se describen los diagramas de colaboración en los que se presentan participantes, unidades de interacción y las tareas necesarias para cada unidad de interacción. Es importante mencionar que los siguientes diagramas de colaboración forman parte de los artefactos de la metodología INGENIAS.

Diagrama de Colaboración Inicializar Modelo Estudiante

En el diagrama de colaboración para la interacción identificar las características del estudiante, participan el agente *tutor* y el agente *concepto*. Para cumplir el objetivo se necesitan dos unidades de interacción, en la primera el agente *concepto* solicita las características del estudiante y, en la segunda, el agente *tutor* envía la información solicitada. La Figura 13 describe el diagrama de colaboración.

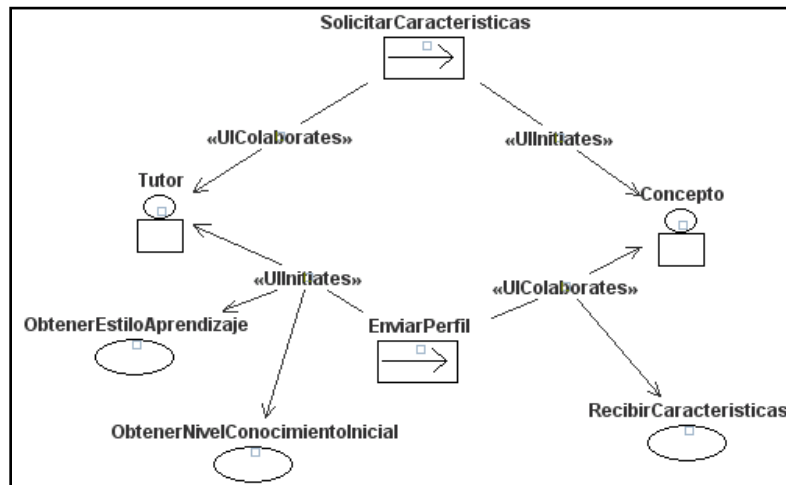


Figura 13. Diagrama de Colaboración: Identificar Perfil del Estudiante

Diagrama de Colaboración Actualizar Modelo Estudiante

En el diagrama de colaboración para la interacción Actualizar Modelo del Estudiante, participan el agente *tutor* y el agente *concepto*. Para cumplir el objetivo se necesitan dos unidades de interacción, en la primera el agente *tutor* solicita el nivel de conocimiento obtenido por el estudiante después de estudiar un concepto determinado y, en la segunda, el agente *concepto* calcula el nivel de conocimiento y lo envía al agente *tutor*. La Figura 14 describe el diagrama de colaboración.

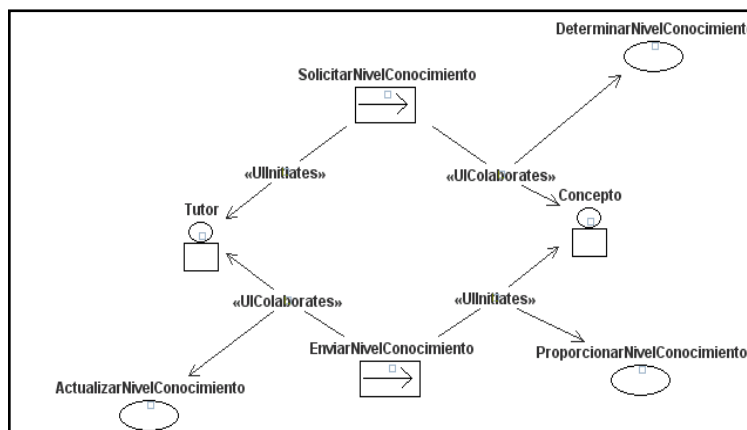


Figura 14. Diagrama de Colaboración: Actualizar Modelo del Estudiante

5.2.3 Modelo de la Organización

El modelo de organización representa la arquitectura del sistema multi-agente y define la estructura en la cual van a existir los agentes, recursos, tareas y objetivos.

Teniendo en cuenta el modelo de interacción entre agentes, se procede a identificar los elementos de la organización. Para el sistema se identifican dos agentes: *tutor* y *concepto*.

A continuación se descompone la organización en grupos y a cada grupo se le asocia por lo menos un miembro. En este caso, la organización se descompone en un grupo: *grupo tutor*, al que pertenece el agente *tutor* y el agente *concepto*.

Después se identifican y se asignan los recursos. En este caso particular, se necesita una base de datos que es manipulada por los miembros del *grupo tutor*, un modulo de diagnostico que permite determinar el nivel de conocimiento del estudiante y es gestionado por el *grupo tutor*, un repositorio de informes y la interfaz grafica son usados por uso por el agente *tutor* y el agente *concepto* para comunicarse con el estudiante y presentar los contenidos.

La Figura 15 presenta el modelo de la organización, donde la organización se representa como un rectángulo con tres círculos, los grupos como un rectángulo con dos círculos, los agentes como un rectángulo con círculo, los recursos como triángulos y los objetivos como círculos.

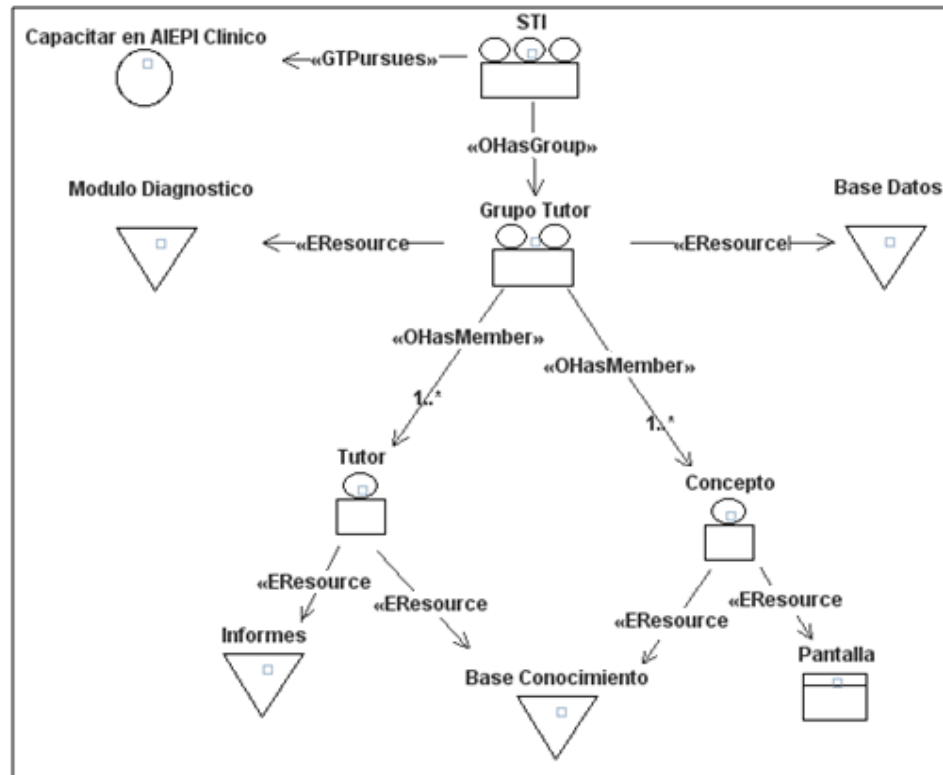


Figura 15. Modelo de la Organización

5.2.4 Modelo del Entorno

Este modelo define las entidades del entorno con las que se puede interactuar, entre los que se encuentran recursos, aplicaciones y agentes.

Para crear este modelo se analizan detalladamente los requisitos. Del análisis de requisitos se identifican los siguientes recursos: base de conocimiento, base de datos, modulo de diagnostico, repositorio de informes, interfaz grafica de usuario. En la Figura 16 se presenta el modelo del entorno, indicando los recursos que utiliza cada agente.

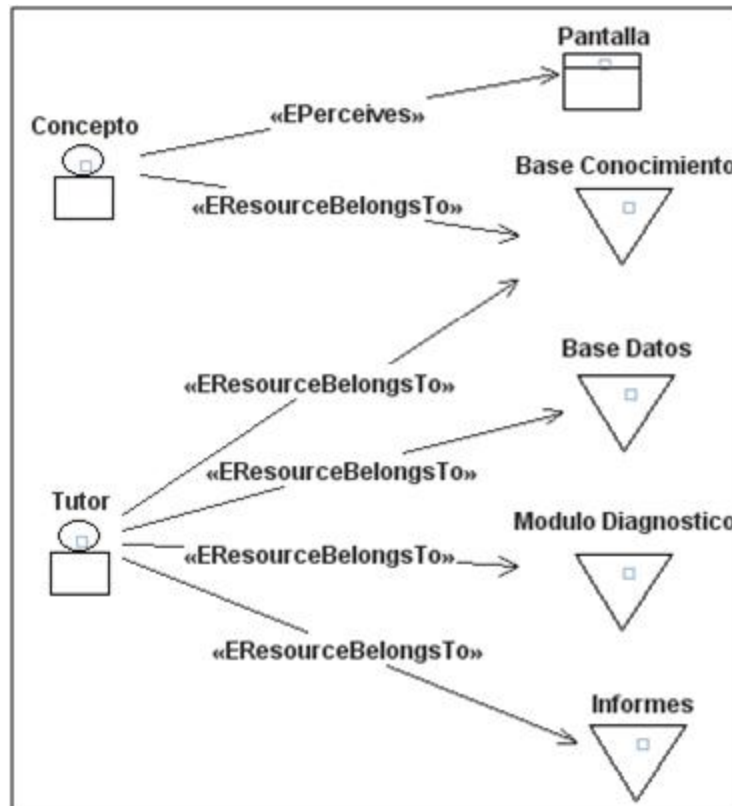


Figura 16. Modelo del entorno

5.3 DISEÑO

A diferencia de SITUA, en este flujo de trabajo se definió el diseño del sistema teniendo en cuenta características de adaptabilidad.

5.3.1 Modelo de Agentes

El sistema desarrollado contará con los siguientes agentes: agente tutor y agente concepto. Cada agente en el sistema juega un rol, el cual le lleva a perseguir unos objetivos. En términos generales, cada agente está definido por su propósito (metas que un agente debe lograr), responsabilidades (que tareas ejecutar), y capacidades (que roles pueden jugar). La forma como se ve reflejado es definiendo el estado mental del

agente, la información que este tiene acerca de la satisfacción de estas metas, y su conocimiento acerca del mundo y los hechos basados en sus experiencias pasadas. Además, el estado mental provee la información que el agente necesita para tomar decisiones.

El estado mental, puede verse como toda aquella información que permite al agente tomar decisiones. Esta información es gestionada y procesada para producir las decisiones del agente. En este sentido, se introducen a nivel conceptual dos entidades que son: Gestor de Estado Mental, y Procesador de Estado Mental. El propósito del Gestor es desarrollar la evolución del estado mental mediante las operaciones de creación, destrucción, modificación y monitorización del conocimiento del agente. Es responsable de mantener la coherencia del conocimiento almacenado y de hacerlo evolucionar. El propósito del Procesador es la toma de decisiones en sí, el control del agente. La ventaja de esta separación entre gestor y procesador es que se desacoplan los mecanismos que implementan la autonomía e inteligencia del agente de la conceptualización del agente.

En la Figura 17 se muestra el modelo para el agente tutor. Los objetivos definidos para este agente son: (i) orientar al estudiante, (ii) identificar las características del estudiante y (iii) actualizar el modelo del estudiante, una vez el estudiante haya incrementado su nivel de conocimiento.

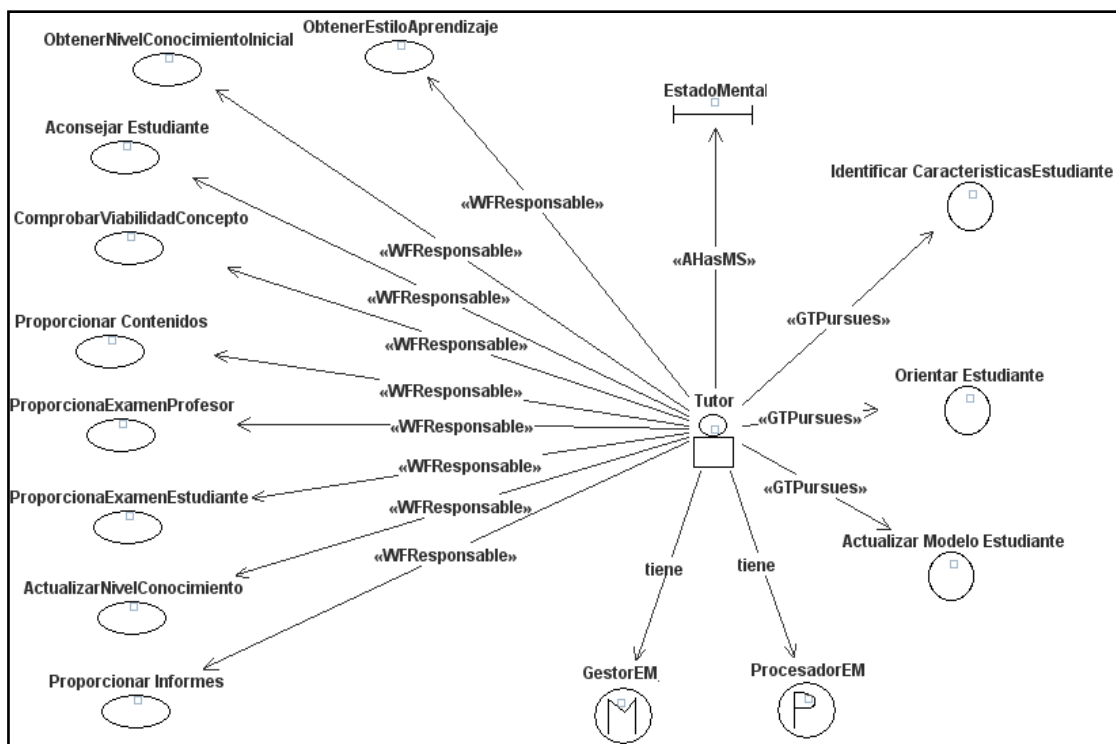


Figura 17. Agente Tutor

En la Figura 18 se muestra el modelo para el agente de concepto. Los objetivos definidos para este agente son: (i) enseñar Concepto y (ii) evaluar concepto, este objetivo se persigue de manera automática, es decir, que la evaluación del concepto es presentada y calificada por el sistema.

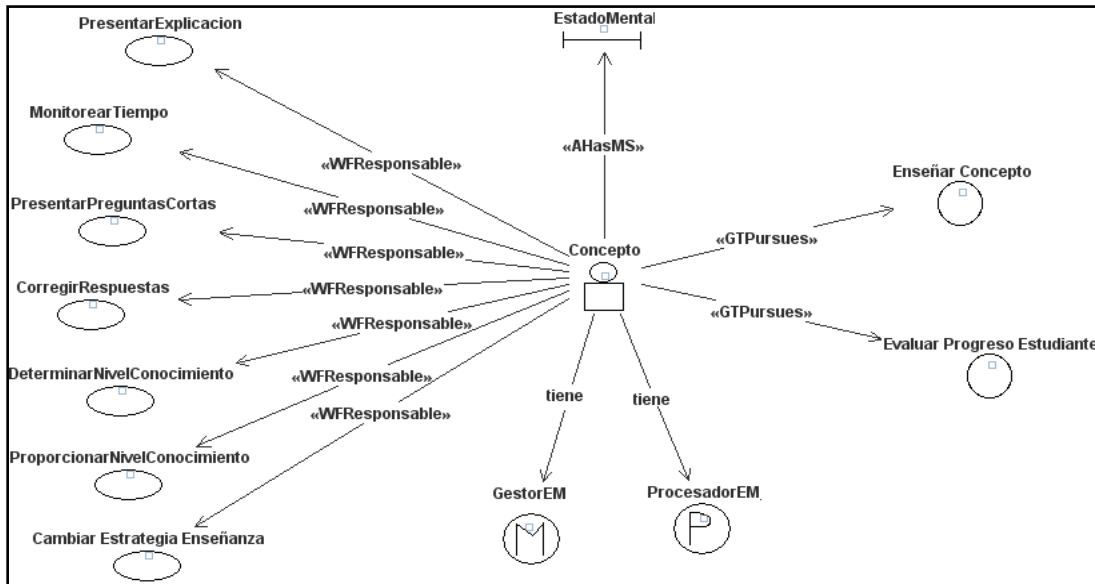


Figura 18. Agente Concepto

5.3.2 Modelo de Tareas y Objetivos

Es muy importante la identificación de las metas que los agentes seguirán y las tareas que deben llevar a cabo para completar cada una de las metas en el diseño de sistemas multi-agente. Cada agente es inicialmente descrito identificando sus responsabilidades (que metas un agente está comprometido a llevar a cabo), y capacidades (que tareas es hábil para desempeñar). En el sistema planteado existe una meta principal y es la de *proveer al estudiante un aprendizaje personalizado* que permita su capacitación en la estrategia AIEPI clínico. Esta meta está compuesta de otras cuatro. La primera es *identificar las características del estudiante*. La segunda es *orientar al estudiante presentándole contenidos adaptables a sus características*, que faciliten de alguna manera el proceso de aprendizaje. La tercera es la de *evaluar el progreso del estudiante* en el proceso de enseñanza-aprendizaje. Y por último, *actualizar el modelo del estudiante*, donde se almacena la información que se ha obtenido como resultado de un aprendizaje. Con el objetivo de llevar a cabo estas metas, se definen diferentes tareas específicas para cada objetivo.

En la Figura 19 se presentan las tareas necesarias para lograr Identificar las características del estudiante. Para lograr este objetivo se requiere Obtener el estilo de aprendizaje del estudiante y el nivel de conocimiento inicial. Con la información del estudiante, el agente *tutor* podrá identificar las características y enviarlas al agente *Concepto*.

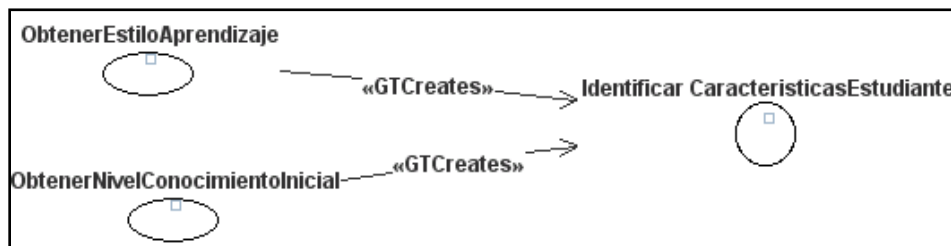


Figura 19. Modelo de Tareas y Objetivos para el objetivo: Identificar las características del Estudiante

El objetivo *orientar al estudiante*, como se muestra en la Figura 20, requiere la ejecución de cinco tareas importantes: (i) comprobar viabilidad del concepto, esto con el fin de presentar el concepto adecuado al estudiante, (ii) proporcionar contenidos, en esta tarea no se presenta el contenido de los conceptos, (iii) proporcionar examen al estudiante, esta tarea se lleva a cabo con el fin de evaluar el conocimiento que tiene el estudiante del tema, pero antes se debe comprobar que el estudiante puede presentar el examen, (iv) proporcionar examen al profesor, esta tarea se lleva a cabo para notificar al profesor que tiene exámenes pendientes por corregir y (v) aconsejar al estudiante.

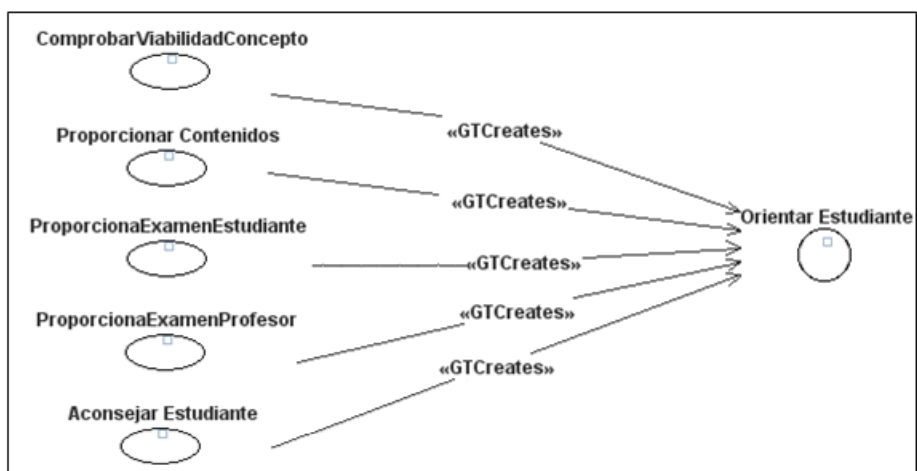


Figura 20. Modelo de Tareas y Objetivos para el objetivo: Orientar al Estudiante

En la Figura 21 se presentan las tareas necesarias para lograr Evaluar el progreso del estudiante. Para lograr este objetivo se requiere realizar las siguientes tareas: (i) Presentar preguntas cortas, esta actividad se lleva a cabo una vez el estudiante termine de estudiar un concepto, (ii) Corregir respuestas, (iii) Determinar nivel de conocimiento, esta actividad se lleva a cabo después de corregir las respuestas del estudiante a las preguntas cortas, (iv) Monitorear tiempo, con esta actividad el sistema puede determinar la velocidad de aprendizaje del estudiante y (v) Cambiar estrategia de enseñanza, esta actividad solo se ejecuta cuando el sistema detecta que el estudiante no adquiere el nivel de conocimiento esperado en algún concepto .

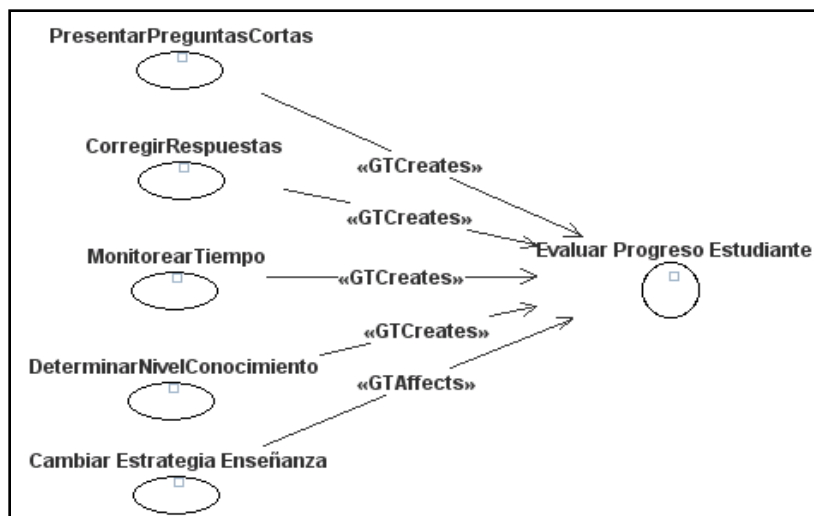


Figura 21. Modelo de Tareas y Objetivos para el objetivo: Evaluar Progreso Estudiante

En la Figura 22 se presentan las tareas necesarias para lograr Actualizar el modelo del estudiante. Para lograr este objetivo se requiere actualizar el nivel de conocimiento adquirido por el estudiante en cada concepto del tutorial. Con los resultados de esta tarea el agente Tutor podrá tomar decisiones que permitan orientar al estudiante de una forma adecuada.



Figura 22. Modelo de Tareas y Objetivos para el objetivo: Actualizar Modelo del Estudiante

5.3.3 Diagrama de Clases del sistema

Después de completar el modelo de la organización, modelo de agentes y el modelo de tareas y objetivos, se realiza el diseño del diagrama de clases. En este trabajo se ha tomado como punto de partida el diagrama de clases del sistema SITUA [91], al cual se le han adicionado las clases asociadas al dominio del problema, las cuales fueron identificadas a partir de los casos de uso.

La Figura 23 muestra el diagrama de clases del sistema. Las clases que se encuentran resaltadas son aquellas que se incluyeron en el diagrama.

La clase *Estilo* como su nombre lo indica, representa el estilo de aprendizaje que identifica a un determinado estudiante. En esta clase se incluyen los dos estilos que hayan obtenido el mejor puntaje como resultado del test CHAEA[106]. Esta información permite al sistema variar sus estrategias de acuerdo a los estilos principal y opcional, almacenados. Así mismo, se almacena el estado del test. El dato correspondiente al estado, permite al sistema verificar si un estudiante ha concluido o no el cuestionario.

La clase *TestInicial*, representa el nivel de conocimiento con el cual un estudiante inicia su aprendizaje. Aquí se incluye el estado del test de conocimiento inicial para que el sistema pueda verificar si el estudiante a terminado a no, de responder el test. El atributo de *calificación* permite obtener información acerca de qué tanto conocimiento tiene el estudiante antes de iniciar su proceso de aprendizaje.

La clase *Preguntas test inicial*, representa el test de conocimiento inicial. Aquí se incluyen todas las preguntas de selección de una forma estructurada de acuerdo a sus respuestas.

La clase *Pregunta test inicial*, representa cada pregunta de selección que hace parte del test de conocimiento inicial, es decir, almacena solamente los enunciados de cada pregunta.

La clase *Respuesta test inicial*, representa cada respuesta que forma parte de una pregunta de selección del test de conocimiento inicial.

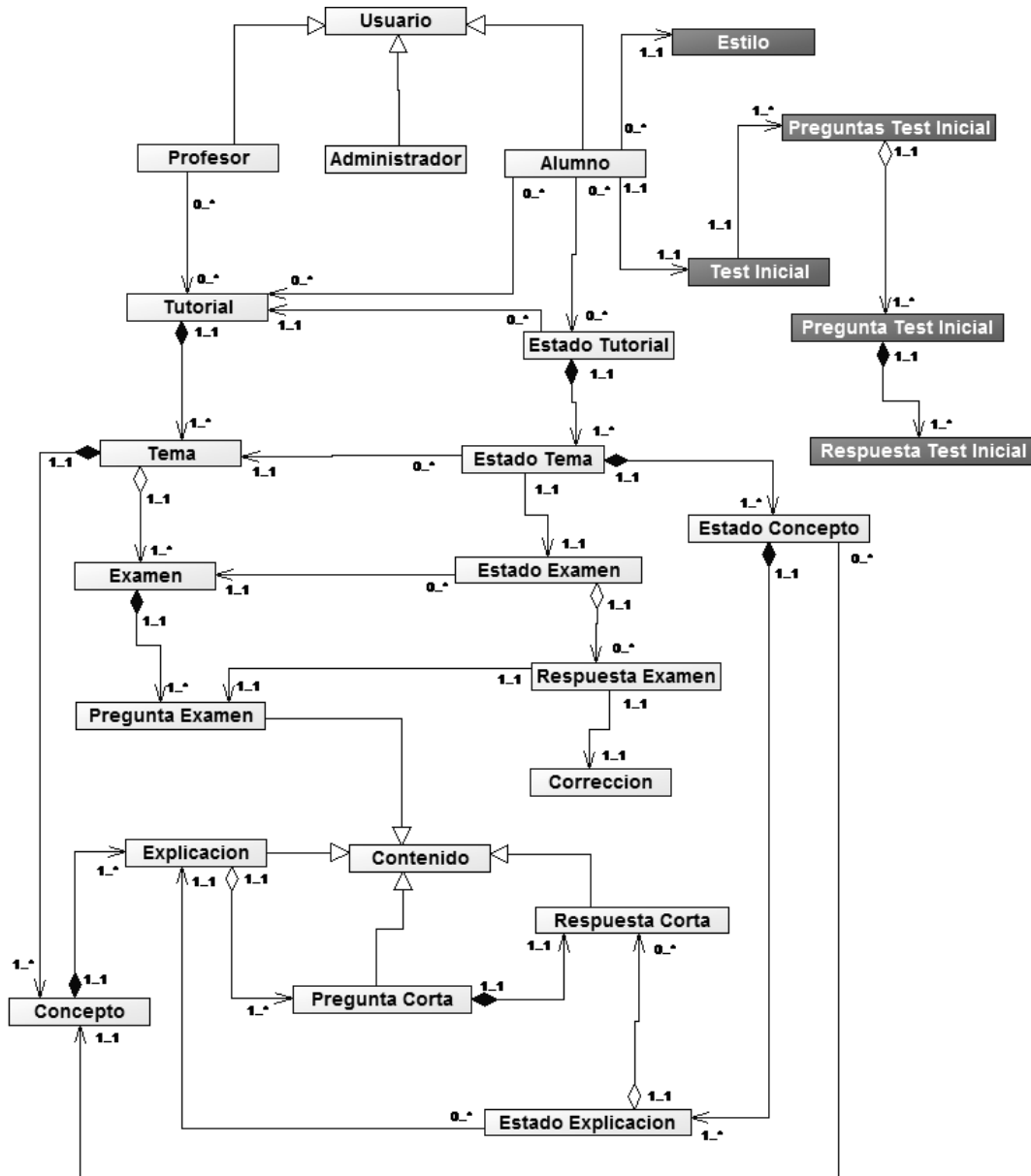


Figura 23. Diagrama de Clases del Sistema

En el ANEXO C, se describen los detalles de cada clase correspondiente al diagrama presentado anteriormente.

5.3.4 Patrones de Diseño

Para el desarrollo del prototipo se ha optado por utilizar el patrón de diseño **MVC** (Modelo Vista Controlador), patrón ampliamente utilizado y de eficacia probada en el desarrollo de aplicaciones Web [114].

En el patrón MVC, el **controlador** es el encargado de redirigir o asignar un elemento que desempeña una funcionalidad concreta (un modelo) a cada petición; el controlador debe poseer de algún modo, un mapa de correspondencias entre peticiones y elementos (modelo) que se les asignan.



El **modelo** son los elementos que desempeñan la funcionalidad para la que se ha desarrollado el sistema. Forman la *lógica de negocio* del sistema. Cuando el *modelo* realiza las operaciones necesarias para atender una petición, el flujo de control es devuelto al *controlador* y éste devuelve los resultados a la **vista** asignada.

Para el desarrollo del sistema, se utilizó el framework **Struts** [115][116], que es una de las implementaciones del patrón MVC más utilizadas en el ámbito del desarrollo de aplicaciones J2EE.

Teniendo en cuenta que la totalidad de los datos se almacenan en una base de datos, es importante definir muy bien el sistema de acceso a la misma para obtener el correcto funcionamiento del sistema. Por esta razón, para implementar el sistema de acceso a la base de datos, se hizo uso del patrón de diseño **DAO (Data Access Object)** de la plataforma J2EE [117][118]. Este patrón de diseño es de uso frecuente en el ámbito de las aplicaciones J2EE, es fácil de implementar y se destaca por su flexibilidad a la hora de migrar la aplicación de un SGBD a otro.

Para conocer más detalles acerca de la implementación de los patrones mencionados, refiérase al ANEXO C.

5.4 IMPLEMENTACIÓN

5.4.1 Diagrama de componentes

En este trabajo se ha tomado como punto de partida el diagrama de componentes del sistema SITUA [91], indicando la relación entre los ficheros de código y componentes de desarrollo.

Cada componente tiene una función y un objetivo bien definido. Para cumplir este objetivo los componentes cuentan con un conjunto de clases que lo implementan. En la Figura 24 se muestra el diagrama de componentes, en donde *Estilos*, *dao.Estilos*, *testInicial*, *dao.TestInicial*, *Inicialización* y *Agentes*, fueron los paquetes que se modificaron y adicionaron al modelo existente para incluir en ellos las clases que permiten implementar la lógica para definir los estilos de aprendizaje de cada estudiante, el conocimiento inicial, el proceso de inicialización del modelo del estudiante y el comportamiento de los agentes.

Descripción de Paquetes

A continuación se tratará cada uno de los componentes resaltados en la Figura 24, por separado, para ver la funcionalidad que satisfacen. Los demás paquetes se describen con más detalle en el ANEXO D.

Paquete *testInicial*

El paquete *testInicial* contiene las clases que almacenan los datos correspondientes al conocimiento con el que inicia cada estudiante, así como también el estado en el cual se encuentra el test en cualquier momento. Al mismo tiempo contiene el paquete *testInicial.actions* donde se incluyen las clases del controlador encargadas de gestionar las peticiones procedentes de los usuarios del sistema. Las peticiones que gestionan estas clases son las que están relacionadas con la realización del test de conocimiento

inicial, así como también las operaciones que permiten analizar los resultados para el posterior registro de usuarios en los tutoriales.

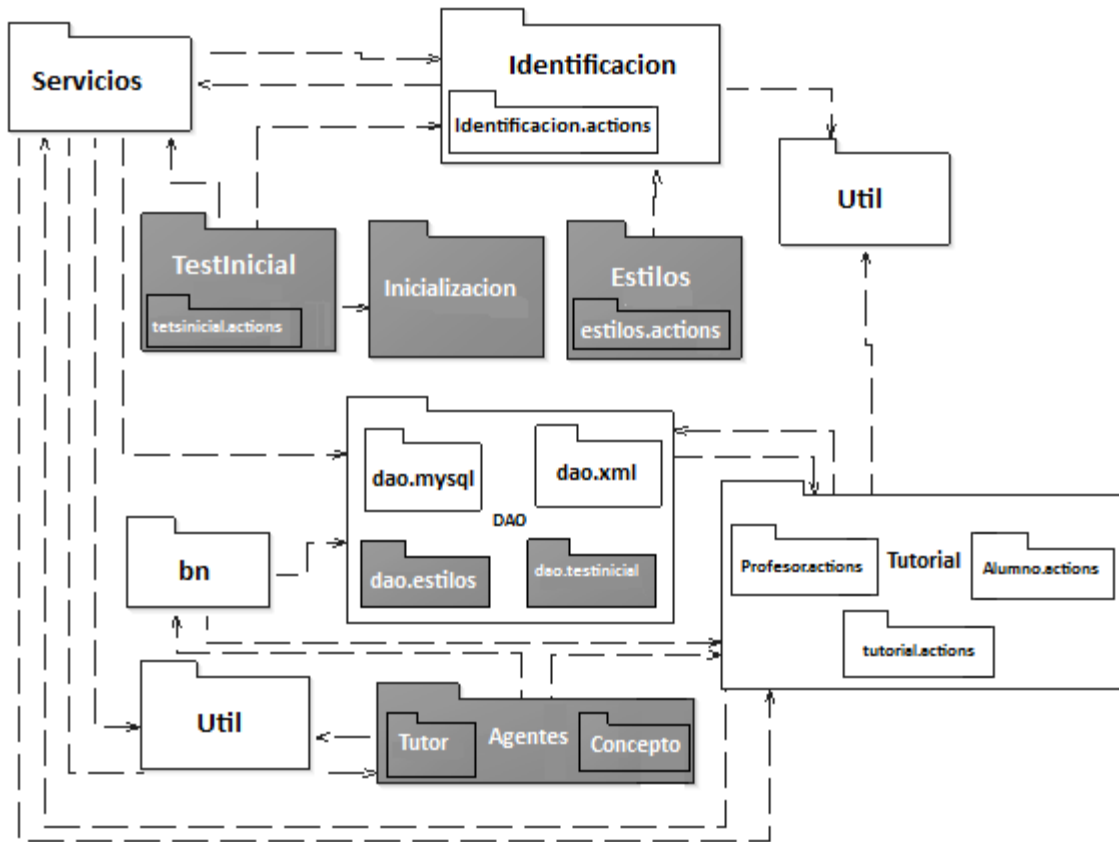


Figura 24. Diagrama de Componentes

Las clases contenidas en el paquete *testInicial* son:

TestInicial
testInicialForm
PreguntaTestInicial
RespuestaTestInicial

Las clases contenidas en el paquete *testInicial.actions* son:

GestionTestInicial
testInicialAction
ImportarPreguntasTestInicial
ObtenerPreguntasTestInicial
MostrarPreguntasTestInicial
ObtnerRespuestasTestInicial

Paquete dao.TestInicial



Las clases contenidas en este paquete están relacionadas con el acceso a la base de datos del sistema e implementan los métodos definidos en los interfaces del paquete *dao*. Estas clases permiten realizar operaciones de inserción, borrado y modificación de la información de los resultados del test de conocimiento inicial y el estado en el que se encuentra el mismo. De igual forma permiten acceder a los datos del test que se encuentran almacenados en la base de datos.

Las clases contenidas en el paquete *dao.TestInicial* son:

TestInicial

TestInicialDAO

TestInicialDAOImpl

PreguntaTestInicialDAO

RespuestaTestInicialDAO

Paquete Estilo

El paquete *Estilos* contiene las clases que almacenan los datos correspondientes al estilo de aprendizaje de cada estudiante, así como también el estado en el cual se encuentra el test de CHAEA en cualquier momento. Al mismo tiempo contiene el paquete *Estilos.actions* donde se incluyen las clases del controlador encargadas de gestionar las peticiones procedentes de los usuarios del sistema. Las peticiones que gestionan estas clases son las que están relacionadas con la realización del test de CHAEA, así como también las operaciones que permiten analizar los resultados para obtener el estilo predominante y el opcional para cada estudiante.

Las clases contenidas en el paquete *Estilos* son:

Test

TestForm

Las clases contenidas en el paquete *Estilos.actions* son:

GestiónEstilos

TestAction

Paquete dao.Estilos

Las clases contenidas en este paquete están relacionadas con el acceso a la base de datos del sistema e implementan los métodos definidos en los interfaces del paquete *dao*. Estas clases permiten realizar operaciones de inserción, borrado y modificación de la información de los resultados del test de CHAEA y el estado en el que se encuentra el mismo. De igual forma permiten acceder a los datos del test que se encuentran almacenados en la base de datos.

Las clases contenidas en el paquete *dao.Estilos* son:

Estilo

EstiloDAO

EstiloDAOImpl

Paquete Inicializacion



El paquete *Inicialización* incluye todas aquellas clases que permiten inicializar el modelo del estudiante. Para obtener el modelo inicial del estudiante, se determina el nivel de conocimiento que tiene el estudiante antes de iniciar el tutorial y teniendo en cuenta la estructura de los contenidos del tutorial se procede a inicializar el modelo del estudiante.

El diagrama de componentes completo y el diagrama de despliegue, se presenta de manera más detallada en el ANEXO D.

5.5 PRUEBAS DE SOFTWARE

El proceso de las pruebas software se basa en el prototipo del sistema tutor inteligente para aprendizaje en salud desarrollado. Mediante pruebas de unidad se evalúan las funciones que debe soportar el prototipo software para dar apoyo a los servicios que proporciona.

Los elementos software evaluados fueron los paquetes de estilos, test inicial, identificación y tutorial que componen el prototipo software, de acuerdo a las siguientes características:

- Funcionalidad
- Interfaz de usuario

Para poder verificar cada una de las características se efectuaron pruebas de unidad. Su planteamiento y desarrollo se especifican en el ANEXO D.

La realización de las pruebas de unidad estuvo basada en el estándar IEEE 829-1998 [119], de esta forma se partió de un conjunto estable de parámetros a evaluar, para los cuales ya existía una definición aceptada como estándar.

5.6 MANUAL DE USUARIO Y DE INSTALACIÓN

En el manual de usuario se explican las funcionalidades del Sistema Tutor Inteligente SIAS, que se crearon o modificaron como parte de los requerimientos de este proyecto. Debido a que existen tres tipos de usuarios en SIAS, se describen por separado las opciones disponibles para los **estudiantes**, **profesores** y **administradores**. El manual de usuario se encuentra en el ANEXO F.

SIAS contiene funcionalidades que requieren una previa instalación de herramientas necesarias para su ejecución y que constituyen la base del entorno donde fue creado dicho sistema. En el manual de instalación se describen los pasos a seguir para la correcta instalación y posterior utilización del software mencionado. Para conocer los detalles de este manual, refiérase al ANEXO F.



6. EVALUACIÓN DEL SISTEMA TUTOR PROPUESTO FRENTE A UN SISTEMA DE ENSEÑANZA TRADICIONAL

6.1 DEFINICIÓN DE INDICADORES DE EVALUACIÓN

Los sistemas de enseñanza tradicional asistida por computador, se caracterizan por mostrar el conocimiento de una manera lineal. Es decir, ningún factor puede cambiar el orden de enseñanza establecido en su momento por el instructor. En este sentido, induce en el estudiante la respuesta correcta mediante una serie de estímulos que han sido cuidadosamente planificados [120]. Por lo anterior, en este trabajo se ha considerado la plataforma Moodle como un sistema que apoya los métodos de enseñanza tradicional.

En los últimos años, se han realizado diferentes estudios comparativos sobre evaluación de plataformas e-learning. Por ejemplo: Graf y List [121] evalúan la capacidad de adaptación de nueve plataformas e-learning. El trabajo de Monti y San Vicente [122], considera tres categorías para evaluar diferentes plataformas, las categorías son: didáctico-funcional, tecnológica y financiera. También, expone las experiencias de orientar un curso de aprendizaje lingüístico en modalidad e-learning. Al-Ajlan y Zedan [123], describen las fortalezas y limitaciones de la plataforma moodle frente a otras plataformas. En dichos estudios, la plataforma Moodle ha presentado los mejores resultados. Teniendo en cuenta lo anterior, en este trabajo se decidió comparar el sistema tutor desarrollado SIAS frente a Moodle por ser una plataforma representativa, lo que permite determinar el aporte obtenido con el uso de STI frente a sistemas de enseñanza tradicional asistido por computador.

Para definir los indicadores de evaluación, se realizó un estudio sobre trabajos relacionados con evaluación de plataformas virtuales y sistemas e-learning. Algunos de estos trabajos se mencionan a continuación: Martínez y Gallego [124] presentan el e-learning como una alternativa frente a la enseñanza tradicional, analizan la relación entre edad, género, conocimientos informáticos, experiencia previa en programas e-learning, tamaño del grupo y variedad de medios con la satisfacción de los usuarios de diferentes programas. El estudio realizado por Pichastor, Nieto y Navarro [20] evalúa las características del entorno virtual Human Site¹, para lo cual considera la motivación intrínseca, el aprendizaje y el disfrute. Además, examina el papel de la auto-eficacia y el valor percibido del entorno. Jiménez [125], plantea una propuesta en la cual pretende evaluar la satisfacción del estudiante de cursos de formación continua teniendo en cuenta las características persono-lógicas, capacidad de adaptación a las necesidades y calidad de formación y evalúa al estudiante desde tres puntos de vista: estudiante, trabajador y cliente.

En el trabajo de Arias [126], se clasifica el material virtual y/o multimedia como: plataformas e-learning, cursos virtuales, unidades didácticas y programas de apoyo y se desarrolla un cuestionario común basado en indicadores para evaluar cada tipo de sistema. Así mismo, se define un conjunto de indicadores los cuales determinan qué

¹ Entorno Virtual de Aprendizaje Interactivo, aplicado a asignaturas relacionadas con el factor humano.



preguntas aplicar para evaluar cada tipo de sistema. Estos indicadores se agrupan en cinco categorías: Calidad Pedagógica, Calidad Técnica, Gestión, Usabilidad y Valoración General. La escala de valoración es de 0.0 a 4.0, siendo 0.0 la puntuación más baja y 4.0 la más alta puntuación que un sistema puede obtener, tanto en la valoración general como en cualquier categoría o indicador.

Teniendo en cuenta que la fiabilidad y pertinencia de los cuestionarios desarrollados por Arias [126] han sido validadas por diferentes expertos en informática, psicopedagogía, enseñanza primaria, entre otros [127]. En este trabajo se ha decidido adecuar uno de estos cuestionarios, específicamente el cuestionario de evaluación de cursos virtuales teniendo en cuenta principalmente la categoría pedagógica y la categoría de usabilidad. No obstante, para cada categoría se evalúan todos los indicadores definidos en el cuestionario original y se mantiene la misma escala de valoración. A continuación se presenta la definición de cada categoría según Arias [126]:

- **Calidad Pedagógica:** En esta categoría se debe medir las pretensiones del sistema, la estructura de los contenidos, las actividades, el sistema de evaluación, el aprendizaje modular etc.
- **Usabilidad:** Esta categoría medirá indicadores que hagan referencia a la facilidad de navegación, interacción con el usuario, calidad estética, etc.

Los indicadores definidos para evaluar las categorías mencionadas anteriormente son:

- **Guía didáctica:** Evalúa si están claras las pretensiones del curso, unidad didáctica o programa de apoyo, de forma que se pueda usar fácilmente por alguien que no sea el autor del sistema
- **Metodología:** Evalúa la calidad didáctica de los elementos básicos que deben estar definidos: objetivos, contenidos, actividades y evaluación.
- **Organización de los contenidos:** Evalúa la arquitectura de los contenidos que se muestran.
- **Calidad de los contenidos:** Evalúa el grado de adecuación de los conocimientos que se explican dentro del sistema.
- **Recursos didácticos:** Evalúa la versatilidad que tiene el sistema para enseñar lo mismo de distintas formas.
- **Capacidad de motivación:** Evalúa el grado de motivación que puede lograr el sistema.
- **Elementos multimedia:** Evalúa los elementos multimedia usados en el sistema.
- **Estilo del lenguaje:** Evalúa la sintaxis y semántica de los textos usados.
- **Discriminación y valores:** Evalúa el grado de discriminación y valores de todo el sistema.
- **Singularidad del usuario:** Evalúa si el sistema se puede acoplar a las características y circunstancias personales del Usuario final.
- **Facilidad de navegación:** Al hacer referencia a la facilidad de navegación sobre una plataforma o curso, se hace en realidad sobre la cantidad de problemas que ese material ayuda a resolver al alumno en relación con su proceso de auto estudio.
- **Legibilidad de la información:** Dado que la mayoría de la interface de usuario llegará a este a través de la lectura del texto, consideramos un indicador necesario el evaluar la visualización del texto.
- **Calidad estética del interface General:** Evalúa los elementos básicos de la interface de usuario, a saber: iconos, menús, formularios, barras de navegación, enlaces, títulos, ventanas, botones, espacio de texto e imágenes



La Tabla 3, presenta los indicadores asociados a la categoría Calidad Pedagógica y Usabilidad de los cursos virtuales.

Los aspectos que se van a evaluar en la plataforma Moodle y en el prototipo SIAS, son: (i) nivel de conocimiento de cada estudiante, (ii) la velocidad de aprendizaje, (iii) las estrategias de enseñanza utilizadas y (iv) la usabilidad del sistema. Es importante aclarar que la adecuación del cuestionario propuesto por Arias, se utiliza para evaluar solo dos de los cuatro aspectos mencionados: las *estrategias en enseñanza* y la *usabilidad* del sistema. Este cuestionario se aplica a los estudiantes del curso orientado a través de la plataforma de enseñanza *Moodle* y a los estudiantes del curso orientado usando el prototipo del sistema tutor desarrollado SIAS.

Categoría	Nombre Indicador
Calidad Pedagógica	Guía didáctica
	Metodología
	Organización de los contenidos
	Calidad de los contenidos
	Recursos didácticos
	Capacidad de motivación.
	Elementos multimedia
	Estilo del lenguaje
	Discriminación y valores
	Singularidad del usuario
Usabilidad	Facilidad de navegación
	Legibilidad de la información
	Calidad estética de la interfaz general

Tabla 3. Relación Categoría-Indicador

El diseño de los cuestionarios para evaluar cada aspecto mencionado, se presenta en el Anexo E.

6.2 DISEÑO DEL CURSO “ESTRATEGIA AIEPI: MÓDULO RECIÉN NACIDO”

En esta sección se describe el diseño del curso “Estrategia AIEPI: Módulo Recién Nacido” en la plataforma Moodle y en el sistema tutor inteligente SIAS.

6.2.1 Especificación de la estructura del curso en la plataforma Moodle.

Moodle es una plataforma de formación académica que promueve un enfoque pedagógico constructivista social [128]. Esta plataforma sirve como herramienta de soporte a los docentes en la creación de comunidades de aprendizaje en línea y ofrece diferentes actividades para los cursos existentes en la plataforma [22][129].

Por otro lado, moodle se ha convertido en una de las plataformas más utilizada por la comunidad educativa para impartir diferentes cursos en distintos idiomas [130]. Diferentes estudios con distinto nivel de complejidad se han realizado para evaluar

diferentes plataformas entre las que se encuentra moodle. Sin embargo, muchos de estos estudios son generales y no han considerado un ámbito de aprendizaje específico [131].

El proceso de diseño del curso virtual utilizando la plataforma, se llevó a cabo en tres etapas. En la primera se seleccionó la población objetivo. En la segunda, se estableció el proceso de aprendizaje, se definieron los objetivos de aprendizaje y el material educativo. Finalmente en la tercera etapa, se hizo la propuesta de evaluación y la puesta en marcha del curso.

En la Figura 25 se presenta la población objetivo, que estuvo conformada por 22 estudiantes de noveno semestre del programa de enfermería de la Universidad del Cauca. De los cuales, 72,7% pertenece al género femenino y 27.3% pertenece al género masculino. La edad promedio de los estudiantes fue de 23 años, con un rango de edad entre los 20 y 29 años. Éste grupo de estudiantes en cursos anteriores, ya habían adquirido una experiencia previa usando el entorno virtual de aprendizaje EVA [7].



Figura 25. Estudiantes de noveno semestre del programa de Enfermería

Por otra parte, el contenido del curso se estructuró por temas teniendo en cuenta las sugerencias de profesores expertos en el dominio, los cuales participaron en el proceso de diseño. Cada tema tiene asociados diferentes materiales educativos, tales como: texto, fotografías y videos. Además, con la ayuda del profesor encargado de orientar el curso, se planteó la propuesta de evaluación para determinar el nivel de conocimiento adquirido por los estudiantes.

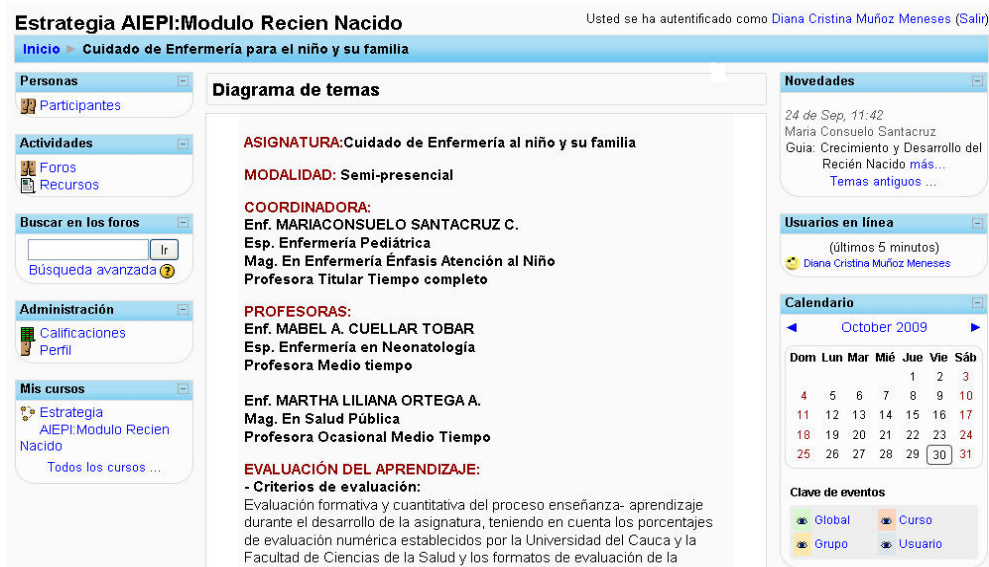
La Figura 26 muestra la información general del curso una vez terminado su diseño. La estructura de los contenidos se dividió en 6 temas principales: (i) introducción, (ii) crecimiento y desarrollo del recién nacido, (iii) evaluar y clasificar al lactante menor de dos meses, (iv) determinar el tratamiento, (v) reanimación y (vi) tratar al niño de cero a cinco años. Donde cada tema tiene su respectivo material de estudio y recursos audiovisuales.

6.2.2 Especificación de la estructura del curso en el Sistema Tutor Inteligente SIAS.

SIAS es el prototipo inicial de sistema tutor inteligente desarrollado para capacitar a diferentes estudiantes de ciencias de la salud en la estrategia AIEPI Clínico. SIAS hace uso de tests previos para determinar el nivel de conocimiento de cada estudiante al iniciar el tutorial y aplica el test de CHAEA [106] para determinar el estilo de aprendizaje de cada

estudiante. Con la información obtenida, SIAS determina la mejor forma de presentar el contenido de acuerdo al perfil de cada estudiante.

El proceso de diseño del tutorial se realizó siguiendo las tres fases planteadas en el apartado anterior. En la primera se seleccionó la población objetivo. En la segunda, se estableció el proceso de aprendizaje, se definieron los objetivos de aprendizaje y el material educativo. Finalmente en la tercera etapa, se hizo la propuesta de evaluación y la puesta en marcha del curso.



The screenshot shows a web interface for a virtual course. At the top, it says 'Estrategia AIEPI: Módulo Recién Nacido' and 'Usted se ha autenticado como Diana Cristina Muñoz Meneses (Salir)'. Below this is a navigation bar with 'Inicio' and 'Cuidado de Enfermería para el niño y su familia'. The main content area is titled 'Diagrama de temas' and lists course details: 'ASIGNATURA: Cuidado de Enfermería al niño y su familia', 'MODALIDAD: Semi-presencial', 'COORDINADORA: Enf. MARIACONSUELO SANTACRUZ C. Esp. Enfermería Pediátrica Mag. En Enfermería Énfasis Atención al Niño Profesora Titular Tiempo completo', and 'PROFESORAS: Enf. MABEL A. CUELLAR TOBAR Esp. Enfermería en Neonatología Profesora Medio tiempo' and 'Enf. MARTHA LILIANA ORTEGA A. Mag. En Salud Pública Profesora Ocasional Medio Tiempo'. It also includes 'EVALUACIÓN DEL APRENDIZAJE: - Criterios de evaluación: Evaluación formativa y cuantitativa del proceso enseñanza- aprendizaje durante el desarrollo de la asignatura, teniendo en cuenta los porcentajes de evaluación numérica establecidos por la Universidad del Cauca y la Facultad de Ciencias de la Salud y los formatos de evaluación de la'. On the left, there are navigation menus for 'Personas', 'Actividades', 'Buscar en los foros', 'Administración', and 'Mis cursos'. On the right, there are sections for 'Novedades', 'Usuarios en línea', and 'Calendario'.

Figura 26. Estructura del curso virtual: Estrategia AIEPI: Módulo Recién Nacido

Para definir la población objetivo, se eligió un grupo de 18 estudiantes de décimo semestre del programa de medicina de la Universidad del Cauca (ver Figura 27). De los cuales, 44,4% pertenecen al género femenino y 55.6% pertenecen al género masculino. La edad promedio de los estudiantes fue de 24 años, con un rango de edad entre los 23 y 30 años.



Figura 27. Estudiantes de décimo semestre del programa de Medicina



En la definición del proceso de aprendizaje y material educativo, participaron profesores expertos en la estrategia AIEPI clínico, específicamente pediatras especialistas en la implementación de la estrategia AIEPI para recién nacidos. El tutorial se compone de cuatro módulos: módulo del estudiante, módulo del experto, módulo del tutor y módulo del entorno. A continuación se describe el proceso de diseño de cada módulo del tutorial:

Módulo del estudiante: para modelar al estudiante se hizo uso de una red bayesiana, la cual permite representar cada uno de los elementos de la estructura de contenidos del tutorial y facilita el proceso de diagnóstico sobre la evolución del estudiante en el tutorial.

Esta red se compone de 9 nodos de concepto:

1. Evaluar y clasificar al menor de dos meses.
2. Evaluar y determinar riesgo durante el embarazo que afecta el bienestar fetal.
3. Primera valoración del recién nacido.
4. Determinar si tiene enfermedad grave o infección local.
5. Diarrea.
6. Problemas de alimentación.
7. Otros problemas.
8. Evaluar el desarrollo.
9. Estado de vacunación.

Con los resultados obtenidos por cada estudiante en el test de conocimiento inicial y las restricciones en el itinerario de estudio, se realiza el proceso de inicialización de los nodos de la red bayesiana.

Módulo del experto: considerando que los estudiantes tienen diferentes estilos de aprendizaje, fue necesario diseñar para cada estilo un tutorial con estrategias de enseñanza acordes a las características de cada estilo. No obstante, el contenido en todos los casos, se estructuró por temas. Cada tema tiene asociados diferentes conceptos y un examen formado por preguntas abiertas relacionadas con el contenido del tema.

Los conceptos están formados por una serie de explicaciones, las cuales se presentan de acuerdo a la estrategia de enseñanza seleccionada. Las explicaciones se presentan de acuerdo a un orden preestablecido dentro del concepto con formato HTML. Al terminar el estudio de cada explicación, el sistema presenta un conjunto de preguntas cortas para determinar el nivel de conocimiento adquirido por los estudiantes. Sin embargo, algunas explicaciones pueden no contener preguntas cortas.

En la Tabla 4 se presenta la estructura del tutorial. El tutorial se denominó Estrategia AIEPI: Módulo Recién Nacido, el cual contiene un único tema conformado por nueve conceptos. Cada concepto está conformado por un número variable de explicaciones. Con la ayuda de los expertos en el dominio se determinó el orden de presentación de los conceptos y las restricciones. En este sentido, el concepto 1 es prerrequisito del concepto 2 y el concepto 2 es prerrequisito para el concepto 3. El concepto 3 es prerrequisito para el concepto 4, concepto 6, concepto 7, concepto 8 y concepto 9. El concepto 4 es prerrequisito del concepto 5.

En el Anexo B, se presentan ejemplos de contenidos estructurados de acuerdo a las diferentes estrategias de enseñanza.



Módulo del tutor: se estructura como un sistema multi-agente formado por dos tipos de agente: el agente tutor y el agente concepto. Las tareas de estos agentes se detallan en el apartado 4.3.

Estrategia AIEPI: Módulo Recién Nacido		
Tema 1: Evaluar y clasificar al lactante menor de 2 meses	Concepto 1: Evaluar y clasificar al menor de 2 meses	Evaluar y clasificar al lactante menor de 2 meses.
	Concepto 2: Evaluar y determinar riesgo durante el embarazo	Evaluar riesgo durante el embarazo
		Determinar riesgo durante el embarazo
		Clasificar el riesgo de embarazo que afecta el bienestar fetal
		Estudio de Casos
	Concepto 3: Primera valoración del recién nacido	Cómo realizar la primera valoración del recién nacido
		Determinar la edad gestacional
		Clasificación por peso y edad gestacional
		Cómo clasificar la primera evaluación del recién nacido
	Concepto 4: Determinar si tiene enfermedad grave o infección local	Cómo evaluar a un niño con enfermedad muy grave o infección local (parte I)
		Cómo evaluar a un niño con enfermedad muy grave o infección local (parte II)
		Cómo clasificar a un niño con enfermedad muy grave o infección local
	Concepto 5: Diarrea	Cómo evaluar a un niño menor de 2 meses con diarrea
		Cómo clasificar la diarrea
Concepto 6: Problemas de alimentación	Cómo evaluar a un niño para detectar bajo peso	
	Evaluar el amamantamiento	
	Cómo clasificar la alimentación	
Concepto 7: Otros problemas	Evaluar otros problemas	
Concepto 8: Evaluar el desarrollo	Cómo evaluar a un niño	
	Cómo clasificar los problemas de desarrollo	
Concepto 9: Estado de vacunación	Evaluar estado de vacunación	

Tabla 4. Estructura del Tutorial

Módulo del entorno: la interfaz de usuario se diseñó teniendo en cuenta tres tipos de usuarios, con el objetivo de facilitar a cada tipo de usuario el desarrollo de las actividades. A manera de ejemplo, en la Figura 28 se presenta la interfaz para el usuario profesor.

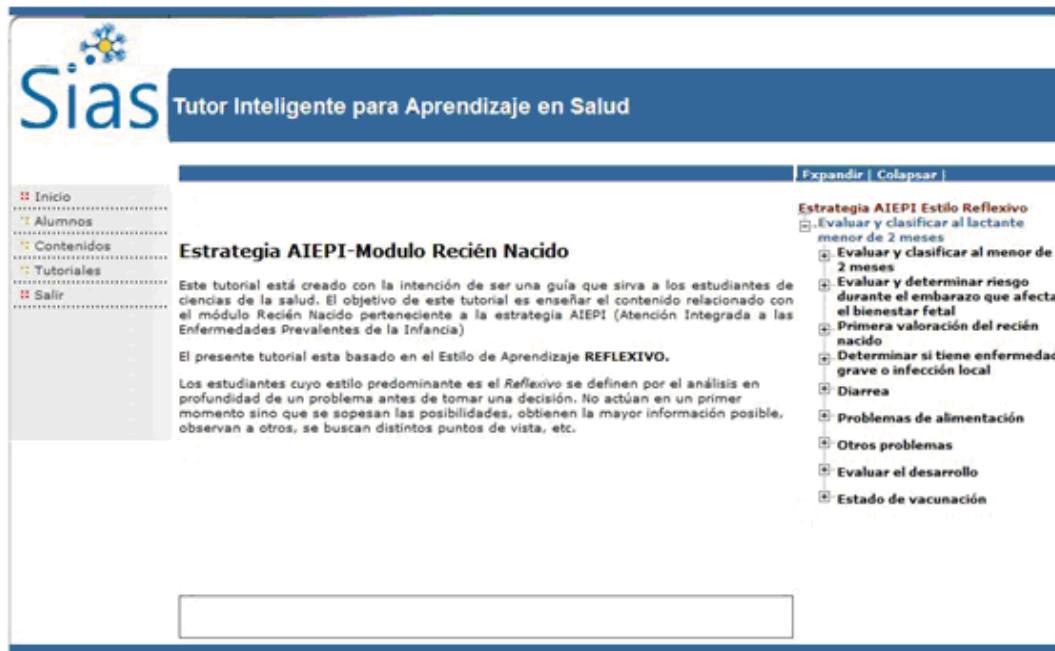


Figura 28. Interfaz para el Profesor

6.3 EVALUACIÓN DE LA PLATAFORMA MOODLE EN LA ENSEÑANZA DE UN CURSO DE CAPACITACIÓN EN AIEPI

Después de concluir la etapa de diseño del curso, se realizó una sesión presencial con los estudiantes en la que se hizo la inducción a la plataforma y el proceso de registro. Así mismo, se les proporcionó una guía sobre el funcionamiento básico de la plataforma. Durante el desarrollo del curso, la comunicación con los estudiantes se estableció por medio de correo electrónico y el chat que proporciona la plataforma moodle. Por otro lado, se usó el módulo *cuestionario* para evaluar el tema reanimación neonatal y pediátrica.

Con el objetivo de evaluar la plataforma Moodle en la enseñanza del curso “**Estrategia AIEPI: Módulo Recién Nacido**”, se tienen en cuenta tres de los cuatro aspectos mencionados anteriormente en el apartado 6.1: (i) el nivel de conocimiento adquirido por los estudiantes al iniciar y finalizar el curso virtual, (ii) las estrategias de enseñanza utilizadas y (iii) la usabilidad de la plataforma. A continuación se presenta el análisis de los resultados obtenidos en cada aspecto anteriormente descrito.

6.3.1 Evaluación Nivel de Conocimiento

Para determinar el nivel de conocimiento adquirido por los estudiantes durante el desarrollo del curso, se aplicó un test preliminar llamado *Test de conocimiento inicial* y posteriormente, se aplicó un *Test de conocimiento final*. El primer test permitió determinar el nivel de conocimiento de los estudiantes al iniciar el curso, y el segundo, para determinar el nivel de conocimiento de los mismos estudiantes pero, al finalizar el curso.

El test de conocimiento inicial estuvo conformado por 9 preguntas de selección, las cuales fueron redactadas teniendo en cuenta el contenido del curso y bajo la supervisión del profesor encargado de orientarlo. Así mismo, con la asesoría del profesor se asignó el puntaje para cada pregunta. Es importante mencionar que el test de conocimiento inicial se realizó utilizando el modulo *cuestionarios* de la plataforma moodle. No obstante, la plataforma no tiene en cuenta el nivel de conocimiento previo de los estudiantes en el dominio a enseñar, ni las características de aprendizaje, por lo tanto todos los estudiantes inician el curso con el mismo nivel de conocimiento y los contenidos son presentados de la misma forma.

Por otra parte, las preguntas que conformaron el test de conocimiento final fueron 13 preguntas abiertas, las cuales fueron planteadas y diseñadas de manera diferente a las preguntas planteadas en el test de conocimiento inicial pero, conservando en cierta medida el mismo nivel de complejidad. Por la estructura de algunas preguntas éste test, no se realizó haciendo uso de Moodle.

La escala de calificación usada en el test de conocimiento inicial y final fue de 0.0 a 5.0, siendo 0.0 la calificación más baja y 5.0 la más alta. Además, se considera que el estudiante aprueba el test con una calificación mayor o igual a 3.0.

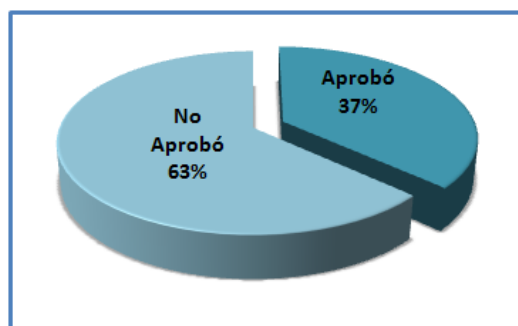


Figura 29. Resultados Test Conocimiento Inicial

De los 22 estudiantes inscritos en el curso, 19 presentaron el test de conocimiento inicial y 3 no presentaron la prueba. En la Figura 29, se observa que de los 19 estudiantes que presentaron el test inicial, el 37% aprobó el test y el 63% lo reprobó. Teniendo en cuenta lo anterior, se puede concluir que todos los estudiantes inician el curso con diferente nivel de conocimiento.

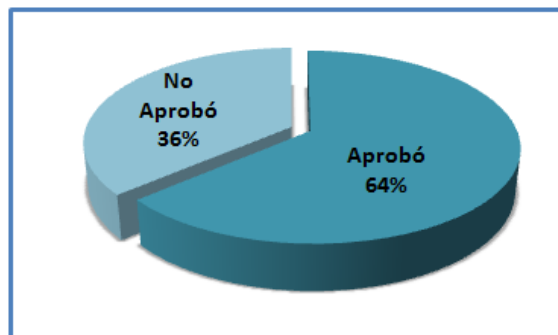


Figura 30. Resultados Test Conocimiento Final

El test de conocimiento final fue presentado por todos los estudiantes inscritos en el curso. Los resultados presentados en la Figura 30, muestran que el 64% de los estudiantes aprobaron el test y el 36% no obtuvieron la calificación mínima para aprobarlo. En la Figura 31, se aprecia que el 55% de las calificaciones oscilan entre 3.0 y 4.0 y solo el 9% de las calificaciones es mayor que 4.0. Además, el 36% de los estudiantes reprobó el test con una calificación inferior a 3.0.

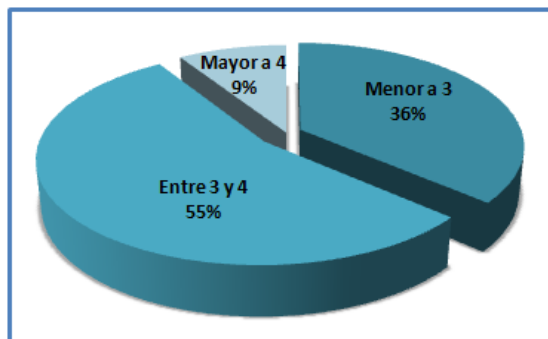


Figura 31. Consolidado Calificaciones Test Conocimiento Final

Por otro lado, es importante destacar que no se aprecia una variación significativa entre la calificación promedio del test de conocimiento inicial, la cual fue de 2,73 y la calificación promedio del test de conocimiento final, que fue de 3,29, lo cual indica que el nivel de conocimiento en cada estudiante no se incrementa significativamente. No obstante, se puede determinar que la utilización de la plataforma ha influido positivamente en el 63.6% de los estudiantes.

6.3.2 Evaluación de las estrategias de enseñanza

Las estrategias de enseñanza utilizadas en el curso, se evaluaron por medio de la categoría Calidad Pedagógica del cuestionario propuesto por Arias [126], el cual se encuentra descrito en el apartado 6.1. Es importante señalar que los estudiantes dieron respuesta a este cuestionario al finalizar el curso.

En la Figura 32 se presentan gráficamente los resultados de la evaluación de las estrategias del curso, teniendo en cuenta los indicadores correspondientes a la categoría *Calidad Pedagógica*.

Los resultados presentados muestran que las principales ventajas al evaluar las estrategias de enseñanza se encuentran en el indicador relacionado con la *guía didáctica* del curso virtual. Así mismo, indican que las principales ventajas de usar la plataforma Moodle, es que permite identificar claramente los objetivos de enseñanza y actividades de evaluación. No obstante, los resultados evidencian que la mayor limitación de la plataforma se encuentra en el indicador *recursos didácticos*². Esto debido a que la plataforma no proporciona diferentes guías de aprendizaje que sean adaptables a las características propias de cada estudiante. Así mismo, los resultados de la evaluación

² Recursos Didácticos: hace referencia a la versatilidad que tiene el sistema para enseñar lo mismo de distintas formas.

indican que se deben mejorar diferentes aspectos relacionados con la calidad y organización de los contenidos, teniendo en cuenta el perfil del estudiante.

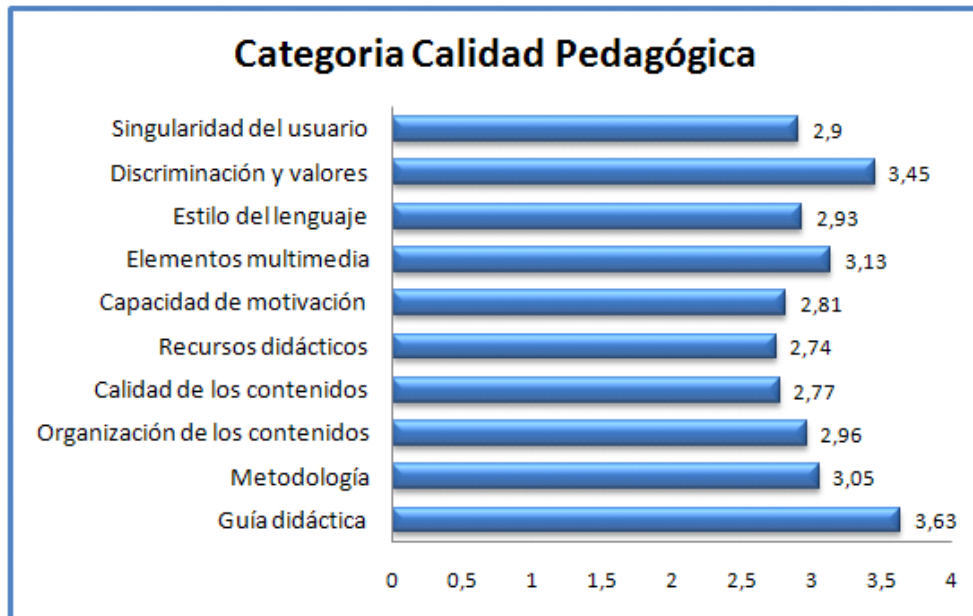


Figura 32. Resultados categoría Calidad Pedagógica discriminados por indicador

6.3.3 Evaluación de usabilidad

La usabilidad del curso, se evaluó por medio de la categoría Usabilidad del cuestionario propuesto por Arias [126], el cual se encuentra descrito en el apartado 6.1.

En la Figura 33 se presentan gráficamente los resultados de la evaluación del curso, teniendo en cuenta los indicadores correspondientes a la categoría *Usabilidad*.

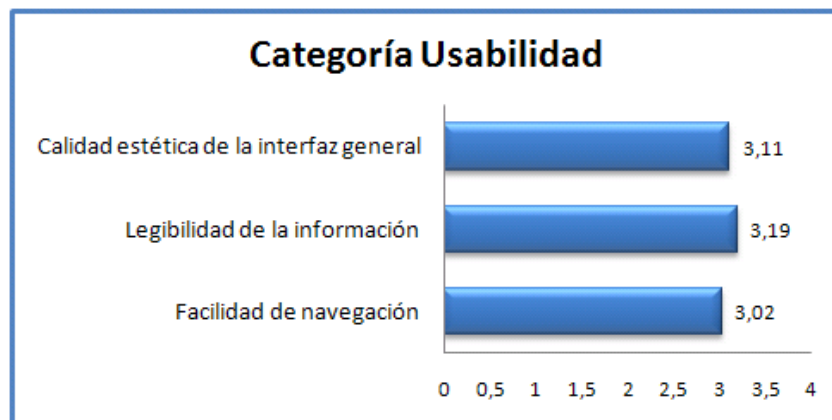


Figura 33. Resultados categoría Usabilidad discriminados por indicador

Los resultados presentados en la Figura 33, evidencian que el indicador con la calificación más alta es *legibilidad de la información*. Así mismo indican que la interfaz de usuario mantiene un aspecto agradable para los estudiantes, lo que les permite moverse por el

curso libre y rápidamente. Sin embargo, se deben mejorar aspectos relacionados con la interacción de los estudiantes con la plataforma.

En el Anexo E se detallan los resultados obtenidos en el test de conocimiento inicial, el test de conocimiento final y el cuestionario de evaluación de las estrategias de enseñanza y usabilidad del curso, usando la plataforma Moodle.

6.4 EVALUACIÓN DEL SISTEMA TUTOR INTELIGENTE SIAS EN LA ENSEÑANZA DE UN CURSO DE CAPACITACIÓN EN AIEPI

Para iniciar el curso se realizó una sesión presencial, en la cual se hizo la inducción al sistema. Posteriormente, se realizó el proceso de registro y cada estudiante respondió el test de conocimiento inicial y el test de CHAEA [106] sin límite de tiempo en cada test.

Los resultados obtenidos con el test de CHAEA se muestran en la Figura 34 e indican que en el estilo predominante en los estudiantes de décimo semestre de medicina es el estilo reflexivo, el segundo estilo predominante es el estilo teórico, seguido del estilo pragmático. El estilo predominante en este grupo de estudiantes coincide con los resultados presentados por [108]. Teniendo en cuenta lo anterior, se puede concluir que todos los estudiantes aprenden de diferentes formas y por tanto es importante conocer el estilo de aprendizaje predominante con el objetivo de presentar el contenido de acuerdo a sus características de aprendizaje.

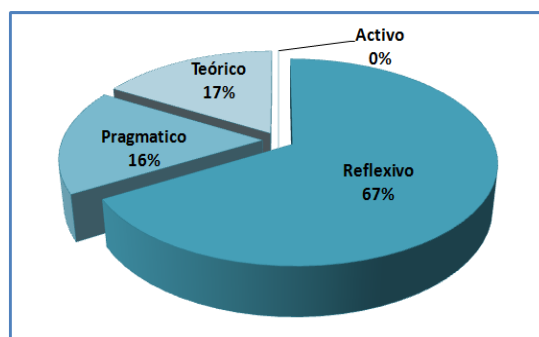


Figura 34. Estilo de Aprendizaje

Este aspecto no se evidenció en la plataforma Moodle a pesar de que la plataforma provee un módulo para la realización de cuestionarios. Esto se debe a que Moodle no posee un mecanismo que permita presentar diferentes estrategias de enseñanza de acuerdo a los resultados del test de estilo para cada estudiante. Por tanto, los contenidos en Moodle siempre se presentan de la misma forma para todos los estudiantes, independiente del estilo de aprendizaje que predomine en cada uno de ellos.

Por otra parte, el prototipo SIAS permitió monitorear el desempeño de cada estudiante durante el desarrollo del curso, comprobar la viabilidad del estudio de cada concepto y de acuerdo a la evolución del estudiante proporcionó diferentes consejos. Por ejemplo, si el estudiante no respondió correctamente a las preguntas cortas de una explicación, el sistema le indica que debe estudiar el contenido nuevamente.



Al finalizar el curso, se llevo a cabo una sesión presencial, en la cual 15 estudiantes que habían terminado el estudio de todos los temas presentaron el test de conocimiento final. Adicionalmente, se realizó una nueva sesión con los 3 estudiantes que no habían presentado el test durante la primera sesión. Esto con el fin de asegurar que todos los estudiantes presentaran el test de conocimiento final.

Con el objetivo de conocer los beneficios de usar el prototipo SIAS en la enseñanza de la estrategia AIEPI Clínico, se evaluó el nivel de conocimiento adquirido por los estudiantes al inicio y al finalizar el curso, velocidad de aprendizaje, estrategias de enseñanza utilizadas y la usabilidad, teniendo en cuenta los indicadores presentados en el apartado 6.1. A continuación se presenta el análisis de resultados de cada uno de los aspectos evaluados.

6.4.1 Evaluación Nivel de Conocimiento

Para determinar el nivel de conocimiento adquirido por los estudiantes durante el desarrollo del curso, se aplicó un test preliminar llamado *Test de conocimiento inicial* y un *Test de conocimiento final*. El primer test permitió determinar el nivel de conocimiento de los estudiantes al iniciar el curso. El segundo, permitió determinar el nivel de conocimiento de los mismos estudiantes pero, al finalizar el curso.

El test de conocimiento inicial estuvo conformado por 9 preguntas de selección, las cuales fueron redactadas teniendo en cuenta los conceptos que componen el curso y bajo la supervisión de un experto en el dominio. Así mismo, se asignó el puntaje para cada pregunta. Es importante mencionar que los resultados de este test son utilizados por el sistema para determinar el itinerario de estudio más adecuado con el cual cada estudiante debe iniciar su proceso de aprendizaje. Es decir, los estudiantes no iniciarán el curso partiendo del mismo nivel de conocimiento.

Por otra parte, el test de conocimiento final estuvo conformado por 13 preguntas las cuales fueron planteadas y diseñadas de manera diferente a las preguntas planteadas en el test de conocimiento inicial pero, conservando en cierta medida el mismo nivel de complejidad. Es importante mencionar que, tanto el test de conocimiento inicial como el test de conocimiento final, se aplicaron utilizando el prototipo SIAS.

Es importante mencionar que se aplicó el mismo test de conocimiento inicial y el mismo test de conocimiento final, tanto en Moodle como en SIAS.

La escala de calificación usada en el test de conocimiento inicial y final fue de 0.0 a 5.0, siendo 0.0 la calificación más baja y 5.0 la más alta. Además, se considera que el estudiante aprueba el test con una calificación mayor o igual a 3.0.

Los 18 estudiantes que iniciaron el curso, presentaron el test de conocimiento inicial. En la Figura 35, se observa que de ellos, el 61% aprobó el test y el 39% lo reprobó. Teniendo en cuenta lo anterior, se puede concluir que todos los estudiantes inician el curso con diferente nivel de conocimiento. Es importante aclarar que los estudiantes al inicio manejaban algunos conceptos relacionados con las temáticas que componen el curso, esto debido a cursos anteriormente vistos dentro del currículum de su programa académico. Debido a esto, los resultados presentan un porcentaje alto de estudiantes que aprobaron el test de conocimiento inicial. No obstante, ningún estudiante obtuvo la

calificación máxima, lo cual indica que ninguno de los participantes tenía total conocimiento sobre las temáticas que componen el curso.

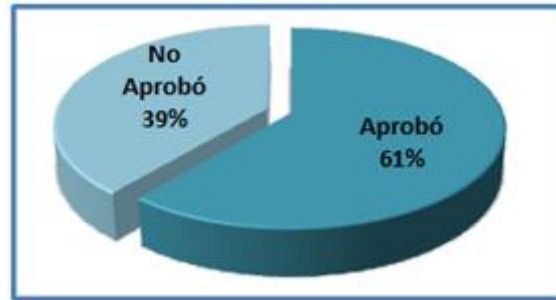


Figura 35. Resultados Test Conocimiento Inicial

El test de conocimiento final fue presentado por los 18 estudiantes que desarrollaron el curso. Los resultados presentados en la Figura 36, muestran que el 89% de los estudiantes aprobaron el test y el 11% no obtuvieron la calificación mínima para aprobarlo. En la Figura 37, se aprecia que el 33% de las calificaciones oscilan entre 3.0 y 4.0, el 56% de las calificaciones es mayor que 4.0 y solo el 11% de los estudiantes obtuvo una calificación inferior a 3.0.

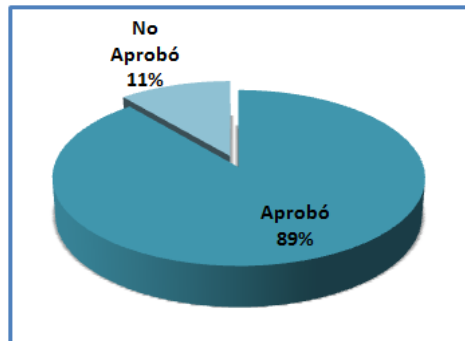


Figura 36. Resultados Test Conocimiento Final

Es importante destacar que se aprecia una variación significativa entre la calificación promedio del test de conocimiento inicial, la cual fue de 3,22 y la calificación promedio del test de conocimiento final, la cual fue de 4,06. Estos resultados indican que hubo un incremento en los resultados de la nota final, por tanto se puede decir que el tutor contribuyó de manera positiva en el 88.9% de los estudiantes que desarrollaron el curso.

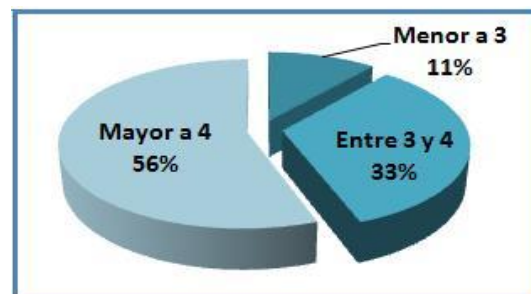


Figura 37. Consolidado Calificaciones Test Conocimiento Final

6.4.2 Evaluación velocidad de aprendizaje

La velocidad de aprendizaje se mide en términos del esfuerzo requerido para desarrollar tareas dentro de un curso, el tiempo empleado por los estudiantes en desarrollar una sola tarea y su progreso durante el proceso de aprendizaje.

Para determinar la velocidad de aprendizaje de los estudiantes, se estimó el tiempo necesario para que un estudiante aprenda los contenidos del curso. Este proceso se realizó con la ayuda de un experto en el dominio. Durante el desarrollo del curso, se contabilizó el tiempo empleado por cada estudiante en aprender los contenidos y se comparó con el tiempo estimado. Adicionalmente, esta información permitió proporcionar consejos adecuados a cada estudiante según su evolución en el curso.

Evaluar este aspecto representa una ventaja en el momento de asignar tareas o temáticas a los estudiantes. Esto, debido a que lo docentes tiene la posibilidad de ajustar el tiempo que asignan a sus estudiantes para realizar un conjunto de tareas, de tal forma que se adecue al tiempo real que ellos ocupan en realizarlas.

La Figura 38 presenta de manera gráfica los resultados del tiempo promedio en minutos empleado por los estudiantes para aprender un concepto, en contraste con el tiempo estimado para el mismo fin. Así mismo se presenta la comparación gráfica del tiempo total estimado para culminar el estudio del curso y el tiempo promedio que los estudiantes efectivamente usaron para terminar el estudio del curso.

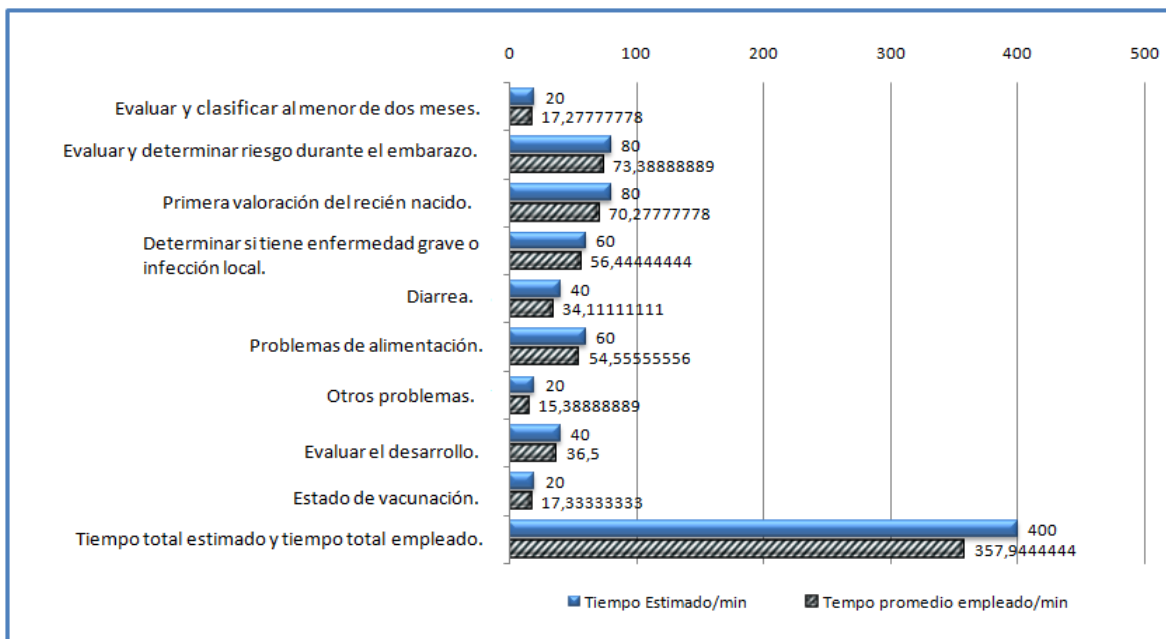


Figura 38. Tiempo empleado y tiempo estimado para el aprendizaje de los conceptos.

Los resultados presentados anteriormente, muestran que los estudiantes dedicaron en promedio menos tiempo que el estimado para cada explicación. Sin embargo el tiempo dedicado se considera suficiente para aprender cada concepto del curso.

6.4.3 Evaluación de las estrategias de enseñanza

Las estrategias de enseñanza utilizadas en el curso, se evaluaron por medio de la categoría *Calidad Pedagógica* del cuestionario propuesto por Arias [126], el cual se encuentra descrito en el apartado 6.1. Es importante señalar que los estudiantes dieron respuesta a este cuestionario al finalizar el curso

En la Figura 39 se presentan gráficamente los resultados de la evaluación del curso, teniendo en cuenta los indicadores correspondientes a la categoría *Calidad Pedagógica*.

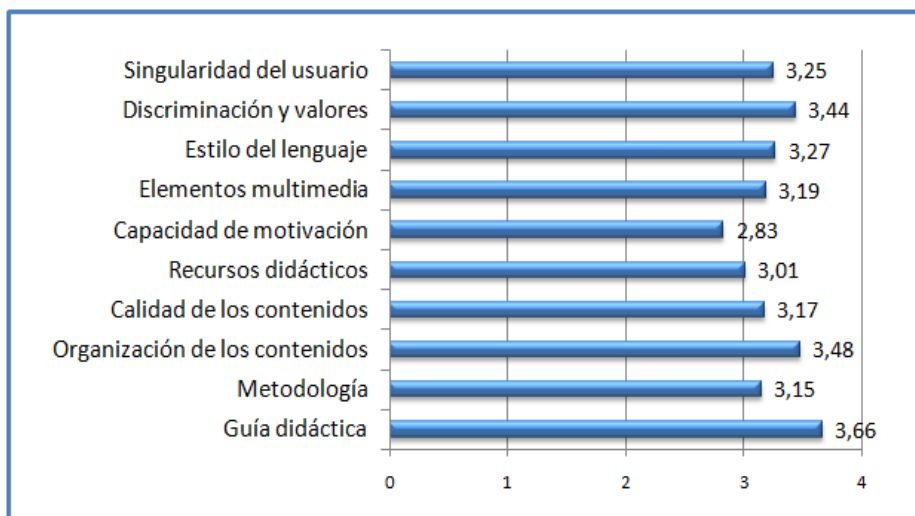


Figura 39. Resultados para cada indicador de la categoría *Calidad pedagógica*

De acuerdo a los resultados presentados en la Figura 39, los aspectos más fuertes del prototipo en cuanto a las estrategias de enseñanza, esta enmarcados en aspectos tales como: la organización apropiada de los contenidos, el estilo del lenguaje, la adaptación de los contenidos teniendo en cuenta las características particulares de cada estudiante, y una guía didáctica apropiada. Así mismo, se evidenció que hace falta mejorar un poco en los aspectos que tienen que ver con la motivación del estudiante. Sin embargo, es importante aclarar que la motivación es un aspecto relativo en el cual se involucran muchos factores que influyen de manera positiva o negativa en el estudiante.

6.4.4 Evaluación de usabilidad

La usabilidad del curso, se evaluó por medio de la categoría *Usabilidad* del cuestionario propuesto por Arias [126]. Como se mencionó anteriormente, el cuestionario se encuentra descrito en el apartado 6.1.

En la Figura 40 se presentan gráficamente los resultados de la evaluación del curso, teniendo en cuenta los indicadores correspondientes a la categoría *Usabilidad*. Los resultados que se presentan, indican que la interfaz general del prototipo software es agradable estéticamente y que el texto se presenta de forma clara. Sin embargo es importante disminuir un poco la cantidad de información que se presenta y se debe mejorar un poco el menú de opciones.

En el Anexo E, se detallan los resultados obtenidos en el test de conocimiento inicial, el test de conocimiento final, el cuestionario de evaluación de las estrategias de enseñanza y usabilidad del curso, y se presentan los resultados en detalle del tiempo de estudio que empleó cada estudiante en cada concepto del curso, haciendo uso del prototipo SIAS.

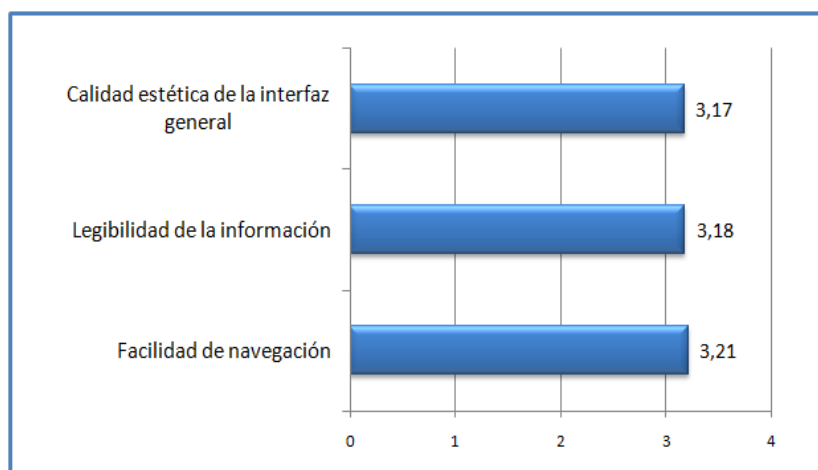


Figura 40. Resultados para cada indicador de la categoría *Usabilidad*

6.5 ANÁLISIS DE LOS RESULTADOS DE LA EVALUACIÓN DEL SISTEMA SIAS FRENTE A MOODLE

En esta sección se describe el análisis comparativo de los resultados obtenidos en el desarrollo del curso “Estrategia AIEPI: Módulo Recién Nacido”, utilizando la plataforma Moodle y el sistema tutor inteligente SIAS. Es importante mencionar que el curso se llevo a cabo teniendo en cuenta que tanto a los estudiantes del programa de medicina como al programa de enfermería, se les orienta en el contenido programático las mismas temáticas referentes a la estrategia AIEPI.

6.5.1 Evaluación Nivel de Conocimiento

A continuación se realiza el análisis de los resultados que corresponden al nivel de conocimiento inicial y final, de los estudiantes. Para dicho proceso, se analizó el porcentaje de estudiantes que aprobaron el test de conocimiento inicial y final, cuando emplearon la plataforma Moodle. Posteriormente, se realizó una comparación con el porcentaje de estudiantes que aprobaron el test de conocimiento inicial y final, cuando utilizaron el sistema tutor inteligente SIAS.

Con base en los resultados presentados en la Figura 41, se evidencia que el nivel de conocimiento final, cuando los estudiantes emplearon el sistema SIAS, fue superior a los resultados obtenidos cuando emplearon la plataforma Moodle. De acuerdo con este hecho, el uso del sistema tutor inteligente SIAS, influyó de manera positiva, mejorando el nivel de conocimiento adquirido por los estudiantes.

Una de las posibles razones por las que mejoraron los resultados con la plataforma SIAS, se debe a que los estudiantes no tenían que volver a repasar temas que ya conocían. Lo anterior se debe a que SIAS, utiliza los resultados del test de conocimiento inicial para establecer un itinerario de estudio apropiado al nivel de conocimiento con el que inicia un estudiante. Por tanto, los estudiantes podían utilizar ese tiempo para profundizar en temas que eran totalmente nuevos. En cambio, la plataforma Moodle no dispone de un mecanismo que le permita establecer un itinerario de estudio adaptado al nivel de conocimiento con que inicia un estudiante, es decir, se asume que todos empiezan su proceso de aprendizaje con el mismo nivel de conocimiento.

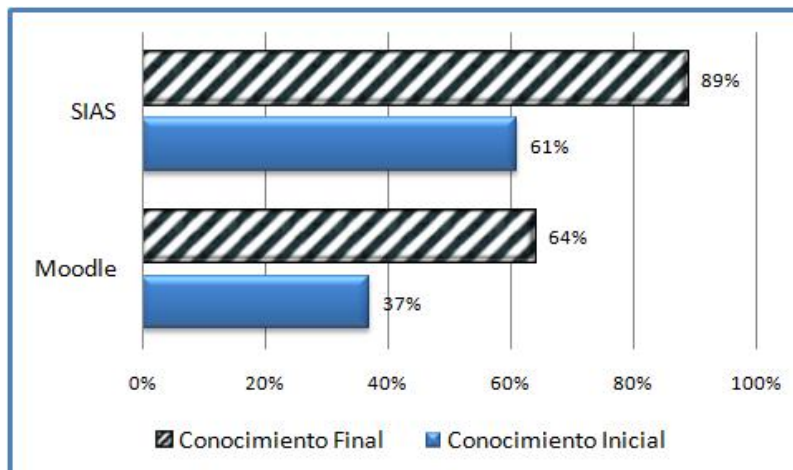


Figura 41. Resultados generales del nivel de conocimiento en Moodle y SIAS

6.5.2 Evaluación Velocidad de Aprendizaje

Generalmente, los docentes estiman un tiempo para la realización de ciertas tareas que según su experiencia, consideran adecuado. Pero, en la mayoría de los casos, no es posible que los estudiantes terminen en el tiempo estimado por el docente. Esto puede deberse a muchos factores, entre ellos, las características específicas de cada estudiante. Sin embargo, es importante tener en cuenta el tiempo real que el estudiante dedica y de esta manera, poder ajustar de una forma adecuada el tiempo que se asigna para la realización de tareas o estudio de diferentes temáticas. Como se ilustró en el apartado 6.4, la velocidad de aprendizaje tiene que ver con el tiempo que emplean los estudiantes en realizar las tareas o en estudiar una temática determinada, la cual se ve reflejada en el progreso de cada uno.

En la plataforma Moodle no es posible medir de manera implícita la velocidad de aprendizaje. Esto se debe a que Moodle solo proporciona los registros de uso de un recurso en particular y estos datos por si solos, no garantizan que el estudiante haya empleado ese tiempo en estudiar los contenidos del curso. Cuando un docente asigna un tiempo fijo para la realización de una tarea, no tiene la posibilidad de ajustar de la manera más adecuada, ese tiempo. Esto debido a que no posee información detallada acerca del tiempo que los estudiantes emplearon en realizar la tarea asignada.

A diferencia de Moodle, SIAS proporciona un completo informe acerca del tiempo que emplea un estudiante en aprender un tema determinado o en realizar ciertas tareas. Así mismo, proporciona un completo informe de la evaluación de los estudiantes indicando

sus debilidades y fortalezas, lo cual permite monitorear el progreso de los estudiantes en el tutorial. En otras palabras, la información que provee la plataforma Moodle acerca de la velocidad de aprendizaje, es muy limitada en comparación con SIAS.

6.5.3 Evaluación Estrategias de Enseñanza

En esta sección se hace un análisis de los resultados obtenidos por el sistema SIAS y la plataforma Moodle en la evaluación de la categoría *Calidad pedagógica*. Esto con el objetivo de determinar el nivel de adaptación de las estrategias de enseñanza utilizadas en cada contexto. En la Tabla 5 se presentan los resultados obtenidos y la diferencia en cada indicador, entre el sistema SIAS y la plataforma Moodle.

Indicador	SIAS	MOODLE	Diferencia (%)
Guía didáctica	3,66	3,63	1
Metodología	3,15	3,05	3
Organización de los contenidos	3,48	2,96	13
Calidad de los contenidos	3,17	2,77	10
Recursos didácticos	3,01	2,74	7
Capacidad de motivación	2,83	2,81	1
Elementos multimedia	3,19	3,13	2
Estilo del lenguaje	3,27	2,93	9
Discriminación y valores	3,44	3,45	0
Singularidad del usuario	3,25	2,9	9

Tabla 5. Resultados discriminados por indicador señalando la diferencia porcentual

A partir de los resultados anteriores, se identifican las características más relevantes del sistema SIAS frente a la plataforma Moodle:

- *Organización de contenidos:* en cuanto a la evaluación de la arquitectura de los contenidos, se evidencia que la información presentada en ambos contextos es actualizada. Sin embargo, en el sistema SIAS la clasificación y la organización de los contenidos, permiten a los estudiantes identificar con mayor facilidad lo relevante de la información presentada. Por lo tanto, se evidencia que la organización de los contenidos en el sistema SIAS, es 13% mejor que en la plataforma Moodle.

- *Calidad de contenidos:* en cuanto al grado de adecuación de los contenidos del sistema SIAS, se evidencia que presentar los contenidos teniendo en cuenta las preferencias de aprendizaje, permite plantear diferentes actividades para estimular en los estudiantes su proceso formativo. Por el contrario, en la plataforma Moodle las actividades propuestas son las mismas para todos los estudiantes. Por lo tanto, los resultados demuestran que la calidad de contenidos mejora 10% en el sistema SIAS.

- *Singularidad del Usuario:* en cuanto a la evaluación de la capacidad que tiene el sistema para acoplarse a las características y circunstancias personales del usuario final, los resultados evidencian que el sistema SIAS, contempla las características particulares de los estudiantes, específicamente, sus características de aprendizaje. Por el contrario, en la plataforma Moodle no se determinan las características de cada estudiante ya que se dispone únicamente de su información personal. Así mismo, los resultados evidencian que para los estudiantes es más fácil organizar su tiempo de estudio con el sistema SIAS

que con la plataforma Moodle. Por lo tanto, se detectó en el estudio que la singularidad del usuario en el sistema SIAS, es 9% mejor que en la plataforma Moodle. No obstante, en ambos contextos se deben contemplar las características físicas de los estudiantes, como algunas discapacidades.

- *Recursos didácticos*: en cuanto a la evaluación del grado de adaptación que tiene el sistema SIAS, se evidencia que el sistema proporciona distintas guías de aprendizaje de acuerdo a las características de los estudiantes, los nuevos conceptos se presentan con la estrategia de enseñanza adecuada y según el caso, se emplea un código comunicativo³ diferente. Además, el sistema proporciona diferentes consejos según la evolución del estudiante. Con todo lo anterior, se les permite a los estudiantes explorar todas sus fortalezas. Por el contrario, en la plataforma Moodle, el contenido se enseña a todos los estudiantes utilizando la estrategia de enseñanza seleccionada por el profesor. Así mismo, esta plataforma no proporciona ayudas a los estudiantes según su evolución en el curso. Por lo tanto, se evidencia que los recursos didácticos en el sistema SIAS son 7% más adaptables que en la plataforma Moodle.

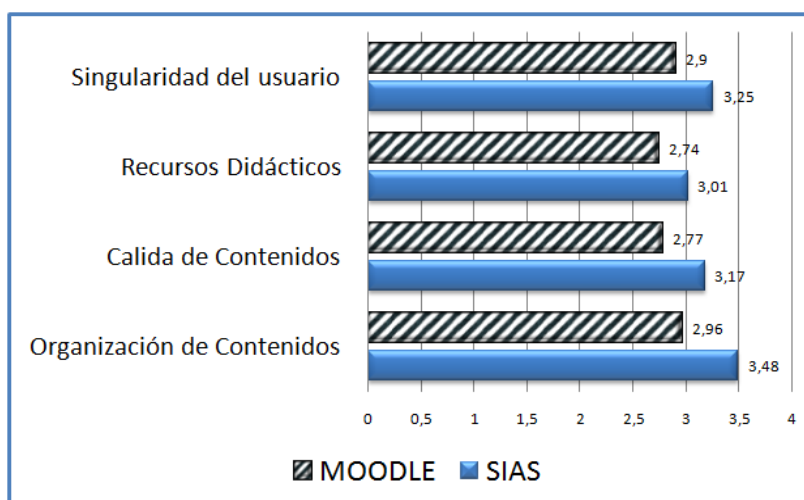


Figura 42. Resultados calidad pedagógica en SIAS y Moodle discriminados por indicador.

En la Figura 42 se presentan los resultados comparativos de los indicadores mencionados anteriormente, los cuales están relacionados con el nivel de adaptación. Estos resultados muestran que SIAS supera la calificación obtenida por la plataforma Moodle en los cuatro indicadores. Esto se debe a que, en el sistema SIAS los contenidos son presentados utilizando las estrategias de enseñanza adecuadas al estilo de aprendizaje de los estudiantes. Por el contrario, en la plataforma Moodle los contenidos se presentan de la misma forma a todos los estudiantes sin tener en cuenta sus preferencias de aprendizaje.

6.5.4 Evaluación de Usabilidad

En esta sección se hace un análisis de los resultados obtenidos por el sistema SIAS y la plataforma Moodle en la evaluación de la categoría *Usabilidad*. A continuación, se presentan los resultados obtenidos por cada indicador que pertenece a la categoría.

³ Código comunicativo, hace referencia a la presentación de la información de diferentes formas (verbal, simbólico, gráfico).

Indicador	SIAS	MOODLE	Diferencia (%)
Facilidad de navegación	3,21	3,02	5
Legibilidad de la información	3,18	3,19	0
Calidad estética de la interfaz general	3,17	3,11	2

Tabla 6. Resultados discriminados por indicador señalando la diferencia porcentual

Teniendo en cuenta los resultados que se presentan en la Tabla 6, se evidencia que el indicador que presentó mayor diferencia porcentual fue la facilidad de navegación. Esto indica que a los estudiantes se les dificulta más navegar en los contenidos presentados a través de Moodle que en los contenidos presentados por medio de SIAS.

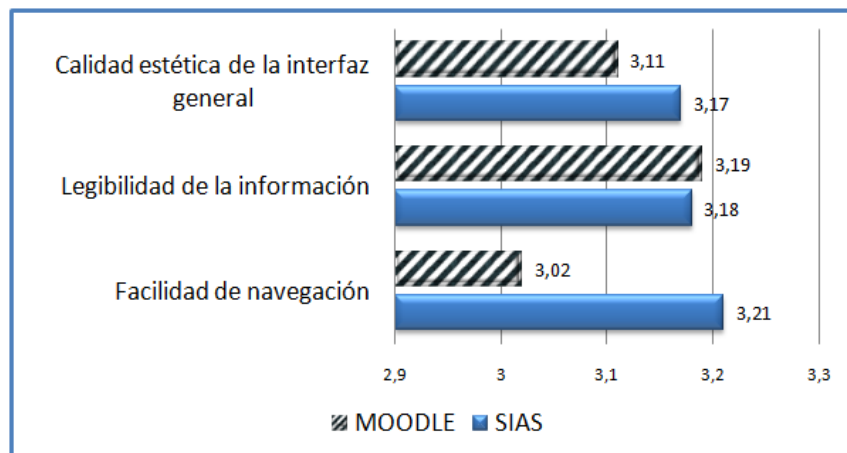


Figura 43. Resultados usabilidad en SIAS y Moodle discriminados por indicador

Los resultados presentados en la Figura 43, muestran de manera gráfica, que el sistema SIAS supera la calificación obtenida por la plataforma Moodle en la mayoría de los indicadores, a excepción del indicador *legibilidad de la información*. Esto se debe a que Moodle maneja gran cantidad de información en su interfaz y ofrece muchas funcionalidades que en generalmente no son usadas por los estudiantes. A diferencia de Moodle, en SIAS hay poca carga cognitiva y los contenidos están bien distribuidos en una interfaz sencilla.

Finalmente, teniendo en cuenta los resultados de la evaluación realizada para cada aspecto (nivel de conocimiento, velocidad de aprendizaje, estrategias de enseñanza, usabilidad) y por cada indicador, se puede evidenciar que el sistema SIAS ha influido positivamente en el proceso de aprendizaje en la implementación de la estrategia AIEPI Clínico.



7. RESULTADOS OBTENIDOS

Durante el desarrollo del proyecto, se obtuvo una serie de resultados producto de la investigación y el desarrollo del sistema tutor inteligente, en donde se evidencia el cumplimiento de cada uno de los objetivos planteados al inicio del proyecto. Los productos obtenidos son:

Monografía: en este documento se realizó una descripción completa y detallada de todo el trabajo desarrollado, documentación de referencia, y demás aspectos relacionados con el proyecto.

Artículos: se realizaron dos artículos de investigación con los resultados obtenidos del proceso de investigación.

El primero hace referencia a la primera evaluación del sistema de enseñanza tradicional Moodle. Este artículo se encuentra en etapa de evaluación para ser publicado en la *Revista Colombiana de Tecnologías de Avanzada, categoría C en Colciencias*.

En el segundo, se presentan los resultados obtenidos de evaluar el sistema tutor inteligente desarrollado SIAS, frente al sistema tradicional de enseñanza Moodle. Este artículo ha sido presentado al *International Council on Medical & Care Compunetics (ICMCC 2010)*, fue aceptado y será publicado en el volumen “*Medical and Care Compunetics 6*” en la serie “*Health Technology and Informatics*”, indexada en *Medline*, categoría B en Colciencias. Es importante mencionar que *Medline* es la base de datos médica más amplia que existe y ha sido producida por la biblioteca nacional de medicina de los Estados Unidos.

Prototipo: es el prototipo software de implementación de un Sistema Tutor Inteligente para Aprendizaje en salud.

Manual de instalación: es el documento donde se identifican claramente los pasos a seguir para una correcta implantación de la aplicación web, en el servidor.

Manual de usuario: es el documento donde se presentan los pasos que debe seguir cada tipo de usuario para lograr un buen uso del sistema.

Documento de evaluación y validación de los resultados obtenidos: inicialmente se había planteado hacer un documento adicional a la monografía, donde se presentarían los resultados obtenidos en la investigación. Pero, teniendo en cuenta que este aspecto es muy importante dentro de la investigación, se decidió incluir todo el proceso de la evaluación en el capítulo 7 de la monografía y estructurar dos artículos de investigación, mencionados anteriormente.

Anexos: este documento permite ampliar aspectos que no se hayan profundizado en la monografía.

Los principales aportes obtenidos con la realización del presente proyectos están directamente relacionados con los objetivos planteados al inicio del proyecto:



- *Diseñar modelos de agentes para la posterior implementación del prototipo de Sistema Tutor Inteligente propuesto.* El cumplimiento de este objetivo se evidencia en el capítulo 6, donde se plantean los diferentes modelos de agentes que se diseñaron y fueron necesarios para la implementación del módulo del tutor teniendo en cuenta características de adaptabilidad, como un sistema multi-agente en el prototipo software. En el sistema se identifican dos tipos de agentes, el agente *tutor* y el agente *concepto*, cada agente se describe por medio de un modelo de agentes específico, donde se identifican los objetivos, responsabilidades y capacidades de cada uno de ellos. Además de los modelos de agentes, fue necesario considerarlos siguientes modelos: modelo de interacción entre agentes, modelo de la organización, modelo del entorno y, modelo de tareas y objetivos, para representar todos los aspectos del módulo del tutor. Para la realización de estos modelos se hizo uso del editor Ingenias Development Kit – IDK [132].

Es importante mencionar que para realizar el diseño de los modelos de agentes que corresponden al módulo del tutor se tuvieron en cuenta las características de adaptabilidad que se definieron para éste módulo.

- *Desarrollar un prototipo de Sistema Tutor Inteligente que, usado para orientar cursos en el dominio de salud, contribuya al mejoramiento del proceso de enseñanza-aprendizaje.* El prototipo inicial SIAS ha sido usado para capacitar a estudiantes de ciencias de la salud en la estrategia AIEPI Clínico y se ha desarrollado integrando técnicas de inteligencia artificial. El módulo del tutor se ha implementado con agentes inteligentes y el módulo del estudiante se ha implementado con redes bayesianas. En el sistema SIAS, se lleva a cabo un proceso de inicialización del modelo del estudiante, que consiste en determinar mediante un test previo, el nivel de conocimiento inicial de un estudiante en el dominio a enseñar e identificar dos estilos de aprendizaje predominantes en cada estudiante por medio del test de CHAEA [106]. De esta manera, el sistema SIAS puede seleccionar el itinerario de estudio y las estrategias adecuadas para el proceso de enseñanza -aprendizaje de cada estudiante en particular.
- *Definir y aplicar un esquema de evaluación basado en indicadores para determinar el aporte obtenido con el uso de Sistemas Tutores Inteligentes, frente a sistemas de enseñanza tradicional.* Con el fin de conocer los beneficios de usar el sistema tutor inteligente desarrollado SIAS frente al sistema de enseñanza tradicional Moodle, se realizó un proceso de evaluación donde se definió un conjunto de características e indicadores a evaluar y se realizó el análisis de los resultados comparando los dos sistemas de enseñanza. Con los resultados obtenidos de la evaluación se evidenció que el sistema tutor inteligente desarrollado, contribuyó al mejoramiento del proceso de enseñanza-aprendizaje en la capacitación de la estrategia AIEPI.



8. CONCLUSIONES Y RECOMENDACIONES

8.1 CONCLUSIONES

El uso de agentes representó una ventaja en el desarrollo del prototipo software. Esto debido a que los agentes *Tutor* y *Concepto* que forman parte del prototipo, proporcionaron mayor flexibilidad en el tratamiento del modulo tutor, permitieron monitorear a los estudiantes constantemente y presentar los contenidos con las estrategias de enseñanza adecuadas, además permitieron mayor tolerancia a fallos por ser entidades autónomas que trabajan en forma distribuida.

El prototipo de STI desarrollado utiliza redes bayesianas para almacenar el estado de conocimiento actual de cada estudiante. El uso de esta técnica de Inteligencia Artificial en el desarrollo del prototipo, permitió determinar con anticipación un itinerario de estudio adecuado teniendo en cuenta su nivel de conocimiento inicial. Así mismo, permitió determinar los conocimientos que debe ir adquiriendo un estudiante a medida que éste avanza en su proceso de aprendizaje.

En este trabajo se presentó un conjunto de indicadores que permitieron establecer un punto de comparación entre el sistema de enseñanza tradicional Moodle y el sistema tutor inteligente SIAS. Estos indicadores permitieron determinar las ventajas obtenidas con el uso de Sistemas Tutores Inteligentes, frente a sistemas de enseñanza tradicional, en el mejoramiento del proceso de enseñanza-aprendizaje en la estrategia AIEPI clínico.

Los resultados obtenidos en la evaluación del curso en el sistema tutor inteligente SIAS, demuestran que los aspectos más fuertes del prototipo esta enmarcados en aspectos tales como: la organización apropiada de los contenidos, el estilo del lenguaje apropiado, la adaptación de los contenidos teniendo en cuenta las características particulares de cada estudiante, y una guía didáctica apropiada. Esto indica que el prototipo SIAS tiene en cuenta los diferentes perfiles de los estudiantes, ofreciéndoles alternativas pedagógicas altamente compatibles con sus características cognitivas y su evolución durante el curso.

Teniendo en cuenta que SIAS, a diferencia de Moodle, utiliza la información del nivel de conocimiento inicial para establecer un itinerario de estudio adaptado al nivel de conocimiento con que inicia un estudiante, los resultados de evaluar el nivel de conocimiento, evidenciaron que el sistema tutor inteligente SIAS obtuvo mejores resultados frente a la plataforma Moodle.

En el proceso de evaluación de la velocidad de aprendizaje, se evidenció que no fue posible medir este aspecto en la plataforma Moodle. Esto se debe a que las tareas son programadas por el docente, quien dispone un tiempo límite en el cual los estudiantes deben terminar las tareas que se asignen. A diferencia de Moodle, SIAS proporciona un completo informe acerca del tiempo que emplea un estudiante en aprender un tema determinado o en realizar ciertas tareas y permite monitorear las tareas y temáticas que los estudiantes se encuentran estudiando. Como consecuencia, se hizo un aporte utilizando SIAS, debido a que los docentes pueden ajustar el tiempo que estiman para la



realización de actividades, de una forma más adecuada, teniendo en cuenta el progreso de cada estudiante.

En el sistema SIAS, los contenidos son presentados utilizando las estrategias de enseñanza adecuadas al estilo de aprendizaje de los estudiantes y en la plataforma Moodle, los contenidos se presentan de la misma forma a todos los estudiantes, sin tener en cuenta sus preferencias de aprendizaje. Por tanto, el aporte se ve reflejado en que, los resultados de evaluar las estrategias de enseñanza, muestran que el sistema SIAS presenta mejoras con respecto a la plataforma Moodle en los aspectos relacionados con la capacidad de adaptación a las características de cada estudiante.

8.2 RECOMENDACIONES Y TRABAJO FUTURO

Tan importante como es el proceso de enseñanza lo es el proceso de evaluación. Por lo tanto, se recomienda mejorar este aspecto en el prototipo desarrollado, de tal manera que el mecanismo de evaluación también adopte comportamientos adaptativos que permitan ajustar el método de diagnóstico del nivel de conocimiento, al estilo de aprendizaje y a las características de cada estudiante. Una de las técnicas actuales más importantes de evaluación adaptativa se conoce como Test Adaptativos Informatizados – TAI [34] y son ampliamente usados en el campo de la educación.

Es necesario plantear evaluaciones posteriores con el fin de determinar con mayor exactitud, si el sistema tutor inteligente desarrollado, contribuye en el mejoramiento del proceso de enseñanza-aprendizaje en la capacitación de la estrategia AIEPI Clínico, esta vez teniendo en cuenta aspectos psicológicos y culturales. Para ello, se propone diseñar varias experiencias que permitan realizar un análisis estadístico detallado por medio de estudios comparativos empleando el sistema tutor inteligente SIAS frente a otros sistemas tutores inteligentes para el aprendizaje en el dominio de salud.

El uso de Razonamiento Basado en Casos – CBR [133] en el desarrollo de cada uno de los componentes que integran la arquitectura de los STI, presenta un nuevo paradigma de resolución de problemas usando aprendizaje incremental. Teniendo en cuenta lo anterior, se propone como trabajo futuro el uso de CBR para diseñar el esquema general del STI.

En el sistema desarrollado, los agentes trabajan de forma autónoma y concurrente, por lo tanto, se propone incluir mecanismos para control de concurrencia [134] con el objetivo de equilibrar el mantenimiento de la consistencia de la información y mantener un alto nivel de concurrencia.

Con el fin de mejorar la interoperabilidad entre el Sistema Tutor Inteligente desarrollado y diferentes STI existentes, se propone como trabajo futuro diseñar modelos de ontologías para definir y organizar los atributos relevantes del dominio pedagógicamente, permitiendo escribir y compartir las estrategias de enseñanza en términos de esos atributos.



BIBLIOGRAFIA

- [1] Organización Mundial de la Salud, "*Investigaciones Operativas sobre Atención Integrada a las Enfermedades Prevalentes de la Infancia (AIEPI)*", Organización Mundial de la Salud, 2001. [Online]. Available: <http://www.who.int/es/>. [Accessed: Abril. 28, 2009].
- [2] D. Palacio Betancourt, R. Guerrero Carvajal, E. J. Alvarado Santander, J. L. Sanchez Mesa, I. M. Neira Nuñez, and I. E. Urquijo Velasquez, *Atención Integrada a las enfermedades prevalentes de la infancia - AIEPI - : curso clínico para profesionales de la salud*, Ministerio de la Protección Social. 2005.
- [3] Learning for Health an Social Care. NHSU. Elearning for Health: A discussion paper. Towards a strategy for NHSU, SHAs, partners and stakeholders. London. NHSU, 2004.
- [4] Department of Health. Working Together – Learning Together. A Framework for Lifelong Learning in the NHS. London: Department of Health. Available: http://www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_4009558
- [5] M. Wooldridge and N. Jennings, *Intelligent Agents. Theory and Practice*. Knowledge Engineering Review. 1995.
- [6] A. Rendón, M. P. Restrepo, M. Solarte, "*EHAS-AIEPI - Fortalecimiento de la estrategia AIEPI con el apoyo de Tecnologías de la Información y Comunicación en los municipios de Silvia y Jambaló, Departamento del Cauca, Colombia. Financiado por la Dirección de Cooperación para el Desarrollo de la Universidad Politécnica de Madrid,*" 2007-2008. [Online]. Available: <http://git.ucauca.edu.co/> [Accessed: Abril 28, 2009].
- [7] M. Solarte and F. Urbano. *Perspectivas de Investigación para el Cauca. Capítulo 8: Desarrollo de Entornos Virtuales para la educación: ELANE*. 2006.
- [8] A. Finkelstein, J. Kramer, *Software engineering: a roadmap*. New York, NY, USA: ACM Press. 2000, pp.3-24.
- [9] K. VanLehn, Student modeling, *Intelligent Tutoring Systems*, en C. Polson, J.J. Richardson, Editors, pp. 55 -77, 1988.
- [10] B. Wolf, "Context Dependent Planning in a Machine Tutor". Ph.D. Dissertation, University of Massachusetts, Amherst, Massachusetts. 1984.
- [11] L. M. M. Giraffa, *Seleção e adoção de estrataguas de ensino em Sistemas Tutores Inteligentes*. Porto Alegre: CPGCC/UFRGS. 1997.
- [12] E. Parra Castrillón, "*Sistemas Tutoriales Inteligentes, un Aporte de la Inteligencia Artificial para la Mediación Pedagógica*". Universidad Nueva Esparta. 2004.
- [13] M. Urretaurcaya, *Sistemas Inteligentes en el ámbito de la Educación*, *Revista Iberoamericana de Inteligencia Artificial*, pp. 5-12. 2001.
- [14] K. Duccki and M. Youngsong, "Notes in Computer Science," en "*An Experiment and Design of Web-based Instruction Model for Collaborative Learning*", Vol. 3043. 2004, pp. 378-385.
- [15] J. López Ostio, "Sistemas Tutoriales Inteligentes" en *Conferencia mecanografiada, 1993, San Sebastián, España*.
- [16] M. V. Corredor, *Sistemas tutoriales inteligentes*. Boletín de Informática Educativa. Proyecto SIIE, Colombia, vol. 2, no. 1, 1989.
- [17] J. R. Carbonell, AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction. *IEEE Transactions on Man-Machine Systems*, 11. 1970. pp. 190-202.
- [18] G. P. Brown, Report No. TR-182, *Laboratory for computer Science MIT*.
- [19] E.B Koffman and S.E. Blount, *Artificial Intelligence* 6. 1975, 215-234.



- [20] R. Pichastor, S. Nieto, and M. A. Navarro, “Entornos virtuales de aprendizaje: El papel del valor del entorno virtual y la auto-eficacia en los resultados de los estudiantes” in *Internacional Conference on Multimedia and ICT in Education, April 22-24, 2009, Lisbon, Portugal*.
- [21] A. E. López, S. Escalera, R. Ledesma, “Ambientes Virtuales de Aprendizaje,” in *Presimposio Virtual Somece 2002*. 2002 pp. 1- 12.
- [22] Moodle, available at: www.moodle.org/.
- [23] A. Grandío. I Jornadas de Innovación Educativa de la Universidad Jaume I, 2000.
- [24] Claroline, available at: <http://www.claroline.net/>
- [25] eCollege, available at: <http://www.ecollege.com/index.learn>
- [26] Blackboard, available at: <http://www.blackboard.com/>
- [27] Dokeos, available at: <http://www.dokeos.com/es>
- [28] M.Campbell Polson, J. Richardson, *Foundations of Intelligent Tutoring Systems*. Ed. Lawrence Erlbaum Associates, Inc, 1988.
- [29] C. González, J.C. Rial, and M. Llamas, “Contribuciones al diseño de sistemas tutores inteligentes usando razonamiento basado en casos”. Tesis Doctoral. Universidad de Vigo, España, 2008.
- [30] B. Morgan, “An Introduction to Bayesian Statistical Decision Processes”. Prentice-Hall Inc., Englewood Cliffs, N.J. 1968. pp. 15
- [31] J.R. Anderson, *The architecture of cognition*. Harvard University Press: Cambridge, Massachusetts -EE.UU. 1983.
- [32] C. Carmona, E. Millán, J. Pérez de la Cruz, M. Trella, and R. Conejo, *Introducing Prerequisite Relations in a Multi-layered Bayesian Student Model*. *User Modeling*. 2005. pp. 347-356.
- [33] M. Cook, “Student modelling in intelligent tutoring systems” in *Artificial Intelligence Review*, 7, 1993. pp. 227-240.
- [34] E. Millán, “Sistema Bayesiano Para Modelado del Alumno”. Tesis Doctoral. Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga. España. 2000
- [35] C. Zhao, Z. Sun, Q. Liu, C. Shang, and D. Shen, *Research on Initializing Student Model*. *Fifth IEEE International Conference on Advanced Learning Technologies ICALT*. 2005 pp. 691-693.
- [36] V. Tsiriga and M. Virvou, “A Framework for the Initialization of Student Models in Web-based Intelligent Tutoring Systems”. *Journal User Modeling and User- Adapted Interaction*. Springer Netherlands. 14 (4), 2004. pp. 289 – 316.
- [37] P. Sánchez, *Computer-Human Learning*. *Computers and Education in the 21st Century 2000*. pp. 297- 312.
- [38] D. Chin, *KNOME: Modeling What the User Knows in UC*. In A. Kobsa and W. Wahlster (eds.): *User Models in Dialog Systems*, Berlin-Heidelberg: Springer Verlag. 1989. pp. 74-107.
- [39] S. Dudani, *The distance-weighted k-nearest-neighbor rule*. *IEEE Transactions on Systems, Man and Cybernetics*. 6, 1976. pp. 325-327.
- [40] P. Brusilovsky and D. Cooper, *ADAPTS: Adaptive hypermedia for a Web-based performance support system*. *Proceedings of Second Workshop on Adaptive Systems and User Modeling on World Wide Web*. Computer Science Report 99-07, Eindhoven University of Technology. 1999.
- [41] G. Weber and M. Specht, *User modeling and adaptive navigation support in WWW-based tutoring systems*. *Proceedings of User Modeling*. 1997. pp. 289-300.
- [42] A. Mitrovic and S. Ohlsson, *Evaluation of a Constraint-Based Tutor for a Database Language*. *International Journal of Artificial Intelligence in Education*, 10. model and their interaction in an ITS authoring tool. *Proceedings of the Eighth International*



- Conference on User Modeling, Lecture Notes in Artificial Intelligence*, 2109, 1999, pp. 158-167.
- [43] A. Garland and R. Alterman, Learning Procedural Knowledge to Better Coordinate. *IJCAI*. 2001. pp. 1073 – 1083.
- [44] L. Giraffa, M. Da Costa, and R. Vicari, Modeling the MCOE Tutor Using a computational Model. 1998. pp. 50 – 60.
- [45] C. Woo, Instructional Planning in an Intelligent Tutoring System: Combining Global Lesson Plans with Local Discourse Control. Ph. D. Department of CSAM. Illinois Institute of Technology, Chicago, Illinois. 1991.
- [46] B. Skinner, Are theories of learning necessary? *Psychological Review*, 57. 1950. pp. 193 – 216.
- [47] E. Harskamp and N. Ding, Structured collaboration versus individual learning in solving physics problems. *International Journal of Science Education*, 28 (14). 2006. pp. 1669-1688.
- [48] J. Elorriaga and C. Fernandez, Using Case-Based Reasoning in Instructional Planning: Towards a Hybrid Self-improving Instructional Planner. Spain. 1998.
- [49] R. Burton, The Environment Module of Intelligent Tutoring Systems. *Foundations of Intelligent Tutoring Systems*. 1988.
- [50] L. Alem and M. Lee, “A task oriented intelligent tutoring system,” in *First New Zealand International Two-Stream Conference*, Issue , 24-26, Nov 1993, pp. 196 - 200.
- [51] G. Kearsley and R. J. Seidel, *Automation in training and education*, 27 (1). 1985, 61-74.
- [52] M. W. Evens, S. Brandle, Ru-Charn Chang, R. Freedman, M. Glass, Y. Hee Lee, L. Seop Shim, C. Woo Woo, Y. Zhang, Y. Zhou, J. A. Michael, and A. A. Rovick, “CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue”, *Twelfth Midwest AI and Cognitive Science Conference, MAICS 2001, Oxford, OH*, pp. 16-23.
- [53] R.A. Khuwaja, M. W. Evens, A. A. Rovick, and J. A. Michael, “Architecture of CIRCSIM-Tutor (v.3): A Smart Cardiovascular Physiology Tutor” in *Seventh Symposium Computer-Based Medical Systems*, 1994. pp. 158 – 163.
- [54] S. Suebnukarn and P. Haddawy, *COMET: A collaborative Tutoring System for Medical Problem-Based Learning*. Thammasat Univ., Bangkok, Vol. 22, Issue 4, *Intelligent Systems*, IEEE. 2007. pp. 70-77
- [55] B. Ferrero, A. Arruarte, I. Fernández Castro and M. Urretavizcaya, *Herramientas de Autor para enseñanza y diagnóstico: IRIS-D*. *Inteligencia Artificial* 12. 2001, pp.13-28.
- [56] G.M.J. Curilem, R. Espinosa, R. Sandoval, and F. de Azevedo Saupe, “Aportes de la Ingeniería Biomédica a la educación en salud”, *Memoria II Congreso Latinoamericano de Ingeniería Biomédica*, La Habana Cuba, 2004.
- [57] L. Martín Viera, L.M. Medina Hernández, P.A. Díaz Rodríguez, C. Paz Paula, and O.G. Chong, “Sistema tutorial para el estudio de la fisiología del aparato respiratorio del cuerpo humano”, *Educ Med Sup*, 2004. [Online]. Available: http://www.bvs.sld.cu/revistas/ems/vol18_3_04/ems04304.htm. [Accessed Abril. 08, 2009].
- [58] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.
- [59] G. M. El Saadawi, E. Tseytlin, E. Legowski, D. Jukic, M. Castine, J. Fine, R. Gormley, and R. S. Crowley, “A natural language intelligent tutoring system for training pathologists: implementation and evaluation”, *Journal: Advances in Health Sciences Education*, vol. 13, pp. 709-722, 2008.



- [60] V. Luengo, "Take into Account Knowledge Constraints for Design of TEL Environments in Medical Education" in *Eighth IEEE International Conference on Advanced Learning Technologies*, 2008, pp. 839 – 841.
- [61] A. Direne, M. Sunye, M. Castilho, F. Silva, L. Bona, L. Garcia, and D. Scott, "Acquiring Expertise in Medical Radiology through Long-Term Interactions" in *IEEE International Symposium on Computer-Based Medical Systems*, 2008 pp. 403-408.
- [62] W. Muller-Wittig, U. Bockholt, J.L.L Arcos, and G. VossI, "Enhanced training environment for minimally invasive surgery" in *Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2001. *WET ICE* pp. 269 – 272
- [63] B. Hayes, "Agents on Stage: Advancing the State of the Art of AI." in *Fourteenth International Joint Conference on Artificial Intelligence*. Morgan, 1995, pp. 967-971
- [64] P. Maes, "Situated Agents Can Have Goals, Robotics an Autonomous Systems", *Designing Autonomous Agents Book*. MIT Press. pp. 49-70, 1990.
- [65] S. Russell and P. Norving, "Artificial Intelligence: A Modern Approach", Prentice-Hall, 1996.
- [66] S. Russell and P. Norving, "Artificial Intelligence: A Modern Approach", Prentice-Hall, 1995.
- [67] Z. Hong, "Developing Formal Specifications of MAS in SLABS. A Case Study of Evolutionary Multi-Agent Ecosystem" in *Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems - AOIS (AAMAS)*, 2002, pp.59 (2).
- [68] Anónimo, "Estudio y Diseño de Agentes Móviles," in *Seminario de Desarrollo de Proyectos de Investigación*, I. T. d. C. Madero, Ed., 2002.
- [69] M. Wooldridge, "Introduction to MultiAgent Systems. John Wiley and Sons." 2002.
- [70] R. Flores, "Hacia una estandarización de los marcos de trabajo para Sistemas Multi-Agentes", Septiembre 10, 2002. [Online]. Available: <http://www.acm.org/crossroads/espanol/xrds5-4/multiagent.html>.
- [71] A. Chávez, D. Dreilinger, R. Guttman, and P. Maes, "A Real-Life Experiment in Creating an Agent Marketplace", in *Second Conference on Practical Applications of Agents and Multi-Agent Systems*, 1997, pp. 159-178.
- [72] W. Brenner, R. Zarnekow, and H. Wittig, *Intelligent Software Agents. Foundation an Aplications*. SpringerVerlag, 1998.
- [73] U. Wajid, N. Mehandjiev , "Agent Interaction Protocols and Flexible Agent Interaction in Dynamic Environments," *15th IEEE International Workshops on Enabling Technologies*, Manchester, 2006.
- [74] A. Omicini, "Coordination, models, languages applications track chair messages" in *ACM Symposium on Applied Computing*, 2001, pp. 105 – 106.
- [75] The Foundation for Intelligent Physical Agents FIPA. 2005, available at: www.fipa.org/
- [76] M. R. Genesereth and R. E. Fikes, "Knowledge Interchange Format, Version 3.0", 1992.
- [77] R. Neches, R. Fikes, T. Finin, T. Gruber, and T. Patil, W. Swartout, "Enabling technology for knowledge sharing", *AI Magazine*. 12 (3). pp. 36-56, 1991.
- [78] T. Finin, Y. Labrou, and J. Mayfield, "KQML as an agent communication language", in *3rd International Conference on Information and Knowledge Management*, 1995 pp. 456-463.
- [79] R. Rizo, F. Llorens and M. Pujol, *Arquitecturas y Comunicación entre Agentes*, Departamento de Ciencia de la Computación e Inteligencia Artificial. Universidad de Alicante. España.
- [80] C. B. D'Amico, R. M. Viccari, and L. O. Alvares, "A Framework for Teaching and Learning Environments," en *SIMPÓSIO DE INFORMÁTICA NA EDUCAÇÃO, VIII*, Sao Paulo, SP.



- [81] C. Buche, R. Querrec, P. De Loor, and P. Chevaillier, "MASCARET: pedagogical multi-agents systems for virtual environment for training", in *International Conference*, 2003, pp. 423-430.
- [82] J. Vassileva, G MacCalla, and J. Greer, Multi-agent Multi-User Modeling in IHelp. *Journal of User Modelling and User Adapted Interaction*. 13 (1-2), pp. 179-210, 2004.
- [83] C. González, J. Burguillo, J. Vidal, and M. Llamas, "SINCO: Intelligent System in Disease Prevention and Control. An Architectural Approach." in *V International Symposium of Biological and Medical Data Analysis*. ISBMDA. Lecture Notes in Artificial Intelligence, 3337, 2004, pp. 129 – 140.
- [84] J. Gascueña and A. Fernandez-Caballero, An Agent-based Intelligent Tutoring System for Enhancing E-learning /E-teaching. *International Journal of Instructional Technology & Distance Learning*. 2 (11), pp. 11-24, 2005.
- [85] J. Grunspenkis, Conceptual Framework for Integration of Multiagent and Knowledge Management techniques in Intelligent Tutoring Systems. *Advances in Theory, Practice and Education*. Springer Verlag, 2005.
- [86] C. Gonzalez, J.C Burguillo, and M. Llamas, "A Case-Based Approach for Building Intelligent Tutoring Systems", in *7th International Conference on Information Technology Based Higher Education and Training*, 2006.
- [87] C. González, J.C. Burguillo-Rial, and M. Llamas, "Integrating Intelligent Tutoring Systems and Health Information Systems". *IEEE. DEXA*. 2007, pp. 633-637
- [88] R. Imbert, L. Sánchez, A. de Antonio, G. Méndez, and J. Ramírez, "A Multiagent Extension for Virtual Reality Based Intelligent Tutoring Systems", in *Seventh IEEE International Conference on Advanced Learning Technologies*. ICALT. 2007
- [89] C. Linqin, M. Tao, and S. Yining, "Multi-agent based intelligent virtual training system" in *7th World Congress on Intelligent Control and Automation*, 2008, pp. 4226-4230.
- [90] L. Jia-Ke, W. Xuan, J. Wei, Z. Xian-Chun, and W. Chao-Fu, "Design and Evaluation of a Multi-agent System for Web Intelligent Tutoring" in *International Conference on Computer Science and Software Engineering*, 2008 vol. 5 pp. 885-888.
- [91] E. González, R. Laza and R. Pavón, "Sistema de Tutorización basado en Agentes: SITUA," ed: I Taller en Desarrollo de Sistemas Multiagente (DESMA 2004).
- [92] J. Gómez-Sanz and J. Pavón, "Methodologies for Developing Multi-Agent Systems", *Journal of Universal Computer Science*, Vol. 10, no. 4, pp. 359-374, 2004
- [93] G. Caire, F. Leal, P. Chainho, R. Evans, F. Garijo, J. J. Gomez-Sanz, J. Pavon, P. Kerney, J. Stark, and P. Massonet, Agent Oriented Analysis using MESSAGE/UML. *Lecture Notes in Computer Science*, 2222, pp. 119-135, 2002.
- [94] E. Rich and K. Knight, *Artificial Intelligence*. Libro completo. McGraw-Hill. 1990.
- [95] D. Kinny, M. Georgeff, and A. Rao, A Methodology and Modelling Technique for Systems of BDI Agents. 1997.
- [96] I. Jacobson, G. Booch, and J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*. Libro completo. Addison Wesley, 2000.
- [97] I. Sánchez Peral, "Desarrollo de un Prototipo Software para la Ejecución de Servicios Grid Semánticos Basado en Tecnología de Agentes". Facultad de Informática, Universidad de Murcia, Febrero 2008.
- [98] B. Bauer, J. P. Müller, and J. Odell, Agent UML: A Formalism for Specifying Multiagent Interaction, *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, 11 (3). 2001.
- [99] J.M Gascueña and A. Fernandez Caballero. *The INGENIAS Methodology for Advanced Surveillance Systems Modeling*. Lecture Notes in Computer Science. SpringerVerlag. Vol. 4528, 2007. pp. 541-550.



- [100] C.J Butz, S.Hua, and R.B. Maguire Bits, “a Bayesian Intelligent Tutoring System for Computer Programming”. Department of Computer Science, University of Regina. Regina, SK, Canada.
- [101] H. F. Gamboa, “Designing Intelligent Tutoring Systems: A Bayesian Approach”. Instituto de Telecomunicacoes, Instituto Superior Tecnico, Lisboa Portugal. Escola Superior de Tecnología de Setúbal, Campo do IPS, Estefanilha, Setúbal Portugal.
- [102] E. Jiménez Rey, M. Grossi and G. Perichinsky, Una Aplicación de la Tecnología de MultiAgentes a los Sistemas Tutores Inteligentes: Enseñanza de Computación en Carreras de Ingeniería. Departamento de Computación, Facultad de Ingeniería, Universidad de Buenos Aires.
- [103] N. Rajadell Puiggròs, “Los procesos formativos en el aula: estrategias de enseñanza-aprendizaje”. Facultad de Pedagogía, Universidad de Barcelona. 2001
- [104] C. M. Alonso, J. G. Domingo and P. Honey, “Estilos de aprendizaje. Procedimientos de diagnóstico y mejora,” Universidad de Deusto. Ed. Mensajero, Bilbao. 1999.
- [105] F. J.Ordóñez Muñoz, M. Rosety-Rodríguez, M. Rosety-Plaza. “Análisis de los Estilos de Aprendizaje Predominantes entre los Estudiantes de Ciencias de la Salud,” *Enfermería global*, no. 3, Noviembre 2003.
- [106] C. Alonso, D. Gallego, and J. García, “Cuestionario Honey-Alonso de Estilos de Aprendizaje,” *CHAEA Estilos de Aprendizaje*, 2006. [online], Available: www.estilosdeaprendizaje.es.
- [107] C.M. Alonso. “Estilos de aprendizaje: Análisis y Diagnóstico en Estudiantes Universitarios.” Universidad Complutense. Madrid. 1992.
- [108] C. López Fernández and C. Ballesteros Benjumedá, "Evaluación de estilos de aprendizaje en estudiantes de enfermería" *Enfermería Global*, no. 3, Noviembre 2003.
- [109] C. Monereo, M. Montserrat Castelló, M. Clariana, M. Palma and M. L. Pérez. *Estrategias de enseñanza y aprendizaje, formación del profesorado y aplicación en la escuela*. México D.F: Colofón enero 2007. [online] Available: gao.com
- [110] A. R. Martínez and O. Bonachea, ¿Estrategias de enseñanza o Estrategias de aprendizaje?
- [111] C. Alonso, D. Gallego, and J. García, “Características de los Estilos de Aprendizaje,” *CHAEA Estilos de Aprendizaje*, 2006. [online], Available: www.estilosdeaprendizaje.es.
- [112] J. E. Díaz, and A. Martins. *Estrategias de enseñanza-aprendizaje, orientaciones didácticas para la docencia universitaria. Instituto Interamericano de Cooperación para la agricultura. San José, Costa rica*: 1982. [online] Available: Amazon.com.
- [113] W. Espinosa de la Pava, “¿Qué Hacer con la Comprensión de Textos Argumentativos por parte de Estudiantes de Grados Superiores?. Ejercicios prácticos,” *Cali: Cátedra UNESCO para el Mejoramiento de la Calidad y Equidad de la Educación en América Latina con base en la Lectura y la Escritura y Universidad del Valle*, 2.005.
- [114] Java BluePrints Model-View-Controller, available at: <http://java.sun.com/blueprints/patterns/MVC-detailed.html>
- [115] J. Carnell, et al., *Pro Struts Applications: Building Web Sites with Struts, Object-Relational Bridge, Lucene, and Velocity*: APress L. P., 2003.
- [116] Struts, available at: <http://struts.apache.org/>
- [117] C. Nock, *Data Access Patterns: Database Interactions in Object-Oriented Applications*: Prentice Hall Professional Technical Reference, 2003.
- [118] Core J2EE Patterns - Data Access Object, available at: <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>
- [119] *IEEE Standard for Software Test Documentation*, IEEE 829, 1998.



- [120] D. H. Johanssen and T.C Reeves, Learning with technology: Using as cognitive tools, in *D.H Johanssen, Handbook of research for communications and technology*, pp. 693-719. 2004.
- [121] S. Graf and B. List, "An evaluation of open source e-learning platforms stressing adaptation issues," *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on*, pp. 163-165, 2005.
- [122] S. Monti and F. San Vicente, "Evaluación de plataformas y experimentación en Moodle de objetos didácticos (nivel A1/A2) para el aprendizaje E/LE en e-learning," *RedELE, Revista electrónica de didáctica/español lengua extranjera*, vol. 8, 2006.
- [123] A. Al-Ajlan and H. Zedan, "Why Moodle," *2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pp. 58-64, 2008.
- [124] E. Martínez and A. Gallego, "La creación y distribución del conocimiento a través del e-learning: ¿qué factores determinan el éxito?," *Decisiones basadas en el conocimiento y en el papel social de la empresa: XX Congreso anual de AEDEM: Ponencias.*, vol. 1, pp. 74, 2007.
- [125] M^a. P. Jiménez, "La Satisfacción del Alumnado de la Teleformación Continua," *Etic@net*, vol. 5, 2005. Available: http://www.ugr.es/~sevimeco/revistaeticanet/numero5/Articulos/Formateados/Pilar_No_guera.pdf.
- [126] J. Arias Masa, "Evaluación de la calidad de Cursos Virtuales: Indicadores de calidad y construcción de un cuestionario de medida. Aplicación al ámbito de asignaturas de Ingeniería Telemática" Tesis Doctoral. Universidad de Extremadura, España, 2008.
- [127] J. Arias Masa, "Anexo II. Validación por Expertos". Universidad de Extremadura, España, 2008.
- [128] M. Zenha-Rela y R. Carvalho, "Work in Progress: Self Evaluation through Monitored Peer Review Using the Moodle Platform", in *Frontiers in Education Conference, 36th Annual, 2006, San Diego*. CA: IEEE.
- [129] M. Berry, *An investigation of the effectiveness of Moodle in primary education. Deputy Head*. 2005, Haslemere.
- [130] J. Cole and H. Foster, *Using Moodle: Teaching with the Popular Open Source Course Management System*. O'Reilly Media, Inc, 2007.
- [131] S. Monti and F. San Vicente, "Evaluación de plataformas y experimentación en Moodle de objetos didácticos (nivel A1/A2) para el aprendizaje E/LE en e-learning," *RedELE, Revista electrónica de didáctica/español lengua extranjera*, vol. 8, 2006.
- [132] J. A. Botia, *et al.*, "IWPAAMS2007-07: The Ingenias Project: Methods And Tool For Developing Multiagent Systems," *IEEE Latin America Transactions*, vol. 6, pp. 529-534, 2008.
- [133] J. Kolodner, Maintaining Organization in a Dynamic Long-Term Memory. *Cognitive Science*, vol 7. pp. 243 -280, 1983.
- [134] E. N. Ko, "An Intelligent QOS Model for Adaptive Concurrency Control Running on Ubiquitous Computing Environments" in *Intelligent Control and Automation*. vol. 344/2006, S. B. Heidelberg, Ed., 2006, pp. 1012-1021.