

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 9



**Modelo para diseñar Mapas Conceptuales Inteligentes utilizando
el Razonamiento Basado en Casos**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

AUTOR: Sindy Soria Francis

TUTOR: Dra. Natalia Martínez Sánchez

Ciudad de la Habana, Julio del 2010

“Año 52 de la Revolución”

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Sindy Soria Francis

Natalia Martínez Sánchez

Resumen

Las Tecnologías de la Informática y las Comunicaciones ofrecen las condiciones necesarias para apoyar las labores docentes e investigativas por lo que actualmente se ha hecho cotidiano el uso de las mismas de forma auxiliar en el proceso de Enseñanza/Aprendizaje.

Los mapas conceptuales son técnicas para la representación y organización del conocimiento, y a menudo los docentes toman partido de las ventajas que implica su utilización. Los Sistemas Tutoriales Inteligentes son programas software que incorporan técnicas de inteligencia artificial para representar el conocimiento que portan sobre determinada materia y su principal objetivo es transmitir este conocimiento a través de un proceso interactivo individualizado.

La construcción de Sistemas Tutoriales Inteligentes implica un intenso y complejo trabajo de ingeniería del conocimiento y requiere además de conocimientos en el dominio de aplicación, de conocimientos en otras ramas del saber. Por esta razón actualmente su construcción solo ha sido posible a través de un enfoque multidisciplinario. Además dicha tarea demanda el uso de técnicas de Inteligencia Artificial con el objetivo de no solo representar el conocimiento sino también de mostrar adquisición del mismo y posibilitar que se simule de forma acertada como los tutores guían al estudiante en el proceso de Enseñanza/Aprendizaje. Se hace necesario dar la posibilidad a los docentes, no expertos en el campo informático, de construir sus propios Sistemas Tutoriales Inteligentes sin que tengan que depender de todo un equipo de desarrollo.

Considerando los aspectos antes mencionados se propone el diseño de una herramienta que permita la creación de Sistemas Tutoriales Inteligentes que incorporen mapas conceptuales adaptados al estado cognitivo particular de cada estudiante, utilizando Razonamiento Basado en Caso para obtener dicho carácter adaptado.

Este documento atesora la fundamentación teórica que sustenta los resultados de la presente investigación. Además expone la realización del análisis del sistema y de un modelo de diseño para éste, con vistas a sentar las bases para la futura implementación y prueba del mismo.

Palabras claves

Sistemas Tutoriales Inteligentes, Mapas Conceptuales, Razonamiento Basado en Casos.

Índice de Contenidos

INTRODUCCIÓN.....	1
CAPÍTULO 1. REFERENTES TEÓRICOS SOBRE MAPAS CONCEPTUALES Y SISTEMAS TUTORIALES INTELIGENTES.....	6
1.1. SISTEMAS TUTORIALES INTELIGENTES.	6
1.2. MAPAS CONCEPTUALES.	8
1.3. RELACIÓN ENTRE LA INTELIGENCIA ARTIFICIAL Y LOS MAPAS CONCEPTUALES.	12
1.4. HERRAMIENTAS PARA LA CONSTRUCCIÓN DE MC.	13
1.5. SISTEMAS BASADOS EN EL CONOCIMIENTO.....	17
1.6. RAZONAMIENTO BASADO EN CASOS.....	19
1.7. ANÁLISIS DE SOLUCIONES EXISTENTES.....	23
1.8. CONSIDERACIONES PARCIALES.....	27
CAPÍTULO 2. MODELO PARA DISEÑAR MCI UTILIZANDO EL RAZONAMIENTO BASADO EN CASOS.....	28
2.1. MAPAS CONCEPTUALES INTELIGENTES	28
2.2. MODELO PARA EL TRABAJO CON RAZONAMIENTO BASADO EN CASOS.....	29
2.3. MÓDULO DE ACCESO Y RECUPERACIÓN.....	29
2.4. MÓDULO DE ADAPTACIÓN.....	30
2.5. MÓDULO DE AUTOAPRENDIZAJE.....	31
2.6. CONSIDERACIONES PARCIALES.....	31
CAPÍTULO 3. TENDENCIAS, HERRAMIENTAS Y TECNOLOGÍAS.....	32
3.1. SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO. JUSTIFICACIÓN.....	32
3.2. SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN A UTILIZAR EN LA IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN. 33	
3.3. SELECCIÓN DEL ENTORNO DE DESARROLLO INTEGRADO (IDE).....	34
3.4. SELECCIÓN DE LA METODOLOGÍAS DE DESARROLLO DE SOFTWARE (MDS).....	35
3.5. LENGUAJE UNIFICADO DE MODELADO (UML): SOPORTE A LA MODELACIÓN DE LA PROPUESTA DE SOLUCIÓN.....	40
3.6. RATIONAL ROSE ENTERPRISE EDITION. HERRAMIENTA CASE PARA LA PROPUESTA DE SOLUCIÓN.....	41
3.7. SELECCIÓN DE UN SOPORTE PARA EL ALMACENAMIENTO Y REPRESENTACIÓN DE INFORMACIÓN PERSISTENTE.....	41
3.8. SELECCIÓN DE UN API PARA PROCESAR XML.....	42
3.9. CONSIDERACIONES PARCIALES.....	44
CAPÍTULO 4. PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA	45
4.1. MODELO DE DOMINIO.....	45
4.2. DEFINICIÓN DE LOS CONCEPTOS Y LAS ENTIDADES PRINCIPALES.....	45
4.3. DIAGRAMA DEL MODELO DE DOMINIO.....	47
4.4. REQUERIMIENTOS FUNCIONALES (RF).....	47
4.5. REQUERIMIENTOS NO FUNCIONALES (RNF).....	48

Índice de Contenidos

4.6.	DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.	49
4.7.	CASOS DE USO DEL SISTEMA (CUS).	50
4.8.	DESCRIPCIÓN DE LOS CUS.	51
4.9.	CONSIDERACIONES PARCIALES.	55
CAPÍTULO 5. ANÁLISIS Y DISEÑO DEL SISTEMA		56
5.1.	DIAGRAMA DE CLASES DEL ANÁLISIS POR CADA CU.	56
5.2.	DIAGRAMA DE COLABORACIÓN DEL ANÁLISIS POR CADA CU.	59
5.3.	ARQUITECTURA PROPUESTA.	61
5.4.	MODELO DEL DISEÑO.	62
5.5.	DIAGRAMA DE CLASES DEL DISEÑO POR CADA CU.	62
5.6.	DIAGRAMA DE SECUENCIA DEL DISEÑO POR CADA CU.	64
5.7.	PRINCIPIOS DE DISEÑO.	65
5.8.	CONSIDERACIONES PARCIALES.	66
CONCLUSIONES.....		67
BIBLIOGRAFÍA Y REFERENCIAS BIBLIOGRÁFICAS		68

Introducción

En la actualidad los sistemas educativos se enfrentan al desafío de utilizar las Tecnologías de la Información y la Comunicación (TIC) para proveer a sus alumnos de las herramientas y conocimientos que se requieren en el siglo XXI.

El incremento de información y el desarrollo tecnológico han influido en la renovación de las tendencias educativas, quedando obsoletas una serie de teorías y prácticas pedagógicas. En este contexto la Informática Educativa está convocada desde la interdisciplinariedad, a llenar el vacío existente en el campo de las tecnologías de la información aplicadas a la educación. Sus objetivos pueden materializarse interrelacionando y proponiendo formas que permitan introducir este avance tecnológico en las actividades y modalidades educacionales.

Consecuentemente se han diseñado e implementado ambientes de aprendizaje, que se articulan con las características de los procesos de Enseñanza/Aprendizaje en forma efectiva, lo que favorece la difusión de diferentes tipos de medios de Enseñanza Asistida por Computadora (EAC) (Vitalia 1989), según su estructura: Tutoriales, Laboratorios Virtuales, Simuladores, Entrenadores, Buscadores de Información, entre otros (Pérez 2002), (Pontes 2005).

Características propias de la EAC en lo que se refiere a la consideración de las características individuales del estudiante, el diagnóstico de las causas de sus errores y al tratamiento de los mismos en el proceso de Enseñanza/Aprendizaje, han conllevado a que los lenguajes y sistemas de autor estén evolucionando en conexión con los avances sobre Sistemas Tutoriales Inteligentes (STI) (Arias, Jiménez et al. 2007), (Ovalle 2007), (Ricucci 2008), que incorporan técnicas de Inteligencia Artificial (Aamodt 1996) (Rich 1988), (Bello 2002) mejorando las posibilidades de interacción del alumno con los programas, el acceso a la información, la presentación de contenidos y el diseño de actividades de aprendizaje (Macias 2001).

Por otro lado los Mapas Conceptuales (MC¹) son instrumentos que combinan el rigor científico con la sencillez y flexibilidad produciendo siempre una acogida unánime en los auditorios donde se han tratado, tanto de alumnos como de profesionales, pues ofrecen un nexo importante de vinculación entre la

¹ *En este trabajo se utilizará MC tanto para el plural como para el singular.*

pedagogía y la tecnología y un vasto campo de aplicaciones. Constituyen una ayuda para el que genera, trasmite, almacena y divulga, información y conocimientos, siendo importantes herramientas para lograr un mayor valor práctico en los sistemas de Enseñanza/Aprendizaje.

No solo es importante el conocimiento que debe ser facilitado por el docente hacia el alumno, sino que cobra importancia la forma en la que este conocimiento es presentado. Por lo cual el fundamento de este trabajo es aprovechar la potencia de los MC y los STI, de los primeros en la representación del conocimiento y de los segundos en la adaptación del mismo, proponiéndose el diseño de STI utilizando un modelo híbrido que resulte más potente y que combine las ventajas de los MC para la representación del conocimiento con un alto nivel de aprendizaje significativo y la utilización de técnicas de IA que faciliten realizar un proceso de Enseñanza/Aprendizaje de forma personalizada y adaptado al estado cognitivo-afectivo del estudiante.

En este contexto se tiene como **Situación Problemática** que la construcción de STI implica un complejo e intenso trabajo de ingeniería del conocimiento, que impide su uso general y aprovechamiento óptimo. Requiere además de conocimientos en el dominio de aplicación, de conocimientos en Programación e Inteligencia Artificial, y éstos no siempre coexisten en los especialistas, lo que ha motivado que el desarrollo de tales sistemas hasta ahora haya sido posible sólo con un enfoque multidisciplinario.

Teniendo en cuenta los aspectos descritos con anterioridad en términos de preguntas de investigación el **Problema a resolver** se formula:

¿Cómo facilitar a profesores no necesariamente expertos en el campo informático diseñar sus propios Sistemas Tutoriales Inteligentes en cualquier dominio de aplicación?

Por tanto en la presente investigación los Sistemas Tutoriales Inteligentes constituyen el **Objeto de estudio**.

Se tiene como **Campo de acción** los Mapas Conceptuales adaptados al estado cognitivo del estudiante como aporte a los Sistemas Tutoriales Inteligentes.

Para la realización del trabajo se trazó como **Objetivo general** diseñar una herramienta computacional que permita la navegación de Mapas Conceptuales adaptados a las características cognitivas de cada estudiante.

Para lograr este objetivo general se plantean los siguientes **objetivos específicos**:

1. Formalizar un modelo computacional donde se aplique el Razonamiento Basado en Casos en la consideración del estudiante como entidad principal.
2. Determinar una función de semejanza que permita recuperar los modelos de estudiantes más similares para diagnosticar qué necesita el estudiante y cómo enseñar.

El desarrollo de la investigación propone como **Idea a defender** que realizando el diseño adecuado de una herramienta computacional que permita la navegación de MC adaptados a las características cognitivas de cada estudiante se obtendrá una aplicación que facilita a profesores no necesariamente expertos en el campo informático construir sus propios Sistemas Tutoriales Inteligentes en cualquier dominio de aplicación.

Para dar cumplimiento al objetivo general descrito, las **Tareas generales** propuestas son:

1. Describir el estado actual de los STI y de las Herramientas para elaborar MC.
2. Describir las técnicas de IA más utilizadas en el desarrollo de los STI.
3. Analizar diferentes tipos de sistemas basados en el conocimiento, en cuanto a la forma de representar el mismo, el método de solución del problema utilizado y la complejidad en el proceso de Ingeniería del Conocimiento implícito en cada uno de los mismos.
4. Describir el modelo de sistema experto a utilizar.
5. Analizar y diseñar la solución propuesta.

Para el desarrollo de las tareas científicas se utilizan **Métodos de investigación** en la búsqueda y procesamiento de la información. Los mismos se dividen en teóricos y empíricos.

Los métodos teóricos son factibles en el estudio de las características poco observables del objeto de investigación. Dentro de este grupo se utilizan:

El método Análisis Histórico-Lógico, permitió estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. El método permitió realizar la primera parte de la investigación, al

hacer un análisis bibliográfico de los STI, MC y técnicas de IA más utilizadas en el desarrollo de STI. Dio paso a la exploración de trabajos realizados en el campo de los Sistemas Inteligentes con fines educativos y de soluciones previas existentes a problemas similares al actual. Se utilizó para determinar a través de la evaluación de la bibliografía conceptos de esta temática, que permiten conocer el estado de la evolución actual del fenómeno e identificar posibles mejoras y alternativas de solución.

El Analítico-Sintético, se utilizó para el estudio a partir de fuentes bibliográficas seguras de los STI, los MC y las técnicas de IA en el proceso de Enseñanza/Aprendizaje. Permitió además descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución propuesta.

Los métodos empíricos describen y explican las características fenomenológicas del objeto basando su contenido en la experiencia. Dentro de este otro grupo se utilizan:

Las Entrevistas, fueron dirigidas a profesores para recopilar información sobre la utilización y aplicaciones de los STI y obtener información referente al objeto de estudio que no aparece en bibliografía, pero que sí es de conocimiento de personas capacitadas. También se utilizó en el levantamiento de los requerimientos funcionales y no funcionales que precisa la propuesta de solución, diseño de la Herramienta para la Navegación de Mapas Conceptuales Inteligentes. En este caso se desea estudiar:

- En qué medida los profesores están familiarizados con el uso de los STI
- Las mejoras que los profesores proponen se le incorporen a los STI
- La aceptación de los MC como forma de organización del conocimiento en los STI

Para realizar estas entrevistas se escogió como población al conjunto de profesores, de la Facultad # 9 de la Universidad de Ciencias Informáticas, con categoría científica: Máster en Ciencias o Doctor, para un total de 13 profesores y se seleccionó como muestra 4 de ellos, lo que representa un número mayor del 30% de la población. Para la realización del muestreo, se utilizó la técnica de tipo: Muestreo aleatorio simple, que se basa en la equiprobabilidad. En este caso todos los elementos de la población (profesores) tienen la misma probabilidad de ser escogidos para formar parte de la muestra y se garantiza una mayor seguridad de que la muestra reproduzca las particularidades de la población.

La estructura de la investigación está compuesta por:

El Capítulo 1 se refiere a los conceptos teóricos asociados al dominio del problema dígase Mapas Conceptuales y los Sistemas Tutoriales Inteligentes. Además se describen aspectos de relevancia para la investigación en relación con el Razonamiento Basado en Casos y se realiza el análisis de soluciones existentes en la actualidad.

En el Capítulo 2 se expone la descripción de un Modelo para construir Mapas Conceptuales Inteligentes utilizando el paradigma de Razonamiento Basado en Casos. En este acápite se describen las características de los componentes del modelo propuesto, así como los algoritmos requeridos en los Módulos de Acceso-Recuperación y Adaptación.

El Capítulo 3 aborda sobre las tendencias y tecnologías actuales que se utilizan en el desarrollo de software, se hace énfasis en las funcionalidades y ventajas que aportan desde el punto de vista de su propósito u objetivo de creación. Este análisis permite la selección de las herramientas, como: metodología de desarrollo y lenguaje de programación para el desarrollo de la propuesta de solución.

El Capítulo 4 expone las principales características que presenta la herramienta computacional que se propone como solución al problema de investigación. En él se explican los procesos críticos para el entendimiento del entorno en el cual se desarrolla el sistema. Tomando en consideración la metodología de desarrollo escogida, se realiza la modelación de la aplicación, comenzando con la captura de requisitos, definiendo requisitos funcionales y no funcionales, así como las descripciones textuales de cada Caso de Uso.

En el Capítulo 5 se exponen los artefactos obtenidos como resultado de la realización del flujo de trabajo de Análisis y Diseño tales como diagramas de clases del análisis, diagramas de clases del diseño y diagramas de interacción.

Este documento contiene además Bibliografía, Anexos y Glosario de términos.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Capítulo 1. Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

En este capítulo se conceptualizan los Sistemas Tutoriales Inteligentes y los Mapas Conceptuales. Se analiza cómo el proceso de modelado del alumno ha motivado la aplicación de técnicas de Inteligencia Artificial. Se muestran además ejemplos de herramientas computacionales para la elaboración de MC que existen en la actualidad. Además se describe el Razonamiento Basado en Casos como una alternativa para la elaboración de los STI, valorándose las ventajas de éste.

1.1. Sistemas Tutoriales Inteligentes.

Los STI son programas que portan conocimientos sobre cierta materia y cuyo propósito es transmitir estos conocimientos a los alumnos mediante un proceso interactivo individualizado, intentando simular la forma en que un tutor o profesor guiaría al alumno en el proceso de Enseñanza/Aprendizaje (Alpigini, Peters et al. 2002), (Salgueiro, Cataldi et al. 2005), (Shneiderman 2006), (Sierra 2006).

El término inteligente se refiere a la habilidad del sistema para decidir qué enseñar, cuándo enseñar y cómo enseñar, simulando la actividad de un profesor real. Para lograrlo, un STI debe encontrar la información relevante sobre el proceso de aprendizaje del estudiante y aplicar el mejor medio de instrucción según sus necesidades individuales (Huapaya, Arona et al. 2005), (Ovalle, Jiménez et al. 2005), (Tella 2006), (Jiménez and Ovalle 2008), (Gómez 2008), (Cataldi and Lage 2009).

Un STI es un sistema de enseñanza asistida por computadora, que utiliza técnicas de Inteligencia Artificial, principalmente para representar el conocimiento y dirigir una estrategia de enseñanza. Es capaz de comportarse como un experto, tanto en el dominio de conocimiento que enseña (mostrando al alumno cómo aplicar dicho conocimiento), como en el dominio pedagógico (donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje) (Jiménez 2006).

“Es un sistema de software que utiliza técnicas de Inteligencia Artificial para representar el conocimiento e interactúa con los estudiantes para enseñárselo” (Wenger 1987).

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

En (Wolf 1984) se definen 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 1997) los delimita como: *“un sistema que incorpora técnicas de IA a fin de crear un ambiente que considere los diversos estilos cognitivos de los alumnos que utilizan el programa”*.

Definitivamente los STI son programas que utilizan técnicas de IA para representar el conocimiento que portan sobre determinada materia, mejorando las posibilidades de interacción de los estudiantes con los programas y la presentación de contenidos, pues dotan al proceso de Enseñanza/Aprendizaje de un carácter personalizado.

1.1.1. Arquitectura de un Sistema Tutorial Inteligente.

La arquitectura descrita en (Ovalle 2007) reúne los elementos comúnmente encontrados en la literatura consultada y se resumen en el criterio que plantea que un STI está compuesto por un módulo del dominio, un módulo del alumno y el módulo pedagógico, que operan de forma interactiva y se comunican a través de un módulo central que suele denominarse módulo entorno.

Módulo del Estudiante

El módulo del estudiante está presente en todos los trabajos en los que se describe la arquitectura básica de un STI. Generalmente solo se diferencian entre sí por las características a incluir para representar el modelo del estudiante.

Puede afirmarse que el modelado del estudiante es un problema de investigación que debe enfocarse desde todas sus aristas, con el fin de obtener una representación de las características del estudiante completa y precisa.

Algunos autores toman en consideración características tales como: el estilo de aprendizaje, el nivel de conocimiento, la información personal o la combinación de algunas de ellas.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Módulo del Dominio

El módulo del dominio, denominado también por muchos autores como módulo experto, contiene el conocimiento acerca del área de estudio. Satisface dos propósitos diferentes. En primer lugar, presentar la materia de forma adecuada para que el alumno adquiera las habilidades y conceptos, lo que incluye la capacidad de generar preguntas, explicaciones, respuestas y tareas para el alumno. En segundo lugar, el módulo del dominio debe ser capaz de resolver los problemas generados, corregir las soluciones presentadas y aceptar aquellas soluciones válidas que han sido obtenidas por medios distintos (Hatzilygeroudis, Karatrantou et al. 2004), (Hatzilygeroudis and Prentza 2004), (Ming 2007).

Módulo Pedagógico

Decide qué, cómo y cuándo enseñar los contenidos del tutor, adaptando sus decisiones pedagógicas a las necesidades del estudiante (Jiménez 2004). Algunos autores le denominan módulo tutor, ya que es el encargado de comparar las características de los estudiantes con el contenido a enseñar y elegir la mejor forma de tomar las decisiones pedagógicas oportunas, adaptándose en cada momento al estudiante.

Módulo Entorno

El módulo entorno o interfaz gestiona la interacción de las otras componentes del sistema y controla la interfaz persona-computadora.

Especifica y da soporte a las actividades del estudiante y a los métodos que se usan para realizar dichas actividades. Los entornos deben ser fáciles de utilizar y atractivos, de forma tal que el alumno pierda el mínimo tiempo posible en aprender a utilizar el entorno y pueda centrar toda su atención en el proceso de Enseñanza/Aprendizaje del contenido.

La necesidad de crear ambientes computacionales capaces de mantener el interés de sus usuarios, implica el desarrollo de interfaces personalizadas para ofrecer un servicio que permita un trato individualizado a un usuario en particular (Medina, Martínez et al. 2007), adaptando su interacción con el sistema a sus necesidades e intereses personales (Paredes 2001). (Fischer 2000), (Costa, Salgueiro et al. 2005).

1.2. Mapas Conceptuales.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Los orígenes de los mapas conceptuales datan de las Investigaciones realizadas durante los años 1960 por Joseph Novak², sobre psicología del aprendizaje, basadas en las teorías de David Ausubel³.

Según el propio Novak, estos constituyen una técnica que representa, simultáneamente, una estrategia de aprendizaje, un método para captar lo más significativo de un tema y un recurso esquemático para representar un conjunto de significados conceptuales, incluidos en una estructura de proposiciones (Ojeda Cabrera 2007).

Los MC son instrumentos para la representación del conocimiento, sencillos y prácticos, que permiten transmitir con claridad mensajes conceptuales complejos y facilitar tanto el aprendizaje como la enseñanza. Para mayor abundamiento, adoptan la forma de grafos (Eduteka 2006).

Según Handerson (1994) el MC aplicado al campo de las multimedia interactivas, es una estrategia de metaprendizaje efectiva, así como una herramienta de refuerzo en el aprendizaje transcultural; por otra parte señala el mismo autor: *“En cuanto que el mapa conceptual fomenta la retrospcción y tiene un carácter reactivo, podría constituir un aspecto esencial a tener en cuenta a la hora de crear buenas herramientas de navegación”* (Ríos Rodríguez 2007).

Siendo los MC un medio que permite la representación de conceptos o conocimientos, se utilizan para expresar relaciones entre ideas o conceptos y para estructurar argumentos. Las gráficas de conocimiento, según muestra la Figura 1, forman redes de conceptos consistentes en nodos y ligas. Los nodos representan los conceptos y las ligas representan la relación entre dos nodos conectados. Los conceptos generalmente son expresados por medio de sustantivos y la relación entre dos conceptos por medio de verbos o preposiciones (Fernández Beltrán 2000).

² Donald Joseph Novak (n. 1932) es un educador de América, y el profesor emérito en la Universidad de Cornell y científico de investigación en el IHMC . Es conocido por su desarrollo de mapas conceptuales en la década de 1970.

³ David Paul Ausubel (Nueva York, 1918 - 2008), psicólogo y pedagogo estadounidense, una de las personalidades más importantes del constructivismo.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Son las palabras o frases que sirven para unir los conceptos y expresar el tipo de relación existente entre ellos. Por ejemplo, para, se conoce como, posee, expresa, está formado por, es, entre otros.

Las palabras de enlace se escriben en la línea que une a dos nodos, pueden ser unidireccionales, bidireccionales o simplemente asociativas (Dürsteler 2004).

Las proposiciones.

Constituyen dos o más conceptos unidos por palabras de enlace para formar una unidad semántica más simple que tiene valor real (Ojeda Cabrera 2007).

1.2.2. Aplicaciones de los MC en la Enseñanza.

Los MC han adquirido popularidad en el ámbito educacional, en especial, porque se consideran como una herramienta que permite asociar, interrelacionar, describir y ejemplificar los contenidos de determinada rama del saber mediante el elemento visual, lo que, sin dudas, constituye una estrategia eficaz para lograr aprendizajes significativos.

Se ha promovido su uso en el campo educativo para investigar el entendimiento de un tema particular por parte de los estudiantes, para construir conocimiento y para evaluarlo. Extendiéndose cada vez más no sólo en el marco de la enseñanza presencial tradicional, sino también en las modalidades semipresencial y a distancia. Algunas de las aplicaciones de los MC en la pedagogía moderna son las siguientes:

- En la organización de planes de estudio y programas de asignaturas.
- En la elaboración de secuencias de instrucción, que no son más que la planificación de la secuencia de pasos a seguir por el profesor para enseñar un contenido, una vez que ha explorado los esquemas conceptuales de sus alumnos.
- En la enseñanza y aprendizaje de la solución de problemas.
- En el desarrollo de competencias cognitivas, para lograr el dominio y manejo lingüístico; así como para desarrollar el pensamiento crítico de los estudiantes.
- Como una herramienta para la presentación de nuevos contenidos.
- Como instrumento de evaluación para el diagnóstico.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Es importante mencionar que un MC jamás se termina pues cada vez que se revise puede ser enriquecido (Ojeda Cabrera 2007).

1.3. Relación entre la Inteligencia Artificial y los Mapas Conceptuales.

Los MC son comúnmente criticados por la comunidad de Inteligencia Artificial por la deficiencia de formalismo que está intrínseco en esta técnica. Sin embargo, sin comprometer la flexibilidad propuesta en los MC, es posible diseñar herramientas de software inteligente que influyan en la estructura, jerarquía y contexto de los mapas para llevar a cabo tareas de inferencia complejas.

La construcción de propuestas flexibles de MC pueden ser enriquecidas con la ayuda de técnicas de la Inteligencia Artificial con el objetivo de aprovechar las facilidades que brinda cada uno, dígase la representación y las vías de adquisición del conocimiento (Cañas 2004).

En el Instituto para la Cognición Humana y Mecánica (IHMC) se han implementado herramientas inteligentes en CmapTools, algunas de estas herramientas sirven de ejemplo, de cómo una herramienta inteligente puede tomar ventaja de las características de los MC (Cañas 2004). A continuación se describen algunas de ellas.

- **Sopas de conocimiento.**

Las Sopas permiten para un tipo único de colaboración entre un grupo de usuarios, usualmente estudiantes, construir cada uno un mapa en un mismo tema. Cada proposición es expresada por una sentencia simplificada la cual puede ser extraída de un mapa siguiendo los arcos, los nodos iniciales y finales, es decir, tomando pares de conceptos y sus frases conectoras. Cuando el estudiante construye el mapa, el sistema automáticamente descompone este mapa en proposiciones que son listadas en la parte superior de la ventana a la derecha del mapa. Esto permite dos diferentes representaciones de las ideas del estudiante: uno es la estructura gráfica del Cmap y otro es la manera textual.

Las sopas del conocimiento tienen muchas interpretaciones y se pueden exhibir de varias maneras. Pueden ser pensadas en como cuerpo del texto, una codificación de un grupo más grande Cmap, o una colección anotada de discusiones entre los estudiantes. El sistema tiene heurística sobre lo mal redactado de las consideraciones que publica un estudiante (Cañas 2004).

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

- **El gigante (Giant).**

El gigante se basa en un pequeño sistema de reglas que se pueden clasificar en tres categorías: transitividad; cuantificador, calificador y dependencias; y clasificación y extensión. De acuerdo con estas reglas el gigante genera sus propias conclusiones. Las reglas asocian los conceptos que emparejan, y también sirven para saber si hay pares de las palabras que se ligan que indican conclusiones razonables.

Al agente se le refiere como "gigante" porque en un sentido "sabe" mucho pero carece totalmente de conocimiento del sentido común y ha limitado las capacidades de razonar que inducen a veces un tonto comportamiento de diversión. Pero como Gigante no pretende enseñar al estudiante, realmente no importa si él no tiene siempre la razón. (Cañas 2004).

Los algoritmos genéticos, los sistemas apoyados en el razonamiento basado en casos, los sistemas expertos y otras técnicas de Inteligencia Artificial hacen posible que hoy se fomenten los sistemas adaptativos, sistemas de entrenamiento, con diagnóstico y corrección precisa, que ayuden al estudiante a mantener su conocimiento correcto y actualizado. Las Sopas de conocimiento y Gigante demuestran cómo las herramientas inteligentes pueden aprovechar la naturaleza proposicional de los MC e incluso pueden ayudar a los estudiantes a construir mejores MC.

1.4. Herramientas para la Construcción de MC.

Existen programas de cómputo para la construcción de MC que permiten economizar tiempo y esfuerzo, y obtener diseños de calidad que pueden incluir recursos visuales como el color, las imágenes, entre otros.

A continuación se hace referencia a herramientas informáticas para la elaboración de los MC:

CmapTools.

CmapTools es una herramienta desarrollada por el IHMC, de la Universidad de West Florida (Estados Unidos), con la finalidad de propiciar la construcción, navegación, crítica y compartición de modelos de conocimiento representados como mapas conceptuales. Se diseña con el objetivo de apoyar la construcción de modelos del conocimiento representados en forma de MC, pero también se pueden

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

elaborar telarañas, mapas de ideas y diagramas causa-efecto. Permite a los usuarios construir los mapas en la computadora y después publicarlos en servidores Web públicos en Internet o en servidores propios.

Entre las principales ventajas está su carácter gratuito para instituciones sin fines de lucro y las posibilidades de gestión de los mapas subidos al servidor (Cabrera 2006).

Posee un entorno de trabajo sencillo, claro e intuitivo; ventana de estilos que facilita el trabajo; posibilidad de ilustrar los conceptos con símbolos, imágenes, colores, formas, sombras, fuentes y estilos; facilidades para relacionar conceptos en forma sencilla; relaciones que se explican con un texto en los enlaces; entre otras ventajas.

Permite exportar los gráficos elaborados en forma de: imagen (jpg, gif, png, bmp, entre otros.), página Web, texto o formato XML. Es compatible con los sistemas operativos (SO) Windows, Mac OSX, Linux (Intel) y Solaris (Sparc) (Eduteka 2006).

Inspiration.

Es una herramienta de aprendizaje visual, para estudiantes de 6º - 11º, muy utilizada por los docentes de todo el mundo. Especialmente diseñada para la creación de diagramas en forma de telaraña, mapas de ideas y MC. Permite exportar los mapas creados a formatos gráficos como jpg, gif y bmp. Compatible con los SO Windows y Macintosh (Ojeda Cabrera 2007).

Cmap Toolkit.

Herramienta de software abierto para construir, compartir, navegar y debatir modelos de conocimiento representados en forma de MC. Está habilitada para el trabajo en red, permite a los usuarios construir y colaborar con sus colegas durante la construcción del MC, por medio de Internet. Es muy intuitiva y fácil de utilizar. Compatible con el SO Windows (Ojeda Cabrera 2007).

Smart Ideas.

Facilita la elaboración de mapas de ideas, telarañas, MC, diagramas de flujo, diagramas causa-efecto, organigramas, entre otros.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Ofrece un entorno de trabajo que se configura de acuerdo con el tipo de diagrama que se elabore; es un programa sencillo, claro e intuitivo. Permite exportar los diagramas creados a formatos como jpg, gif, png, bmp, entre otros. Ofrece librerías, plantillas y ejemplos; los diagramas se pueden elaborar partiendo de cero o basándose en una plantilla o un ejemplo. Compatible con SO Windows y Mac (Ojeda Cabrera 2007).

Axon2002.

Esta herramienta para la presentación y organización de ideas se vale de atributos como: color, forma, tamaño, escala, posición, profundidad, sombras, enlaces e iconos, para facilitar la memorización, asociación y el descubrimiento. Soporta estructuras jerárquicas y de redes. Posee un generador de ideas. Las ideas se muestran como objetos gráficos y sus relaciones como enlaces. Se pueden adicionar plantillas de fondo, texturas e imágenes. Soporta hipertexto y texto enriquecido. Exporta hacia html, texto plano, y texto enriquecido. Compatible con el SO Windows (Ojeda Cabrera 2007).

OpenOffice Draw.

Este programa gratuito forma parte de la suite de oficina de OpenOffice.org, y se diseñó especialmente para elaborar gráficos y diagramas en general. Es apropiado para que los estudiantes realicen organigramas, telarañas, mapas de ideas, MC y diagramas causa-efecto. Su instalación es sencilla, pero es necesario instalar toda la suite de oficina de OpenOffice.org. Compatible con los SO Windows, Linux y Solaris (Ojeda Cabrera 2007).

Macosoft.

Software que integra un conjunto de requerimientos importantes que permiten de forma fácil y rápida la elaboración de MC. La construcción, organización, almacenamiento y evaluación del conocimiento, son elementos esenciales en todo proceso de gestión del conocimiento y aprendizaje, y la integración automatizada de estos componentes en MACOSOFT contribuye a que estos procesos se realicen de forma más efectiva.

Es un software que automatiza el proceso de elaboración de MC, con mecanismos fáciles y cómodos de inserción, eliminación de conceptos y enlaces, modificación de sus propiedades, al igual que mecanismos

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

de modificación de la interfaz. Uno de los elementos novedosos que se han incorporado en esta herramienta es la definición de forma automática del nodo-concepto principal o más inclusivo del MC, se permite la identificación de forma significativa de este nodo-concepto principal. Los conocimientos que se representen en el MC, con esta herramienta pueden ser almacenados en un fichero con extensión propia (MCF) y además tiene implementado la salva en diferentes formatos, tales como: imagen (BMP), HTML y XML.

La Tabla 1 muestra una comparación entre herramientas que permiten la construcción de MC y se concluye que CmapTools es la más apropiada para la edición de los MC, requeridos como entrada al sistema que se propone como solución. Esta selección se basa principalmente en la facilidad que brinda el CmapTools de salvar los MC en formato XML y en la posibilidad de incorporar material didáctico que enriquezcan y eleven la calidad del contenido presentado por los MC.

Tabla 1. Comparación entre distintas herramientas existentes para el diseño de MC.

Criterios de comparación.	Inspiration (v.7.5)	CmapTool(v.3.0)	Smart Ideas(v.1.2.7)	MACOSOFT
Representación de los conceptos (uso de imágenes)	A través de figuras prediseñadas. Brinda una gran Biblioteca de figuras prediseñadas con posibilidades de inserción y creación.	A través de rectángulos y permite asociar imágenes. No hay Biblioteca de imágenes.	A través de figuras prediseñadas. Biblioteca de figuras con posibilidades de inserción y creación.	A través de figuras prediseñadas. Biblioteca de imágenes con posibilidades de personalizarla. Representación del concepto más inclusivo. Definición del tipo de enlace.
Vínculo de recursos a los nodos-concepto	Vincula un único elemento del tipo: doc, mapas, ppt, video, imagen, html.	Vincula: doc, mapas, ppt, video, imagen, html, exe.	Vincula un único elemento del tipo: doc, mapas, ppt, video, imagen, html.	Vincula: doc, mapas, ppt, video, imagen, html, exe.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Portabilidad del MC, salva en diferentes formatos	HTML, GIF, JPG, BMP y WMF	HTML, PDF, JPG, SVG, TXT y XML, extensión propia(.CMAP).	HTML	JPG, BMP, HTML y XML, extensión propia (.MCF).
Idioma utilizado en la interfaz	Inglés	Configurable el idioma: Español, Inglés, Italiano, Francés, Alemán, Ruso, entre otros	Configurable el idioma: Español, Inglés, Italiano, Francés y Alemán	Español

1.5. Sistemas Basados en el Conocimiento.

Los sistemas basados en el conocimiento constituyen técnicas de la Inteligencia Artificial válidas para enfrentar la construcción de STI dado por sus aspectos afines. Al utilizar un sistema basado en el conocimiento la solución que se obtiene es similar a la obtenida por una persona experimentada en el dominio del problema. Por su parte los STI utilizan la información almacenada sobre las características del estudiante para adaptar el proceso de Enseñanza/Aprendizaje del mismo a la materia a enseñar.

Una característica distintiva de los sistemas basados en el conocimiento es la separación del conocimiento (base de conocimiento) del método de solución del problema (máquina de inferencia). La construcción de la base de conocimiento lleva implícito un arduo proceso de adquisición del conocimiento y es particular para cada sistema, por lo que será necesario construirla para cada aplicación. Sin embargo, la máquina de inferencia puede reutilizarse en la construcción de varios sistemas basados en el conocimiento siempre que el tipo de conocimiento y el tipo de razonamiento sea similar. La máquina de inferencia es el método implementado que utiliza el conocimiento de la base para resolver los problemas del dominio.

Diferentes tipos de conocimiento dan lugar a diferentes tipos de sistemas basados en el conocimiento, entre ellos los sistemas basados en reglas(Hand 1997), (Rich 1988), los sistemas basados en probabilidades (Pearl 1988), (Castillo 1997), (Minka 2001), (Lerner 2002), sistemas expertos conexionistas o redes expertas (Barr and Mani 1994), (Hilera and Martínez 1995), (Ohno-Machado 1996) y los sistemas

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

basados en casos (Dutta 1991), (Kolodner 1992), (Aamodt 1996), (García and Bello 1997) (Gutiérrez and Bello 2003). El tipo de conocimiento determina qué método de solución de problemas es posible utilizar.

La envergadura del proceso de adquisición del conocimiento depende del tipo de conocimiento.

En los **Sistemas Basados en Reglas** se desarrolla un proceso complejo y prolongado pues la extracción se refiere a la formalización de reglas y el pensamiento humano no siempre está regido conscientemente por las reglas de la lógica; en ocasiones es básicamente un procesamiento de información recuperada con el tiempo. Éstos utilizan reglas de producción como forma de representar el conocimiento y usualmente la búsqueda en profundidad con dirección hacia delante o dirigida por datos y hacia atrás o dirigida por objetivo como método de solución de problemas.

En los **Sistemas Basados en Probabilidades** la adquisición del conocimiento consiste en coleccionar muestras y realizar un procesamiento estadístico que produzca las probabilidades o frecuencias que forman la base de conocimiento. No son factibles para todo tipo de dominio, pues se dificulta construir las redes con ayuda de expertos humanos cuando existen carencias de conocimiento. No son viables para explicar el razonamiento, ya que los métodos y modelos que utiliza están aún lejos de ofrecer explicaciones comprensibles. Éstos utilizan probabilidades o frecuencias como forma de representar el conocimiento y generalmente el Teorema de Bayes como método de solución de problemas.

En las **Redes Expertas** la adquisición del conocimiento incluye la selección de los ejemplos, el diseño de su topología y el entrenamiento de la red para hallar el conjunto de pesos. Facilitan el trabajo con información incompleta y brindan algoritmos poderosos de aprendizaje para crear la base de conocimiento; pero requieren de muchos ejemplos y son cajas negras que no explican cómo se alcanza la solución. Éstas generalmente utilizan pesos como forma de representar el conocimiento y el Cálculo de niveles de activación de las neuronas como método de solución de problemas.

En los **Sistemas Basados en Casos** la adquisición del conocimiento se reduce a la selección de un conjunto de ejemplos o casos resueltos y su organización en la base de casos. Argumenta una solución mediante los casos que son relevantes al nuevo problema. Cada caso es la experiencia anterior almacenada. Su dificultad radica en la definición adecuada de la función de semejanza, al no existir una función de semejanza general apropiada para cualquier problema. Los Sistemas basados en casos

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

utilizan casos como forma de representar el conocimiento y el paradigma de Razonamiento basado en casos como método de solución de problemas.

Se concluye que no todos los paradigmas para crear sistemas basados en el conocimiento facilitan la concepción de un STI. Por tanto, teniendo en cuenta que la información y conocimientos que se manipulan en la propuesta de solución pueden enmarcarse en cualquier dominio de aplicación y considerando además que la fuente de conocimiento de la misma puede funcionar sobre la base de la concepción de ejemplos o vivencias, resultaría oportuna la utilización de un Sistema Basado en Casos (SBC).

1.6. Razonamiento basado en casos

El Razonamiento Basado en Casos (RBC) es definido por (Schank 1996) como *“el proceso de resolver problemas nuevos a través de la adaptación de soluciones que fueron usadas para resolver problemas viejos”*.

Según (Agnar 1994) el RBC es el proceso de solucionar nuevos problemas basándose en las soluciones de problemas anteriores. Es una manera de razonar haciendo analogías.

Es una tecnología derivada de la Inteligencia Artificial que representa el conocimiento como casos que fueron usados para resolver problemas pasados (Kolodner 1993).

Se ha argumentado que el RBC es más que un método poderoso para el razonamiento de computadoras, pues es usado por las personas para solucionar problemas cotidianos. Más radicalmente se ha sostenido que todo razonamiento es basado en casos porque está basado en la experiencia previa.

El razonador basado en casos depende de una base de casos en la cual se almacena el conocimiento necesario para resolver los problemas del dominio de aplicación para el cual se desarrolla el SBC. Por tanto este sistema de razonamiento se basa en una unidad mínima llamada caso, el cual tiene dos componentes: rasgos predictores (descripción del problema) y rasgos objetivos (solución del problema).

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Un caso se puede definir como la representación de una experiencia anterior o una vivencia (Agnar 1994). Un caso puede ser visto como una pieza en la representación de la memoria o como una experiencia pasada, que es recordada para ser utilizada en el futuro.

En fin el RBC es un paradigma para la resolución de problemas nuevos partiendo del análisis y adaptación de soluciones que fueron dadas a problemas previos, que se encuentran almacenados y organizados en una biblioteca o base de casos. El RBC consiste básicamente en el procesamiento de la información que se recupera, en el momento oportuno. Las anteriores definiciones sirven de base para las futuras etapas en las que se realiza la integración de estos tres conceptos: Base de casos, Caso y RBC.

Con el propósito de describir de forma más clara el funcionamiento básico de los sistemas RBC a continuación se utiliza como ejemplo: el sistema CHEF (Hammond 1986), que aunque opera en el dominio de la cocina china es uno de los ejemplos más conocidos e ilustrativos.

CHEF tiene como tarea construir nuevas recetas a partir de un recetario, en base a un conjunto de requisitos del usuario y se inicializa con una biblioteca de aproximadamente 20 recetas. La entrada del sistema es una lista de objetivos tales como: "plato caliente con pollo y brécol rehogados". Los objetivos se refieren a diferentes gustos, texturas, ingredientes y tipos de platos y la salida de CHEF es una receta única que satisface todos los objetivos. Antes de buscar un plan de elaboración de una receta para modificarlo, CHEF examina los objetivos en su entrada y trata de anticipar cualquier problema que pueda surgir mientras tiene lugar la planificación. Una vez obtenida la receta el usuario la evalúa y si encuentra algún problema realimenta el sistema con un informe de fallos. En este último caso CHEF repara la receta y también modifica la biblioteca de casos para evitar cometer el mismo tipo de error en situaciones futuras similares.

Los procesos medulares involucrados en el RBC son recodar y adaptar. Partiendo del ejemplo anterior se identifican como las etapas básicas de un razonamiento basado en casos:

1. Recuperación de casos similares a la situación actual.
2. Adaptación de la solución o soluciones previas al problema actual.
3. Revisión de la solución.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

4. Almacenamiento del caso resuelto.

Recuperación de casos

Constituye la primera etapa en el ciclo de funcionamiento de un sistema RBC, donde dada la descripción de una situación o problema y un conjunto de objetivos o tareas que hay que llevar a cabo, se trata de encontrar el caso o un pequeño conjunto de casos que por su similitud pueden resultar de utilidad. Además, ya que en muchas ocasiones las características correspondientes de los casos no siempre son comparables se hace necesario definir y especificar en este módulo un método para comparar estas características y que permita asesorar acerca de la similitud entre casos.

Adaptación de casos

Un adaptador de casos toma la descripción de un problema actual, la descripción de un problema similar y la solución a este, para construir con estos elementos la solución del problema actual.

El proceso de adaptación se compone de dos fases: identificación de aquello que hay que adaptar y aplicación de un método de adaptación. Además en éste se pueden seguir dos estrategias: adaptación estructural, que consiste en aplicar las reglas de adaptación directamente a la solución almacenada en un caso, y adaptación derivacional, que consiste en utilizar y a la vez modificar las reglas que generaron la solución original, partiendo de las características del nuevo problema para generar la solución a éste. De estas dos estrategias la más adecuada para ser utilizada en la propuesta de solución es la estrategia de adaptación estructural, considerando que no existirá un historial con reglas de generación de soluciones pues, para obtener las nuevas soluciones será suficiente con adaptar directamente las soluciones que proponen los casos antiguos.

Los métodos de adaptación implican operaciones tales como: la inserción, borrado, sustitución y división, tanto de valores como de estructuras. Los principales son:

- Adaptación nula.

Consiste en transferir la solución que propone un problema similar recuperado directamente como solución al problema actual.

- Adaptación transformacional.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

Consiste en reutilizar soluciones recuperadas después de aplicárseles determinadas transformaciones que generen la solución buscada. Estas transformaciones se pueden realizar ajustando parámetros o substituyendo valores.

- Adaptación derivacional.

Consiste en realizar la adaptación de algún elemento de la solución que no se ajusta correctamente reaplicando, para el problema actual, el procedimiento que fue utilizado para calcular dicho elemento de la solución en el caso previo.

- Sistemas cooperativos.

Al adaptar utilizando esta técnica se establece una cooperación entre el sistema y sus usuarios, donde el sistema únicamente presenta las soluciones relevantes y el experto humano es quien se encarga de realizar la adaptación de éstas.

Revisión

La fase de revisión consiste en dos tareas: 1. evaluación de la solución generada y 2. si la solución se identifica como incorrecta se repara recurriendo a los métodos de adaptación.

Existen distintas formas de comprobar si la adaptación tuvo éxito, puede ser comprobando las restricciones de entrada o buscando casos similares ya computados o simulando las soluciones o aplicando la solución en el entorno real.

Almacenamiento

En la fase de aprendizaje o almacenamiento se procede a almacenar el caso recién resuelto en la base de casos para uso futuro. Este proceso comprende tres fases: determinar que se necesita almacenar, definir como será indexado el nuevo caso y finalmente integrar el caso a la memoria de casos.

Atendiendo al conjunto de facilidades que proporcionan, existentes tres clasificaciones de RBC: Sistemas consejeros, Sistemas con recuperación y adaptación y Sistemas con recuperación, adaptación y reparación. En esta investigación se utiliza la segunda de estas clasificaciones, en la cual el usuario

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

describe una situación, y el sistema encuentra y adapta casos previos similares, y obvia la evaluación y reparación. La Figura 3 esquematiza dicho proceso.

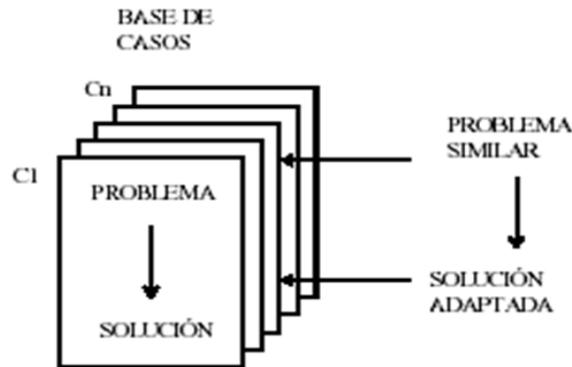


Figura 2. Recuperación y adaptación de casos.

La utilización de esta técnica en la solución propuesta es fundamental, debido a que la tarea de adquisición del conocimiento se simplifica dado que les permite a los expertos hablar de casos en lugar de tener que hacer el esfuerzo de extraer reglas analizando y generalizando. Además la utilización de una base de casos evita tener que derivar de nuevo soluciones ya obtenidas y recordar problemas previos.

1.7. Análisis de soluciones existentes.

Actualmente, se está dando un proceso de generalización y aprovechamiento de la experiencia ganada en el desarrollo de STI y, al mismo tiempo, está ocurriendo una transición de sistemas experimentales a comerciales junto con el desarrollo de herramientas para el diseño de dichos sistemas (Romero 2000). A continuación se describen herramientas que permiten el desarrollo de aplicaciones de STI.

IRIS

IRIS (IRakaste-Ikaste Sistema; Sistema de Enseñanza/Aprendizaje) es un entorno de construcción de Sistemas Tutores Inteligentes. Sin embargo, aunque los STI que genera IRIS tienen la capacidad de realizar una instrucción adaptada al alumno, carecen de los componentes necesarios para llevar a cabo el diagnóstico de su actividad durante los procesos de resolución de ejercicios procedimentales. IRIS facilita el desarrollo de nuevos tutores mediante la reutilización y personalización de toda la arquitectura de Intza

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

(Gutiérrez 1996). IRIS trabaja en dos fases que ocurren de forma secuencial. En la primera fase, se genera la estructura tutor sobre la base de las necesidades, y en la segunda fase, los módulos definidos son llenados con contenido. Los tutores generados por IRIS se basan en el paradigma basado en reglas (Arruarte 1997).

IRIS-D

La herramienta de autor IRIS-D es el resultado de la ampliación del sistema IRIS (Arruarte 1997) una vez integrado a éste el sistema de diagnóstico genérico: DETECTive (Ferrero 1997). IRIS-D es un entorno que ayuda a construir Sistemas Tutores Inteligentes, el objetivo principal es ayudar a los profesores que no son expertos con la tecnología a construir sistemas adaptativos de Enseñanza/Aprendizaje. El proceso de IRIS-D consiste en adaptar, modificar y completar una arquitectura básica o genérica de un tutor para poder cumplir los requerimientos del profesor. La arquitectura de IRIS-D está integrada por 4 componentes fundamentales que son (Ferrero 2001):

- Interfaz. Gestiona todo el proceso de comunicación de IRIS-D con el exterior (instructor).
- Generador de la estructura del tutor. Elige la arquitectura del nuevo sistema tutor. Esto se hace con base a los requerimientos del profesor y a partir de arquitecturas de tutores ya construidas.
- Generador de objetos. Adapta, completa y almacena la estructura de cada componente.
- Generador de reglas. Dentro de cada componente del módulo pedagógico deben existir reglas de decisión pedagógica.

ETG

ETG es un software educativo, entrenador y generador de tutoriales, producto resultante del proyecto de investigación PI N° 530 "Generador de entrenadores basados en componentes reusables" de la Universidad Nacional de la Patagonia San Juan Bosco, Argentina. ETG facilita la tarea de crear tutoriales como recurso didáctico, que colaboren y potencien los procesos de aprendizaje en la educación superior y sirvan de apoyo al proceso de Enseñanza/Aprendizaje. Al mismo tiempo, el alumno al interactuar con esos tutoriales, puede analizar el problema y la solución propuesta, tantas veces como lo desee, incrementando así su capacidad de asimilación y fijación de conceptos. También permite que el alumno realice prácticas

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

sobre el tema, y en caso de cometer errores, puede intervenir guiándolo hacia la solución correcta, sin necesidad de la permanente presencia del docente, a la vez que le brinda un mecanismo de auto evaluación (Rosanigo 2007).

HEDEA

HEDEA (Herramienta de Autoría) es una herramienta de autoría que trabaja con un laboratorio virtual y permite desarrollar un STI a partir de la definición temática de un curso. Dichos STI se basan en modelos probabilísticos partiendo del temario pesado de un curso. La generación del modelo del estudiante se realiza de manera automática y es transparente al usuario. Los STI generados por la herramienta son asociados a un laboratorio virtual existente. El modelo del estudiante toma en cuenta los valores de experimentos previos lo cual permite darle mayor valor al historial o a su último resultado (Romero 2009).

PLAGENSTI-IIE1

Plataforma para la generación de Sistemas Tutoriales Inteligentes en la capacitación de personal técnico de centrales eléctricas. Esta plataforma genérica para el desarrollo de aplicaciones de STI en capacitación y entrenamiento fue diseñada principalmente tomando en cuenta su reutilización o su grado de generalidad. Para representar el conocimiento utiliza la representación en forma de nodos, que simbolizan unidades de aprendizaje, lecciones o conceptos y arcos, lo que da lugar a una especie de redes tipo árbol. La arquitectura de PLAGENSTI-IIE1 está integrada por 4 componentes fundamentales que son (Romero 2000): módulo del domino, modelo estudiante, módulo tutor y módulo interfaz.

E-Vitro

Es una herramienta que permite a los usuarios (estudiantes) construir seres virtuales o agentes que interactúan independientemente en un entorno controlado. Cada estudiante define el comportamiento de su agente o agentes según los conceptos aprendidos en la clase. Su objetivo principal es proveer al estudiante de un mundo virtual donde pueden aplicar toda la teoría explicada en clase y además pueden utilizar una biblioteca de programación donde se encuentran los algoritmos implementados elementales para la solución de diferentes problemas. Con el apoyo de e-Vitro se realizan dos tipos de análisis para obtener información acerca del aprendizaje de los estudiantes. El primero es un análisis automatizado de los agentes que el estudiante desarrolla y el segundo, se basa en el seguimiento de cómo el estudiante

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

interactúa con e-Vitro. Los resultados arrojados por e-Vitro en los dos aspectos anteriores son entregados al docente para que él los analice y determine si necesita hacer alguna observación al estudiante. (Martinez 2005).

AMBAR

AMBAR es un sistema generador de ambientes de Enseñanza/Aprendizaje basados en objetos de aprendizaje. El objetivo del proyecto de investigación y desarrollo AMBAR es desarrollar una plataforma tecnológica que soporte el almacenamiento, generación, uso y reúso de Objetos de aprendizaje y diseños de aprendizaje en ambientes instruccionales bajo enfoques cognitivo-constructivistas. Este sistema se caracteriza por: estar basado en la Web, fácil uso e ilimitado acceso, participación activa en los procesos de enseñanza y aprendizaje (Lopez 2008).

MAS-PLANG

MAS PLANG es un sistema multiagente que permite por medio de tecnología Web, mejorar aspectos adaptativos de la plataforma USD⁴. MAS-PLANG fue desarrollado utilizando una arquitectura con dos niveles de agentes (agentes de información y de soporte). Estos agentes identifican el comportamiento y permiten personalizar el material de estudio ofrecido al estudiante y las herramientas de navegación. MAS- PLANG se divide en 3 plataformas: (Velez 2007):

- Cliente USD dentro del que se encuentra un navegador estándar.
- Servidor USD dentro del que se encuentra un Servidor Web Estándar, la Base de Datos (Historial de actividades del estudiante, modulo del dominio, bases de conocimiento de los Agentes) y Contenidos (Paginas HTML).
- HabitatPro⁵ dentro del que se encuentra un Java Application Server (maneja JSP, Servlets y Java Beans) y la Java Virtual Machine.

⁴ Plataforma educativa utilizada para el soporte a la educación a distancia a través de la web, basándose en los estilos de aprendizaje. Fabregat, R. M., J.; Peña, C. (2000). "Teaching Support Units."

⁵ Herramienta diseñada para la personalización de contenidos y sondeo de mercados utilizando técnicas de Razonamiento Basado en Casos y Reglas de Lógica Difusa. Habitat-Pro, E. (2001). Agents Inspired Technologies S.A. Girona, España, Universidad de Girona.

Capítulo 1: Referentes Teóricos sobre Mapas Conceptuales y Sistemas Tutoriales Inteligentes

SCOMAX

La Herramienta SCOMAX es un Sistema Inteligente basado en MC evolucionados en la Automatización de un Aprendizaje Significativo para la enseñanza universitaria de la asignatura de Jerarquía de Memoria. Fue elaborada por especialistas de la Facultad de Física y Matemáticas, Universidad de La Laguna, Tenerife, España.

Esta herramienta presenta una evolución de los tradicionales MC, a los que denomina Grafo Instruccional Conceptual Pedagógico (PCIG). Su funcionamiento básicamente consiste en:

El profesor describe el conocimiento de la materia mediante uno de estos grafos evolucionados, como entrada requerida al sistema. Este sistema inteligente hace uso de un cuestionario adaptativo, generando varias preguntas sobre los conceptos incluidos en ese PCIG. Luego de las respuestas que se obtengan por parte de los estudiantes, se construye un PCIG personalizado para cada alumno conteniendo la información de qué conceptos conoce a priori y, no menos importante, con qué profundidad (Moreno 2007).

Del análisis realizado a estos sistemas se concluye que son valiosas herramientas para la elaboración de STI; pero a su vez no están al alcance de todos, debido a su poca difusión, complejidad en el manejo y carácter propietario. No obstante el estudio de las mismas sirve de base a la definición de las características del sistema propuesto y promueve la incorporación en éste de aspectos comunes que resultan de utilidad, como es el caso de estructuración de la arquitectura y los elementos que generalmente componen a cada uno de los módulos de dicha arquitectura.

1.8. Consideraciones Parciales.

Los MC son herramientas gráficas eficaces para la representación y organización del conocimiento. Se propone complementar el concepto MC con los STI para lograr una navegación inteligente, surgiendo así el nuevo concepto de Mapas Conceptuales Inteligentes. Estos últimos constituyen un apoyo al proceso de Enseñanza/Aprendizaje y un nuevo modelo para diseñar los actuales STI.

Capítulo 2: Modelo para diseñar MCI utilizando el Razonamiento Basado en Casos

Capítulo 2. Modelo para diseñar MCI utilizando el Razonamiento Basado en Casos

Retomando los aspectos descritos en los epígrafes 1.8 y 1.9, en este capítulo se presenta el modelo que facilita la elaboración de MCI a partir de las componentes fundamentales de los SBC. Se describen los algoritmos que conforman el modelo para el acceso y recuperación de casos de la Base de Casos, así como el algoritmo de adaptación que se le aplica a las soluciones que proponen el conjunto de casos semejantes.

2.1. Mapas Conceptuales Inteligentes

La esencia del modelo de un MCI es elaborar un STI a través de un MC, que permita una navegación adaptada al estado cognitivo del estudiante. Los MCI no son más que los MC descritos a lo largo del epígrafe 1.4 pero con la característica particular de poseer un carácter adaptativo, pues a partir de un MC genérico pueden derivarse otros, que restringen su navegación en correspondencia con los conocimientos previos del estudiante específico, al cual va dirigido y describe lo que conoce a priori de la materia a aprender. Tienen un fuerte basamento en los conocidos STI, de los cuales importan un gran número de características y como tal son el resultado de combinar los MC con los Sistemas Basados en Casos para lograr una navegación personalizada.

La arquitectura general de un MCI está conformada por los tres módulos principales de todo STI. El módulo del estudiante y el modelado del estudiante se implementan a través de un Sistema Basado en Casos. El módulo del dominio y el módulo pedagógico se representan a través de un MC, cuyos nodos contienen materiales didácticos elaborados para la materia que aborda el sistema tutorial utilizando las estrategias pedagógicas necesarias. Estos dos módulos conforman el rasgo objetivo de los casos que conforman la Base de Casos.

El diseño de un Mapa Conceptual Inteligente utilizando Razonamiento Basado en Casos comprende dos fases:

La primera fase consiste en la edición del MC utilizando CMapTools como herramienta para editar MC. Esta fase implica también la confección de un conjunto de cuestionarios relacionados con la materia

Capítulo 2: Modelo para diseñar MCI utilizando el Razonamiento Basado en Casos

definida en el módulo dominio del MC, que permitan captar el estado cognitivo del estudiante. Los materiales didácticos que se elaboran para incluir en los nodos del MC llevan implícita las estrategias pedagógicas necesarias para lograr un proceso de Enseñanza/Aprendizaje personalizado.

La segunda fase comprende la interacción del estudiante con el MCI. En esta fase a partir de la aplicación de un cuestionario y la evaluación de éste, como estrategia para captar su estado cognitivo, y utilizando RBC se realiza el diseño dinámico de un Mapa Conceptual Inteligente.

2.2. Modelo para el trabajo con Razonamiento Basado en Casos

En este modelo los casos son almacenados en una Base de Caso (BC) la cual posee una estructura lineal. Los casos representan el estado del conocimiento y comportamiento del estudiante, así como el entrenador o material didáctico que se adapta a las características del estudiante. Éstos serán indexados combinando el identificador del MC y del cuestionario al cual se encuentre asociado.

Los rasgos predictores representan el modelo del estudiante descrito por la evaluación obtenida en cada pregunta al realizar un cuestionario, éstos toman valores binarios (0,1) donde 1 significa respuesta correcta y 0 significa respuesta incorrecta. El rasgo objetivo contiene el conjunto de nodos que conforman el material didáctico que se adapta al estado cognitivo del estudiante

2.3. Módulo de acceso y recuperación.

El primer paso para dar solución a un nuevo caso es la recuperación de los casos más semejantes a él. Precisamente, como se expuso anteriormente, este módulo se concibe con la finalidad de acceder y recuperar dichos casos semejantes. La función de semejanza que se utiliza en este modelo, tomada de (Morales 1999), es:

$$\beta(O_0, O_T) = \frac{\sum_{i=1}^n w_i * \partial_i(O_0i, O_Ti)}{\sum i} \quad (1.1)$$

Capítulo 2: Modelo para diseñar MCI utilizando el Razonamiento Basado en Casos

Donde w_i es la importancia del rasgo y cuyo valor se define por criterio del experto, O_0i y O_Ti son los valores que el rasgo i tiene en el caso nuevo y en el caso ya almacenado respectivamente, y δ_i , tomada de (Bello 2002), es la función de comparación para el rasgo i y se define como:

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \quad (1.2)$$

Se realiza una comparación del nuevo caso con cada uno de los casos almacenados en la BC utilizando la función de semejanza (1.1) y considerando un valor umbral (μ) tomado como patrón para medir la semejanza y predefinido por el experto, se conforma el conjunto de casos semejantes.

Algoritmo 1. Acceso a los casos más semejantes

Entrada: Caso nuevo (O_0) y Base de casos (BC').

Salida: Conjunto de casos más semejantes (C_S).

A1: Calcular $\beta(O_0, O_T)$ para cada $O_T \in BC'$.

A2: Si $\beta(O_0, O_T) \geq \mu$ entonces:

-- $C_S \leftarrow O_T$.

Sino

-- Descartar O_T

2.4. Módulo de adaptación.

Después de seleccionados y recuperados los casos más semejantes, las soluciones que proponen dichos casos pueden utilizarse directamente como solución al nuevo problema o si fuese necesario se realiza una modificación de éstas, este método de solución del problema es conocido como adaptación de las soluciones. El modelo propone que la adaptación de la solución se realice utilizando la adaptación nula y la adaptación transformacional como métodos de adaptación según se requiera. La adaptación transformacional se realizará a través de la intercepción de los MC que proponen como solución los rasgos objetivos de los casos recuperados. Este proceso se describe a través del siguiente algoritmo.

Capítulo 2: Modelo para diseñar MCI utilizando el Razonamiento Basado en Casos

Algoritmo 2. Adaptación de casos semejantes

Entrada: Conjunto de casos más semejantes (C_S) y Caso nuevo (O_0).

Salida: Caso adaptado (O_A).

A1: $U \leftarrow$ Obtener \bigcap de los rangos objetivos de los elementos de C_S .

A2: Si $U \neq \emptyset$

-- Rasgos predictores de $O_A \leftarrow$ Rasgos predictores de O_0 .

-- Rasgo objetivo de $O_A \leftarrow U$

Sino

-- $U' \leftarrow$ Obtener rasgo objetivo del elemento de $\max(\beta(O_0, O_T))$ de los elementos de C_S .

-- Rasgos predictores de $O_A \leftarrow$ Rasgos predictores de O_0 .

-- Rasgo objetivo de $O_A \leftarrow U'$

2.5. Módulo de autoaprendizaje.

No todos los casos obtenidos deben ser incorporados a la BC. Los casos obtenidos como resultado final luego de la aplicación de los algoritmos antes descritos son almacenados temporalmente en una lista de casos pendientes de análisis por expertos, los cuales tienen la responsabilidad de decidir si tiene sentido o no incorporarlos a la BC.

2.6. Consideraciones Parciales.

En el presente capítulo se expresa la utilización del paradigma de Razonamiento Basado en Casos como apoyo en la toma de decisiones y la integración de éste con la presentación del conocimiento en forma adaptativa a través de Mapas Conceptuales Inteligentes. De esta manera quedan descritas las características de los elementos que lo componen, dígame Casos, Base de Casos, Módulo de Acceso-Recuperación, Módulo de Adaptación y el no menos importante Módulo de Autoaprendizaje.

Capítulo 3. Tendencias, Herramientas y Tecnologías

Las tecnologías influyen en el progreso social y económico del hombre. Se presentan cada vez más como una necesidad en el contexto de la sociedad donde los rápidos cambios, el aumento de los conocimientos y las demandas de la información constantemente actualizada se convierten en una exigencia permanente. La principal finalidad de las tecnologías es transformar el entorno, para adaptarlo mejor a las necesidades y deseos humanos. En este capítulo se abordan las tecnologías en el desarrollo del software haciéndose énfasis en las funcionalidades y ventajas que aportan desde el punto de vista de su propósito u objetivo de creación. Este análisis permite la selección de las herramientas para desarrollar la propuesta de solución en cuestión.

3.1. Selección de las herramientas de desarrollo. Justificación.

Existen dos tipos de aplicaciones informáticas: las aplicaciones de Escritorio y las aplicaciones Web.

La aplicación Desktop o de Escritorio no es más que una solución completa de interfaz gráfica de usuario que ofrece íconos, barras de herramientas, programas e integración entre aplicaciones con habilidades para ofrecer al usuario una interacción amigable. Entre sus ventajas se pueden encontrar que el usuario está más acostumbrado a su uso y manejo y que como aplicación pesada debe correr en el lado del cliente (Nadereau 2009).

No obstante se selecciona como herramienta de desarrollo una Aplicación Web pues son programas que corren en servidores Web y utilizan páginas Web como interfaz de usuario. Para la interacción con el usuario promedio, este tipo de software será sencillo, económico, móvil, confiable.

Para el desarrollo de la Herramienta en cuestión ofrece las ventajas:

- Otorga la flexibilidad de determinar niveles de acceso según la confidencialidad de los datos así como la posibilidad de realizar determinados procesos según privilegios de accesos, pues en el caso particular de la solución propuesta es necesario un sistema cliente-servidor para que simule el trabajo administrador (profesor), usuario simple (estudiante).
- Solo existe una versión en el servidor lo que implica que no hay que distribuirla entre las demás computadoras.

Capítulo 3: Tendencias, Herramientas y Tecnologías

- No requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones, solo un computador con un buen navegador Web.

3.2. Selección del lenguaje de programación a utilizar en la implementación de la propuesta de solución.

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se ponen a disposición del programador para que éste pueda comunicarse con los dispositivos hardware y software existentes (Anónimo 2009).

Inicialmente fue LISP un lenguaje de alto nivel, quien se convirtiera en la lengua dominante de programación de IA. Luego surgió el Prolog que es uno de los lenguajes que más se destaca en este ámbito, pues trabaja fuertemente orientado al desarrollo de la IA. Por otro lado Java es un lenguaje que actualmente está presente en muchas aplicaciones, algunas incluso relacionadas con la inteligencia artificial. No obstante los algoritmos utilizados en sistemas que se basan en inteligencia artificial pueden ser programados en cualquier lenguaje de programación.

Considerando que la solución a la problemática actual se basa en la utilización de RBC y por tanto de su unidad básica: los casos, a continuación se realiza una comparación entre los posibles lenguajes a utilizar para la implementación, como muestra la Tabla 2, considerando solo los aspectos que son significativos para la futura construcción de la solución propuesta.

Tabla 2. Comparación entre lenguajes LISP, Prolog y Java.

Criterios	LISP	Prolog	Java
Tipos de datos disponibles	Solo las listas	Listas y tipos estructurados	A parte de los tipos básicos cuenta también con un conjunto completo de estructuras de datos complejas
Forma de ejecución	Recursividad	Recursividad	Soporta la recursividad e iteración

Capítulo 3: Tendencias, Herramientas y Tecnologías

Claridad y comprensión del código	Código difícil de comprender	Dificultad para comprender los programas	Código de marcada simpleza
Dificultad en la depuración	Presenta dificultad para depurarlo	Su depuración es complicada	Verifica su código al mismo tiempo que lo escribe, y una vez más antes de ejecutarse

Del análisis comparativo antes hecho se concluye que es Java el lenguaje de programación más adecuado para implementar la aplicación Web propuesta. Además para el desarrollo de la Herramienta en cuestión java ofrece otras ventajas como:

- Programas transportables: el código, cuando está bien generado, deriva en archivos capaces de correr en cualquier tipo de sistema operativo.
- Orientado a objetos: lo que permite una mayor flexibilidad de implementación.

A continuación se listan algunos ejemplos de trabajos con sistemas inteligentes que han sido desarrollados utilizando tecnología Java.

- Agente Reflejo Simple y Agente de Búsqueda (Flores 2009).
- Agente Reflejo Simple e Informado (Melgarejo 2009).
- Weka (Witten 2005).
- Sistema para Detección de Problemas en el Aprendizaje (Fiel 2009).

3.3. Selección del Entorno de desarrollo integrado (IDE).

Al escoger un IDE se pueden valorar varios aspectos. Para unos el mejor IDE es el que pueda ser expansible mediante miles de plugins y que tenga soporte para cualquier lenguaje y tecnología, otros pensarán que el mejor IDE es aquel que tenga mejor interfaz gráfica y otros que es aquel que tenga mejor documentación. Pero el mejor IDE es aquel que ayuda a resolver tareas de la forma más fácil, es aquel con el que el desarrollador se sienta más cómodo y seguro para trabajar, y teniendo eso en cuenta el mejor IDE no es uno solo, solo que todos tienen sus ventajas y desventajas.

Capítulo 3: Tendencias, Herramientas y Tecnologías

En la presente investigación se propone utilizar Java como lenguaje de programación; por tanto son de interés solo los IDE dirigidos al desarrollo con Java. Netbeans y Eclipse son actualmente los IDE para Java más difundidos, ambos son libres y más similares que diferentes.

La Tabla 3 muestra una comparación entre ambos IDE, considerando solo los aspectos más significativos para la implementación de la solución propuesta. Para ello se seleccionan 4 indicadores y una escala valorativa definiéndose los siguientes valores: 1- muy bajo, 2- bajo, 3- medio, 4- alto y 5- muy alto. A partir de elementos expuestos en la bibliografía consultada, artículos electrónicos (Eclipse.org 2010) (Netbeans.org 2010) (Molpeceres 2002), otros trabajos de tesis (Pérez 2009) y tomando también como base opiniones de desarrolladores de todo el mundo plasmadas en foros, se asignaron valores a dichos indicadores.

Tabla 3. Comparación entre IDE Netbeans y Eclipse.

Indicadores	Netbeans	Eclipse
Portabilidad	5	5
Flexibilidad	3	5
Experiencia con aplicaciones Web	3	5
Análisis de código	4	5

Atendiendo al análisis realizado se selecciona Eclipse como la plataforma a utilizar para implementar la solución propuesta. Eclipse es además el más utilizado por la comunidad de desarrolladores Web a nivel mundial.

3.4. Selección de la Metodologías de desarrollo de software (MDS).

El software es el intermediario cada vez más grande entre la información y la inteligencia humana. El poder acceder a la información, conocer quién controla a este intermediario y las garantías de su transparencia y confiabilidad constituyen tema actuales de preocupación. Uno de los métodos efectivos para obtener estos resultados es sin dudas el uso de las metodologías de desarrollo. Las metodologías se basan en la planificación, inspiradas por otras disciplinas de la ingeniería, para lograr que el software sea predecible y eficiente. No existe una metodología universal para hacer frente con éxito a cualquier

Capítulo 3: Tendencias, Herramientas y Tecnologías

proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto: recursos, técnicos y humanos, tiempo de desarrollo, tipo de sistema, entre otros.

3.4.1. Metodologías Robustas o Pesadas.

Las metodologías robustas se centran especialmente en el control del proceso. Las mismas establecen rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán durante el proceso de desarrollo.

Rational Unified Process (RUP).

La metodología RUP, llamada así por sus siglas en inglés, divide en 4 fases el desarrollo del software:

- Inicio, el objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración, determinar la arquitectura óptima.
- Construcción, obtener la capacidad operacional inicial.
- Transición, obtener el reléase del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, el cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. El ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

- **Disciplina de Desarrollo**
 - Ingeniería de Negocios: Entendiendo las necesidades del negocio.
 - Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
 - Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
 - Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
 - Pruebas: Asegurándose que el comportamiento requerido es el correcto.
- **Disciplina de Soporte**
 - Configuración y administración del cambio: Guardando todas las versiones del proyecto.

Capítulo 3: Tendencias, Herramientas y Tecnologías

- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.
- Distribución: Hacer todo lo necesario para la salida del proyecto.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software (Mendoza 2004).

3.4.2. Metodologías ágiles.

Las metodologías ágiles ofrecen una solución para gran cantidad de proyectos pequeños. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo.

Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo y equipos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. La metodología XP se basa en:

- Pruebas Unitarias: las pruebas realizadas a los principales procesos, teniendo en cuenta el funcionamiento, para en el futuro desarrollar pruebas de fallas que pudieran ocurrir, o sea obtener los posibles errores.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa. Incrementa la productividad y la calidad del software desarrollado.
- Define cuatro variables para proyectos de software: coste, tiempo, calidad y ámbito.

RUP Ultra Light

Capítulo 3: Tendencias, Herramientas y Tecnologías

Versión simplificada de RUP que describe fácilmente el aseguramiento del negocio de aplicaciones de software usando técnicas y conceptos ágiles. Divide el proceso de desarrollo en 10 pasos fundamentales inmersos en cuatro fases de desarrollo al igual que RUP: Inicio, Elaboración, Construcción y Transición.

Para desarrollar la MDS RUP Ultra Light se deben seguir los siguientes pasos:

1. Realizar un Diagrama de Casos de Uso: Una vez identificadas todas las funcionalidades del software a construir, se debe representar todos estos casos de uso en un diagrama teniendo en cuenta además, a los actores involucrados y las relaciones existentes entre ellos (actor-caso de uso, actor- actor, caso de uso- caso de uso).
2. Priorizar los Casos de Uso a trabajar: Luego de identificar los casos de uso del sistema, se debe hacer una lista de prioridad de casos de uso, donde la prioridad es el riesgo que conllevan. El riesgo es todo lo que puede afectar el buen desarrollo del sistema como puede ser: necesidad de cambiar la arquitectura, no escoger los requisitos adecuados, no construir un sistema correcto, entre otros. Se debe determinar cuáles son los requerimientos más importantes a desarrollar en las primeras iteraciones y cuáles deben dejarse para más tarde.
3. Generar los Documentos de Caso de Uso: Para un mejor entendimiento de la funcionalidad asociada a cada caso de uso se necesita un documento que describa de forma breve o extendida lo que hace el caso de uso, más allá de la representación gráfica de su Diagrama de Casos de Uso.

El documento de Caso de Uso debe ser generado por el Analista del proyecto y debe tener de una forma u otra la siguiente estructura:

- Descripción Breve. De lo que hace el Caso de Uso.
- Precondiciones. Condiciones que deben ser cumplidas antes de ejecutar el Caso de Uso.
- Flujo Básico. Descripción paso a paso de las acciones a realizar por el Usuario cuando trabaja de forma correcta en este Caso de Uso.
- Flujo Alternativo. Detalle de los pasos que seguirá el usuario cuando no trabaje de forma correcta en este requerimiento, ejemplo: validaciones, entre otros.

Capítulo 3: Tendencias, Herramientas y Tecnologías

- Postcondiciones. Las acciones que se deben realizar después de que se ha terminado de ejecutar el Caso de Uso.
 - Interfaz Gráfica. Prototipo de cómo debe quedar la pantalla del caso de uso para ser vista por los usuarios”.
4. Generar los Diagramas de Secuencia: Un diagrama de secuencia permite conocer la forma en la que los objetos se comunicarán en una pantalla para cumplir su objetivo a través de la ordenación temporal de mensajes durante su ciclo de vida. No es indispensable hacer este gráfico pero ayudará a que el Arquitecto de Software comprenda mejor lo que debe hacer, y los Implementadores podrán hacer mejor su trabajo.
 5. Diseñar el Framework del proyecto: El Arquitecto de Software del proyecto diseñará las clases que serán usadas en todo el Software. Es un trabajo bastante delicado ya que el mal diseño de las clases involucra que no sean implementadas las funcionalidades de cada clase de forma correcta, lo cual conllevará a escribir un “Spaghetti Code”, que significa que el código estará difuso. Esta etapa de diseño debe ser revisada cuidadosamente y se recomienda la utilización de Patrones de Diseño de Software de ser posible.
 6. Creación de la Base de Datos: El diagrama de clases del diseño desarrollado constituye el punto de partida para el diseño de la Base de Datos del proyecto pues se puede utilizar la Capa de Datos del mismo, donde se alojan las clases Entidad.
 7. Construir la máscara la WebSite o Aplicación Windows: Simultáneamente al desarrollo de los pasos 4, 5 y 6 se pueden ir realizando las plantillas para la creación de las páginas Web ayudándose de los gráficos de las GUI (Graphic User Interface o Interfaz Gráfica de Usuario) que se encuentran en los Documentos de Casos de Uso.
 8. Programar las funcionalidades de los Casos de Uso: Una vez terminadas las clases, se comienzan a programar las funcionalidades de los Casos de Uso. Para ello, los programadores se apoyan en los documentos de casos de uso desarrollados por los analistas y basándose en el Diagrama de Secuencia y en las clases diseñadas por el Arquitecto escriben el código que se necesita para que el caso de uso funcione.

Capítulo 3: Tendencias, Herramientas y Tecnologías

9. Probar los Requerimientos del Software: Independientemente que en el Documento de Casos de Uso se describa detalladamente cómo debe funcionar el requerimiento y que se hayan realizado los diagramas, siempre se escapan algunos detalles que se deben corregir en una etapa de pruebas exhaustivas, que no deben ser hechas por las mismas personas que programaron los casos de uso.
10. Integrar los requerimientos concluidos: Al finalizar es indispensable unir lo que se ha realizado por diferentes programadores de forma tal que el sistema funcione como un todo y ponerlo a disposición de los usuarios.

Se deben repetir de los pasos 3 al 10 por cada iteración que se haya programado para el proyecto, para de esta forma poder controlarlo (Guerrero 2007).

Se selecciona como Metodología de Desarrollo a utilizar RUP Ultra Light, pues siendo una versión simplificada de RUP ofrece las facilidades y ventajas de esta potente metodología, de forma ágil, o sea se adapta al contexto de la aplicación a realizar (software pequeño, y poco complejo en un período de desarrollo corto). Además las bases de la metodología se adaptan a las condiciones de trabajo existentes para el apoyo al desarrollo del software y agiliza la interpretación de los procesos en versiones simplificadas. En este caso particular el equipo de desarrollo se compone solo de una persona, que debe desempeñar todos los roles de la Ingeniería de Software necesarios para el desarrollo de la solución y además se dispone de un período corto de tiempo. Razones por las cuales se necesita reducir el proceso de desarrollo de la herramienta en cuestión con la utilización de la antes mencionada metodología.

3.5. Lenguaje Unificado de Modelado (UML): soporte a la modelación de la propuesta de solución.

Lenguaje Unificado de Modelado (UML) es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. En otras palabras, es el lenguaje en el que está descrito el modelo (Booch 2006).

Capítulo 3: Tendencias, Herramientas y Tecnologías

Siendo un lenguaje de modelado permite describir métodos o procesos, definir el sistema, detallar los artefactos, permitiendo al cliente, a través de diagramas, entender con mayor exactitud el proceso de desarrollo de software de la solución propuesta.

3.6. Rational Rose Enterprise Edition. Herramienta CASE para la propuesta de solución.

Rational Rose es una herramienta de fácil utilización que permite la modelación absoluta de los procesos del negocio y del sistema, o sea permite mostrar de forma gráfica el análisis de los procesos del negocio y los requerimientos del sistema. Además tiene capacidad para proporcionar el desarrollo iterativo y de ingeniería (G.S.I. 2007).

Para el modelado visual se utiliza Rational Rose siendo una herramienta de gran fortaleza en el diseño, despliegue, construcción, pruebas y administración de proyectos en el proceso desarrollo de software. Esta herramienta de diseño de software es el producto más completo de la familia Rational Rose y como fue desarrollada por los creadores de UML es perfecta para el modelado utilizando este lenguaje.

3.7. Selección de un soporte para el almacenamiento y representación de información persistente.

El conocimiento es importante y primordial en todas sus aplicaciones y en cualquiera de sus facetas, y esto funciona de igual manera para el comportamiento inteligente. Su almacenamiento y representación constituye una de las máximas prioridades de la investigación en IA. Los Sistemas Gestores de Bases de Datos (SGBD), las Tablas Hash y los sistemas de Archivos XML son variantes para el almacenamiento y representación de información persistente.

Seguidamente se realiza un análisis de estas 3 variantes, con la finalidad de determinar cuál sería la más adecuada para ser utilizada como almacén de los datos que maneja el sistema propuesto. Para ello se escogen y miden una serie de indicadores los cuales son evaluados usando una escala de 1 a 5, donde cada uno representa los siguientes valores: 1- muy bajo, 2- bajo, 3- medio, 4- alto y 5- muy alto. Es importante destacar que algunos indicadores tienen más peso (están resaltados) que otros; pues son de mayor interés y los de mayor influencia en la decisión final. A continuación se muestra la Tabla 4 en la cual se expresa el resultado del análisis.

Capítulo 3: Tendencias, Herramientas y Tecnologías

Tabla 4. Comparación entre SGBD, Tablas Hash y Archivos XML.

Indicadores	SGBD	Tablas Hash	Archivos XML
Manejo de datos complejos	3	5	5
Accesibilidad y reutilización de la información	5	3	5
Complejidad	5	5	2
Requerimiento de espacio en disco	5	2	2
Vulnerabilidad a los fallos	5	5	2
Flexibilidad	3	3	5

Finalmente como resultado del análisis se decide que se debe hacer uso de Archivos XML. Esta selección se hace teniendo en cuenta principalmente los indicadores 2 y 6 de la tabla anterior que son los aspectos de mayor peso, pues en este caso se necesita lograr la mayor accesibilidad y reutilización posible de la información y precisamente con XML por ser extensible y flexible se logra esto. XML es un Lenguaje de Etiquetado Extensible muy simple, que juega un papel fundamental en el intercambio de una gran variedad de datos. Es muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato ideal para estructurar, almacenar e intercambiar información.

3.8. Selección de un API para procesar XML.

Al momento de desarrollar una aplicación que necesite leer archivos XML, no es conveniente crear piezas de código que hagan éste tipo de tareas desde cero. Un parser XML es una librería de clases que ofrece al programador métodos para manipular documentos XML, además de abstraer el funcionamiento complejo o no particular de XML. Los parsers XML generalmente pueden usar una o más APIs XML para llevar a cabo su misión. A continuación se describen y a su vez se comparan algunas APIs que permiten el trabajo con documentos XML (Castiblanco 2008) (Lou 2002):

DOM

Capítulo 3: Tendencias, Herramientas y Tecnologías

Document Object Model, es un API que modela un documento XML como un *árbol*. DOM analiza el documento completo en una estructura de árbol y luego permite leer y manipular los datos. El análisis de la estructura a menudo requiere la carga de la estructura jerárquica del documento completo antes de hacer cualquier trabajo. DOM puede leer y generar documentos XML. Sin embargo cargando y analizando gramaticalmente documentos grandes puede llegar a ser muy lento.

SAX

Simple API for XML. Es muy poco lo que no se puede hacer con dicha API. Las clases e interfaces de SAX modelan el parser, el flujo desde el cual los documentos son leídos, y la aplicación cliente recibe datos desde el parser. Sin embargo, ninguna clase modela el documento XML; en vez de ello, el parser provee el contenido a la aplicación cliente a través de una interfaz callback. Los requisitos de memoria son usualmente inferiores, porque deja a los desarrolladores decidir cuál etiqueta tratar. Como SAX no tiene que guardar el documento entero en la memoria hace que sea muy rápido, eficiente e ideal para manipular documentos XML demasiado grandes. Pero con el parser SAX la codificación puede ser difícil y cuesta mucho acceder al mismo documento en múltiples y diferentes datos.

JAXP

Java API for XML Processing, es el API estándar de Java que encapsula interfaces tanto de SAX como de DOM. El API trata sobre el patrón de diseño factoría para crear un nuevo analizador SAX, un constructor de documentos DOM o motores de hojas de estilo (SAX, DOM y XSLT respectivamente). Pero JAXP no es un analizador por sí mismo, sino que proporciona una capa de abstracción para un programa de análisis subyacente.

JDOM

Java for XML Document Object Model, es una representación Java de un documento XML lo cual simplifica la interacción con XML, y hace el trabajo más rápido que al utilizar a DOM. JDOM proporciona una forma de representar los documentos XML para una lectura, manipulación y escrituras fáciles y eficientes. JDOM usa clases concretas en vez de simplemente utilizar una interfaz, además hace uso extensivo de colecciones de clases ya conocidas por los desarrolladores java simplificando su uso. JDOM es específico de la plataforma Java y una alternativa para DOM y SAX, aunque se integra bien con

Capítulo 3: Tendencias, Herramientas y Tecnologías

ambos. JDOM no contiene un analizador gramatical por lo que usualmente usa el analizador gramatical del SAX 2.0 para analizar gramaticalmente y validar documentos XML pero JDOM puede utilizar casi cualquier programa de análisis. A diferencia de JAXP, JDOM no es una envoltura alrededor de W3C DOM, sino que éste se vale por sí mismo.

En conclusión y tomando en consideración los aspectos descritos se selecciona JDOM como API para la futura manipulación de los documentos que conforman el sistema de archivos XML que soporta la información persistente. Esta selección se hace teniendo en cuenta principalmente la idea de que dicha API facilita el trabajo con documentos XML de gran tamaño.

3.9. Consideraciones Parciales.

Es necesario establecer y seleccionar las herramientas y tecnologías que se deben utilizar al desarrollar la herramienta propuesta, pues constituyen el soporte de desarrollo para la misma. Esta selección se realiza de forma tal que se garantice una productividad óptima en el proceso de desarrollo, principalmente en la implementación, siendo una de las etapas más complejas.

Se selecciona la Aplicación Web como herramienta de desarrollo. RUP Ultra Light la metodología de desarrollo a utilizar, pues es una de las versiones de RUP más objetiva. Para el soporte de la modelación se decide utilizar el Lenguaje Unificado de Modelado (UML) y la herramienta Rational Rose para el modelado visual.

Se escoge además Java como lenguaje de programación sobre el IDE Eclipse que es la plataforma de desarrollo más conveniente para implementar, debido a su vasta experiencia en el desarrollo de aplicaciones Web. Como medio de almacenamiento para la información persistente se utiliza un sistema de ficheros XML, pues XML es flexible y extensible dando la libertad de crear nuevas etiquetas lo cual posibilita que la información sea más accesible y reutilizable. Además para la manipulación de dichos archivos XML se selecciona el API JDOM.

Capítulo 4: Presentación de la solución propuesta.

Capítulo 4. Presentación de la solución propuesta

Este capítulo resume las principales características que presenta el sistema informático que se propone como solución al problema de investigación. En él se explican los aspectos críticos para el entendimiento del entorno del sistema. Tomando en consideración la metodología de desarrollo escogida, se realiza la modelación de la aplicación, comenzando con la captura de requisitos, definiendo requisitos funcionales y no funcionales así como las descripciones textuales de cada Caso de Uso.

4.1. Modelo de Dominio.

Un Modelo del Dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema (UCI 2009).

Un Modelo del Dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos (Jacobson 2000).

El modelo de dominio presenta un bajo nivel de estructuración y se centra en las tecnologías informáticas, además como el sistema propuesto no va dirigido a un cliente o entidad en específico, se hace difícil la identificación de procesos de negocio bien estructurados, por estas razones se decide realizar este modelo. Para ello se realiza la descripción mediante un diagrama de clases UML, que permite especificar las clases conceptuales de mayor importancia que intervienen en el sistema y que a su vez representan los objetos o eventos que suceden en el entorno en que se trabaja.

4.2. Definición de los conceptos y las entidades principales.

Estudiante: Usuario con determinados permisos dentro del sistema, para visualizar Mapas Conceptuales y para la realización de los respectivos cuestionarios previos a la navegación en Mapas Conceptuales Inteligentes.

Profesor: Usuario con determinados permisos dentro del sistema, para administrar cuestionarios dirigidos a los estudiantes, administrar los casos que conforman la Base de Conocimiento y administrar Sistemas Tutoriales Inteligentes.

Capítulo 4: Presentación de la solución propuesta.

Mapa Conceptual: Herramienta para representación y organización de conocimiento, generalmente compuesta de nodos y las relaciones entre éstos.

Mapa Conceptual Inteligente: Herramienta gráfica para representación y organización de conocimiento, generalmente compuesta de nodos y las relaciones entre estos. Posee la característica particular de que la navegación sobre él es individualizada.

Pregunta: Son las interrogaciones que el estudiante responde previo a visualización de su correspondiente Mapa Conceptual Inteligente, como parte de los cuestionarios, de los cuales son la unidad básica que los compone. Las preguntas son formuladas e insertadas en los cuestionarios por el profesor.

Cuestionario: Es un tipo de evaluación que incluye un conjunto de preguntas diseñadas por el profesor con vistas a que sean respondidas por los estudiantes.

Respuesta Estudiante: Guarda la respuesta a las preguntas de los cuestionarios, son las respuestas insertadas por los estudiantes.

Respuesta Base: Guarda la respuesta a las preguntas de los cuestionarios, son las respuestas formuladas e insertadas por el profesor.

Objeto Aprendizaje: Representa a todo material didáctico, ya sea un documento, presentación, video, imagen, entre otros; o sea se define como cualquier recurso digital que se desee incluir para describir y ampliar el concepto del nodo del Mapa Conceptual en el cual está contenido.

Base de Conocimiento: Representa el conjunto de casos resueltos, o sea el conjunto de pares Rasgos Predictores - Rasgo Objetivo.

Caso: Es el par formado por un Rasgo Objetivo y varios Rasgos Predictores. En ocasiones el rasgo objetivo tiene valor nulo lo que indica que estamos en presencia de un *caso no resuelto* o *caso nuevo*.

Rasgo Predictor: Representa las características de un estudiante. Estas características se expresan a través del puntaje obtenido en cada pregunta de un cuestionario que dicho estudiante haya realizado.

Rasgo Objetivo: Representa el tratamiento que se le da a un conjunto de Rasgos Predictores. Está conformado por el conjunto de nodos de un Mapa Conceptual y los correspondientes objetos de aprendizaje que hayan sido seleccionados como solución a un *caso no resuelto*.

Capítulo 4: Presentación de la solución propuesta.

4.3. Diagrama del Modelo de Dominio.

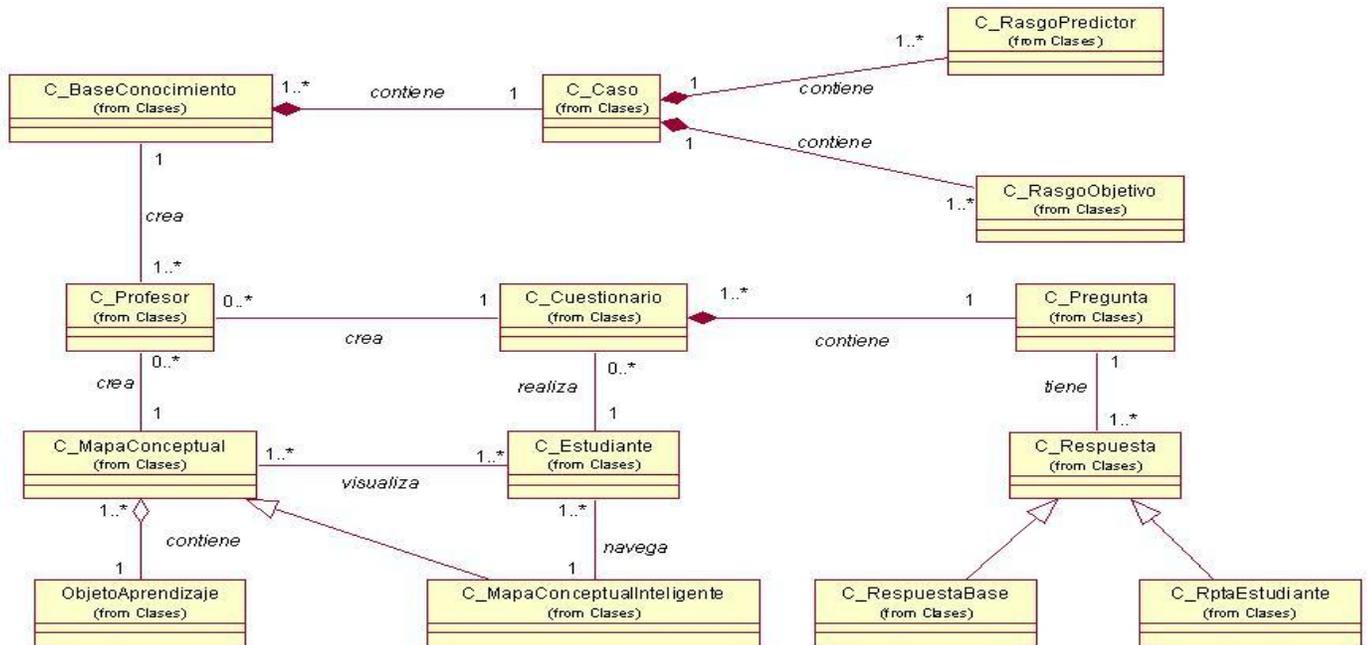


Figura 3. Diagrama del Modelo de Dominio.

4.4. Requerimientos Funcionales (RF).

Una vez que se comienza a realizar la modelación del negocio el siguiente paso es definir qué es lo que debe hacer el sistema, por lo que se hace necesario capturar los requisitos que éste debe cumplir. El levantamiento de requisitos es una de las actividades más importantes en la fase de Inicio de la modelación de un software y mediante él se establecen las funcionalidades y restricciones que el sistema debe cumplir. A continuación se enumeran los RF:

- R1 Autenticar al Usuario.
- R2 Gestionar los datos de los Usuarios.
 - R2.1 Insertar un usuario nuevo.
 - R2.2 Eliminar un Usuario.
 - R2.3 Modificar los datos de un Usuario.
- R3 Modificar la contraseña de un usuario.
- R4 Gestionar los cuestionarios.
 - R4.1 Crear un cuestionario.

Capítulo 4: Presentación de la solución propuesta.

- R4.2 Eliminar un cuestionario.
- R5 Evaluar los cuestionarios.
- R6 Importar Mapas Conceptuales.
- R7 Generar Mapas Conceptuales Inteligentes.
- R8 Guardar Mapas Conceptuales Inteligentes.
- R9 Gestionar los casos de la Base de Conocimiento.
- R9.1 Insertar un nuevo Caso en la Base de Conocimiento.
- R9.2 Eliminar un Caso de la Base de Conocimiento.
- R9.3 Modificar los datos de un Caso.
- R10 Gestionar datos de los Sistemas Tutoriales Inteligentes.
- R10.1 Insertar un nuevo Sistema Tutorial Inteligente
- R10.2 Eliminar un Sistema Tutorial Inteligente

4.5. Requerimientos No Funcionales (RNF).

Los RNF son propiedades o cualidades que el producto debe tener. Estas propiedades deben ser asimiladas como las características que hacen que un producto sea rápido, usable o confiable. Constituyen elementos importantes para que el producto final satisfaga las expectativas de los clientes (UCI 2009). A continuación se enumeran los RNF:

- RNF 1 Apariencia

El diseño de la Aplicación debe ser sencillo, amigable al usuario, con una navegabilidad intuitiva.

- RNF 2 Usabilidad:

La aplicación debe proveer interfaces que puedan ser usadas por cualquier usuario con conocimientos básicos en el manejo de la computadora. La aplicación debe estar disponible las 24 horas del día.

- RNF 3 Confiabilidad:

El sistema debe realizar tratamiento de excepciones, validaciones de las entradas hechas por los usuarios. El sistema debe ser preciso en la información que le suministra al usuario para evitar cualquier tipo de error.

- RNF 4 Seguridad

Identificar al usuario antes de que pueda realizar cualquier acción en el sistema y proteger la información de accesos no autorizados. El sistema debe permitir solo el acceso a cada uno de sus servicios al

Capítulo 4: Presentación de la solución propuesta.

personal requerido para esto. El sistema debe comunicarse usando un protocolo seguro (https) y los datos como la contraseña no pueden viajar de forma transparente por la red por tanto se debe usar el método asimétrico para la encriptación.

- RNF 5 Software

Herramienta CmapTools instalado. Máquina virtual Java. Se debe utilizar un servidor con plataforma de funcionamiento GNU/Linux.

- RNF 6 Portabilidad

El sistema debe funcionar sobre las plataforma GNU/Linux y Windows.

- RNF 7 Funcionalidad

Tiempo de entrenamiento. El sistema debe ser sometido a una fase de pruebas en las cuales los usuarios se familiaricen con éste y a la vez se puedan detectar posibles errores.

- RNF 8 Ejecución.

El sistema debe tener un tiempo de respuesta de a lo sumo 3 minutos ante cualquier solicitud del usuario.

- RNF 9 Hardware.

Se necesitan computadoras conectadas a la red que cuenten con:

PC servidora CPU con procesador Pentium 4 de 2 GHz o superior, con 50 Gbyte de disco duro y 512 Mbyte de RAM o superior.

PC cliente CPU con procesador Pentium 4, con 5 Gbyte de disco duro disponible y 256 Mbyte de RAM o superior.

4.6. Descripción de los actores del sistema.

Tabla 5. Actores del Sistema.

Actores	Descripción
Usuario	Accede al sistema y según su rol tiene privilegios de editar base de conocimientos, sistemas tutoriales inteligentes y cuestionarios o visualizar MC y MCI. Los roles pueden ser: Administrador, Profesor y Estudiante
Administrador	Tiene los mismos privilegios del usuario y además gestiona la información referente a los datos de usuarios.

Capítulo 4: Presentación de la solución propuesta.

Profesor	Representa a la persona que accede al sistema bajo el permiso de profesor. Por lo tanto tiene privilegios para editar la base de conocimientos, sistemas tutoriales inteligentes y los cuestionarios.
Estudiante	Representa a la persona que accede al sistema bajo el permiso de estudiante. Solo tiene privilegios de visualizar MC y MCI.

4.7. Casos de Uso del Sistema (CUS).

Según Pressman un Caso de Uso (CU) modela el sistema desde el punto de vista del usuario y representa una secuencia de actividades que incluyen a los actores y al sistema. Los CUS son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores (UCI 2009). A continuación se representa el Modelo de CUS y se describe cada uno de los CU presentes en éste.

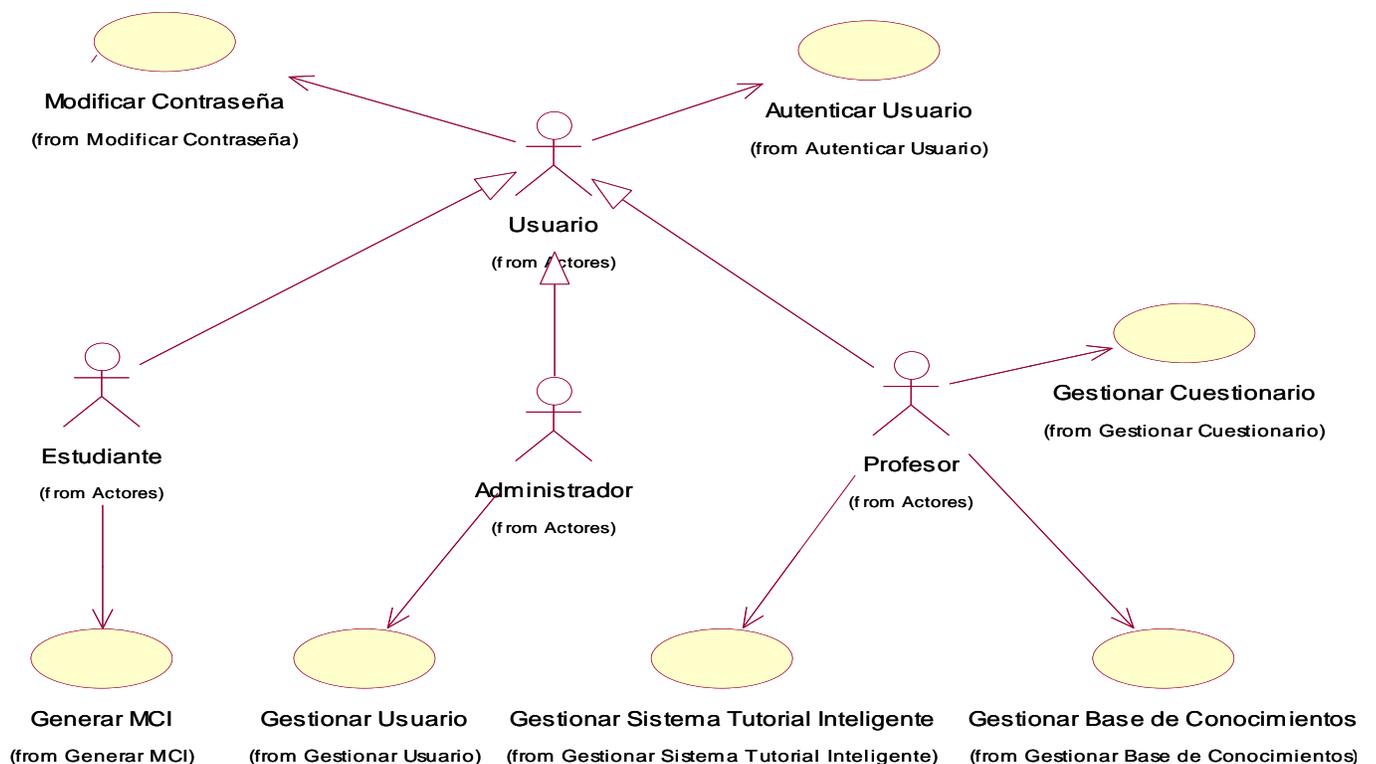


Figura 4. Modelo de CU del Sistema

Capítulo 4: Presentación de la solución propuesta.

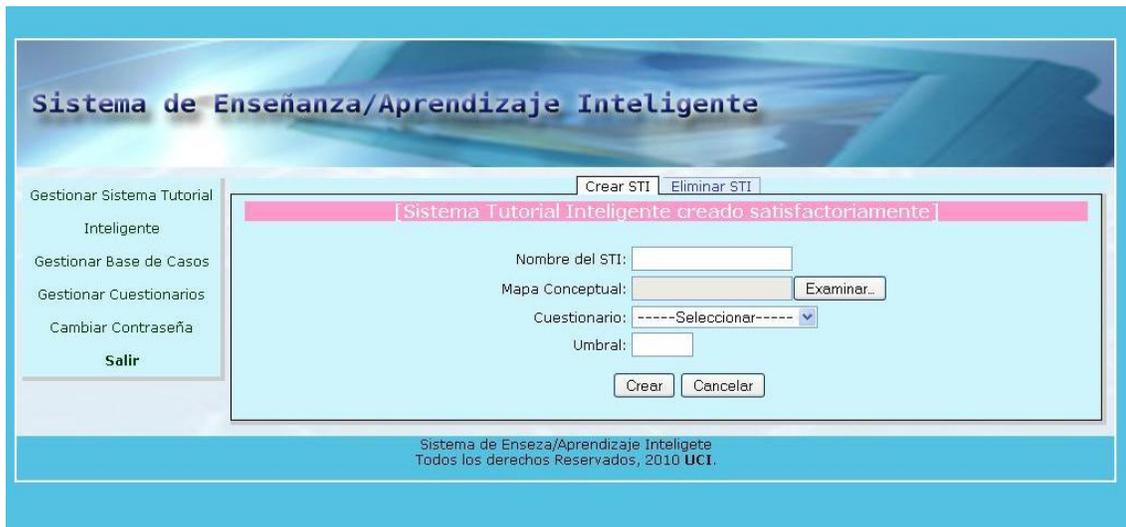
4.8. Descripción de los CUS.

Seguidamente se muestran las descripciones de algunos CU, las restantes descripciones se encuentran en el Anexo A.

Tabla 6. Descripción del CU Gestionar Sistema Tutorial Inteligente.

Caso de Uso	Gestionar Sistema Tutorial Inteligente	
Actores:	Profesor	
Resumen:	El caso de uso se inicia cuando el Profesor selecciona la opción Gestionar STI para llevar a cabo la acción deseada (Crear un nuevo STI o eliminar uno ya existente).	
Precondiciones:	El Usuario debe estar autenticado como Profesor	
Referencias	R10, R10.1 y R10.2	
CU Asociados		
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción de Actor	Respuesta del Sistema	
1. El Profesor selecciona la opción "Gestionar Sistema Tutorial Inteligente"	2. El sistema muestra una vista con la opciones: <ul style="list-style-type: none"> • Crear STI • Eliminar STI 	
3. El Profesor selecciona la opción deseada. En caso de: <ul style="list-style-type: none"> a) Crear STI. Ir a la Sección Insertar STI b) Eliminar STI. Ir a la Sección Eliminar STI 		
Sección "Insertar STI"		
	1. El sistema muestra un formulario con los campos: <i>Nombre del STI, Mapa Conceptual, Cuestionario y Umbral.</i>	
2. El Profesor completa los datos requeridos y presiona el botón "Aceptar"	3. El sistema comprueba los datos ingresados. [si los datos son correctos] 3.1 El sistema guarda el STI y muestra el mensaje "Sistema Tutorial Inteligente creado satisfactoriamente"	
Prototipo de Interfaz		

Capítulo 4: Presentación de la solución propuesta.



Flujos Alternos

2.1. El Profesor completa los campos requeridos y presiona el botón "Cancelar"

2.2. El sistema resetea el formulario de la vista actual.

[si los datos son incorrectos]

3.2. El sistema muestra el mensaje "Complete correctamente los campos" y resalta los campos con valor incorrecto.

Sección "Eliminar STI"

1. El sistema muestra un listado con todos los STI que se encuentran previamente almacenados

2. El Profesor selecciona un STI y presiona el botón

3. El sistema muestra un mensaje: "Está seguro que

Capítulo 4: Presentación de la solución propuesta.

“Eliminar”	desea eliminar este STI”
4. El Profesor presiona el botón “Aceptar”	5. El sistema ejecuta la eliminación, muestra el mensaje: “Sistema Tutorial Inteligente eliminado satisfactoriamente”.

Prototipo de Interfaz



Flujos Alternos	
2.1. El Profesor selecciona un STI y presiona el botón “Cancelar”	2.2. El sistema resetea el formulario de la vista actual.
4.1. El Profesor presiona el botón “Cancelar”	4.2. El sistema muestra la vista anterior (El listado de los STI ya almacenados).

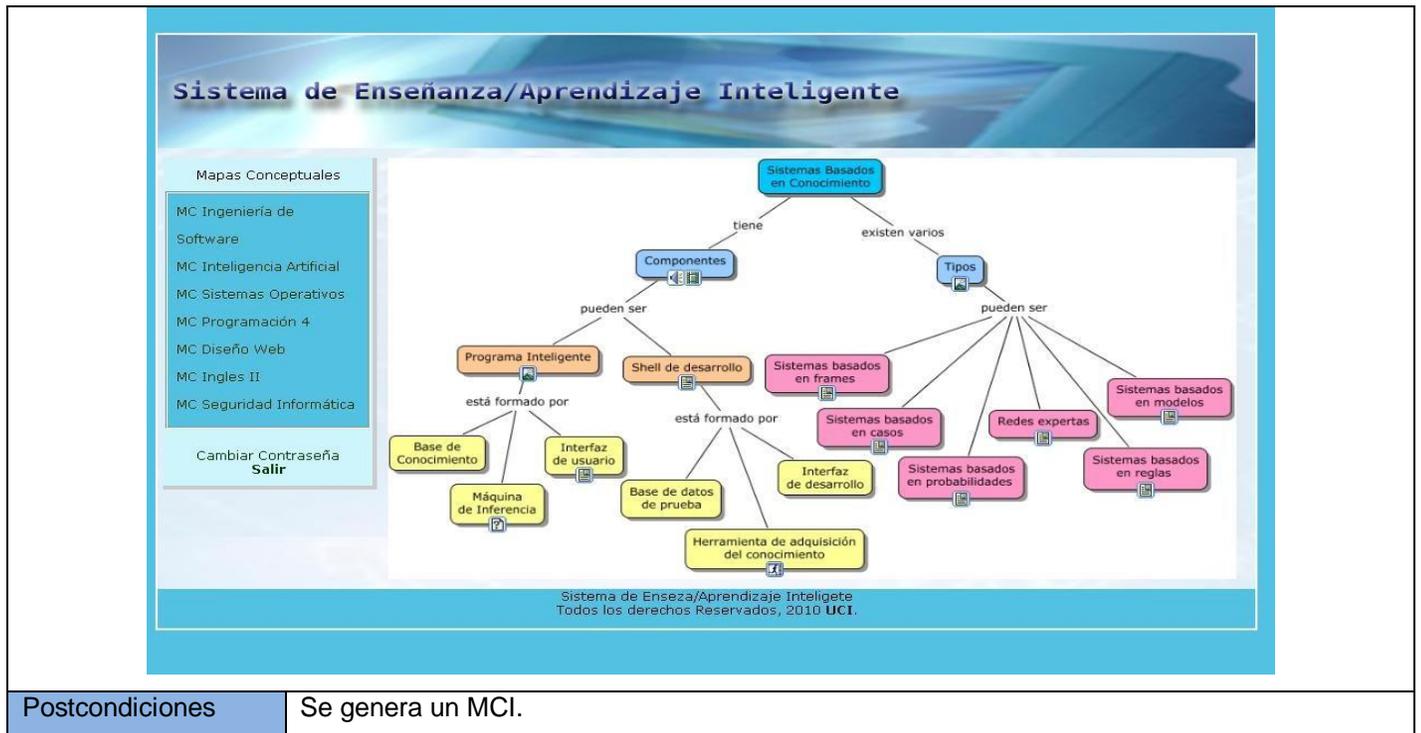
Capítulo 4: Presentación de la solución propuesta.

Postcondiciones	El Sistema logra insertar o eliminar un Sistema Tutorial Inteligente de la base.
-----------------	--

Tabla 7. Descripción del CU Generar MCI.

Caso de Uso	Generar Mapas Conceptuales Inteligentes	
Actores:	Estudiante	
Resumen:	El caso de uso se inicia cuando el Estudiante desea consultar la información de los nodos de un Mapa Conceptual.	
Precondiciones:	El Usuario debe estar autenticado como Estudiante	
Referencias	R7	
CU Asociados		
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción de Actor	Respuesta del Sistema	
1. El Estudiante selecciona un Mapa Conceptual de la lista de MC	2. El sistema muestra en una nueva vista el Mapa Conceptual seleccionado	
3. El Estudiante selecciona algún nodo del Mapa Conceptual	4. El sistema muestra en una nueva vista el cuestionario correspondiente al Mapa Conceptual activo	
5. El Estudiante responde el cuestionario y presiona el botón "Terminar"	6. El sistema evalúa el cuestionario 7. El sistema procesa el nuevo caso y guarda el resultado. 8. El sistema construye un MCI 9. El sistema muestra en la vista anterior el MCI.	
Prototipo de Interfaz		

Capítulo 4: Presentación de la solución propuesta.



4.9. Consideraciones Parciales

Tomando como guía la metodología de desarrollo escogida (RUP Ultra Light) en este capítulo se refleja la realización de la modelación de la aplicación, que constituye el paso inicial en esta metodología. Ésta incluye la captura de requisitos funcionales y no funcionales, la identificación de los actores del sistema, la descripción de los casos de usos que tienen lugar, la confección del modelo de diagramas de casos de uso del sistema, el cual es uno de los artefactos más importantes que se generan. Sumándose además el modelado de los prototipos no funcionales de interfaz de usuario, para alcanzar un mejor entendimiento de las funcionalidades del sistema propuesto.

Capítulo 5. Análisis y diseño del sistema

En este capítulo se modelan los artefactos correspondientes al flujo de Análisis y Diseño. Se presentan los modelos del análisis y del diseño, los cuales incluyen las realizaciones de los CU definidos y descritos en el capítulo anterior. Se muestran los diagramas de clases y colaboración del análisis y de esta manera se describen las clases interfaces, controladoras, entidades y las relaciones que existen entre ellas. Se muestran los diagramas de clases del diseño así como de secuencia. Se presenta además la descripción de la arquitectura que debe soportar la solución propuesta.

5.1. Diagrama de clases del análisis por cada CU.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo (UCI 2008).

Seguidamente se modelan los diagramas de clases del análisis de algunos CU, los restantes diagramas se encuentran en el Anexo B.

Caso de Uso “Gestionar Sistema Tutorial Inteligente”

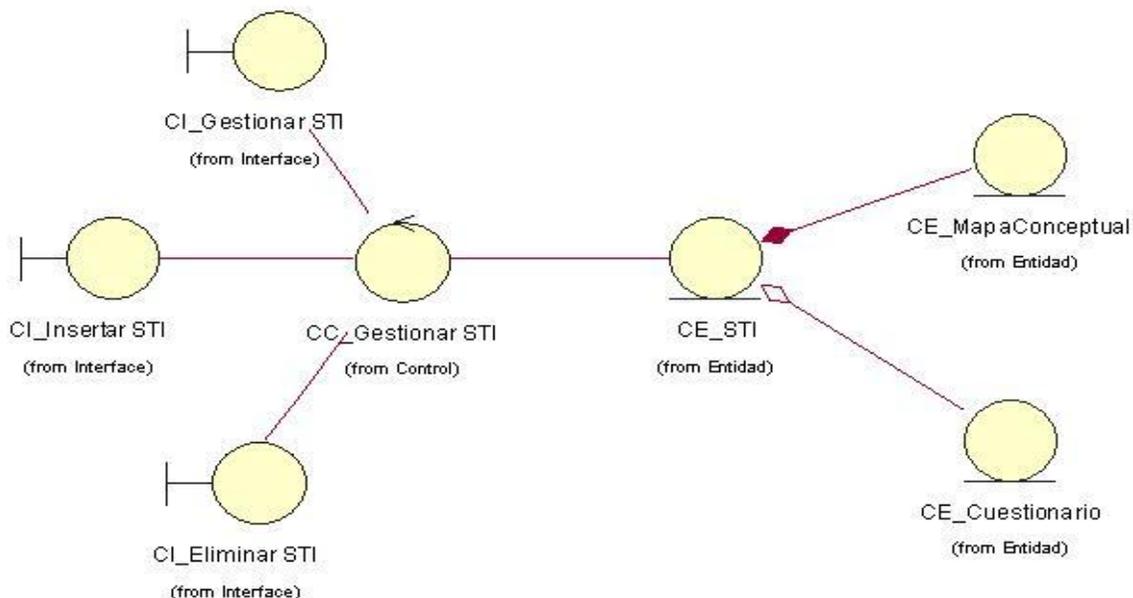
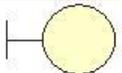
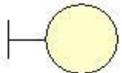
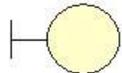
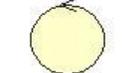
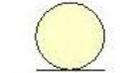
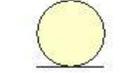
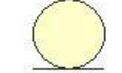


Figura 5. Diagrama de las clases del análisis, CUS "Gestionar Sistema Tutorial Inteligente"

Capítulo 5: Análisis y diseño del sistema

Tabla 8. Descripción de las clases del análisis, CUS "Gestionar Sistema Tutorial Inteligente"

Clase	Descripción
 CI_Gestionar STI (from Interface)	Esta clase modela la interacción que se establece entre el usuario y el sistema. Brinda las opciones de insertar un nuevo Sistema Tutorial Inteligente y de eliminar uno ya existente.
 CI_Insertar STI (from Interface)	Esta clase modela la interacción que se establece entre el usuario y el sistema. Permite al usuario ingresar los datos de un nuevo STI y llevar a cabo la inserción de éste en la base.
 CI_Eliminar STI (from Interface)	Esta clase modela la interacción que se establece entre el usuario y el sistema. Permite ver una lista con todos los STI existentes y llevar a cabo la eliminación de uno de ellos de la base.
 CC_Gestionar STI (from Control)	Esta clase es responsable de responder a las peticiones procedentes de la interfaz de usuario. Efectúa acciones como: la validación de datos, la ejecución de acciones de negocio y devolución de respuestas al usuario.
 CE_Cuestionario (from Entidad)	Esta es una clase persistente y porta toda la información necesaria para la realización de cualquier operación que tenga relación con los cuestionarios.
 CE_MapaConceptual (from Entidad)	Esta es una clase persistente y porta toda la información necesaria para la realización de cualquier operación que tenga relación con los Mapas Conceptuales.
 CE_STI (from Entidad)	Esta es una clase persistente y porta toda la información necesaria para la realización de cualquier operación que tenga relación con los Sistemas Tutoriales Inteligentes.

Capítulo 5: Análisis y diseño del sistema

Caso de Uso "Generar MCI"

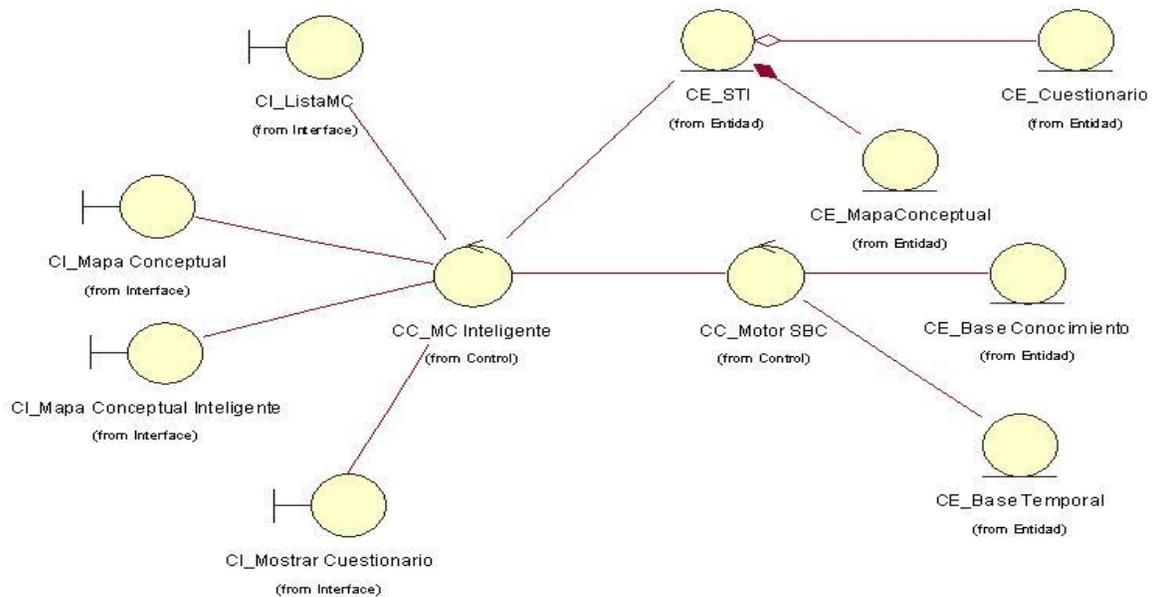
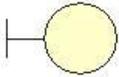
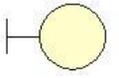
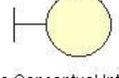
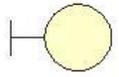


Figura 6. Diagrama de las clases del análisis, CUS "Generar MCI".

Tabla 9. Descripción de las clases del análisis, CUS "Generar MCI".

Clase	Descripción
 CI_ListaMC (from Interface)	Esta clase modela la interacción que se establece entre el usuario y el sistema. Permite ver una lista con todos los Mapas Conceptuales existentes en la base para seleccionar uno y visualizarlo en detalle.
 CI_Mapa Conceptual (from Interface)	Esta clase modela la interacción que se establece entre el usuario y el sistema. Provee una vista completa y detallada de un Mapa Conceptual.
 CI_Mapa Conceptual Inteligente (from Interface)	Esta clase modela la interacción que se establece entre el usuario y el sistema. Provee una vista completa y detallada de un Mapa Conceptual Inteligente.
 CI_Mostrar Cuestionario (from Interface)	Esta clase modela la interacción que se establece entre el usuario y el sistema. Pone a disposición del usuario un cuestionario listo para ser respondido.

Capítulo 5: Análisis y diseño del sistema

 <p>CC_MC Inteligente (from Control)</p>	<p>Esta clase es responsable de responder a las peticiones procedentes de la interfaz de usuario. Efectúa acciones como: la evaluación de los cuestionarios realizados por los usuarios y a partir del resultado la construcción de un Mapa Conceptual Inteligente.</p>
 <p>CC_Motor SBC (from Control)</p>	<p>Esta clase es responsable de responder a las peticiones procedentes de la clase controladora CC_MC Inteligente. Es responsable de darle un tratamiento inteligente a los datos de entrada y de devolver un resultado definitivo para la futura construcción de un Mapa Conceptual Inteligente.</p>
 <p>CE_Mapaconceptual (from Entidad)</p>	<p>Esta es una clase persistente y porta toda la información necesaria para la realización de cualquier operación que tenga relación con los Mapas Conceptuales.</p>
 <p>CE_Cuestionario (from Entidad)</p>	<p>Esta es una clase persistente y porta toda la información necesaria para la realización de cualquier operación que tenga relación con los cuestionarios.</p>
 <p>CE_BaseConocimiento (from Entidad)</p>	<p>Esta es una clase persistente y porta toda la información necesaria para la realización de cualquier operación que tenga relación con los casos que forman parte de la Base de Conocimiento.</p>
 <p>CE_Base Temporal (from Entidad)</p>	<p>Esta es una clase persistente y porta de forma temporal toda la información relacionada con los casos candidatos a formar parte de la Base de Conocimiento.</p>
 <p>CE_STI (from Entidad)</p>	<p>Esta es una clase persistente y porta toda la información necesaria para la realización de cualquier operación que tenga relación con los Sistemas Tutoriales Inteligentes.</p>

5.2. Diagrama de colaboración del análisis por cada CU.

Un diagrama de colaboración se utiliza para modelar la dinámica del sistema a través de los objetos de las distintas clases del análisis y los mensajes que pueden ser enviados entre ellos. Muestra una interacción

Capítulo 5: Análisis y diseño del sistema

organizada en torno a los objetos que efectúan operaciones (UCI 2008). Seguidamente se muestran los diagramas de colaboración de algunos CU, los restantes diagramas se encuentran en el Anexo C.

Caso de Uso "Gestionar Sistema Tutorial Inteligente"

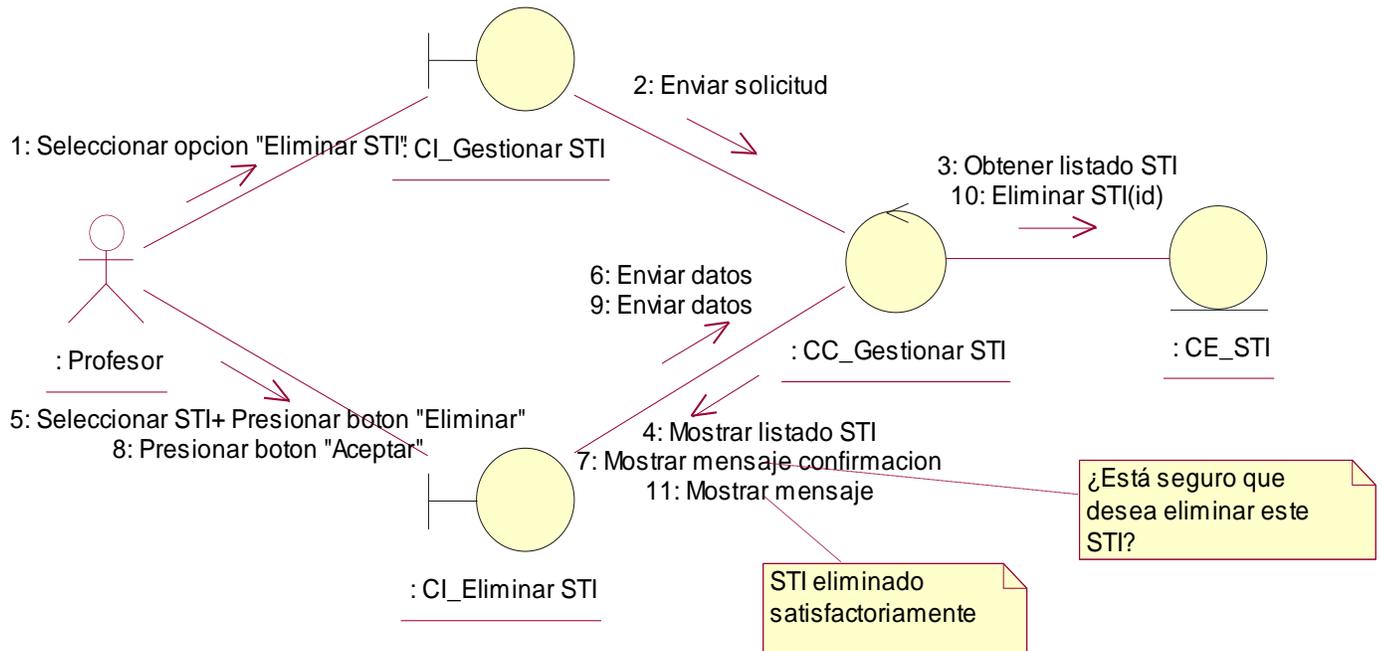


Figura 7. Diagrama de colaboración del análisis Sección "Eliminar STI"

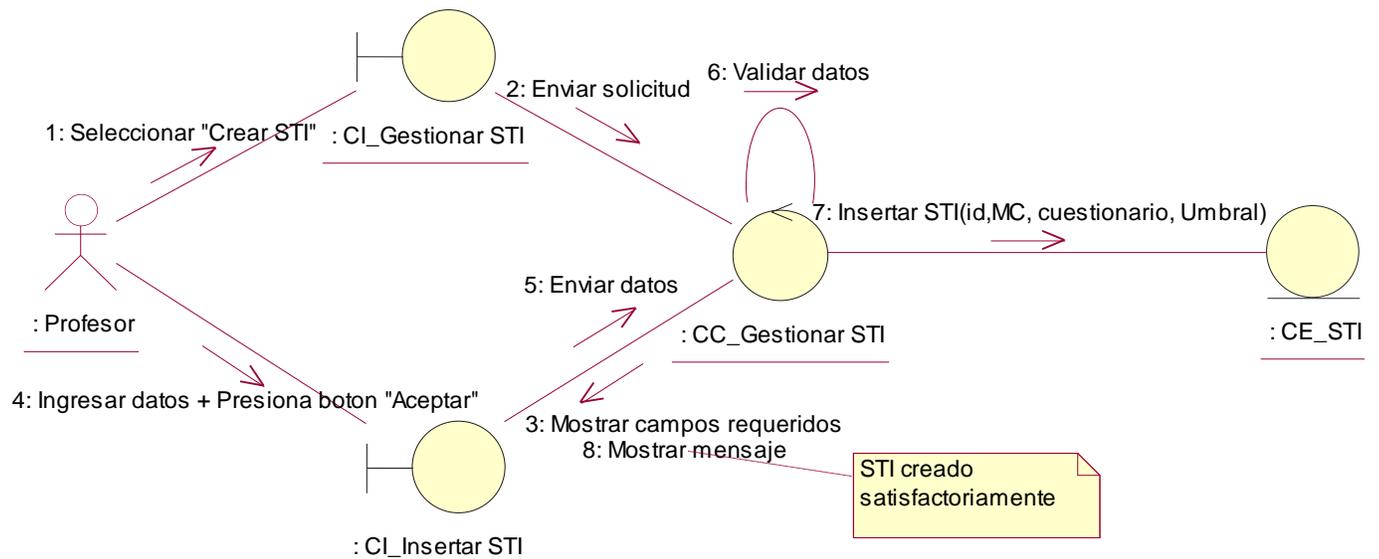


Figura 8. Diagrama de colaboración del análisis Sección "Insertar STI"

Capítulo 5: Análisis y diseño del sistema

Caso de Uso "Generar MCI"

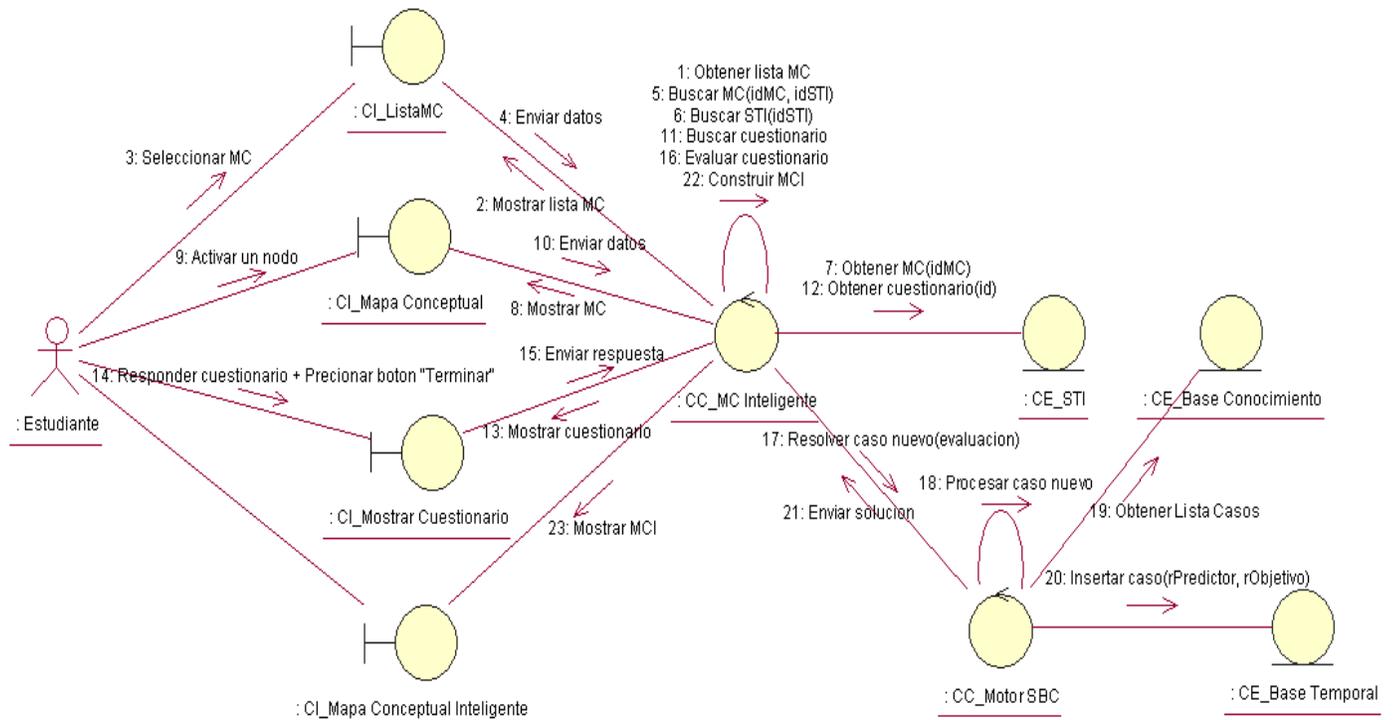


Figura 9. Diagrama de colaboración del análisis CUS "Generar MCI"

5.3. Arquitectura propuesta.

Según la IEEE la Arquitectura es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

La arquitectura es una vista estructural de alto nivel, ocurre muy tempranamente en el ciclo de vida y define los estilos o grupos de estilos adecuados para cumplir con los requerimientos no funcionales (UCI 2009).

5.3.1. Patrón Arquitectónico Modelo - Vista – Controlador (MVC)

El patrón MVC separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes diferentes. Con frecuencia es utilizado en aplicaciones Web, de manera que la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio (UCI 2009).

Capítulo 5: Análisis y diseño del sistema

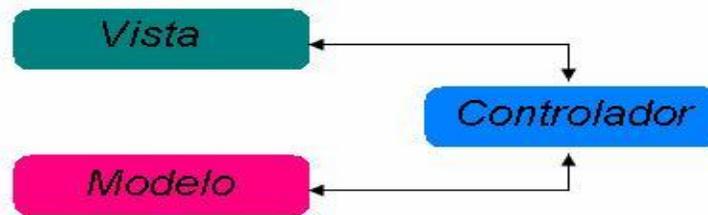


Figura 10. Patrón arquitectónico Modelo-Vista-Controlador

Modelo: Ésta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

Vista: Éste presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Éste responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista (UCI 2009).

Se propone MVC como patrón arquitectónico a utilizar en el desarrollo de la presente investigación teniendo en cuenta que la forma en que se implementa responde y satisface las necesidades que plantea la construcción de la solución propuesta.

5.4. Modelo del Diseño.

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, y es utilizado como una entrada fundamental de las actividades de implementación (Jacobson 2000).

5.5. Diagrama de clases del diseño por cada CU.

Seguidamente se muestran el diagrama de clases del diseño de algunos CU, los restantes diagramas se encuentran en el Anexo D.

Capítulo 5: Análisis y diseño del sistema

Caso de Uso "Gestionar Sistema Tutorial Inteligente"

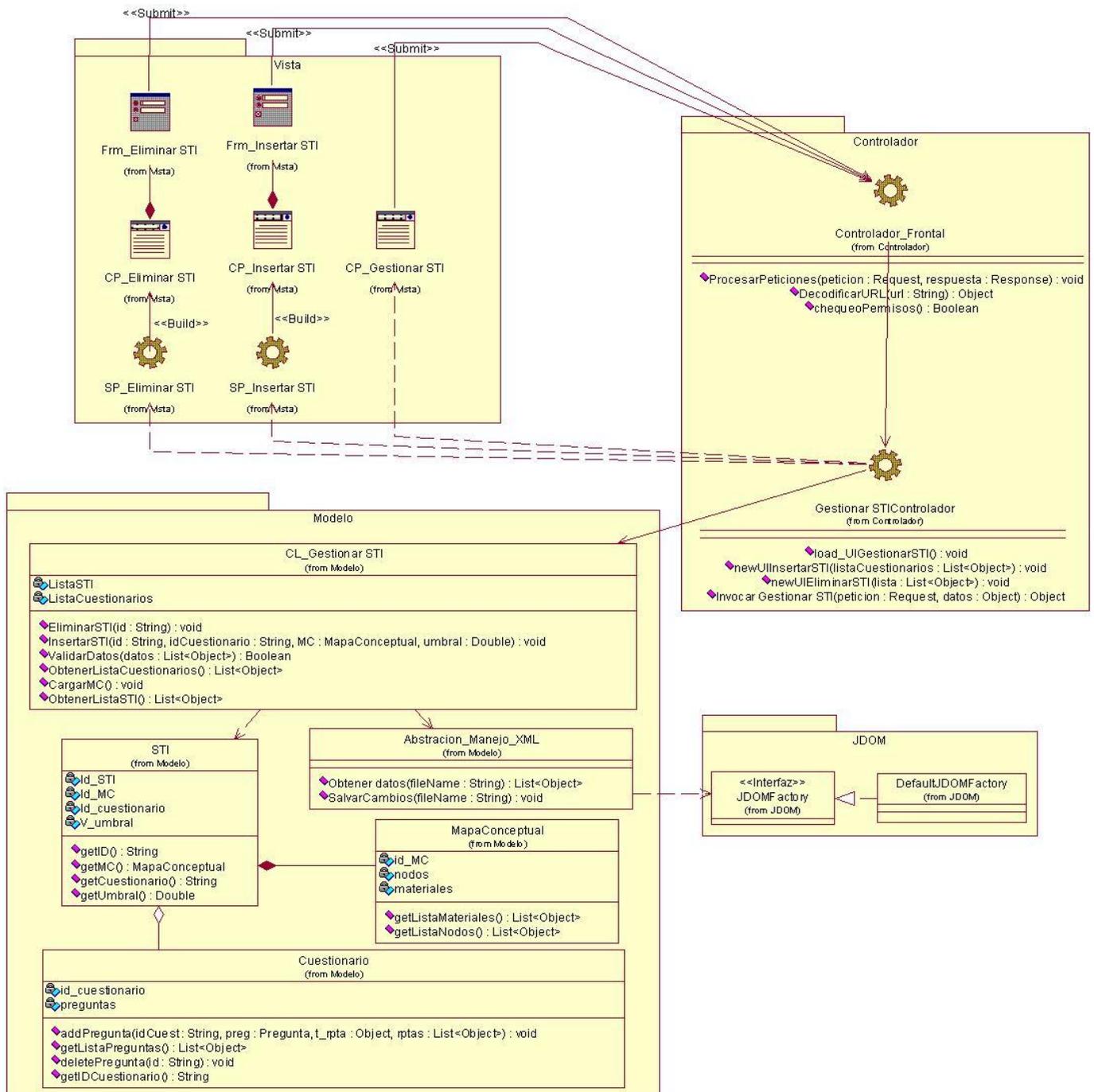


Figura 11. Diagrama de clases del diseño CUS "Gestionar Sistema Tutorial Inteligente"

Capítulo 5: Análisis y diseño del sistema

Caso de Uso "Generar MCI"

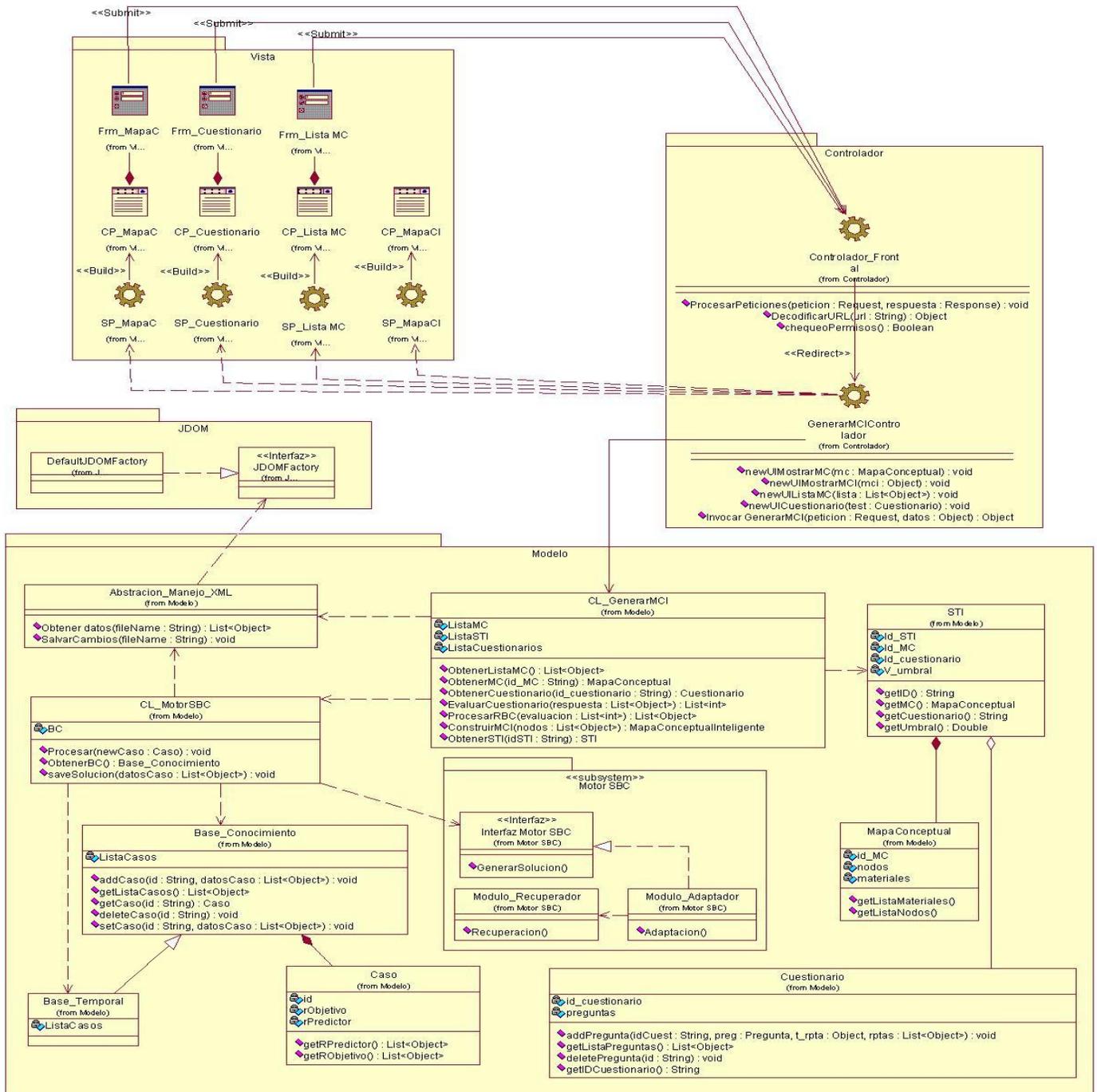


Figura 12. Diagrama de clases del diseño CUS "Generar MCI"

Capítulo 5: Análisis y diseño del sistema

5.6. Diagrama de secuencia del diseño por cada CU.

Una vez obtenidos los diagramas de clases se procede a elaborar los diagramas de secuencia, que son de vital importancia debido a que muestran desde una vista interna del sistema las interacciones entre los objetos, ordenados en una secuencia temporal durante un escenario concreto. Los diagramas de secuencia del diseño de los CU identificados se encuentran en el Anexo E.

5.7. Principios de diseño

El prototipo de Interfaz de Usuario se construye con el objetivo de mostrar una aproximación de cómo quedará el futuro sistema propuesto. El diseño debe ser concebido teniendo en consideración las características de los usuarios que van a trabajar con la herramienta. En este caso se trata de estudiantes y profesores, por lo cual se tienen en cuenta una serie de principios al diseñar las interfaces de la presente aplicación Web.

- La ley de Fitts. Éste hace referencia a las características que tienen que tener los objetos para que sea más fácil interactuar con ellos, teniendo en cuenta la posición en la que se encuentren, el tamaño y la expresividad de dichos elementos.
- Dar feedback al usuario. Consiste en dar información de retorno al usuario sobre las consecuencias de las acciones que realiza.
- Reutilización de la experiencia del usuario. Consiste en exponer al usuario ante lo conocido y plantea que no es recomendable crear interfaces que requieran excesivo aprendizaje.
- Imágenes. Plantea que no es recomendable cargar excesivamente nuestras páginas de imágenes o de lo contrario hacer uso de thumbnails (imágenes de muy reducido tamaño que al pulsarse abren otra ventana en la que se carga la imagen original) (Díaz 2001).

5.7.1. Estándares de la interfaz de la aplicación.

En la interfaz se utiliza como color principal el azul pues este color es fresco, motiva a la meditación, permite una mayor legibilidad y un menor cansancio visual.

Capítulo 5: Análisis y diseño del sistema

Además es importante mantener una misma plantilla para las páginas y así mantenerlas consistentes y ordenadas. La interfaz propuesta se compone de los siguientes elementos:

- Cabecera: En esta sección se muestra el banner que identifica al sistema.
- Menú Principal: En esta sección se muestran todas las opciones a las que tiene acceso el usuario ya autenticado.
- Contenido: En esta sección se muestran los formularios según la opción seleccionada por el usuario.
- Pie. Esta sección cierra cada página.

Se pueden realizar múltiples operaciones en algunas páginas, de forma que el usuario no tenga que moverse tanto dentro de la aplicación, para completar una operación. Por ejemplo, se puede hacer la actualización y eliminación en las páginas donde se muestran listados.

5.8. Consideraciones Parciales.

En este capítulo se realiza una descripción y representación gráfica de los diagramas de clases y de interacción (colaboración y secuencia) para el Modelo de Análisis y el Modelo de Diseño respectivamente, tomando en consideración el patrón arquitectónico Modelo Vista Controlador. Con la modelación del análisis y diseño se obtiene como resultado una vista futura de cómo debe ser implementado el sistema, pues estos artefactos constituyen las bases para la implementación de la solución propuesta.

Conclusiones

Con el desarrollo de este trabajo se arriba a las siguientes conclusiones:

- Las características del actual proceso de desarrollo de Sistemas Tutoriales Inteligentes imponen la necesidad de crear una herramienta inteligente que posibilite a los docentes no expertos en el campo informático construir sus propios Sistemas Tutoriales Inteligentes.
- Se obtuvo un modelo para diseñar Mapas Conceptuales Inteligentes como una nueva perspectiva para el desarrollo de los Sistemas Tutoriales Inteligentes, combinando las ventajas de la Mapas Conceptuales con los Sistemas Basados en Casos.
- Se realizó el análisis y diseño de la propuesta de aplicación Web, definiéndose además las funcionalidades de la misma.

Finalmente dado los aspectos antes mencionados se concluye que el objetivo general para el presente trabajo ha sido cumplido correctamente, obteniéndose el diseño de una Herramienta para la navegación de Mapas Conceptuales Inteligentes y el prototipo no funcional de Interfaz de Usuario para la misma como productos finales.

Bibliografía y Referencias Bibliográficas

Aamodt, A. P., E. (1996). CBR: foundational issues, methodological variations and systems approach. AI Communications.

Agnar, A. P., Enric (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. [Consultado] 13/11/2009 [En línea] <http://www2.iii.csic.es/People/enric/AICom.html>.

Alpigini, J. F., J. Peters, et al. (2002). Rough sets and current trends in computing. Third International Conference, RSCTC 2002. Malvern, USA.

Anónimo. (2009,). definicion.org. [Consultado] 16/01/2010 [En línea] <http://www.definicion.org/lenguaje-de-programacion>.

Arias, F. J., J. Jiménez, et al. (2007). Una Aproximación Metodológica para la Construcción de Sistemas Tutoriales Adaptativos MultiAgente con Énfasis en el Modelo Pedagógico. Revista Avances en Sistemas e Informática, Vol.4 No. 3, Medellín, 78, ISSN 16577663.

Arruarte, A. F., Isabel; Ferrero, Begoña; Greer, Jim (1997). The IRIS Shell: "How to Build ITSs from Pedagogical and Design Requisites". Departamento de Lenguajes y Sistemas Informáticos. Donostia, Universidad del País Vasco UPV/EHU.

Barr, D. S. and G. Mani (1994). Using neural nets to manage investments. AI Expert.

Beck, J. E. (2007). Difficulties in inferring student knowledge from observations (and why you should care).

Bello, R. G., María; García, Zoila; Reynoso, Antonio (2002). Aplicaciones de la Inteligencia Artificial. Mexico, Universidad de Guadalajara.

Booch, G. R., James; Jacobson, Ivar (2006). El lenguaje unificado de modelado, Addison Wesley. ISBN: 8478290761, 2da edición.

Cabrera, J. L., Osmel; Álvarez, Ariane (2006). Las herramientas tecnológicas en el Proceso de Enseñanza Aprendizaje.

Bibliografía y Referencias Bibliográficas

- Cañas, A. J. C., Marco (2004). Concept Maps and AI: an Unlikely Marriage? Medellín, Universidad EAFIT (Escuela de administración finanzas y tecnología): Traducción libre realizada por los alumnos de la clase de ingeniería del Conocimiento en la Universidad EAFIT.
- Castiblanco, C. C. (2008). XML en Java – APIs y Parsers. [En línea] <http://casidiablo.net/xml-java/>
- Castillo, E. G., J.M.; Hadi, H. (1997). Expert Systems and Probabilistic Network Models. Springer. New York.
- Cataldi, Z. and F. J. Lage (2009). Sistemas Tutores Inteligentes Orientados a la Enseñanza para la Comprensión. EDUCTEC. Revista Electrónica de Tecnología Educativa. ISSN:1135-9250. [En línea] <http://edutec.rediris.es/revelec2/revelec28/>
- Costa, G., F. A. Salgueiro, et al. (2005). Sistemas inteligentes para el modelado del estudiante. GCETE'2005, Global Congress on Engineering and Technology Education.
- Díaz , M. D. (2001, 02/04/2010). Principios de diseño Web. [En línea] http://www.unal.edu.co/documentos/servicio_web/principios_de_diseno_web.pdf.
- Duque, N. (2007). Propuesta de Doctorado en Ingeniería de Sistemas. Medellín, Universidad Nacional de Colombia.
- Durán, E. B. (2006). Modelo del Alumno en Sistemas de Aprendizaje Colaborativo. WAIFE.
- Dürsteler, J. C. (2004). Mapas Conceptuales. InfoVis, [En línea] <http://www.infovis.net/printMag.php?num=141&lang=1>
- Dutta, S. B., P. (1991). Integrating Case-Based and Rule-Based Reasoning: the Possibilistic Connection. In Uncertainty in Artificial Intelligence 6. Pág: 281-298. North-Holland.
- Eclipse.org. (2010). [En línea] <http://www.eclipse.org/>
- EduTEKA. (2006). CmapTools, programa para elaborar mapas conceptuales. [Consultado] 2009/11/10, [En línea] <http://www.eduteka.org/Cmap1.php>

Bibliografía y Referencias Bibliográficas

- Fernández Beltrán, A. (2000). Mapas Conceptuales para la creación de Ambientes Virtuales de Aprendizaje. Instituto Politécnico Nacional de México. [Consultado] 13/11/2009, [En línea] <http://www.desarrollo.upev.ipn.mx/marco/B2A126.doc>.
- Ferrero, B. A., A; Fernández-Castro, I ;Urretavizcaya, M. (2001). Herramientas de autor para enseñanza y diagnóstico: IRIS-D. Departamento de Lenguajes y Sistemas Informáticos. Donostia, Universidad del País Vasco/Euskal Herriko Unibertsitatea.
- Ferrero, B. U., M.; Fernández-Castro, I. (1997). Un Sistema de Diagnóstico y Evaluación basado en DETECTive.
- Fiel, A. E., G.; Porreca, C.; Decunto, M.; López, D.; Laurito, F.; Scanavino, L.; Laudalio, A.; Burgos, M. (2009). Sistema para Detección de Problemas en el Aprendizaje. [En línea] <http://www.slideshare.net/Lizeeth/sistema-para-deteccion-de-problemas-en-el-aprendizaje>
- Fischer, G. (2000). User Modeling in Human-Computer Interaction. Contribution to the 10th Anniversary Issue of the Journal. User Modeling and User-Adapted Interaction (UMUAI).
- Flores, J. G., Pedro; Huamani, Edward; Masias, Rodrigo; Tarazona, David; Wong, Alvaro. (2009). Agente Reflejo Simple y Agente de Búsqueda. [En línea] http://www.wiphala.net/courses/intelligent_systems/090666/2009-I/groups/grupo11.informe.pdf.
- G.S.I. (2007). Rational Rose Enterprise. [En línea] <http://www.rational.com.ar/herramientas/roseenterprise.html>
- García, M. and R. Bello (1997). El empleo del razonamiento basado en casos en el desarrollo de Sistemas basados en el conocimiento para el diagnóstico. Trabajo de Tesis en opción al grado científico de Doctor en Ciencias Técnicas., UCLV.
- Giraffa, L. (1997). Seleção e adoção de estratēguas de ensino em Sistemas Tutores Inteligentes. Porto Alegre.

Bibliografía y Referencias Bibliográficas

- González, H., N. Duque, et al. (2008). Modelo del estudiante para sistemas adaptativos de educación virtual. Revista de Avances en Sistemas e Informática, Vol. 5 No. 1, Edición especial Medellín. ISSN 1657-7663. III Congreso Colombiano de Computación - 3CCC 2008.
- Graf, S. (2005). Improving student modeling: The relationship between learning styles and cognitive traits.
- Guerrero, H. (2007). [En línea] <http://hancocchi.net/proceso-desarrollo-software-orientado-objetos/>
- Gutiérrez, I. and R. Bello (2003). Modelo para la Toma de Decisiones usando Razonamiento Basado en Casos en condiciones de Incertidumbre. Trabajo de Tesis en opción al grado científico de Doctor en Ciencias Técnicas. UCLV.
- Gutiérrez, J. (1996). INTZA: un Sistema Tutor Inteligente para Entrenamiento en Entornos Industriales. Donostia, Euskal Herriko Unibertsitatea.
- Hammond, K. J. (1986). CHEF: A Model of Case-Based Planning. Department of Computer Science. Yale University.
- Hand, D. J. (1997). Construction and Assessment of Classification Rules. John Wiley & Sons. UK, Chichester.
- Hatzilygeroudis, I., A. Karatrantou, et al. (2004). An Expert System with Certainty Factors for Predicting Student Success. Springer-Verlag Berlin Heidelberg.
- Hatzilygeroudis, I. and J. I. Prentza (2004). Knowledge Representation Requirements for Intelligent Tutoring Systems. Proceedings of 7th International Conference Intelligent Tutoring Systems, Brasil.
- Hilera, J. and V. Martínez (1995). Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones. Addison-Wesley.
- Huapaya, C. R., G. M. Arona, et al. (2005). Sistemas Tutoriales Inteligentes Aplicados a Dominios de la Ingeniería. JEITICS 2005 - Primeras Jornadas de Educación en Informática y TICS en Argentina.
- Jacobson, I. B., Grady; Rumbaugh, James (2000). El Proceso unificado de desarrollo de software. Madrid. ISBN: 84-7829-036-2. [En línea] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>

Bibliografía y Referencias Bibliográficas

- Jiménez, E. G., María D.; Perichinsky, Gregorio. (2006). Una Aplicación de la Tecnología de Multiagentes a los Sistemas Tutores Inteligentes: Enseñanza de Computación en Carreras de Ingeniería. [En línea] <https://dc.exa.unrc.edu.ar/wicc/papers/InformaticaEducativa/83.pdf>
- Jiménez, J. A. and D. A. Ovalle (2008). Uso de técnicas de Inteligencia Artificial en ambientes distribuidos de enseñanza/aprendizaje. Revista Educación en Ingeniería. ISSN 1900-8260. no. 5.
- Jiménez, J. O., Demetrio (2004). Entorno Integrado de Enseñanza/Aprendizaje basado en ITS & CSCL. Revista Iberoamericana de Sistemas, Cibernética e Informática. USA.
- Kolodner, J. (1993). Case-based reasoning. San Francisco, California, Morgan Kaufmann.
- Kolodner, J. L. (1992). An Introduction to Case-Based Reasoning. Artificial Intelligence Review 6.
- Lerner, U. (2002). Hybrid Bayesian Networks for Reasoning about Complex Systems. [En línea] <http://citeseer.nj.nec.com/lerner02hybrid.html>
- Lopez, M. M., Vanessa; Montañó, Nora (2008). Sistema generador de ambientes de enseñanza-aprendizaje constructivistas basados en objetos de aprendizaje (AMBAR): La interdisciplinariedad en los ambientes de aprendizaje en línea. España.
- Lou, R. (2002). Comparación de las Tecnologías Java para XML. [En línea] <http://www.emagister.com/comparacion-tecnologias-java-para-xml-cursos-1084374.htm>
- Macias, J. C., P. (2001). An Authoring Tool for Building Adaptative Learning Guidance Systems on the Web. Lecture Notes in Computer Science: Active Media Technology-AMT. Heidelberg, Spring-Verlag.
- Martinez, G. G., Fernando (2005). Exploración del aprendizaje de los estudiantes haciendo uso de ambientes colaborativos: Enseñando Inteligencia Artificial. México.
- Medina, D., N. Martínez, et al. (2007). Putting Artificial Intelligence Techniques into a Concept Map to Build Educational Tools. In S. V. B. Heidelberg.

Bibliografía y Referencias Bibliográficas

- Melgarejo, J. M., Carlos; Mori, Juan C.; Oscanoa, María; Quintanilla, Manuel; Tineo, Andrea. (2009). Agente Reflejo Simple e Informado. [En línea] http://www.wiphala.net/courses/intelligent_systems/090666/2009-I/groups/grupo08.informe.pdf
- Mendoza, M. A. (2004). Metodologías de Desarrollo de Software. Informatízate. [En línea] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
- Ming, G. Q., C. (2007). EpiList: An Intelligent Tutoring System Shell for Implicit Development of Generic Cognitive Skills That Support Bottom-Up Knowledge Construction. IEEE Transactions on Systems, Man & Cybernetics.
- Minka, T. (2001). Expectation propagation for approximate Bayesian inference. In Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI).
- Molpeceres, A. P., Martín (2002). Arquitectura empresarial y software libre, J2EE. javaHispano.
- Morales, E. (1999). Razonamiento Basado en Casos (CBR). [En línea] <http://aprenderonline.freeservers.com/cbr.html>
- Moreno, L. G., E.; Piñeiro, J.; Popescu, B.; Hamilton, A.; Sigut, J.; Torres, J.; Toledo, J.; Merino, J.; González, C. (2007). Hacia un Sistema Inteligente basado en Mapas Conceptuales Evolucionados para la Automatización de un Aprendizaje Significativo. Aplicación a la Enseñanza Universitaria de la Jerarquía de Memoria. Universidad de La Laguna. Tenerife.
- Nadereau, Y. (2009). Propuesta de automatización para el cálculo de las Métricas de Calidad generadas en los proyectos productivos de la Facultad 9. Ciudad de la Habana, Universidad de la Ciencias Informáticas (UCI).
- Netbeans.org. (2010). [En línea] <http://platform.netbeans.org/>
- Ohno-Machado, L. (1996). Medical Applications of Neural Networks: Connectionist Models of Survival. Stanford University, Department of Computer Science. STAN-CS-TR-96-1564.
- Ojeda, A. D. C., Fe E.; González Landrián, Lázara; Pinedo, Melisy P.; Hernández Gener, Mary E. (2007). Los mapas conceptuales: una poderosa herramienta para el aprendizaje significativo. ACIMED.

Bibliografía y Referencias Bibliográficas

- Ovalle, D. (2007). Análisis funcional de la estrategia de aprendizaje individualizado adaptativo.
- Ovalle, D. A., J. Jiménez, et al. (2005). Sistemas de Enseñanza/Aprendizaje basados en Agentes Inteligentes Pedagógicos.
- Paredes, R. (2001). Modelado de usuario para un ambiente lifelong learning. Universidad de las Américas – Puebla, México.
- Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. Palo Alto: Morgan Kaufmann.
- Pérez, G. S., José (2009). Arquitectura del Sistema Nacional Público para el Seguimiento de Inversiones y Sectores (SINAPSIS). Cuba, Universidad de las Ciencias Informáticas.
- Pérez, M. M. R. (2002). Enseñanza Asistida por Ordenador: Entrenador en las metodologías de solución de ejercicios típicos del tema "Proyecciones de cuerpos geométricos elementales. Tesis Doctoral. Oviedo.
- Pontes, A. (2005). Aplicaciones de las tecnologías de la información y de la comunicación en la educación científica. Primera parte: Funciones y Recursos." Revista Eureka sobre Enseñanza y Divulgación de las Ciencias (2005), Vol. 2, N° 1, ISSN 1697-011X.
- Riccucci, S. (2008). Knowledge Management in Intelligent Tutoring Systems. Dottorato di Ricerca in Informatica Università di Bologna, Padova December 2007. Department of Computer Science University of Bologna. Italy.
- Rich, E. (1988). Inteligencia Artificial. Barcelona.
- Ríos Rodríguez, L. R. L. R., Elsy; Lescano Brito, Mateo; Hernández Reyes, Aurelio; García Hernández, Addislexy (2007). Los mapas conceptuales, las TIC y el e-learning. Revista Iberoamericana de Educación.
- Romero, G. M., Rogelio. (2000). Plataforma para la generación de sistemas tutoriales inteligentes en la capacitación de personal técnico de centrales eléctricas. [En línea] <http://www.iie.org.mx/simulacion/tutorial.pdf>

Bibliografía y Referencias Bibliográficas

- Romero, M. S., Enrique; Gómez-Gil, Pilar (2009). Diseño de Hedeia: Una herramienta para la construcción de Sistemas Tutores Inteligentes.
- Rosanigo, Z. P., Alicia; Bramati, Pedro; Bramati, Hernán (2007). Diseño de un entrenador. Universidad Nacional de la Patagonia San Juan Bosco.
- Salgueiro, F., Z. Cataldi, et al. (2005). "Sistemas Tutores Inteligentes: Redes Neuronales para selección del protocolo pedagógico." IV Workshop de Tecnología Informática Aplicada en Educación. XI Congreso Argentino de Ciencias de la Computación.
- Schank, R. (1996). Goal-based scenarios: case-based reasoning meets learning by doing. AAI Press/MIT Press.
- Shneiderman, B. (2006). Diseño de interfaces de usuario. Estrategias para una interacción persona computadora efectiva. Addison Wesley. Mexico.
- Sierra, E., García-Martínez, R., Hossian, A., Britos, P. y Balbuena, E. (2006). Providing Intelligent User-Adapted Control Strategies in Building Environments. Research in Computing Science Journal. ISSN 1665-9899. Volumen 19. Pág. 235-241.
- UCI, D. I. (2008, Curso 2008-2009). Fase de Elaboración. Flujo de trabajo de Análisis y Diseño. Conferencia 7 de Ingeniería de Software 1. Universidad de la Ciencias Informáticas (UCI). [En línea] http://eva.uci.cu/file.php/102/Curso_2009-2010/Semana_11/Conferencia_7/Materiales_Basicos/Conferencia_7.doc
- UCI, D. I. (2009, Curso 2009-2010). Arquitectura y Patrones de diseño. Conferencia 2 de Ingeniería de Software 2, [En línea] http://eva.uci.cu/file.php/259/Curso_2009-2010/Conferencia_2/Conferencia_2_de_Arquitectura_2010.doc
- UCI, D. I. (2009, Curso 2009-2010). Fase de Inicio. Disciplina de Requisitos. Conferencia 6 de Ingeniería de Software 1. Universidad de la Ciencias Informáticas (UCI), [En línea] http://eva.uci.cu/file.php/102/Curso_2009-2010/Semana_7/Conferencia_6/Materiales_Basicos/Conferencia_6.doc

Bibliografía y Referencias Bibliográficas

- UCI, D. I. (2009, Curso 2009-2010). Fase de Inicio. Modelo del Negocio. Conferencia 5 de Ingeniería de Software 1. Universidad de la Ciencias Informáticas (UCI), [En línea] http://eva.uci.cu/file.php/102/Curso_2009-2010/Semana_4/Conferencia_5/Materiales_Basicos/Conferencia_5.doc
- Velez, J. F., Ramón (2007). Arquitectura para la Integración de las Dimensiones de Adaptación en un Sistema Hipermedia Adaptativo. España.
- Vitalia, M. (1989). Sistemas Tutoriales Inteligentes. Boletín de Informática Educativa N° 1. Colombia.
- Wenger, E. (1987). Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge. California, Morgan Kaufmann Publishers, Inc.
- Witten, I. F., Eibe; Trigg, Len; Geoffrey, Mark; Jo, Sally. (2005). "Weka: Practical machine learning tools and techniques with java implementations." New Zealand. University of Waikato. [En línea] <http://www.cs.waikato.ac.nz/~ml/publications/1999/99IHW-EF-LT-MH-GH-SJC%-Tools-Java.pdf>
- Wolf, B. (1984). Context Dependent Planning in a Machine Tutor Massachusetts, University of Massachusetts.