

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 2 “TELECOMUNICACIONES Y SEGURIDAD
INFORMÁTICA”**



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

**Título: “Análisis y Diseño de una herramienta para elaborar Mapas
Conceptuales Inteligentes”**

Autora: Yaneidis Hernández Orozco.
Tutora: Dra. Natalia Martínez Sánchez.

**La Habana, 27 de Junio del 2011
“Año 53 de la Revolución”**

DECLARATORIA DE AUTORÍA

Por este medio declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad 2 a que hagan el uso que estimen pertinente con el mismo.

Para que así conste firmo el presente a los ____ días del mes de junio del año 2011.

Firma del Autor

Yaneidis Hernández Orozco

Firma del Tutor

Dra. Natalia Martínez Sánchez



**EL PRINCIPIO DE LA SABIDURÍA ES EL
TEMOR A JEHOVÁ.
PROVERBIOS**

AGRADECIMIENTO

A Dios por su inmenso amor y por su misericordia, por ser el guía de mi vida.

A mis padres por el amor, sacrificio y apoyo cuando más lo necesitaba, por confiar en mí a pesar de mis errores. Por el ánimo, cuando ya no me quedaban fuerza para seguir adelante, por la esperanza siempre... Los amo.

A mi madre por ser mi amiga, mi hermana y mi madre. Por ser como es, ese ángel tan especial.

A mi padre por todas las cosas, por estar junto a mí siempre a pesar de las circunstancias.

A mi tutora por la ayuda y paciencia demostrada, por tener siempre un tiempito a pesar de la cantidad de trabajo. Mil gracias.

A los hermanos en Cristo de la universidad y a toda esa increíble familia cristiana, por acogerme y darme el apoyo y cariño necesarios, por estar presentes siempre, gracias de todo corazón.

A mi familia por existir y ser maravillosa.

A los profesores de la facultad 2, por su apoyo incondicional.

A los profesores de los JCCE de mi municipio, San Luis.

A mis amigos inseparables de todos los tiempos, Manuel y Danieska.

A mis amistades de la universidad Maikel, Yurineisis y Ana Ivis.

A la Revolución por darme la oportunidad de serle útil.

A la UCI por acogerme en su ceno durante estos cinco años.

A todos, gracias...

DEDICATORIA

Al rey de reyes y señor de señores, a nuestro amparo y fortaleza, al Dios todopoderoso.

Y a la insustituible ayuda en la distancia (y en la proximidad) de mis padres, destinatarios por siempre de todo mi amor y mi gratitud.

RESUMEN

Las Tecnologías de la Información y las Comunicaciones ofrecen condiciones para transformar una enseñanza tradicional pasiva, fundamentalmente centrada en la trasmisión del contenido, el profesor y la clase, en otro tipo de educación más personalizada, participativa, centrada en alcanzar aprendizajes diversos y que posea una real significación para cada estudiante.

Tanto los Mapas Conceptuales como los Sistemas Tutoriales Inteligentes, son herramientas valiosas para apoyar el proceso de enseñanza-aprendizaje en cualquier área del saber. Cada uno de ellos tiene características distintivas. Por un lado los Mapas Conceptuales respecto a las destrezas cognitivas son un instrumento poderoso que permiten la organización, asociación, validación, interrelación, discriminación, descripción y ejemplificación de contenidos, con un alto poder de visualización. Sin embargo el proceso de aprendizaje que se desarrolla a través de un Mapa Conceptual no se adapta al estado cognitivo del estudiante. Por otra parte la característica distintiva de los Sistemas Tutoriales Inteligentes es que son programas que portan conocimientos de cierto contenido mediante un proceso interactivo individualizado. En este trabajo se describen los aspectos teóricos que constituyen el fundamento para el análisis y diseño de una herramienta computacional web de apoyo al proceso de enseñanza-aprendizaje.

Se propone un modelo que integra el paradigma de los Sistemas Basados en Reglas, los Sistemas Tutoriales Inteligentes y los Mapas Conceptuales que favorece la concepción de estos sistemas, teniendo en cuenta las facilidades y naturalidad del enfoque basado en reglas.

Palabras Claves:

Mapas Conceptuales, Sistemas Basados en Reglas, Sistema Tutorial Inteligente.

ÍNDICE

AGRADECIMIENTO	III
DEDICATORIA	IV
RESUMEN.....	V
INTRODUCCIÓN.....	4
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL TEMA DE LOS SISTEMAS TUTORIALES INTELIGENTES.....	8
1.1. Introducción.....	8
1.2. Sistema Tutorial Inteligente.....	8
1.2.1. Arquitectura de un Sistema Tutorial Inteligente.....	9
1.3. Mapas Conceptuales	12
1.3.1. Herramientas para la construcción de MC	12
1.4. Técnicas de Inteligencia Artificial. Sistemas Basados en el Conocimiento.....	14
1.5. Sistema Basado en Reglas.....	15
1.5.1. Estructura de un Sistema Basado en Reglas.....	15
1.5.1.1. Tipos de inferencia en un SBR	17
1.6. Análisis de soluciones existentes de STI	18
1.6.1. Resultado del análisis de los STI estudiados	19
1.7. Metodología y Tecnologías.....	19
1.7.1. Metodologías de desarrollo de software	19
1.7.1.1. Metodología Proceso Unificado de Rational (RUP)	20
1.7.1.2. Metodología Extreme Programming (XP)	23
1.7.1.3. Fundamento de la metodología de software seleccionada. RUP	24
1.7.2. Lenguaje modelado	24
1.7.2.1. Lenguaje Unificado de Modelos (UML)	24
1.7.3. Herramientas CASE	25
1.7.3.1. Visual Paradigm	25
1.7.3.2. Rational Rose.....	25
1.7.3.3. Herramienta CASE a utilizar Visual Paradigm	26
1.7.4. Sistemas Gestores de Bases de Datos (SGBD)	26
1.7.4.1. MySQL	26

1.7.4.2. PostgreSQL.....	27
1.7.4.3. Gestor de Base de Datos seleccionado. PostgreSQL.....	28
1.7.5. Servidores Web	28
1.7.5.1. Apache	28
1.7.5.2. Lighttpd	29
1.7.5.3 Servidor Web seleccionado Apache	29
1.7.6. Lenguajes de Programación para el desarrollo web.	30
1.7.6.1. JAVA	30
1.7.6.2. PHP.....	30
1.7.6.3. Lenguaje de Programación seleccionado PHP	31
1.7.7. Lenguajes para la programación de IA	31
1.7.7.1. CLIPS.....	31
1.7.7.2. Prolog.....	32
1.7.7.3. Lenguajes para la programación de IA seleccionado Prolog.....	32
1.7.8. Marco de Trabajo o Frameworks	32
1.8. Conclusiones	34
CAPÍTULO 2: MODELO PARA DISEÑAR MAPAS CONCEPTUALES INTELIGENTES UTILIZANDO UN SISTEMA BASADO EN REGLAS.	35
2.1. Introducción.....	35
2.2. Mapas Conceptuales Inteligentes	35
2.3. Características del modelo para elaborar MCI usando un Sistema Basado en Reglas	36
2.4. Conclusiones	39
CAPÍTULO 3: REALIZACIÓN COMPUTACIONAL DEL MODELO DE DESARROLLO.....	40
3.1. Introducción.....	40
3.2. Propuesta de solución	40
3.3. Modelo de Dominio	42
3.4. Relación de requerimientos	44
3.4.1. Requisitos funcionales.....	44
3.4.2. Requisitos no funcionales	46
3.5. Modelo del Sistema	48
3.5.1. Definición de los actores del sistema.....	48
3.5.2. Diagrama de Casos de Uso del sistema a automatizar.....	49

3.5.3. Patrones de Casos de Uso	49
3.6. Descripciones Textuales de los Casos de Uso del sistema.....	50
3.7. Técnica de validación de requisitos	53
3.7.1. Técnica de validación empleada en el desarrollo del sistema.....	54
3.8. Conclusiones del Capítulo	54
CAPÍTULO 4: ANÁLISIS Y DISEÑO DEL SISTEMA PROPUESTO.....	55
4.1. Introducción.....	55
4.2. Análisis del sistema	55
4.2.1. Diagramas de clases del análisis.....	55
4.3. Diseño del sistema	56
4.3.1. Patrón Arquitectónico Modelo Vista Controlador (MVC).....	57
4.3.2. Patrones de diseño.....	58
4.3.2.1. Patrón Active Record.....	58
4.3.2.2. Patrón Front Controller	59
4.4. Diagrama de diseño.....	59
4.4.1. Diagramas de interacción	61
4.5. Modelo de Base de Datos.....	62
4.5.1. Modelo de Datos.....	63
4.6. Descripción de las tablas del Modelo Físico de la BD	63
4.7. Principios de diseño.....	64
4.8. Tratamiento de errores	64
4.9. Seguridad	64
4.10. Conclusiones del Capítulo.....	65
CONCLUSIONES GENERALES	66
RECOMENDACIONES	67
REREFENCUAS BIBLIOGRÁFICAS	68
REFERENCIAS CONSULTADAS.....	70
GLOSARIO DE TÉRMINOS	73

INTRODUCCIÓN

En la actualidad los sistemas de enseñanza se enfrentan al desafío de utilizar las Tecnologías de la Información y la Comunicación (TIC) para proveer a sus alumnos de herramientas y conocimientos, de forma que se puedan lograr mayores avances en el sistema educacional.

Su impacto, toca muy de cerca a las escuelas y universidades cubanas, lo que propicia modificaciones en las formas tradicionales de enseñar y aprender, siendo posible reconocer múltiples acciones que en los actuales Centros de Educación Superior (CES) se realizan en función de potenciar la inserción de las TIC en la enseñanza.

Se han desarrollado sistemas y lenguajes de autor, que facilitan el uso de la Enseñanza Asistida por Computadora (EAC), por parte de los profesores interesados en esta temática, mediante los cuales se pueden elaborar lecciones, incluyendo diagramas, gráficos, imágenes, textos, cuestionarios y permitiendo la evaluación de las respuestas, mediante la realimentación adecuada.

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 de Enseñanza-Aprendizaje Inteligentes (SEAI), denominados también Sistemas Tutoriales Inteligentes (STI). (1)

Los STI han demostrado su efectividad en diversas aplicaciones de los procesos de enseñanza-aprendizaje. Sin embargo; su construcción implica un complejo e intenso trabajo de ingeniería del conocimiento, que impide un uso más general y aprovechamiento óptimo. (1)

En los últimos años, los Mapas Conceptuales (MC¹) han alcanzado popularidad y en su integración con las tecnologías de la información y las comunicaciones se han convertido en un elemento importante en los planes de perfeccionamiento de los sistemas de enseñanza y han extendido su uso a otras esferas de la actividad humana en las que la gestión y uso del conocimiento ocupe un lugar preponderante. (2)

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

La combinación de los STI y los MC, a esta combinación se le llama Mapa Conceptual Inteligente (MCI) resultan una herramienta poderosa a la hora de transmitir y guiar a los alumnos en su aprendizaje; de los primeros se aprovecha su manera de adaptabilidad de acuerdo a los conocimientos cognitivos del alumno, utilizando para ello, técnicas de Inteligencia Artificial (IA), para lograr tal objetivo. De los segundos, la manera de representar gráficamente el conocimiento y su navegabilidad por los diferentes niveles del MC, dejando claro y por niveles la conceptualización del mismo.

Los MC no son capaces por si solos de guiar al estudiante de acuerdo a sus necesidades en el aprendizaje, pues estos permiten la navegabilidad sin restricciones por todos sus nodos; y en cuanto a los STI, estos no le muestran al alumno de manera gráfica la organización del conocimiento que necesita dominar. Pero hasta el momento, no existe un software capaz de utilizar las potencialidades de ambas herramientas.

Analizado el contexto anterior, se tiene además que el desarrollo de un STI requiere, conocimiento en el dominio de aplicación, técnicas de Programación, Ingeniería de Software e Inteligencia Artificial, y ellos 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 lo antes expuesto, se plantea la siguiente interrogante como **Problema a resolver**:
¿Cómo facilitar a profesores diseñar sus propios Mapas Conceptuales Inteligentes?

Luego de identificado el problema es necesario centrar la investigación en Los Sistemas Tutoriales Inteligentes como **Objeto de Estudio**, enmarcado en el **Campo de acción**: Los Sistemas Tutoriales Inteligentes utilizando MC.

Como **Objetivo general** se define, Realizar un análisis y diseño una herramienta computacional web para elaborar MCI, donde se integren los STI utilizando la técnica de IA los Sistema Basados en Reglas (SBR) y los MC para facilitar a profesores, el desarrollo de este tipo de Sistema de Enseñanza-Aprendizaje en cualquier área del saber. Y a partir de este, se desglosan los siguientes **objetivos específicos**:

- Realizar un estudio de los Sistemas de Representación del Conocimiento Basados en Reglas.
- Modelar un sistema computacional donde se aplique el SBR en la consideración del estudiante como entidad principal.

- Realizar un estudio de los STI y sus formas de integración con los MC, para representar el conocimiento.

Idea a defender

Con el análisis y diseño de la herramienta computacional web se podrá implementar una aplicación facilitando al profesor no necesariamente conocedor del campo informático, diseñar sus propios Sistemas Tutoriales Inteligentes; además, permitirle al estudiante navegar por dicha herramienta utilizando las ventajas de los MC, de acuerdo a sus características cognitivas.

Para poder dar solución a los objetivos definidos anteriormente, se trazan las siguientes **Tareas de la Investigación:**

1. Elaboración del Marco Teórico Conceptual.
2. Identificar los principales sucesos que intervienen en la gestión de los procesos de enseñanza - aprendizaje.
3. Estudio de las técnicas de IA y de los SBR.
4. Selección de las herramientas para la futura implementación de la aplicación.
5. Estudio de las fases, flujos y artefactos que genera la metodología seleccionada
6. Realización del modelamiento del negocio, requisitos, análisis y diseño del sistema.

Para el desarrollo de este trabajo se tienen en cuenta algunos **Métodos de investigación** que a continuación se referencian.

Métodos teóricos

Análisis y síntesis: Se utiliza para realizar un análisis detallado de varios documentos que hacen referencia a los modelos de STI y de los MC.

Histórico-lógico: Se utiliza para conocer los antecedentes y tendencias tributantes a la investigación, seleccionando luego la más adecuada de acuerdo a las características que presentará el sistema que se desea proponer.

Métodos empíricos

Observación: Se utiliza para conocer a través de la percepción directa realizada a otros sistemas, características que necesita o puede tener el sistema, así como las vías para lograrlo.

Entrevista: Se realizan entrevistas al personal interesado en el tema, centrando el objetivo en buscar la manera más eficiente de obtener el producto con la calidad requerida por el usuario.

La tesis está estructurada en cuatro **capítulos**:

Capítulo 1: Contiene la fundamentación teórica del trabajo. Incluye todos los elementos teóricos que sustentan el problema a resolver y los objetivos del trabajo, el estado del arte, la propuesta de la metodología y tecnologías necesarias para la elaboración del sistema propuesto.

Capítulo 2: Contiene el modelo para diseñar MCI utilizando un SBR. Se describe el proceso de los SBR, definiendo las pautas de su funcionamiento.

Capítulo 3: Contiene la realización computacional del modelo para el desarrollo. Se describe una propuesta de la investigación y los procesos que están vinculados al campo de acción. Se plantean los requisitos funcionales y no funcionales que debe tener el sistema para lograr óptimos resultados.

Capítulo 4: Contiene el análisis y diseño del sistema propuesto. Se presenta el modelo de análisis, que incluye los diagramas de clases del análisis, los diagramas de colaboración para cada caso de uso y diagrama de clases de diseño web por caso de uso. Por último, se presenta el modelo de diseño de la base de datos, concluyendo en el desarrollo del análisis y diseño del sistema propuesto.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL TEMA DE LOS SISTEMAS TUTORIALES INTELIGENTES.

1.1. Introducción

En este capítulo se abordan los conceptos relacionados con los Sistemas Tutoriales Inteligentes (STI) y los Mapas Conceptuales (MC). Además, se realiza un estudio del estado del arte de las tendencias y tecnologías existentes, definiendo las herramientas necesarias para llevar adelante el desarrollo del sistema.

1.2. Sistema Tutorial Inteligente

Los STI surgieron como una evolución de los Sistemas de Instrucción Asistida por Computador (CAI), primeramente se les llamó ICAI (Enseñanza Inteligente Asistida por Computadora) nombre que aún se utiliza. Comenzaron a desarrollarse con la idea de poder impartir el conocimiento usando alguna forma de inteligencia para poder asistir y guiar al estudiante en el proceso de aprendizaje. Se buscó emular el comportamiento de un tutor humano, es decir, a través de un sistema que pudiera adaptarse al comportamiento del estudiante, identificando la forma en que el mismo resuelve un problema a fin de poder brindarle ayudas cognitivas cuando lo requiera. (3)

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. (2)

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. (2)

Un tutor inteligente, *“es un sistema de software que utiliza técnicas de IA para representar el conocimiento e interactúa con los estudiantes para enseñárselo”*. (2)

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 estudiantes que utilizan el programa”*. (2)

TEORÍAS DEL APRENDIZAJE Y MÉTODOS DE ENSEÑANZA CON ORDENADORES		
<i>Teoría del aprendizaje</i>	<i>Métodos de enseñanza</i>	<i>Características</i>
CONDUCTISMO	Enseñanza Asistida por Ordenador (EAO), multimedia educativo en CD-ROM, cursos empaquetados on line	Material de enseñanza estructurado. Aprendizaje por recepción. Se aprende como actividad individual del alumno con el ordenador
PROCESAMIENTO INFORMACIÓN	Sistemas tutoriales inteligentes, Hipermedia adaptativos	Metáfora del cerebro como computadora. Aplicaciones de los principios de la Inteligencia Artificial. El ordenador adapta la formación al sujeto

Figura 1.1: Teorías del Aprendizaje y Métodos de Enseñanza con Ordenadores.

Por lo que se puede concluir, que los STI son programas que utilizan técnicas de IA para representar el conocimiento sobre determinada materia, actuando como un tutor o profesor en el proceso de Enseñanza-Aprendizaje de forma personalizada.

1.2.1. Arquitectura de un Sistema Tutorial Inteligente

La arquitectura descrita en la figura 1.2, reúne los elementos más 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. (1)

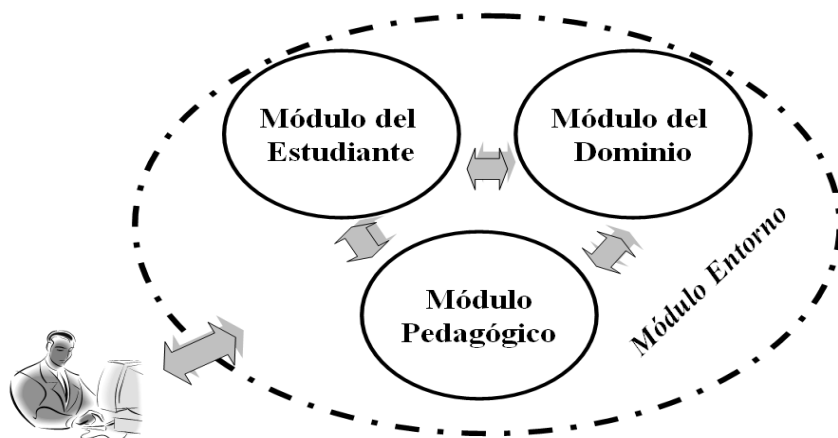


Figura 1.2: Arquitectura de un Sistema Tutorial Inteligente.

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 modelo 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 como se referencia a continuación toman en consideración características tales como: el estilo de aprendizaje, el nivel de conocimiento, la información personal o la combinación de algunas de ellas: (1)

Estilos de aprendizaje: conjunto de características psicológicas, rasgos cognitivos, afectivos y fisiológicos que suelen expresarse conjuntamente cuando una persona debe enfrentar una situación de aprendizaje. Los rasgos cognitivos tienen que ver con la forma en que los estudiantes estructuran los contenidos, forman y utilizan conceptos, interpretan la información, resuelven los problemas, etc. Los rasgos afectivos se vinculan con las motivaciones y expectativas que influyen en el aprendizaje, mientras que los rasgos fisiológicos están relacionados con el biotipo y el biorritmo del estudiante. (1)

Nivel de conocimiento: características propias de cada estudiante referente al grado de conocimiento que posee acerca de conceptos, temas y asignaturas. (1)

Información personal: datos como la edad, género, idioma, y otras informaciones que pueda ser de interés. (1)

Módulo del Dominio

El módulo del dominio, denominado también por muchos autores como módulo experto, proporciona los conocimientos del dominio. Satisface dos propósitos diferentes. En primer lugar, presentar la materia de la 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. (1)

En este módulo, el conocimiento a ser enseñado por el STI debe organizarse pedagógicamente para facilitar el proceso de enseñanza-aprendizaje. (1)

Módulo Pedagógico

Decide qué, cómo y cuándo enseñar los contenidos del tutor, adaptando sus decisiones pedagógicas a las necesidades del estudiante. Algunos autores le denominan módulo tutor, ya que es el encargado de comparar las características de los estudiantes con el contenido a enseñar y elegir la mejor forma de tomar las decisiones pedagógicas oportunas, adaptándose en cada momento al estudiante. (1)

Módulo Entorno

El módulo entorno 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 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. (1)

Modelado del estudiante en los STI

El modelado del alumno es un problema central en el diseño y desarrollo de los SEAI. En efecto, si la característica que distingue a los STI de los Sistema de Enseñanza- Aprendizaje es su capacidad de adaptación al alumno; entonces un STI debe ser capaz de determinar con la mayor precisión y rapidez posible cuál es el estado cognitivo y afectivo-motivacional del estudiante; para poder personalizar el proceso de enseñanza-aprendizaje.

El problema del modelado del alumno está en seleccionar la estructura de datos para representar toda la información relativa al alumno y elegir el procedimiento que se utiliza para realizar el diagnóstico. Evidentemente ambas componentes están estrechamente relacionadas, y por tanto se diseñan y desarrollan simultáneamente. (1)

1.3. Mapas Conceptuales

Novak², creador de los MC, lo define como 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.

Según Handerson (1994) “el MC fomenta la retrospectión 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).

1.3.1. Herramientas para la construcción de MC

En Internet, existen disponibles herramientas que permiten elaborar MC, unas de ellas son descargables, y otras pueden utilizarse en línea sin necesidad de descargarlas o de instalar software alguno. Estas últimas permiten a cualquier usuario elaborar y alojar en línea los organizadores y llegar a ellos desde cualquier lugar con acceso a Internet. Además, posibilita la interacción social mediante la cual otras personas pueden visualizarlos, descargarlos, reutilizarlos, comentarlos o simplemente usarlos como referencia. (4)

Herramientas como: el CmapTools, Inspiration, Smart Ideas, Macosoft, y otras, se encuentran en Internet para facilitar las disímiles ventajas que ofrecen.

Cmaptools

Es una herramienta desarrollada en la Universidad de West Florida (Estados Unidos), con la finalidad de construir y transmitir información representada en forma de MC presentando gráficamente conceptos teóricos. También, se pueden elaborar telarañas, mapas de ideas y diagramas causa-efecto. Permite a los usuarios construir los mapas en la computadora y posibilita, tanto elaborar mapas de forma colaborativa, como publicar y compartir en Internet los mapas elaborados o en servidores propios. Es una herramienta realmente muy completa y útil, además de gratuita.

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.

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.

Es compatible con los Sistemas Operativos (SO) Windows, Mac OSX, Linux (Intel) y Solaris (Sparc). (4)

A continuación se muestra la tabla 1.2, donde se comparan algunas herramientas que realizan entre otras funciones, la creación de MC.

Tabla 1.2: Comparación entre herramientas existentes para el diseño de MC.

Criterios de comparación	<u>Inspiration (v.7.5)</u>	CmapTool(v.3.0)	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 imágenes que pueden personalizarse.
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: doc, mapas, ppt, video, imagen, html, exe.
Portabilidad del MC, salva en diferentes formatos	HTML, GIF, JPG, BMP y WMF	HTML, PDF, JPG, SVG, TXT y XML, extensión propia (.CMAP).	JPG, BMP, HTML y XML, extensión propia (.MCF).
Idioma utilizado en la interfaz	Inglés	Configurable en múltiples idiomas.	Español

Luego de analizadas las diferentes herramientas, se concluye a seleccionar CmapTool, ya que cumple con las necesidades del software. Este permite que los MC puedan ser exportados al formato HTML, es una de las herramientas más utilizadas en todo el mundo y es configurable en múltiples idiomas.

1.4. Técnicas de Inteligencia Artificial. Sistemas Basados en el Conocimiento

La Inteligencia Artificial (IA), es una rama de la Ciencia de la Computación dedicada a la creación de hardware y software que intenta producir resultados similares a los expresados por los humanos. Le conciernen dos ideas básicas: la primera es que esta involucra el estudio de los procesos del pensamiento de los humanos y la segunda que trata de representar estos procesos en una computadora. Conceptualizar estas ideas básicas condujo al desarrollo de los llamados sistemas basados en el conocimiento. (2)

Un Sistema Basado en el Conocimiento se puede definir como: "un sistema computacional capaz de soportar la representación explícita del conocimiento de un dominio específico y de explotarlo a través de los mecanismos apropiados de razonamiento para proporcionar un comportamiento de alto nivel en la resolución de problemas". En otras palabras los sistemas basados en el conocimiento tratan con problemas poco estructurados en los que se pueden encontrar requisitos subjetivos, entradas inconsistente, incompletas o con incertidumbre y que no pueden ser resueltos aplicando los algoritmos clásicos. Además se caracterizan por tener un revolvedor de problemas capaz de manejar conocimiento específico del dominio de aplicación siendo independiente un módulo del otro. (2)

Existen formas de conocimiento y diferentes mecanismos de inferencias en los sistemas basados en el conocimiento, entre ellos los sistemas basados en reglas, los sistemas basados en probabilidades sistemas expertos conexionistas o redes expertas y los sistemas basados en casos.

Razonamiento Basado en Casos

Aborda nuevos problemas tomando como referencia problemas similares resueltos en el pasado. Se aprende de experiencias anteriores en forma de casos. Los Sistemas basados en casos utilizan casos como forma de representar el conocimiento y el paradigma de razonamiento basado en casos como método de solución de problemas.

Sistemas Basados en Probabilidades

Consiste en coleccionar muestras y realizar un procesamiento estadístico que produzca las probabilidades o frecuencias que forman la base de conocimiento y generalmente el Teorema de Bayes, como método de solución de problemas. No son factibles para todo tipo de dominio, pues se dificulta construir las redes con

ayuda de expertos humanos cuando existen carencias de conocimiento, además de que los métodos y modelos que utiliza están aún lejos de ofrecer explicaciones comprensibles.

Sistemas Basados en Reglas (SBR)

Un sistema basado en reglas es una forma de expresar el conocimiento atendiendo al siguiente patrón: Situación - Acción. Las reglas se rigen por este patrón debido a que se ejecutan solo en el caso de ocurrir un determinado suceso provocando una respuesta al mismo. Las reglas representan conocimiento informal o atajos, que permiten a los expertos encontrar rápidamente una solución a un problema sin tener que realizar un análisis detallado de situaciones particulares.

El proceso de solución de problemas en un SBR es crear una cadena de inferencias que constituye un camino entre la definición del problema y su solución.

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 que a través de las reglas se puede llegar de forma sencilla y clara a los resultados, resulta oportuna la utilización de un SBR.

1.5. Sistema Basado en Reglas

“Una regla es una afirmación lógica que relaciona dos o más objetos e incluye dos partes, la premisa y la conclusión. Cada una de estas partes consiste en una expresión lógica con una o más afirmaciones objeto-valor conectadas mediante los operadores lógicos (AND, OR Y NOT). La forma de representación del conocimiento son las reglas de producción.

Se llama producción o regla a un par ordenado (x,y). Se dice que (x) es la parte izquierda de la producción y (y) la parte derecha. **IF** las condiciones se cumplen o son ciertas **THEN** realiza las acciones (o concluye).

Los SBR son SBC en los que la forma de representación del conocimiento usado son las reglas de producción y como método de inferencia utiliza la regla de modus ponens.

1.5.1. Estructura de un Sistema Basado en Reglas

Un SBR es un sistema basado en el conocimiento en el cual se realiza una representación simbólica declarativa de un dominio mediante reglas de producción o reglas condicionales. Su estructura se muestra en la figura 1.3.

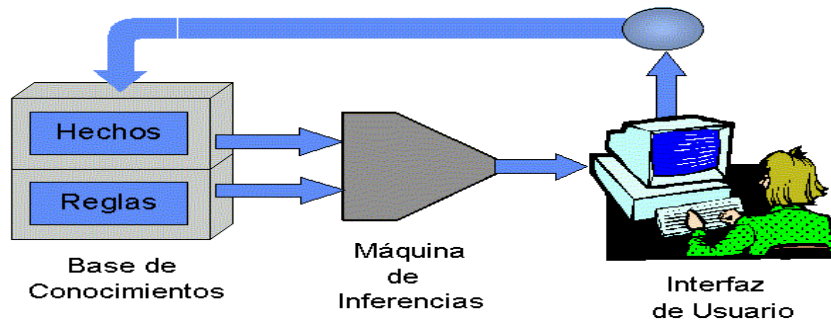


Figura 1.3: Estructura básica de un SBR.

Componentes básicos de los Sistemas Basados en Reglas

Base de conocimiento: Almacena la representación del conocimiento sobre el dominio de aplicación del sistema. Contiene las reglas de producción.

Se divide en la base de hechos y en la base de reglas:

- En la base de hechos: se representa el conocimiento en las variables de entrada y salida del sistema, que forman parte de las reglas semánticas almacenadas en la base de reglas.
- Una regla semántica es una representación del conocimiento en forma de sentencia condicional. En cada regla se combinan variables de la base de hechos con las partículas condicionales IF y THEN, así como con los operadores lógicos AND y OR. La parte de una regla situada entre IF y THEN es el antecedente de la regla, siendo en resto el consecuente. En el antecedente se sitúan las premisas que deben cumplirse para que la regla sea aplicable. El consecuente es el conjunto de acciones derivadas de la aplicación de la regla. (5)

Máquina de inferencia o intérprete de reglas: Es el elemento central del sistema; se encarga de coordinar la información procedente de todos los demás y de enviar los resultados de la inferencia al lugar oportuno. Por cada entrada al sistema, el motor de inferencia utiliza el conocimiento almacenado en forma de hechos y reglas para generar una salida. La inferencia en un RBS se basa en técnicas de la lógica proposicional o booleana, donde cada variable (también llamada hecho) es representada mediante un símbolo (X) y únicamente puede tomar dos valores: presente (X) o ausente (-X). La expresión más habitual de inferencia en la lógica proposicional es el modus ponens (encadenamientos).

Interfaz de usuario: Se encarga de solicitar al usuario la información necesaria y de mostrarle los resultados de la inferencia.

1.5.1.1. Tipos de inferencia en un SBR

Una regla se activa cuando se satisfacen las cláusulas indicadas por su antecedente, implicando la ejecución de su consecuente. Cuando se activan varias reglas para una misma configuración de las variables de entrada, se genera una salida por cada una de dichas reglas. Dichas salidas son hechos que actualizan el conocimiento actual almacenado en la base de hechos, es decir, que la salida de una regla puede provocar un cambio en el conocimiento que active otras reglas (la primera regla puede servir para ejecutar la siguiente). A este proceso se le conoce como encadenamiento de reglas.

Atendiendo al sentido del encadenamiento, se conocen cinco tipos de encadenamiento, estos se mencionan a continuación:

Encadenamiento hacia adelante: El motor de inferencia parte de los hechos para llegar a los resultados, esto es, selecciona las reglas que verifiquen las condiciones de la parte izquierda (premisas).

Encadenamiento hacia atrás: En este caso el motor de inferencia parte de los resultados y trata de volver a los hechos para comprobar si encajan con el problema planteado.

Encadenamiento mixto: Este modo ofrece varias posibilidades que resultan de combinar el encadenamiento hacia adelante y hacia atrás para paliar sus limitaciones y mantener las ventajas.

Algoritmos de búsqueda heurística: En los casos en los que la naturaleza de la base de conocimiento permita construir una estructura de árbol, el proceso de inferencia se convierte en un problema de búsqueda en un árbol. Existen diversos métodos, y la elección dependerá, por lo tanto, de la naturaleza del problema.

Herencia: Es el método de inferencia utilizado en entornos orientados a objetos. Un objeto hijo hereda propiedades y hechos de sus padres. Así, la asignación de nuevas propiedades a un objeto se realiza a través de las relaciones entre ese objeto y el resto. (6)

Pero los más usados son los tipos de inferencia: hacia delante y hacia atrás de ellos de forma aplicada se tiene que el:

Encadenamiento hacia delante o basado en datos: cuando la información introducida en el sistema hace que se ejecute una regla, y la conclusión obtenida permite que se ejecuten otras reglas. Utiliza solamente los datos disponibles y es menos específico que el encadenamiento hacia atrás ya que ejecutará todas las reglas posibles en función de la información introducida. Parte de la observación de hechos en las variables de entrada para, mediante el encadenamiento de reglas, alcanzar un hecho de salida deseado.

Encadenamiento hacia atrás o basado en objetivos: consiste en buscar una regla que permita establecer cierta conclusión y suele solicitar al usuario la información que no ha podido deducir. Lleva implícito un proceso de búsqueda, por lo que es más específico que el encadenamiento hacia delante y por tanto, más eficaz. Trabaja desde el objetivo, encadenando a través de las reglas hasta encontrar los hechos conocidos que soportan la demostración.

1.6. Análisis de soluciones existentes de STI

Con el avance de las tecnologías se encuentran muchísimos STI en todo el mundo, unos al alcance de todos, otros con acceso más restringido, pero todos ofreciendo servicios en los diferentes destinos.

Herramienta de Autoría (HEDEA)

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).

HESEI

HESEI es una herramienta de autor que facilita la elaboración de SEAI a usuarios no expertos en el campo informático; pero sí en dominios donde ejercen su profesión como docentes.

Como filosofía de trabajo de este software se definen dos fases bien delimitadas: diseño del SEAI y trabajo con la herramienta computacional HESEI.

Esto facilita al experto en la materia representar a plenitud su saber humano, lo que resulta un trabajo muy engorroso para un ingeniero del conocimiento debido a la disimilitud de materias y a la complejidad de poder asimilar la experiencia humana.

Esta herramienta está diseñada para realizar STI utilizando MC, pero no es multiplataforma, además es una aplicación de escritorio. (1)

Sistemas Expertos realizados en la Universidad de las Ciencias Informáticas

En la Universidad de las Ciencias Informáticas (UCI) se desarrollan producciones de software para la salud y equipos médicos, educación, telecomunicaciones, bioinformática, entre otras; vinculadas con diferentes instituciones, contribuyendo al desarrollo de la genética médica en el país.

Se han desarrollado Sistemas Expertos para diferentes ramas, siendo la medicina la rama más común y difundida en estos. A continuación se exponen varios ejemplos de sistemas expertos desarrollados en la UCI, la mayoría de estos desarrollados utilizando sistemas basados en reglas.

SEGEDIS (Sistema Experto para el diagnóstico médico de las enfermedades genéticas con Dismorfias): Creado con el objetivo de proporcionar a los genetistas cubanos una herramienta en el apoyo a sus decisiones. Además permite a los genetistas del Centro Nacional de Genética Médica (CNGM) que poseen los permisos de administración sobre la misma, agregar nuevas enfermedades, actualizando sistemáticamente la aplicación, y de esta forma brindar una mejor atención a los pacientes. (7)

SEDIM-SV utilizando la metodología Weiss-Kulikowski. Este es un Sistema Experto de Diagnóstico Médico de Sepsis Vaginales (SEDIM-SV), como medio de ayuda, para la consulta de Ginecología de la Universidad de las Ciencias Informáticas. (8)

1.6.1. Resultado del análisis de los STI estudiados

La mayoría de los STI analizados están diseñados para fines específicos, excepto el STI basado en casos HESEI, que está diseñado para realizar STI utilizando MC.

Con los resultados obtenidos de la investigación, se obtuvo un patrón a seguir sirviendo como punto de partida para la propuesta de solución, ya que tiene características afines como la presencia (cuestionarios para captar el conocimiento cognitivo del estudiante y materiales en apoyo a su aprendizaje), para garantizar la adaptabilidad de acuerdo a los conocimientos cognitivos del estudiante.

1.7. Metodología y Tecnologías

Todo proyecto o sistema de software tiene que estar basado en metodologías y herramientas para su estructuración. En este apartado se muestran las que se analizaron para la propuesta de solución.

1.7.1. Metodologías de desarrollo de software

Existen diversas metodologías que permiten el desarrollo de software. Estas constituyen un conjunto de procedimientos, herramientas y técnicas que ayudan a lograr el desarrollo y la documentación de un

producto. Y se desarrollan con el objetivo de dar solución a los problemas existentes que cada vez son más complejos. No existe una metodología de software universal, y se encuentran clasificadas en ágiles y tradicionales.

Metodologías ágiles: para desarrollo incremental, es decir, entregas pequeñas de software, con ciclos rápidos y de forma cooperativa, lo que implica que el cliente y los desarrolladores trabajan juntos constantemente con una cercana comunicación; permite realizar cambios de último momento.

Metodologías tradicionales o pesadas: requieren de mucha documentación, se centran en el producto y en las formas de que este sea eficiente y de calidad.

Pero a la hora de pensar en una metodología, este término es flexible ya que usted puede definir su propia metodología desarrollo de acuerdo con las características y necesidades propias de su producto de software. Por lo que el estudio realizado demuestra algunas características de las metodologías que a continuación se presentan.

1.7.1.1. Metodología Proceso Unificado de Rational (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, “El *Proceso Unificado de Rational* (RUP), es un proceso de ingeniería de software planteado por Kruchten (1996) con el objetivo de producir software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido. Utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software.

RUP es en esencia, una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo), y tiene como objetivo asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles.

Características principales de RUP

Dirigido por Casos de Uso: Los Casos de Uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.

Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño, constituye la arquitectura del producto a desarrollar.

Iterativo e incremental: Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo. (9)

Cada una de las fases del desarrollo, como lo refleja la figura 1.3, es desarrollada mediante el ciclo de iteraciones, la 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.

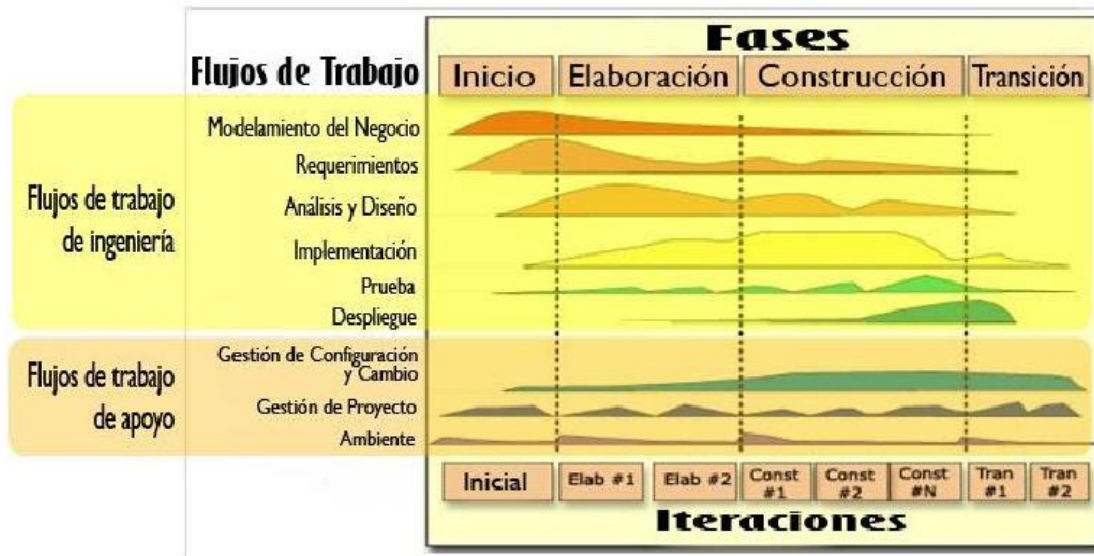


Figura 1.3: Fases e Iteraciones de la Metodología RUP.

Las fases de desarrollo del software son las siguientes:

Inicio: Tiene como propósito establecer el alcance del proyecto y proponer una visión general de la arquitectura de software.

Elaboración: En esta etapa se realiza el análisis del dominio del problema y definir el plan del proyecto donde se planifiquen las actividades necesarias y recursos requeridos. Determina la arquitectura óptima.

Construcción: En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.

Transmisión: El objetivo es llegar a obtener el release del proyecto. Comienza cuando el producto está suficientemente maduro para ser entregado. Se corrigen los últimos errores y se agregan los rasgos pospuestos.

El ciclo de vida que se desarrolla por cada iteración, es llevado por dos flujos de trabajo:

Flujo de trabajo de desarrollo:

Modelamiento del Negocios: Entendiendo las necesidades del negocio.

Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.

Análisis y Diseño: Trasladando 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 y que todo lo solicitado está presente.

Flujo de trabajo de soporte:

Configuración y administración del cambio: Guardando todas las versiones del proyecto.

Administrando el proyecto: Administrando horarios y recursos.

Ambiente: Administrando el ambiente de desarrollo.

Despliegue: Hacer todo lo necesario para la salida del proyecto.

Beneficios que aporta RUP

- Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.
- Permite la producción de software que cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos, con una agenda y costo predecible.
- Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.
- Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar software.
- Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria.
- No solo garantiza que los proyectos abordados serán ejecutados íntegramente sino que además evita desviaciones importantes respecto a los plazos.

1.7.1.2. Metodología Extreme Programming (XP)

La programación extrema o **eXtreme Programming** (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, “*Extreme Programming Explained: Embrace Change*” (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

XP es la primera metodología ágil y la que le dio conciencia al movimiento actual de metodologías ágiles. De la mano de Kent Beck, XP ha conformado un extenso grupo de seguidores en todo el mundo, disparando una gran cantidad de libros a los que comenzó el mismo Beck. (10)

La programación se hace en parejas, pero el código pertenece al equipo completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento, para el final.

Objetivos de XP

- La satisfacción del cliente es lo principal. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, se debe responder a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación.
- Potencia al máximo el trabajo en grupo. Los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

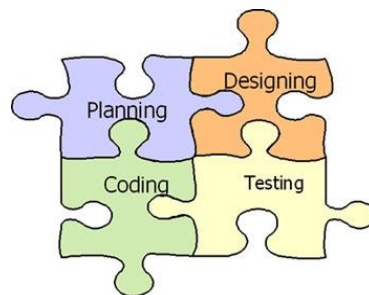


Figura 1.5: Metodología Extreme Programming.

Características de XP, están basadas en:

Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a 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.

1.7.1.3. Fundamento de la metodología de software seleccionada. RUP

RUP resultó ser la seleccionada. Está pensada para adaptarse a cualquier tipo de proyecto, principalmente donde el ciclo de vida está dirigido por casos de uso, que reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.

Se necesita dejar documentado todo lo referente al proceso de análisis y diseño para que pueda ser consultada por los que la necesiten y RUP genera mucha más documentación. Además, es escalable y extensible.

1.7.2. Lenguaje modelado

Los lenguajes de modelado permiten la modelación de objetos mediante un conjunto estandarizado de símbolos y técnicas de modelado. Modelan una simplificación de la realidad donde se toma de los objetos los aspectos significativos brindando así una mejor comprensión del sistema.

1.7.2.1. Lenguaje Unificado de Modelos (UML)

"UML" son las siglas de Unified Modeling Language (Lenguaje Unificado de Modelos), notación con que se construyen sistemas por medio de conceptos orientados a objetos. Es uno de los lenguajes de modelado de sistemas de software más conocido y utilizado en la actualidad. Se utiliza para visualizar, especificar, construir y documentar los artefactos de un software. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

- Permite especificar todas las decisiones de análisis, diseño e implementación.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- No es difícil de aprender ni de utilizar.
- Es libre.

1.7.3. Herramientas CASE

Las herramientas CASE cuyas siglas significan (Computer Aided Software Engineering), y en su traducción al Español significa Ingeniería de Software Asistida por Computación. Estas herramientas ayudan en todos los aspectos del ciclo de vida del software.

1.7.3.1. Visual Paradigm

Es una herramienta CASE que permite modelado UML. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, incluye modelado de negocio, requerimientos, así como otras funcionalidades que facilitan la gestión de proyectos. Permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Tiene integración con diversos IDE's (Entornos de Desarrollo Integrados) como NetBeans (de Sun Microsystems), JDeveloper (de Oracle), Eclipse (de IBM), JBuilder (de Borland).

- Soporta el lenguaje de modelado UML.
- Importación y exportación de ficheros XML.
- Ingeniería inversa, código a modelo, código a diagrama.
- Generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ofrece soporte para el mapeo de objeto relacional (Object Relational Mapping – ORM).
- Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- Fácil de instalar y actualizar.
- Está disponible en versiones de Windows y Linux.
- Presenta licencia gratuita y comercial.

1.7.3.2. Rational Rose

Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

- Mantiene la consistencia de los modelos del sistema software.
- Chequeo de la sintaxis UML.
- Generación Documentación automáticamente.
- Generación de Código a partir de los Modelos.
- Ingeniería Inversa (crear modelo a partir código).

- Apropia los Sistemas Operativos: Windows 2000, Windows NT y Windows XP.
- Integración con otras herramientas de desarrollo.

1.7.3.3. Herramienta CASE a utilizar Visual Paradigm

Se seleccionó la herramienta Visual Paradigm, debido a que soporta el ciclo de vida completo del desarrollo de software, las grandes ventajas que aporta son de mucha utilidad para los desarrolladores, ostenta soporte para Windows y Linux. Es una herramienta fácil de instalar y actualizar, los artefactos que se utilizan en la elaboración de las representaciones ofrecen que el trabajo quede con la mayor claridad posible. Permite crear posibles interfaces de usuario de un sistema en construcción, ahorra tiempo a los desarrolladores y minimiza las posibilidades de cometer errores.

1.7.4. Sistemas Gestores de Bases de Datos (SGBD)

El Sistema de Gestión de Bases de Datos (SGBD), es el responsable de tratar todas las peticiones de información de dichos usuarios. Este sistema es un conjunto de programas de propósito general, que permite a los usuarios controlar el acceso y la utilización de la base de datos, para incluir, modificar o recuperar información, incluyendo prestaciones con el fin de conseguir la independencia, integridad y seguridad de los datos, y la concurrencia de los usuarios. (11)

1.7.4.1. MySQL

Es un sistema SGBD relacional, multihilo y multiusuario que trabaja bajo la licencia GNU GPL Licencia GPL a partir de la versión 3.23.19, para cualquier uso compatible con esta licencia, pero para las empresas que deseen utilizarlo en productos privativos, necesitan comprar la licencia específica que le permita este uso.

- Consume muy pocos recursos, tanto de CPU como de memoria.
- Mayor rendimiento. Mayor velocidad al conectar con el servidor.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- Aunque presente problemas de servicios, no suele perder información ni corromper los datos.
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.

- No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

1.7.4.2. PostgreSQL

Es un SGBD es libre, publicado bajo la licencia BSD (Berkeley Software Distribution). Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

El desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y apoyado por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene mejor soporte para triggers y procedimientos en el servidor.
- Es multiplataforma. Corre en los sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows.
- Extensible, el código fuente está disponible para todos sin costo.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas, llaves foráneas, e incorpora una estructura de datos array, además, incluye herencia entre tablas.
- Dispone de API's en gran cantidad de lenguajes, C, C++, Java y PHP, Perl, etc.
- Soporta hasta 32 índices por tabla.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Soporta el uso de índices, reglas y vistas,
- Se integra con la librería de javascript Querys.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Altamente adaptable a las necesidades del cliente.
- Presenta ofertas de soporte, y una importante comunidad de profesionales.

1.7.4.3. Gestor de Base de Datos seleccionado. PostgreSQL

Teniendo en cuenta las características de los gestores analizados, se opta por usar PostgreSQL. Funciona en todos los principales SO, incluyendo Linux, UNIX y Windows. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios idiomas). También es compatible con el almacenamiento de objetos binarios, incluyendo imágenes, sonidos o vídeo. Tiene interfaces de programación nativa de C/, C++, Java, NET, Perl, Python, etc., y soporte para triggers y procedimientos en el servidor.

1.7.5. Servidores Web

Servidores Web (*Web Servers*): Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red. Los servidores web son aquellos cuya tarea es alojar sitios y aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HTTP (*Hypertext Transfer Protocol*).

1.7.5.1. Apache

Apache es uno de los servidores web más utilizados, posiblemente porque ofrece instalaciones sencillas para sitios pequeños y si se requiere es posible expandirlo hasta el nivel de los mejores productos comerciales. Además el servidor HTTP Apache, es un software libre y de código abierto para las plataformas Windows, Mac OS X y UNIX (GNU, BSD, etc.).

En cuanto a las **características** que posee Apache y que lo llevo al éxito están:

- Ofrece tecnología libre y de código abierto.
- Es un servidor Web configurable y de diseño modular, capaz de extender su funcionalidad y la calidad de sus servicios.
- Trabaja en conjunto con gran cantidad de lenguajes de programación interpretados como PHP, Perl, Java, JSP (Java Server Pages) y otros lenguajes de script, que son el complemento ideal para los sitios web dinámicos.
- Es posible configurar y personalizar cada uno de los mensajes de error que se pueden producir por la utilización del servidor.
- Cuenta con los archivos Log, en donde registra gran cantidad de información global del sistema, errores producidos en un determinado tiempo, en la cual estos archivos son de gran importancia para los

administradores de sistemas y pueden influenciar en las políticas de seguridad debido a la gran cantidad de información que contiene.

- Por su popular y utilización, es posible encontrar gran cantidad de documentos, ejemplos y ayuda en internet en todos los idiomas.
- Puede conectarse directamente a una Base de Datos.
- Posee diversos módulos que le permiten utilizar una gran cantidad de lenguajes y desarrollar funcionalidades avanzadas.

1.7.5.2. Lighttpd

Lighttpd es un servidor http o https, seguro, rápido que respeta estándares y consume muy pocos recursos, todo ello con un código limpio y elegante. Es ideal para ser usado en entornos donde la carga es máxima, se requieren respuestas rápidas y alta escalabilidad.

- Virtual Hosting incluso por medio de MySQL.
- Soporte de varios lenguajes de programación como PHP o Ruby a través de CGI y FastCGI.
- Admite certificados SSL y por lo tanto puede servir https.
- Autenticación con htpasswd, LDAP o MySQL.
- Tiene un módulo de reescritura y de redirección de URLs.
- Permite módulos externos.
- Acepta Webdav.
- Acepta SSI (*Server Side Includes*).

1.7.5.3 Servidor Web seleccionado Apache

Apache es un servidor altamente configurable de diseño modular que por su sencillez permite ampliar sus capacidades. Apache es gratuito y multiplataforma. Además trabaja con gran cantidad de lenguajes como: Perl, PHP y otros lenguajes de script. Apache presenta muchas otras características, entre ellas un elaborado índice de directorios, un directorio de alias, negociación de contenidos, informe de errores HTTP configurable, gestión de recursos para procesos hijos, integración de imágenes del lado del servidor, reescritura de las URL, comprobación de la ortografía de las URL, y manuales online.

1.7.6. Lenguajes de Programación para el desarrollo web.

1.7.6.1. JAVA

JAVA es un lenguaje de programación orientado al objeto y fue pensado para servir como nueva manera de manejar la complejidad del software.

Tiene convertido de las tecnologías más seguras para el desarrollo del Web site y de programas. Los Java applets Son de uso frecuente proporcionar funcionalidad mejorada mientras que hojean el WWW (*World Wide Web*).

- El lenguaje de programación es de alto nivel y muy poderoso.
- Es multiplataforma, prácticamente se usa en todas.
- Es seguro, se ejecuta en una sandbox.
- Independencia de la plataforma.
- Disponibilidad fácil para los usuarios pues es una fuente abierta.
- Se tiene que descargar todo el applet, solo la primera vez.
- Como fue creado por la Sun Microsystems, posee códigos que aun no es han sido liberados.
- Necesita la instalación de una enorme máquina virtual.

1.7.6.2. PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. PHP es utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Como producto de código abierto, PHP goza de la ayuda de una gran comunidad de desarrolladores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- Con PHP se puede hacer cualquier cosa que se desee realizar, procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.
- Sencillo de aprender y utilizar.
- Soporta técnicas de programación orientada a objetos, clases y herencia.

- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Biblioteca nativa de funciones sumamente amplia.
- Tiene manejo de excepciones (desde PHP5).

1.7.6.3. Lenguaje de Programación seleccionado PHP

Es independiente de plataforma, con una gran librería de funciones y mucha documentación. PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente. Además, la excelencia del lenguaje lo demuestra su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de BD; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta. Existe un amplio conocimiento del mismo, por lo que tiene aceptación por el grupo de desarrollo. Contiene una amplia documentación situada en su página oficial y está respaldado por una activa y numerosa comunidad de desarrolladores.

1.7.7. Lenguajes para la programación de IA

1.7.7.1. CLIPS

CLIPS es un acrónimo de C Language Integrated Production System (Sistema de Producción Integrado en Lenguaje C) es una herramienta para el desarrollo de Sistemas Expertos creada por la Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center. Los orígenes de CLIPS se remontan a 1984.

Es un lenguaje programado en C, tiene comunicación con otros lenguajes como C y ADA y mantiene similitudes al lenguaje C y LISP.

Se diseñó para facilitar el desarrollo de software que modele el conocimiento humano:

- Con propósitos específicos: alta portabilidad, bajo coste, y facilidad de integración.
- Puede ser llamado desde un lenguaje procedural, realizando su función y devolver el control al programa que le llamó.

CLIPS es un entorno completo para la construcción de sistemas expertos basados en reglas y/o objetos. La versión estándar de CLIPS proporciona un entorno de desarrollo interactivo orientado a texto, incluyendo una herramienta de depuración, ayuda en línea y un editor integrado, aunque se han

desarrollado interfaces visuales para plataformas Macintosh, Windows 3.x y el sistema X Window. (12)

1.7.7.2. Prolog

Prolog es un lenguaje de programación que se centra alrededor de un conjunto pequeño de mecanismos, incluyendo reconocimiento de patrones, estructuras de datos basadas en árboles y backtraking (retroceso) automático. Este conjunto pequeño constituye una estructura de programación sorprendentemente poderosa y flexible. Prolog es ideal para resolver problemas que involucren objetos, en particular objetos estructurados y relaciones entre ellos.

Entre las características con las que no cuenta Prolog se encuentran:

1. Sentencias de asignación.
2. Sentencias goto.
3. Sentencias if – then – else.
4. Ciclos do, ciclos for y ciclos while.

Sin embargo, Prolog compensa tales características con las siguientes, las cuales lo hacen un lenguaje poderoso para la representación del conocimiento:

- Un método para definir predicados mediante reglas de aserción y hechos.
- Un método para hacer preguntas para comenzar cómputos.
- Un reconocedor de patrones que construye y analiza las estructuras de datos. (13)

1.7.7.3. Lenguajes para la programación de IA seleccionado Prolog

De los dos lenguajes analizados se ha seleccionado Prolog. Acerca de CLIPS y se encontró una extensión de este lenguaje que era para su unión con PHP llamada PHLIPS, sin embargo no es estable, por lo que es descartado. Por su parte, Prolog puede unirse con PHP mediante comandos y ficheros, además este lenguaje es más sencillo de aprender y posee un motor de inferencia implícito que realiza: recursividad, instanciación, verificación, unificación y backtracking; mecanismos ideales para la construcción de un sistema experto.

1.7.8. Marco de Trabajo o Frameworks

El framework es un entorno de trabajo como estructura conceptual y tecnológica, para la organización y desarrollo del software, brindando una arquitectura que facilita la utilización de otras tareas.

Simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener.

Existen muchísimos framework que facilitan las implementaciones en php, y sin lugar a dudas son extremadamente necesarios a la hora de pensar en hacer aplicaciones web con calidad, por las disimiles ventajas que ofrecen al programador. (5)

Symfony

Es un framework muy poderoso y con muchísimas experiencias acumuladas. Está desarrollado en PHP5, se puede utilizar en plataformas nix (Unix, Linux) y Windows. Incorpora el patrón MVC, soporta AJAX, plantillas y un gran número de bases de datos. Es un Framework de desarrollo rápido.

Otras características:

- Posee partes de otros proyectos de código libre como son Creole, para su capa de abstracción de la base de datos o de Prado para su soporte de internacionalización.
- Más complejo y elegante que CodeIgniter, pero es uno de los más completos.
- Está diseñado para utilizarse en cualquier tipo de proyectos, pero fundamentalmente en proyectos grandes.
- Su principal desventaja está en su empinada curva de aprendizaje. (5)

CodeIgniter

CodeIgniter es un framework construido para codificadores PHP que necesitan una herramienta de desarrollo fácil para crear aplicaciones web simples y elegantes, permitiendo el desarrollo de las mismas con mayor rapidez. CodeIgniter se basa en un diseño modular (lo que quiere decir que puedes implementar librerías específicas a voluntad) lo que repercute en una mayor velocidad del sistema.

- Usa pocos recursos pero ofrece un rendimiento excepcional.
- Genera URLs limpias aptas para motores de búsqueda.
- Fácilmente extensible.
- Funciona sobre PHP 4 (a partir de 4.3.2) y PHP 5.
- SQL Server, Postgres, Oracle, MySQL (a partir de 4.1), MySQLi, SQLite y ODBC.
- La seguridad de la aplicación es un objetivo básico.

- La mayoría de módulos sólo se cargan cuando son necesarios, lo que redundará en un drástico recorte de los recursos necesarios.
- Ofrece útiles librerías de código, módulos y atajos que ayudan con las operaciones complejas como email, manipulación de imágenes, validación de formularios, subida de archivos, sesiones, aplicaciones multilenguaje y creación de apis para una aplicación, lo que acelera las tareas tediosas en PHP.
- Presenta helpers, como el de uso de formularios o el de validación de formularios, muy bien desarrollados.

Codeigniter, hace uso del estilo de arquitectura MVC, de archivos de configuración con plantilla, basando gran parte de su potencia en que es el propio desarrollador el que carga o usa lo que realmente necesita y cuando lo necesita. Es uno de los frameworks más fácil de usar, partiendo desde cero. (5)

Marco de trabajo o Frameworks seleccionado Codeigniter

Se ha decidido emplear Codeigniter como entorno de trabajo para el desarrollo futuro de la aplicación que se propone por ser de código fuente abierto, cuya licencia permite la modificación y distribución del sistema. Posee una de las mayores comunidades on-line, muy activa y una gran implantación en Internet, pudiendo encontrarse multitud de tutoriales y soluciones a problemas comunes con otros frameworks. Además de su pequeña curva de aprendizaje, se destaca por su bajo uso de recursos, un rendimiento excepcional, y su alta compatibilidad con varias versiones y configuraciones de PHP.

1.8. Conclusiones

En este capítulo se abordan los conceptos fundamentales sobre la temática tratada y sobre investigaciones tributantes a la misma, dejando fundamentado el tema.

Se han expuesto las herramientas y metodologías necesarias para el análisis y desarrollo de los demás capítulos. Se han seleccionado y justificado las herramientas a tener en cuenta en el desarrollo del software propuesto.

CAPÍTULO 2: MODELO PARA DISEÑAR MAPAS CONCEPTUALES INTELIGENTES UTILIZANDO UN SISTEMA BASADO EN REGLAS.

2.1. Introducción

En este capítulo se presenta un modelo que facilita la elaboración de Mapas Conceptuales Inteligente (MCI) a partir de las componentes fundamentales de los STI, integrando el paradigma de los SBR y los MC. Se describen los algoritmos que conforman el modelo.

2.2. Mapas Conceptuales Inteligentes

Para elaborar un MCI, solo se requiere elaborar un STI utilizando un MC, donde el MC se encarga de representar los conocimientos de manera gráfica y el STI de facilitarle al alumno una navegación adaptada a sus características cognitivas, para ello utiliza el SBR.

Todo MCI está arquitectónicamente conformado por los tres módulos principales de los STI. El módulo del estudiante y el modelado del estudiante se implementan a través de un Sistema Basado en Reglas. 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 Reglas.

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

- La primera fase consiste en la edición del MC utilizando CMapTools que permite editar los MC al formato HTML.

Esta fase implica también la confección de un conjunto de cuestionarios relacionados con la materia definida en el módulo dominio del MCI, que permitan captar el estado cognitivo del estudiante. Los materiales didácticos que se elaboran para incluir en los nodos del MCI 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 SBR para definir los accesos de navegabilidad por el MC, se obtiene una navegación adaptada de acuerdo a los conocimientos cognitivos del estudiante.

2.3. Características del modelo para elaborar MCI usando un Sistema Basado en Reglas

El conjunto de reglas que componen la base de conocimiento representan el estado del conocimiento y comportamiento del estudiante, así como el entrenador o material didáctico más adecuado. Cada inferencia (materiales didácticos más adecuados para ese modelo de estudiante) obtenida a partir de los datos iniciales (modelo del estudiante) es un ejemplo de modelado de estudiante. Dado un nuevo estudiante se diagnostica usando el paradigma de los SBR los entrenadores sugeridos para el mismo, adaptados a sus conocimientos y comportamientos.

La evaluación del cuestionario estará dada de acuerdo a las definiciones que el profesor realiza al hacer el cuestionario. Los resultados se llevan a combinaciones binarias y luego se convierten en regla. Luego se especifica con más detalles en el Módulo de Adquisición del Conocimiento.

Como se muestra en la figura 2.1, un SBR consta de tres módulos.

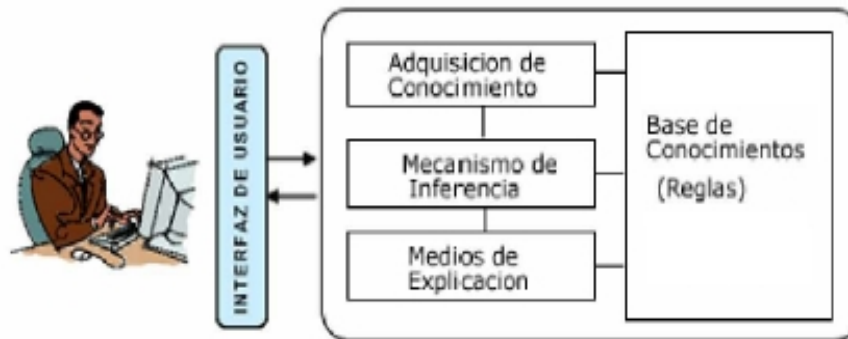


Figura 2.1: Componentes funcionales del sistema.

Algoritmo para el Mecanismo de Inferencia

El mecanismo de inferencia propuesto está basado en reglas con un algoritmo orientado a un objetivo, que permiten ubicar un objetivo (en este caso el área a la que tiene acceso navegar el estudiante), entonces el algoritmo navega a través de las reglas en búsqueda de una conclusión para el objetivo seleccionado. Una vez llegada a la conclusión tiene opciones para seguir navegando por el MC, entonces el algoritmo fuerza preguntar al usuario en busca de nueva información para encontrar otro objetivo, y que el estudiante pueda seguir venciendo las deficiencias del tema.

Módulo de Adquisición del Conocimiento

Le sirve al experto para que pueda construir la base de conocimiento de una forma sencilla, así como disponer de una herramienta de ayuda para actualizar de la base de conocimiento cuando sea necesario. El módulo de adquisición del conocimiento, forman el esqueleto o sistema esencial, y que, separadas de las bases de conocimiento y de hechos, constituyen una herramienta software para el desarrollo de los sistemas basados en el conocimiento. (14)

Para ello se realiza el cuestionario, pues a través de este, se captan los conocimientos que necesita el tutor para confeccionar la base de de conocimientos, conformada por base de hechos y base de reglas.

No.	Variables necesarias	Clasificación
1	Nivel de complejidad de la pregunta	Bajo medio, alta
2	Cantidad de preguntas	[1-...]
3	Número de la pregunta	[1-...]

Tabla 2.1: Variables necesarias para el proceso de adaptación inteligente.

Conjunto de pasos:

1. Realizar el cuestionario.

1.1. Asignarle a cada pregunta el nivel de complejidad al que pertenece.

1.2. Hallar todas las combinaciones de las posibles respuestas de las preguntas por nivel de complejidad de la pregunta en números binarios.

Donde, para hallar las combinaciones denotadas por (C) se tiene que:

➤ Respuesta [0, 1].

Se puede llegar al número de combinaciones de dos formas:

C= combinaciones denotadas.

N= cantidad de preguntas.

P= cantidad de respuestas posibles.

- $C = N^P$

1.3. Transformar las combinaciones del lenguaje binario al natural (formación el antecedente de cada una de reglas) incluyendo en la transformación la presencia de operadores.

1.3.1 Para formar el antecedente de las reglas.

Donde:

- B1, B2, B3,..., Bn. Donde las (B) van a ser los identificadores de las preguntas que tienen nivel de complejidad bajo
- M1, M2, M3,..., Mn. Donde las (M) van a ser los identificadores de las preguntas que tienen nivel de complejidad medio
- A1, A2, A3,..., An. Donde las (A) van a ser los identificadores de las preguntas que tienen nivel de complejidad alto

1.3.2 Para formar el consecuente de las reglas.

Asignarle a cada combinación (cada consecuente), el nivel de navegabilidad al que pertenece en el mapa conceptual.

- Consecuente = nivel de complejidad bajo, nivel de complejidad medio, nivel de complejidad alto.

Luego de haberle asignado los consecuentes a todos los antecedentes, entonces se conforma la Base de Reglas.

1.4. Se insertan cada una de las reglas en la base de reglas y se guardan en la base de datos la base de hechos y la base de reglas.

2. Finaliza así la composición de la base de conocimiento.

Motor de Inferencia o Mecanismos de inferencia

El motor de inferencia es el encargado de supervisar cómo y cuándo puede ser usado el conocimiento que se tiene.

Mecanismo de inferencia:

- Encadenamiento hacia delante o basado en datos.

Se utiliza este mecanismo de inferencia ya que parte de la observación de hechos en las variables de entrada para, mediante el encadenamiento de reglas, alcanzar un hecho de salida deseado.

Proceso de Encadenamiento hacia delante o basado en datos:

1. Se define el hecho a alcanzar, es decir, las variables de salida del sistema cuyo valor se desea inferir.
2. Un conjunto de hechos relativo a las variables de entrada es observado por el sistema, es decir, llega una entrada al sistema.
3. Se busca el subconjunto de reglas de la base de conocimiento cuyos antecedentes son satisfechos por los hechos observados.

4. Si el subconjunto está vacío, se finaliza el proceso. En otro caso se continúa en 5.
 5. El subconjunto de reglas seleccionado se activa y da lugar a un número de hechos nuevos igual al tamaño del subconjunto.
 6. La base de hechos se actualiza con los nuevos hechos.
 7. Si se ha alcanzado el hecho de salida deseado se finaliza el proceso, en otro caso, se vuelve a 3. (5)
- El encadenamiento hacia delante es típico de sistemas en los cuales se desea conocer el valor de una variable de salida atendiendo a una serie de valores de entrada.

Módulo de Explicación (opcional)

Es una utilidad importante en la etapa de desarrollo ya que aporta una ayuda considerable al ingeniero del conocimiento para refinar el funcionamiento del motor de inferencia, y al experto a la hora de construir y verificar la coherencia de la base de conocimiento. Sirve para explicar al usuario tanto las reglas usadas como el conocimiento aplicado en la resolución de un determinado problema. (15)

2.4. Conclusiones

En este capítulo se analizaron los métodos y pasos necesarios para la gestión del acceso del estudiante a navegar por el MC, luego de respondido el cuestionario que se le presenta una vez seleccionado el MC. Se analizó el mecanismo de funcionamiento del sistema experto de la aplicación propuesta, el SBR.

CAPÍTULO 3: REALIZACIÓN COMPUTACIONAL DEL MODELO DE DESARROLLO.

3.1. Introducción

En este capítulo se recogen los elementos necesarios para la solución de la aplicación de software propuesta.

La comprensión del problema planteado será la base para identificar el negocio y para capturar los requisitos funcionales y no funcionales que tendrá la aplicación. A partir de los últimos mencionados se seleccionan los actores y casos de uso del sistema.

3.2. Propuesta de solución

Para dar solución al problema planteado, se propone realizar el análisis y diseño de una herramienta para elaborar un MCI a través de un SBR, que facilite los procesos de enseñanza- aprendizaje de los estudiantes de cualquier centro educativo e inclusive de cualquier entidad que requiera este tipo de enseñanza personalizada. Donde cualquier usuario (como profesor), sin conocimientos avanzados de informática puede crear e interactuar fácilmente con el MCI, facilitando a los estudiantes una herramienta capaz de adaptarse a sus conocimientos cognitivos.

La herramienta contará con tres usuarios (estudiante, profesor y administrador) para lograr su funcionamiento. Los administradores y profesores podrán cambiar su perfil de usuario, donde podrán plasmar sus datos personales incluyendo los datos de contacto. Y teniendo en cuenta las características con que debe contar la herramienta, se tiene que los departamentos contendrán una serie de asignaturas y estas a su vez, contendrán los MCI.

- Los administradores podrán controlar todo el sistema atendiendo a seguridad y mantenimiento, así como la gestión de los usuarios de la aplicación.
- Los profesores estarán encargados de realizar los MC, los cuestionarios y la verificación de las estadísticas de aceptación de los estudiantes por cada MCI.
 - Los MC podrán insertarse en la aplicación y este se realizará con la herramienta externa Cmaptools. Estos en sus nodos tendrán documentación y cuestionarios, donde la documentación es para que el estudiante pueda documentarse del tema fomentando así, sus conocimientos, y el cuestionario es para que una vez vencido el tema en el que se encuentra, pueda acceder a otros niveles del MC.

- Los cuestionarios (iniciales y específicos de los nodos), se realizaran en la aplicación. Se hará uso de estos para definir la navegabilidad o acceso a los distintos niveles (nivel bajo, nivel medio y nivel alto) del MC, y para ello se utiliza la técnica de IA, SBR. A través parámetros definidos por el profesor en la creación del cuestionario (nivel de complejidad de cada pregunta), y la respuesta del estudiante, se define el nivel de acceso a los nodos del MC.
- El estudiante, podrá seleccionar el departamento que desee, y de la misma manera, el MCI dentro del departamento. Una vez que seleccionado el MCI, deberá realizar un cuestionario, luego podrá acceder a navegar por la parte del MCI a la que tiene acceso, definida a través del resultado obtenido en el cuestionario realizado. Se le dejará habilitada solamente, la parte del mapa por la que puede navegar, para garantizar que solo acceda a la información que necesite conocer del tema, aunque podrá ver todo el MC. Podrá cambiar de nivel, una vez vencido todo el contenido del que se encuentra y vencer el cuestionario inicial.

A continuación se presenta la figura 3.1, donde se muestra la propuesta de solución.



Figura 3.1: Propuesta de solución.

3.3. Modelo de Dominio

En la construcción de una aplicación el primer paso es desarrollar el modelo de negocio o de dominio, identificando el proceso y las actividades que requieren automatización. Se puede observar que el mismo tiene un bajo nivel de estructuración.

Un Modelo de 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. Este se describe mediante diagramas de UML, en los cuales se muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones. (16)

Se presenta como uno o más diagramas de clases y contiene, no conceptos propios de un sistema sino de la propia realidad física. Pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Son similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender la lógica de negocio a la que el sistema va a servir.

Por tanto, se realiza el modelo de dominio porque este es utilizado cuando no existe un negocio definido y cuando se pretende ampliar el proceso en análisis. Teniendo en cuenta además de lo antes expuesto, que el proceso en desarrollo está altamente centrado en herramientas y tecnologías informáticas, se determina que no es viable llevar a cabo un modelo de negocio. Se hace necesario realizar la descripción de la aplicación mediante un modelo de dominio o modelo de conceptos del dominio, el cual agrupa los conceptos fundamentales de objetos relacionados con el dominio, sus atributos y sus relaciones.

Conceptos más significativos que se utilizarán en el modelo de dominio:

Base de Conocimiento: Es el conjunto de todas las reglas y hechos.

Hechos: Es el conjunto de parámetros que conforman una regla.

Reglas: Condición que se debe cumplir.

Pregunta: Es la pregunta que elabora el profesor en el cuestionario.

Respuesta: Es la respuesta del cuestionario.

Respuesta del Profesor: Es la respuesta que el profesor define como respuesta correcta.

Respuesta del Estudiante: Es la respuesta del estudiante a la pregunta.

Estudiante: Usuario encargado de realizar el cuestionario, navegar por el MC accediendo a la información contenida en los nodos.

Profesor: Usuario con permisos de gestionar el (MC, cuestionarios, material y los departamentos), así como las estadísticas de los cuestionarios realizados por el/los estudiante.

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.

Mapa Conceptual: Mapa insertado por el profesor donde reflejan los conocimientos del tema que desea abordar.

Mapa Conceptual Inteligente: Es el MC pero ya estructurado en su interior con el Cuestionario y los materiales deseados, para que pueda navegarse por el mismo, de manera adaptativa a los conocimientos cognitivos del estudiante.

Material: Representa el material de aprendizaje, ya sea un doc, ppt, avi, etc. O sea, cualquier recurso que el profesor desee incluir para que sea de apoyo en el estudio del estudiante.

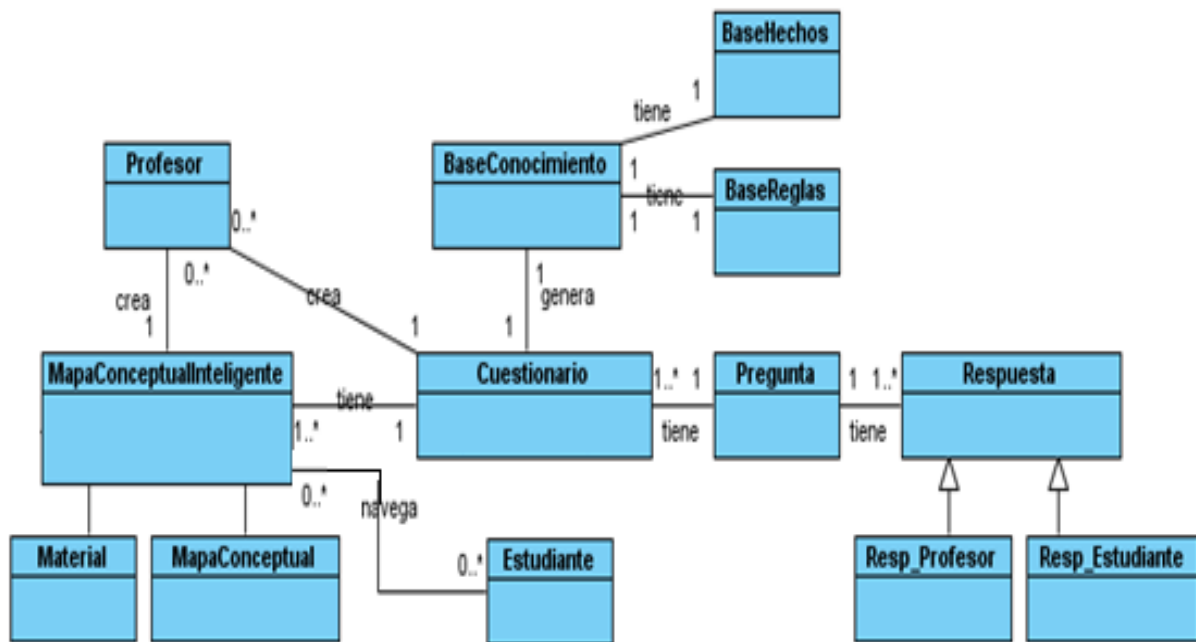


Figura 3.1: Diagrama del Modelo de Dominio.

3.4. Relación de requerimientos

Los requerimientos son las características que va a poseer el producto de software, sintetizados en la definición en detalle de los datos, funciones y el comportamiento del sistema. Los requerimientos deben ser verificados y validados para evitar errores que pueden resultar altamente costosos en una etapa posterior del desarrollo del sistema. Se dividen en dos grupos, requerimientos funcionales y no funcionales.

3.4.1. Requisitos funcionales

Los requisitos funcionales son una descripción de las necesidades de un producto y establecen los comportamientos del sistema. Son capacidades o condiciones que el sistema debe cumplir.

En el sistema a diseñar los requisitos funcionales que se identificaron fueron los siguientes:

Nombre:	RF1- Autenticar Usuario.
Requisito funcional que debe permitir la autenticación del usuario en el sistema.	

Nombre:	RF2- Gestionar Usuario.
Requisito funcional que debe permitir al administrador, la gestión de los usuarios de la aplicación, brinda y distribuye la cadena de mando dentro de la aplicación y proporciona una mayor organización y estructuración de la misma.	

RF 2.1- Insertar Usuario.

RF 2.2- Modificar Usuario.

RF 2.3- Eliminar Usuario.

RF 2.4- Buscar Usuario.

Nombre:	RF3- Gestionar Departamento.
Requisito funcional que debe permitir al administrador, la gestión de los usuarios de la aplicación, brinda y distribuye la cadena de mando dentro de la aplicación y proporciona una mayor organización y estructuración de la misma.	

RF 3.1- Insertar Departamento.

RF 3.2- Modificar Departamento.

RF 3.3- Eliminar Departamento.

RF 3.4- Buscar Departamento.

Nombre:	RF4- Gestionar Cuestionario.
Requisito funcional que debe permitir al profesor, la gestión de los cuestionarios en la aplicación, brinda y la posibilidad de interactuar con los mismos dentro de la aplicación.	

RF 4.1- Crear Cuestionario.

RF 4.2- Modificar Cuestionario.

RF 4.3- Eliminar Cuestionario.

RF 4.4- Buscar Cuestionario.

Nombre:	RF5- Gestionar Mapa Conceptual Inteligente (MCI).
Requisito funcional que debe permitir al profesor, la gestión de los MCI de la aplicación, brinda y distribuye las posibilidades de operar con los mismos dentro de la aplicación.	

RF 4.1- Crear Mapa Conceptual Inteligente.

RF 4.2- Modificar Mapa Conceptual Inteligente.

RF 4.3- Eliminar Mapa Conceptual Inteligente.

RF 4.4- Buscar Mapa Conceptual Inteligente.

Nombre:	RF6- Consultar Material.
Requisito funcional que debe permitir al estudiante, consultar los materiales que se encuentran en cada uno de los nodos del MCI.	

Nombre:	RF7- Mostrar Estadísticas de Cuestionarios.
Requisito funcional que debe permitir al profesor, mostrar las estadísticas por niveles (nivel bajo, nivel medio y nivel alto), de acuerdo a los resultados obtenidos en el cuestionario por cada uno de los MCI de su autoría.	

Nombre:	RF8- Modificar Perfil de Usuario.
Requisito funcional que debe permitir al profesor y al administrador, modificar sus datos de usuario.	

Nombre:	RF9- Responder Cuestionario.
----------------	-------------------------------------

Requisito funcional que debe permitir al estudiante, responder el cuestionario para poder navegar por el MCI seleccionado.

3.4.2. Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades constituyen las características que hacen al producto atractivo, usable, rápido y confiable. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

RNF 1. Restricciones en el diseño y la implementación:

- Usar el lenguaje de programación PHP 5.2.8.
- Usar el gestor de Base de Datos PostgreSQL 8.3.5.
- Utilizar como metodología de desarrollo RUP.
- El modelado UML se realizará con Visual Paradigm 3.4.
- Utilizar como servidor web Apache 2.3.19.
- Desarrollado bajo el Framework de desarrollo, Codeigniter.

RNF 2. Software:

- Plataforma de funcionamiento GNU/Linux y Windows.
- Herramienta CmapTools instalado.
- Usar un navegador web como: Mozilla Firefox, internet Explorer, Google Chrome, etc. Debe estar habilitada en el navegador la opción JavaScript.

RNF 3. Hardware:

- PC Pentium II o superior, con mínimo de 512 MB de RAM.
- Se requiere de un servidor Web y de Base de Datos, con mínimo de 512 MB de RAM y 20 MB de espacio libre en disco duro.
- Tanto la PC servidor como las clientes deben de estar conectadas a la red de al menos 100 Mbps de velocidad.

RNF 4. Apariencia o interfaz externa:

La herramienta contará con:

- Un diseño de interfaz de usuario sencilla.
- Sin muchas imágenes, animaciones ni contrastes entre colores, para facilitarle una vista agradable al usuario.
- Hará uso de banners discretos y con la información referente al nombre de la herramienta.

RNF 5. Usabilidad:

- El sistema podrá ser usado por cualquier persona con conocimientos básicos en el manejo de la computadora.

RNF 6. Rendimiento:

- Debe ser rápido en la navegabilidad, eficiente tanto en los tiempos de respuesta como en la velocidad de procesamiento, permitiendo un rápido acceso a la información y de la manera más sencilla posible.

RNF 7. Soporte:

- El sistema será probado, instalado y configurado por un profesor y un Administrador, el cual se ocupará también de su mantenimiento.
- El producto requiriere de constantes actualizaciones debido a los cambios que pueden ocurrir respecto a los usuarios, estudiantes y profesores del centro.

RNF 8. Portabilidad:

- Este producto podrá ser utilizado en las plataformas GNU/Linux y Windows.

RNF 9. Seguridad:

- Debe permitir identificar al usuario antes de que pueda realizar cualquier acción en la aplicación y además, debe permitir que los usuarios solo tengan acceso a las operaciones de acuerdo a los permisos que tienen.

Confidencialidad: La información almacenada, procesada y generada por el sistema será protegida de acceso no autorizado, ya que será requerida la autenticación de los usuarios para garantizar que solo el personal autorizado pueda acceder a estos datos.

Integridad: La información privada de cada usuario serán objeto de cuidadosa protección contra la corrupción y estados de inconsistencia.

Disponibilidad: La aplicación estará disponible en todo momento para aquellas personas con acceso a la información.

RNF 10. Políticos-culturales:

- El sistema estará completamente identificado con los principales conceptos del campo de trabajo, para que el usuario lo asocie a sus situaciones reales. El mismo responderá a los principios e intereses del área en el que se encuentre.

RNF 11. Ayuda y documentación en línea:

- El sistema contará con una ayuda que indica qué operación realiza cada componente, además de mostrarle la forma de trabajo con la herramienta.

3.5. Modelo del Sistema

Aquí se especifica toda la parte referente al sistema. Se definen los usuarios del mismo y se detalla en el diagrama de Casos de Uso del Sistema (CUS), las especificaciones que va a posibilitar el software.

Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones. El modelo de casos de uso permite que los desarrolladores y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema (16). El modelo de casos de uso describe lo que hace el sistema para cada tipo de usuario.

3.5.1. Definición de los actores del sistema

Un actor del sistema es aquella persona o sistema que interactúa con el sistema.

Tabla 3.1: Tabla de actores del sistema.

Actores	Descripción
Usuario	Usuario que accede al sistema y según su rol tiene privilegios administrativos, de profesor o los simples privilegios del estudiante.
Estudiante	Usuario que accede al sistema pero solo tiene acceso a ver los MC, realizar los cuestionarios para navegar por el MC, y ver la documentación contenida en los nodos.

Profesor	Usuario que accede al sistema para gestionar los MC, los cuestionarios y la documentación que contendrán los nodos del MC. Así como controlar los resultados de los cuestionarios.
Administrador	Usuario encargada de administrar el sistema y de gestionar los usuarios.
Usuario Privilegiado	Usuario que accede al sistema como usuario (profesor o administrador).

3.5.2. Diagrama de Casos de Uso del sistema a automatizar

Un diagrama de CUS es la representación gráficamente de los procesos y su interacción con los actores. Un Caso de Uso (CU), es la especificación del comportamiento del sistema y de algo externo (personas, hardware, otros software) que usa sus servicios.

Los CU son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

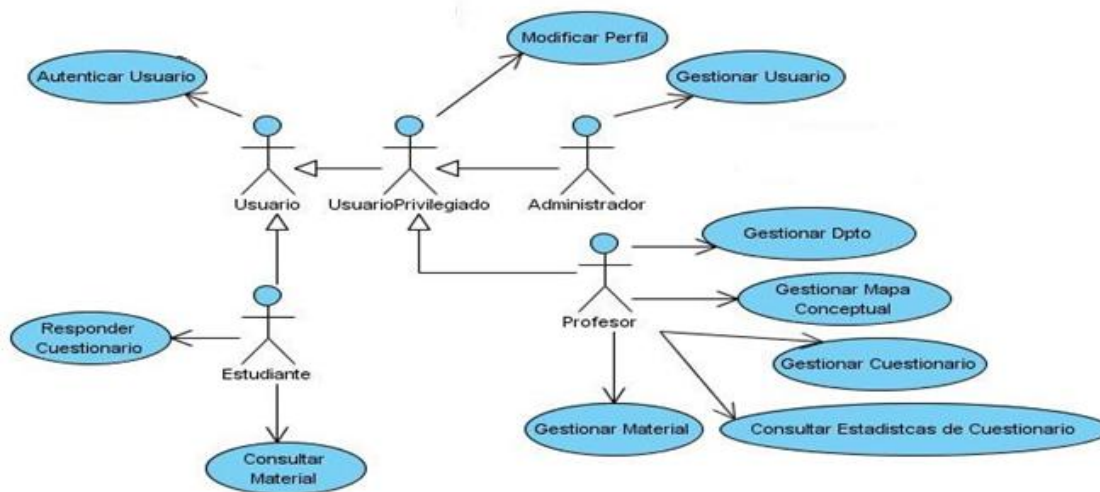


Figura 3.2: Diagrama de Casos de Uso del Sistema.

3.5.3. Patrones de Casos de Uso

Un patrón es una pareja de problema/solución con un nombre, que codifica (estandariza) buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Para desarrollar el presente modelo de casos de uso, se aplicaron patrones que permitieron reflejar con más

precisión los requisitos reales y lograr mejores resultados de forma más rápida, haciendo más fácil el trabajo, optimizando con su aplicación los problemas que se presentaron en la modelación del sistema, los mismos se muestran a continuación:

CRUD (Crear, Obtener, Actualizar, Eliminar): Este patrón se utiliza para fusionar casos de uso simples, formando una unidad conceptual y presenta dos especificaciones, pero solo se utilizó:

CRUD Parcial: para modelar una parte de las operaciones que pueden ser realizadas sobre una parte de la información de un tipo determinado; tales como (creación, modificación eliminación).

Múltiples actores (Roles comunes): Este patrón se utiliza para representar actores que interpretan un mismo papel en determinado caso de uso. Este rol es representado por otro actor, que contiene de forma hereditaria los actores que comparten este rol.

3.6. Descripciones Textuales de los Casos de Uso del sistema

En este apartado se especifican las descripciones textuales de los CU, dejando definido toda la información de los CU del sistema, para una mejor comprensión de la del mismo.

Aquí se encuentra la descripción textual del CU: “Gestionar Mapa Conceptual Inteligente”, las restantes se encuentran en el [Anexo A](#).

Tabla 3.1: Descripción del CU “Gestionar Mapa Conceptual Inteligente”.

Caso de uso	
CU- 4	Gestionar Mapa Conceptual Inteligente.
Propósito	Elaborar un Mapa Conceptual Inteligente que se adapte a los conocimientos cognitivos del estudiante.
Actores: Profesor	
Resumen: El caso de uso se inicia cuando el Profesor selecciona la opción Mapa Conceptual Inteligente, y una vez dentro de este, puede Gestionar MCI (Crear, Eliminar, Modificar, Buscar) un MCI.	
Referencias	RF4
Prioridad	alta
Acción del actor	Respuesta del sistema
1. El Profesor selecciona la opción Gestionar MCI.	2. El sistema muestra una vista con las opciones como: <ul style="list-style-type: none"> • Crear MCI.

	<ul style="list-style-type: none"> • Eliminar MCI. • Modificar MCI. • Buscar MCI.
<p>3. El Profesor selecciona la opción deseada. En caso de:</p> <p>a) Crear Mapa Conceptual Inteligente. Ir a la Sección Crear MCI.</p> <p>b) Modificar Mapa conceptual. Ir a la Sección Modificar MCI.</p> <p>c) Eliminar MCI. Ir a la Sección Eliminar MCI.</p> <p>d) Buscar Mapa Conceptual. Ir a la Sección Modificar MCI.</p>	
Prototipo de Interfaz	
SC 1.1 Crear Mapa Conceptual Inteligente.	
<p>3. El Profesor selecciona la opción Crear MCI.</p>	<p>4. El sistema muestra una ventana de trabajo con herramientas para crear un MCI, y los botones: Crear y Cancelar.</p>
<p>5. El Profesor realiza el MCI y presiona el botón Aceptar.</p>	<p>6. El sistema guarda el MCI y muestra el mensaje “El Mapa Conceptual Inteligente se ha creado satisfactoriamente”.</p>
Flujo alternativo	
Acción del actor	Respuesta del sistema

5.1. El Profesor realiza el MC y presiona el botón Cancelar.

5.2. El sistema cancela la operación y regresa a la vista anterior.

Prototipo de Interfaz

CREAR MAPA CONCEPTUAL

 Guardar Cancelar

▶ Area del Mapa Conceptual

▶ Información de Nodos

▶ Mapa Conceptual

Grupos de Nodos

Complejidad Nodo

Nodos Complejidad Baja **Nodos Complejidad Media** **Nodos Complejidad Alta**

▼ Información de Nodos

Cuestionario de Navegación

Tipo de Selección:

Cuestionario:

Mapa Conceptual

Mapa Conceptual:

Información de Nodo

Nodo: Complejidad:

Documentación:

Tabla de Nodos

Nombre de Nodo	No. de Documentos
Nodo 1	00
Nodo 2	00
Nodo 3	00
Nodo 4	00

<< 1 2 3 ... >>

SC 1.2 Modificar Mapa Conceptual Inteligente.	
3. El Profesor selecciona la opción Modificar MCI.	4. El sistema muestra una ventana de trabajo con herramientas para crear un MCI, y los botones: Crear y Cancelar.
5. El Profesor modifica el MCI y presiona el botón Aceptar.	6. El sistema guarda los datos modificados y muestra el mensaje “El Mapa Conceptual Inteligente se ha modificado satisfactoriamente”.
Flujo alternativo	
Acción del actor	Respuesta del sistema
5.1. El Profesor modifica el MCI y presiona el botón Cancelar.	5.2. El sistema cancela la operación y regresa a la vista anterior.
SC 1.3 Eliminar Mapa Conceptual Inteligente.	
3. El Profesor selecciona la opción Eliminar MCI.	4. El sistema muestra una ventana de trabajo con herramientas para crear un MCI, y los botones: Crear y Cancelar.
5. El Profesor elimina el MCI y presiona el botón Aceptar.	6. El sistema guarda los datos modificados y muestra el mensaje “El Mapa Conceptual Inteligente se ha modificado satisfactoriamente”.
Flujo alternativo	
Acción del actor	Respuesta del sistema
5.1. El Profesor elimina el MCI y presiona el botón Cancelar.	5.2. El sistema cancela la operación y regresa a la vista anterior.
Puntos de extensión.	
(11)	

3.7. Técnica de validación de requisitos

Los requisitos cuando se definen necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente.

Existen algunas técnicas que pueden aplicarse para ello:

- Revisiones de Requisitos.
- Generación de Casos de Prueba.
- Construcción de Prototipos.

3.7.1. Técnica de validación empleada en el desarrollo del sistema

Después de haber realizado una investigación de las técnicas de validación de requisitos existentes para la aprobación de los requisitos funcionales identificados para el desarrollo de un sistema informático, se escogieron las siguientes técnicas:

Construcción de prototipos: fue desarrollada por el equipo de análisis de la línea de desarrollo, donde después de haber identificado y especificado los requisitos de software, se realizan los prototipos de interfaz de usuario, el cual permite tener una visión de cómo debe de quedar el sistema.

3.8. Conclusiones del Capítulo

En el presente capítulo se analizaron las características fundamentales con las que va a contar el sistema, además de un esquema detallado con todo el flujo a realizar. A partir del análisis de los procesos de negocio se obtuvo un listado con los requerimientos funcionales y no funcionales para el desarrollo del sistema, que se representaron en el diagrama de CUS y finalmente se realizaron las descripciones textuales de los mismos.

CAPÍTULO 4: ANÁLISIS Y DISEÑO DEL SISTEMA PROPUESTO.

4.1. Introducción

La metodología seleccionada para el desarrollo de la herramienta propuesta, documenta muy ampliamente los detalles de ambas fases, la identificación de las clases que describen la realización de los CU, los atributos y las relaciones entre ellas. Es necesario destacar que el análisis y el diseño, es imprescindible a la hora de desarrollar un sistema teniendo en cuenta cómo será implementado el software a partir de las funcionalidades previstas y las restricciones impuestas.

4.2. Análisis del sistema

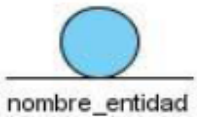
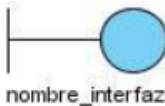
Definición del modelo de análisis. Modelo de clases del análisis.


El análisis se centra en comprender perfectamente los requisitos del software y no a precisar cómo se implementará la solución. Define realizaciones de casos de uso, y cada una de ellas representa el análisis de un CU del modelo de casos de uso. Durante esta fase, se analizan los requisitos que fueron descritos en la captura de requisitos, refinándolos y estructurándolos. No se tiene en cuenta el lenguaje de programación a usar en la construcción de la aplicación, los componentes prefabricados o reusables de otras aplicaciones, entre otras características que afectan al sistema.

4.2.1. Diagramas de clases del análisis

Los diagramas de clases de análisis, expresan la definición y relación entre las clases, están compuestos por clases que pueden ser de tres tipos fundamentales: interfaz, controladora y entidad.

Tabla 4.1: Estereotipos y su descripción.

Estereotipo	Descripción
	Modelan la interacción entre el sistema y sus actores.
	Modelan información que posee larga vida y que es a menudo persistente.

 nombre_control	Coordinan la realización de uno o unos pocos Casos de Uso coordinando las actividades de los objetos que implementan su funcionalidad.
-----------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------

Seguidamente se muestra en la Figura 4.1, el diagrama de clases del análisis del CU: Gestionar Mapa Conceptual Inteligente, los restantes diagramas se encuentran en el [Anexo B](#).

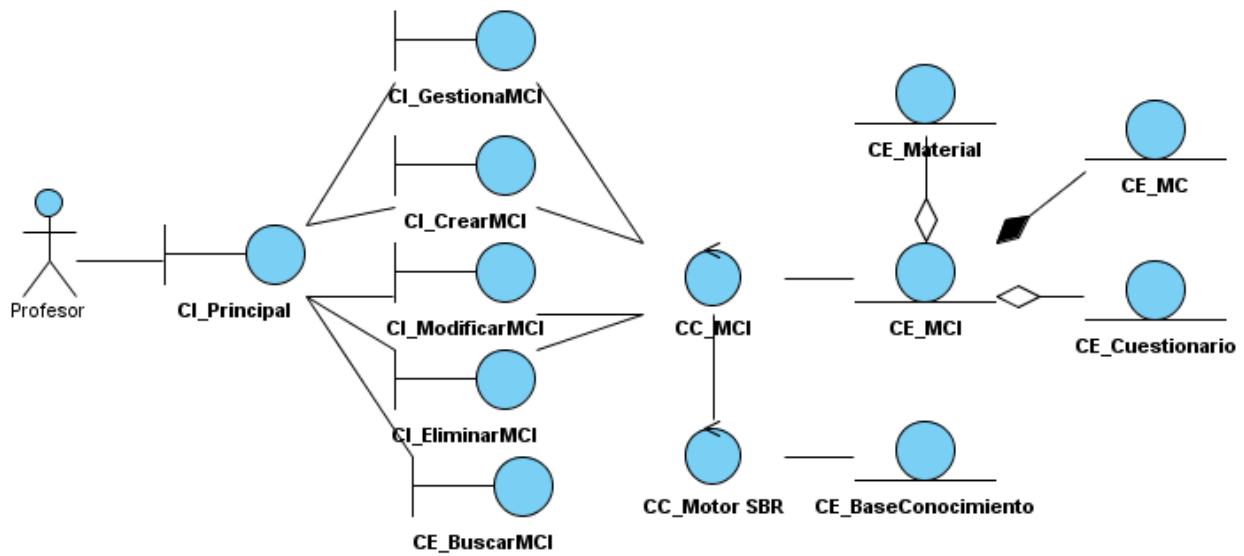


Figura 4.1: Diagrama de las clases del análisis, CU "Gestionar Mapa Conceptual Inteligente".

4.3. Diseño del sistema

El modelo de diseño es un modelo de objetos que describe la realización de los casos de uso, y se utiliza como abstracción del modelo de implementación y el código fuente. Es usado como una entrada inicial en las actividades de implementación y prueba.

La Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del Software.

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.

Para la realización de la aplicación se selecciona el framework de PHP CodeIgniter, el cual está basado en el patrón Modelo Vista Controlador además de nutrir su arquitectura con otros patrones como Active Record y Front Controller, los mismos se describen a continuación.

4.3.1. Patrón Arquitectónico Modelo Vista Controlador (MVC)

El Modelo-Vista-Controlador (Model-View-Controller), como se muestra en la figura 4.2, es un patrón de desarrollo que separa la parte lógica de una aplicación de su presentación. Básicamente sirve para separar el lenguaje de programación del HTML lo máximo posible y para poder reutilizar componentes fácilmente.

El Modelo: representa las estructuras de datos, la lógica de la aplicación, que se encuentra inmersa en los datos. Las clases del modelo contendrán funciones que ayudarán a recuperar, insertar y actualizar información en la base de datos.

La Vista: Hace referencia a la interfaz de usuario, a la presentación de la información. Normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un pie de página.

El Controlador: sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web. CodeIgniter tiene un enfoque bastante flexible del MVC, ya que los Modelos no son requeridos. (17)

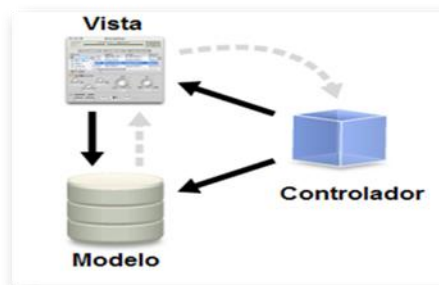


Figura 4.2: Patrón arquitectónico Modelo-Vista-Controlador.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

Características:

- Clara separación entre interfaz, lógica de negocio y de presentación, que además provoca parte de las ventajas siguientes.
- Sencillez para crear distintas representaciones de los mismos datos.
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables.

4.3.2. Patrones de diseño

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio, y aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software, son denominados, patrones de diseño.

- Proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo.
- Están basados en la recopilación del conocimiento de los expertos en desarrollo de software.

4.3.2.1. Patrón Active Record

CodeIgniter viene con una clase de base de datos llena de características la que posibilita un rápido acceso a la base de datos gracias a su sintaxis simple y clara. Soporta tanto las estructuras tradicionales así como el patrón Active Record.

El patrón Active Record es un patrón de diseño. Es un enfoque al problema de acceder a los datos de una base de datos. Donde una fila en la tabla de la base de datos se envuelve en una clase, de manera que se asocian filas únicas de la base de datos con objetos del lenguaje de programación usado. Cuando se crea uno de estos objetos, se añade una fila a la tabla de la base de datos. Cuando se modifican los atributos del objeto, se actualiza la fila de la base de datos.

Gran parte de este patrón viene de un modelo de dominio y esto significa que las clases están muy cercanas a la representación en la base de datos. Cada Active Record es responsable de sí mismo, tanto en lo relacionado con persistencia como en su lógica de negocio. (18)

CodeIgniter usa una versión modificada del Patrón de Base de Datos Active Record. Este patrón permite obtener, insertar y actualizar información en la base de datos con una mínima codificación. Existen casos en que solo se necesita una o dos líneas de código para realizar una acción de base de datos. CodeIgniter no requiere que cada tabla de la base de datos sea un propio archivo de clase. Se permite una interface más simplificada.

Más allá de la simplicidad, un beneficio mayor de usar la Active Record es que te permite crear una aplicación independiente de la base de datos que usa, ya que la sintaxis de consulta es generada por cada adaptador de base de datos. También permite consultas más seguras, ya que los valores son escapados automáticamente por el sistema. (17)

4.3.2.2. Patrón Front Controller

Es un patrón de diseño que se basa en usar un controlador como punto inicial para la gestión de las peticiones del usuario en una aplicación. El controlador gestiona estas peticiones, y realiza algunas funciones como: comprobación de restricciones de seguridad como la autenticación y autorización, manejo de errores, mapear y delegación de las peticiones a otros componentes de la aplicación que se encargarán de generar la vista adecuada para el usuario, además del control de la selección de estrategias de creación de contenido.

Es exclusivo en cada aplicación. Este realiza tareas comunes a los demás controladores que puedan existir en la aplicación como el manejo de las peticiones del usuario, el manejo de la seguridad y la carga de la configuración de la aplicación. Entre sus principales ventajas se encuentra la de constituir un punto de entrada único para la gestión de peticiones de toda la aplicación, en el caso de CodeIgniter ese controlador frontal es el index.php del framework. Así, en caso de que sea necesario impedir el acceso a la aplicación, solamente es necesario editar el script correspondiente al controlador frontal. Otra de las ventajas son los aumentos de reusabilidad de código que este patrón representa.

4.4. Diagrama de diseño

Un diagrama de clases de diseño muestra la especificación para las clases software de una aplicación. Incluye clases con (asociaciones y atributos), métodos, navegabilidad y dependencias.

Para una mejor organización del sistema en cuestión, se decide agrupar por paquetes las funcionalidades a fines, como muestra en la figura 4,3.

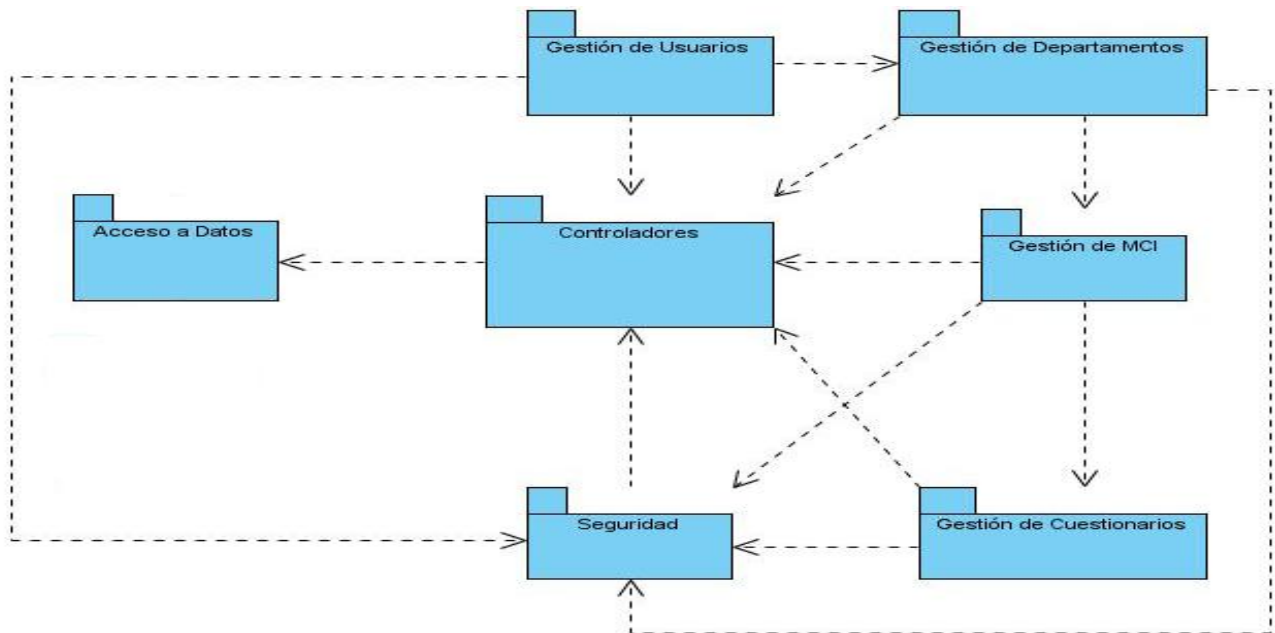


Figura 4.3: Diagrama de Paquetes del Diseño.

El diseño de un sistema está ligado a la arquitectura escogida para su desarrollo, una arquitectura con componentes que cumplen funciones específicas asociados a cada una de ellas. La siguiente figura muestra la arquitectura general del sistema basado en el patrón Modelo Vista Controlador.

En la figura 4.4, se muestra la Vista de la Arquitectura de CodeIgniter.



Figura 4.4: Vista General de la Arquitectura.

1. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter.
2. El Enrutador examina la petición HTTP para determinar la acción a realizar con la misma.
3. Seguridad, antes de que se cargue el controlador se filtra los datos enviados para que estos sean fiables.

4. El controlador comprueba si un archivo de cache existe, es enviado directamente al navegador, sobrepasando el sistema de ejecución normal, si no existe un archivo cache carga el modelo, librerías, plugins, helpers y todos los recursos necesarios para satisfacer la petición. La caché se puede configurar y si lo deseamos, incluso deshabilitar.
5. Una vez la Vista está finalizada es enviada al navegador si la cache está habilitada se almacena el resultado en cache para la próxima vez que esta URL sea servida.

A continuación se muestra en el diagrama 4.1, el principal Diagrama de Clases del Diseño. Los restantes diagramas se encuentran en el [Anexo C](#).

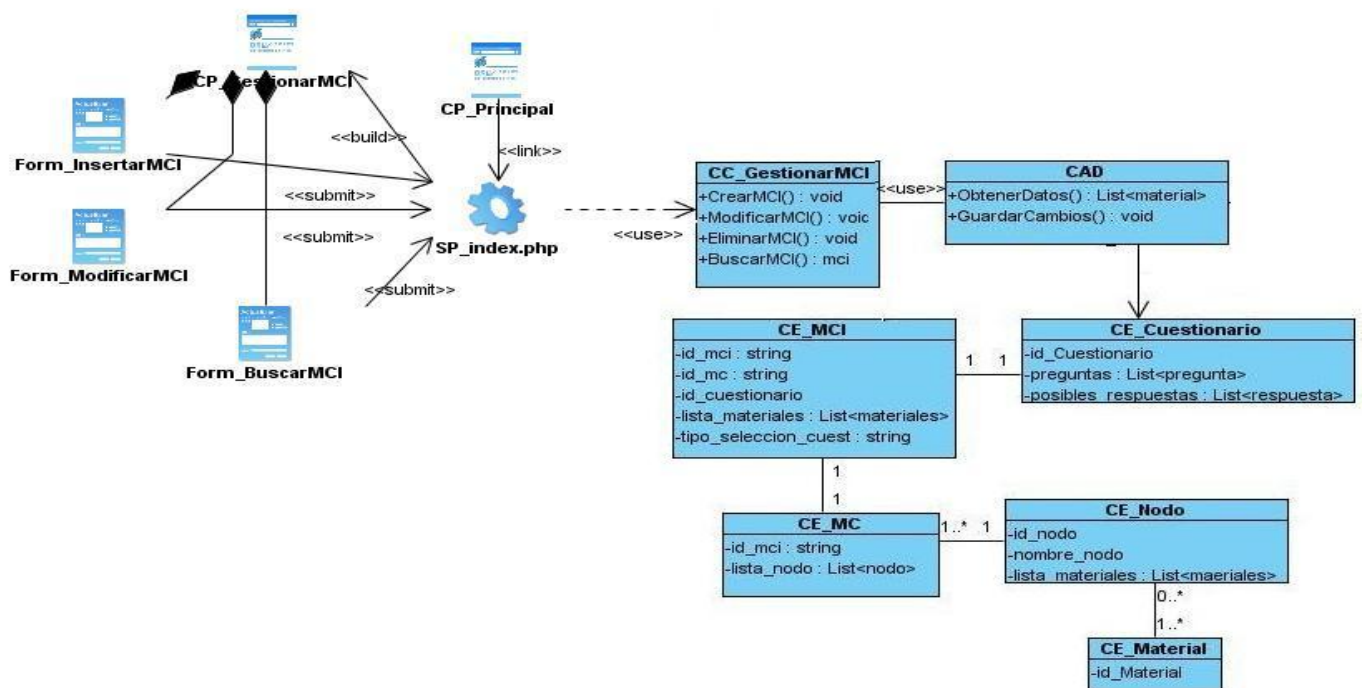


Figura 4.5: CU “Gestionar Mapa Conceptual Inteligente”.

4.4.1. Diagramas de interacción

Los objetos interactúan para realizar colectivamente los servicios ofrecidos por las aplicaciones. Los diagramas de interacción muestran cómo se comunican los objetos en una interacción.

Diagramas de colaboración o de secuencia: Son intercambiables y dependerá de que desees que se vea mejor; la interacción entre objetos o la secuencia temporal de los mensajes. Si tienes que tratar con clases

activas mejor los de colaboración y también te vendrá aquí bien para reflejar determinadas interacciones los de objetos.

Diagramas de secuencia

Los Diagramas de Secuencia son más adecuados para observar la perspectiva cronológica de las interacciones. Muestran la secuencia de mensajes entre objetos durante un escenario concreto.

Por medio de los diagramas de secuencia se proporciona una visión más detallada de la funcionalidad del sistema, partiendo de las descripciones textuales de los distintos casos de uso, atendiendo al flujo producido en la interacción del usuario con el sistema a través de la interfaz.

A continuación se muestra el diagrama de secuencia correspondiente al CU: Gestionar mapa conceptual. Los demás se pondrán en el [Anexo D](#).

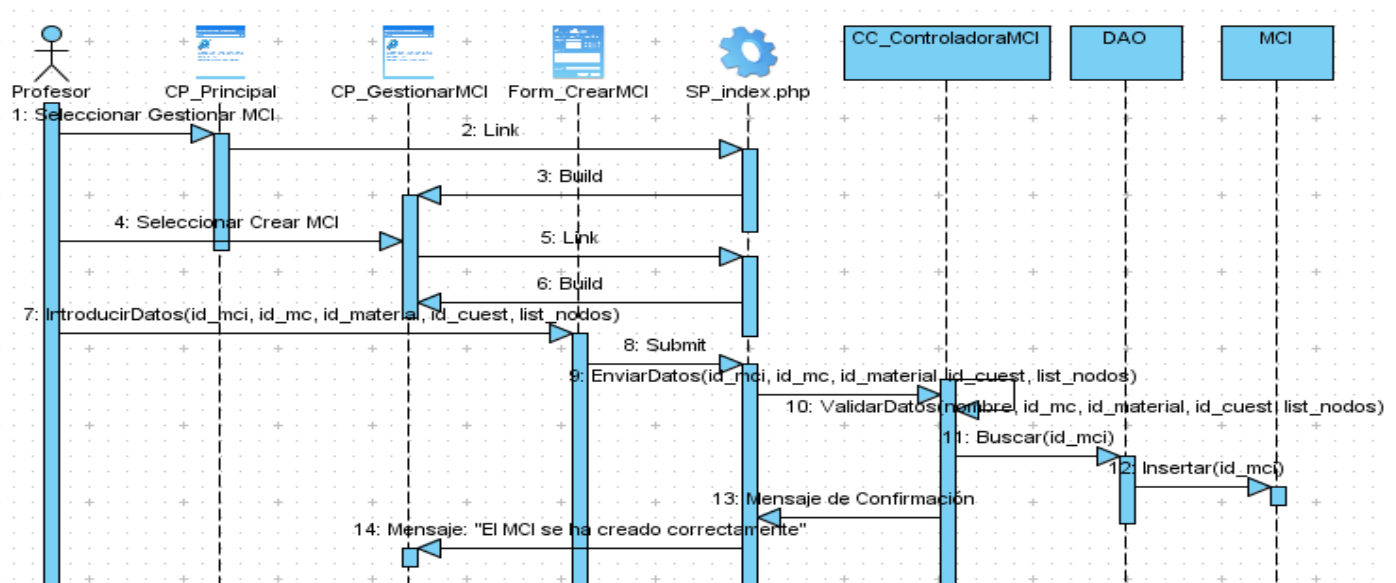


Diagrama 4.6: Crear Mapa Conceptual Inteligente.

4.5. Modelo de Base de Datos

El modelo de datos es el enfoque utilizado para la representación de las entidades y sus características dentro de la Base de Datos (BD). El objetivo fundamental de este modelo es definir las estructuras permitidas y las restricciones, lo cual constituye un elemento básico para el posterior diseño de la base de datos.

4.5.1. Modelo de Datos

Para la construcción del modelo físico de datos reflejado en la figura 4.4, se realizó antes el diagrama entidad relación (modelo lógico de datos). Luego se realizaron las transformaciones necesarias para obtener el modelo relacional, y por último se normalizó parcialmente el esquema obtenido hasta la tercera forma normal. Fue utilizada la herramienta ERStudio.

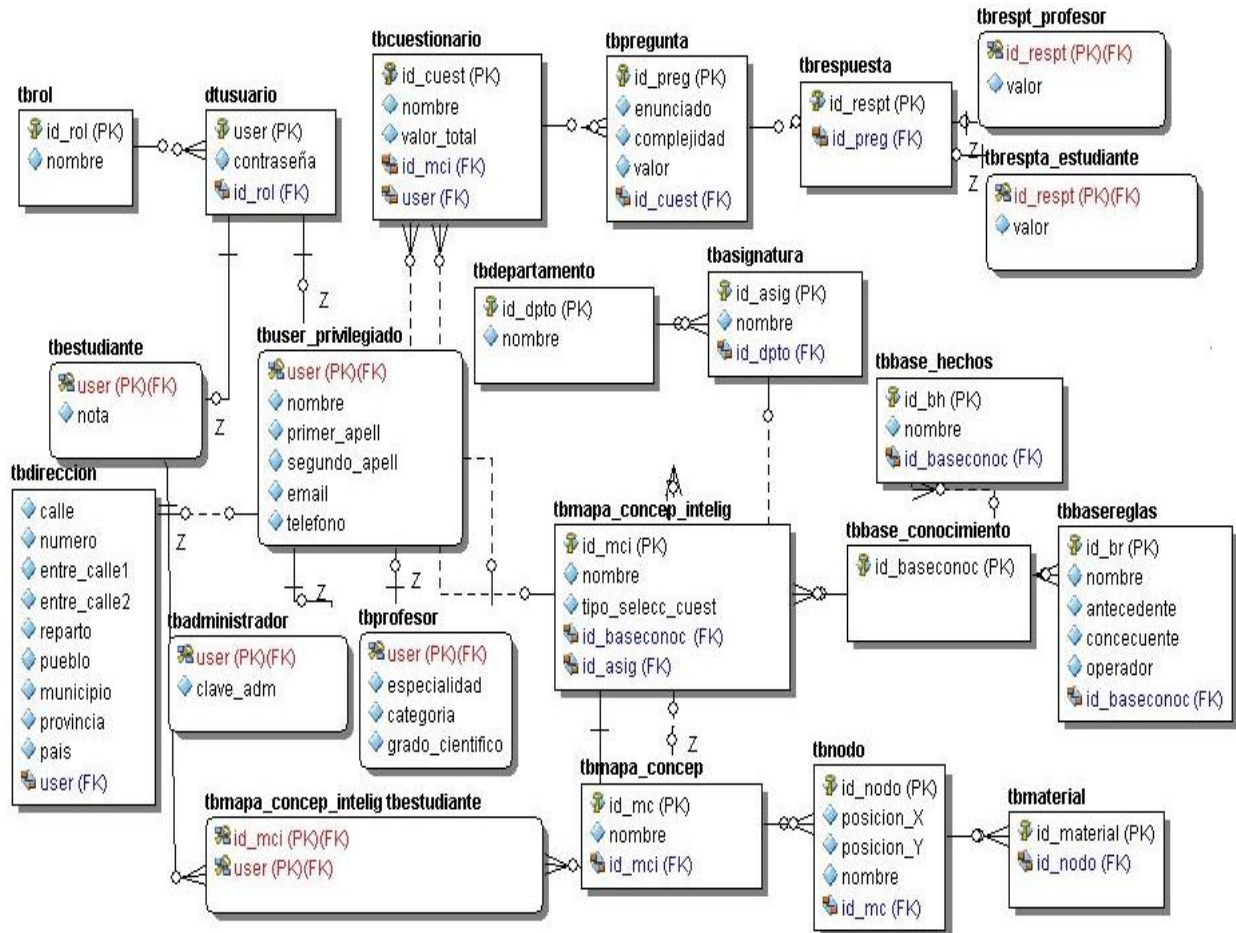


Figura 4.7: Modelo Físico de la Base de Datos.

4.6. Descripción de las tablas del Modelo Físico de la BD

Una descripción de tabla es un análisis detallado de las características de todos los atributos que contienen las tablas. Las descripciones de las tablas de la BD del sistema propuesto, se encuentran en el [Anexo E](#).

4.7. Principios de diseño

Estándares de la interfaz de la aplicación

Es importante mantener una misma plantilla para las páginas y así mantenerlas consistentes y ordenadas. La interfaz a utilizar se compone de los siguientes elementos:

Cabecera: En esta sección se muestra el banner que identifica al sistema.

Descripción de Usuario: en esta sección se encuentran los datos pertenecientes al usuario autenticado.

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 las páginas, de forma que el usuario no tenga que moverse dentro de la aplicación para completar una operación. Por ejemplo, se puede hacer la creación, modificación, eliminación, búsqueda de información y generación de reportes en las páginas donde se muestran listados.

4.8. Tratamiento de errores

El tratamiento de los errores es un paso indispensable para el buen funcionamiento del sistema. Los errores más comunes que se pueden presentar están relacionados con el trabajo con la base de datos, fundamentalmente en los procesos de inserción, eliminación y modificación de datos. Es por esto que se han establecido mecanismos que visualicen la información para evitar en la medida de lo posible la introducción manual de los datos y así minimizar los errores que pueda tener la aplicación. Además se muestran mensajes indicando al usuario los errores cometidos, permitiéndole a la vez corregir su entrada con facilidad y continuar.

4.9. Seguridad

En el mundo, la información constituye un elemento de vital importancia, por lo que es necesario que existan mecanismos de seguridad que garanticen la protección de los datos que se manipulan en la aplicación y que los usuarios según su rol solo puedan realizar solo las acciones que necesiten y por tanto, obtener información únicamente de las tareas para las que tienen autorización.

A partir de esta idea, para mantener la seguridad de la información que se maneja, se tuvo en cuenta para el acceso al sitio, una autenticación previa, donde el sistema comprueba que tanto el usuario como la

contraseña coincidan con los almacenados en la BD, estos datos, previamente entrados por el usuario en registro de su perfil. En caso contrario se re-direcciona a la página de autenticación y se brinda la posibilidad de autenticarse nuevamente.

4.10. Conclusiones del Capítulo

Con el desarrollo del capítulo, se garantizó el análisis y diseño del sistema propuesto. En el análisis, se plantearon todas las especificaciones detalladas que presentará el sistema. Y en el diseño, se fomentaron las clases del diseño, el modelo de diseño de la BD, los patrones utilizados y sus principios, y algunas consideraciones finales.

CONCLUSIONES GENERALES

Con la elaboración de este Trabajo de Diploma se pueden abordar las conclusiones siguientes:

- Las características del actual proceso de desarrollo de Sistemas Tutoriales Inteligentes y las ventajas de los MC, imponen la necesidad de crear una herramienta inteligente que posibilite a los profesores, construir sus propios MCI; además, donde los estudiantes puedan aprovechar tanto las ventajas de los STI como la de los MC.
- Se obtuvo un modelo para diseñar MCI como una nueva perspectiva para el desarrollo de los Sistemas Tutoriales Inteligentes, combinando las ventajas de la Mapas Conceptuales con los Sistemas Basados en Reglas.
- Se exponen la metodología, las herramientas y las técnicas que se utilizan para el desarrollo de dicha aplicación informática.
- Se realizó el análisis y diseño del software propuesto, definiéndose además las funcionalidades del mismo.
- Se obtuvieron los artefactos que generan las fases de análisis y diseño de la herramienta propuesta.

Finalmente dado los aspectos antes mencionados se concluye que el objetivo general para el presente trabajo ha sido cumplido correctamente, obteniéndose el análisis y diseño de una Herramienta para la elaboración de Mapas Conceptuales Inteligentes utilizando para la navegabilidad por la misma las facilidades ventajas de los STI y las ventajas de los MC.

RECOMENDACIONES

Con el objetivo de incorporar mejoras significativas a la Herramienta para Elaborar Mapas conceptuales Inteligentes se recomienda:

- Hacer uso de los artefactos generados en el desarrollo del Trabajo de Diploma en la implementación de la Herramienta para elaborar Mapas Conceptuales Inteligentes con el fin de explotar todas las funcionalidades diseñadas.
- Continuar con el estudio de las soluciones existentes para poder identificar las debilidades que puedan existir y además, añadir funcionalidades en versiones posteriores a la versión que se implementará.

REREFENCUAS BIBLIOGRÁFICAS

1. **Martínez Sánchez, Natalia.** Modelo para diseñar Sistemas de Enseñanza-Aprendizaje Inteligentes utilizando el Razonamiento Basado en Casos. Santa Clara : s.n., 2009.
2. **Soria Francis, Sindy.** Modelo para diseñar Mapas conceptuales Inteligentes utilizando el Razonamiento Basado en Casos. [Online] 2010. <http://biblioteca.uci.cu/sbd/biuci/index.html>.
3. Tutores_inteligentes . [Online] 2011. http://www.ecured.cu/index.php/Tutores_inteligentes.
4. **EDUTEKA.** RESEÑA DE HERRAMIENTAS WEB 2.0 PARA ELABORAR: MAPAS CONCEPTUALES – MAPAS DE IDEAS – TELARAÑAS. [Online] 2002. [Cited: enero 10, 2011.] <http://www.eduteka.org>.
5. TESIS DE MÁSTER MÁSTER DE INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL. [Online] http://oa.upm.es/1064/1/JOSE_MARIA_FONT_FERNANDEZ.pdf.
6. Sistemas Expertos. [Online] [Cited: junio 16, 2011.] http://html.rincondelvago.com/sistemas-expertos_1.html.
7. **Gutiérrez Rodríguez, Marianela y Bedoya Rusenko, Jorge.** Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias (SEGEDIS). [Online] 2010. 78.
8. **Mena Jorge, Ernesto Ferrer y García Suárez-Arango, Adrián.** Desarrollo del sistema SEDIM-SV utilizando la metodología Weiss-Kulikowski . [Online] Junio 2008. 94.
9. **Ivar Jacobson, Grady Booch, James Rumbaugh.** [Online] [Cited: junio 10, 2011.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>. El Proceso Unificado de desarrollo de Software.
10. **Beck, K.** Extreme Programming Explained. [Online]
11. **Núñez Camallea, Noel L.** Gestión de Base de Datos con ADO.NET. [Online] Científico- Técnica, 2004. <http://bibliodoc.uci.cu/pdf/reg04117.pdf>.
12. **RYBARCZYK, MARTIN.** PROYECTO FIN DE CARRERA. [Online] [Cited: junio 15, 2011.] <http://scalab.uc3m.es/~sfernandez/PFC/AI-LIVE/memorias/Memoria2-clips.pdf>.
13. tutorial de Prolog. [Online] [Cited: mayo 10, 2011.] <http://proton.ucting.udg.mx/tutorial/prolog/index.html>
14. ARQUITECTURA DE UN SISTEMA EXPERTO. [Online] [Cited: junio 15, 2011.] <http://maricelamaldonado.wordpress.com/2011/03/23/arquitectura-de-un-sistema-experto/>
15. Módulo de adquisición del Conocimiento ARQUITECTURA DE UN SISTEMA EXPERTO. [Online] [Cited: junio 15, 2011.] <http://maricelamaldonado.wordpress.com/2011/03/23/arquitectura-de-un-sistema-experto/>

16. Sistemas expertos. [Online] [Cited: junio 15, 2011.] http://html.rincondelvago.com/sistemas-expertos_1.html
17. **Ivar Jacobson, Grady Booch, James Rumbaugh.** El Proceso Unificado de desarrollo de Software. [Online] [Cited: junio 9, 2011.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>
18. **Martínez, Pablo, Díaz, Pablo Ruiz y Waisbrot, Sebastián.** Manual de CodeIgniter en Español.
19. Salceda, Nestor. La red para los Profesionales IT. La red para los Profesionales IT. [Online] [Cited: mayo 29, 2011.] <http://esdebugmodeon.com/articulo/active-record>.

REFERENCIAS CONSULTADAS

1. **Edith Ariza Gómez y Jorge Rouquette Alvarado.** Uso de un sistema tutorial inteligente en el ámbito educativo. . [Online] www.somece.org.mx/simposio2004/memorias/grupos/.../080.doc.
2. **Martinez, Natalia.** Conceptual Maps and Cases Based Reasoning: A perspective for the Intelligent Teaching- Learning Systems. [Online] september 2010.
3. **Romero, M. S., Enrique and Gómez-Gil, Pilar.** *Diseño de Hedeia: Una herramienta para la construcción de Sistemas Tutores Inteligentes.* [Online] 2009.
4. <http://www.rational.com.ar> . [Online]
5. [Online] <http://java.sun.com/javase/technologies/jta/index.jsp>.
6. **Lora, O. y Cedeño,** “Análisis y Diseño Módulo Salud Ambiental del proyecto Control Sanitario Internacional”. [Online]UCI 2008. Tesis de diploma.
7. **Font Fernández, José María.** GENERACIÓN DE SISTEMAS BASADOS EN REGLAS MEDIANTE PROGRAMACIÓN GENÉTICA. [Online] Universidad Politécnica de Madrid Facultad de Informática. Madrid, junio 2008. http://oa.upm.es/1064/1/JOSE_MARIA_FONT_FERNANDEZ.pdf.
8. **Soria Francis, Sindy.** Tesis: Modelo para diseñar Mapas Conceptuales Inteligentes utilizando el Razonamiento Basado en Casos. [Online] 2010. <http://biblioteca.uci.cu/sbd/biuci/index.html>.
9. Sistemas Expertos. [Online] [Cited: junio 10, 2011.] <http://html.rincondelvago.com/sistema-experto.html>.
10. Tutorial de Prolog. [Online] <http://proton.ucting.udg.mx/tutorial/prolog/index.htm>.
11. **Casares Charles, Juan Pablo.** AMIVA: Ambiente para la Instrucción. [Online] Instituto Tecnológico Autónomo de México. México, 1999.
12. Bello Pérez, Rafael Esteban. Aplicaciones de la Inteligencia Artificial (Español). [Online] Guadalajara, Universidad de Guadalajara,, 2002. <http://bibliodoc.uci.cu/pdf/reg03294.pdf> .
13. Sistemas Expertos, Sistemas Basados en el Conocimiento. Conferencia de IA. [Online] UCI. <http://eva.uci.cu/mod/resource/view.php?id=310574>.
14. Sistemas Basados en Reglas, Tema 2. Sistemas Basados en el Conocimiento. Conferencia de IA. [Online] UCI. http://eva.uci.cu/file.php/455/Tema_3_Sistemas_Expertos/Bibliografia/Principal/Sistemas_Basados_en_el_Conocimiento.rar
15. **María del C Sánchez-Quevedo, María A Cubero, Miguel Alaminos, Pascual Vicente Crespo y Antonio Campos.** El mapa conceptual. Un instrumento educativo polivalente para las ciencias de la

-
-
- salud. Su aplicación en histología. [Online] Departamento de Histología.de Histología. Facultad de Medicina y Odontología. Universidad de Granada. Educación Médica.
16. *APLICACIÓN DE REDES NEURONALES PARA LA DETECCIÓN DE INTRUSOS EN REDES Y SISTEMAS DE INFORMACIÓN* . [Online]
17. Sistemas-expertos. [Online] [Cited: junio 14, 2011.] <http://www.monografias.com/trabajos16/sistemas-expertos/sistemas-expertos.shtml>.
18. [Online] [Cited: abril 19, 2011.] <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
19. [Online] [Cited: abril 19, 2011.] <http://www.elwebmaster.com/articulos/frameworks-php-recomendados-guia-para-principiantes>.
20. [Online] <http://www.dsic.upv.es/docs/bib-dig/tesis/etd-12152008-114615/TesisAndresTomas.pdf>.
21. **Lio, Dr. Daniel Gálvez.** *Curso de Sistemas Basados en el Conocimiento. Grupo de Investigación en Inteligencia Artificial,*. Universidad Central “Martha Abreu” de Las Villa : s.n.
22. **Martínez Sánchez, Natalia.** Modelo para diseñar Sistemas de Enseñanza-Aprendizaje Inteligentes utilizando el Razonamiento Basado en Casos. Santa Clara : s.n., 2009.
23. **Soria Francis, Sindy.** Modelo para diseñar Mapas conceptuales Inteligentes utilizando el Razonamiento Basado en Casos. [Online] 2010. <http://biblioteca.uci.cu/sbd/biuci/index.html>.
24. Tutores_inteligentes . [Online] 2011. http://www.ecured.cu/index.php/Tutores_inteligentes.
25. **EDUTEKA.** RESEÑA DE HERRAMIENTAS WEB 2.0 PARA ELABORAR: MAPAS CONCEPTUALES – MAPAS DE IDEAS – TELARAÑAS. [Online] 2002. [Cited: enero 10, 2011.] <http://www.eduteka.org>.
26. TESIS DE MÁSTER MÁSTER DE INVESTIGACIÓN EN INTELIGENCIA ARTIFICIAL. [Online] http://oa.upm.es/1064/1/JOSE_MARIA_FONT_FERNANDEZ.pdf.
27. Sistemas Expertos. [Online] [Cited: junio 16, 2011.] http://html.rincondelvago.com/sistemas-expertos_1.html.
28. **Gutiérrez Rodríguez, Marianela y Bedoya Rusenko, Jorge.** Sistema Experto para el diagnóstico médico de las enfermedades genéticas con dismorfias (SEGEDIS). [Online] 2010. 78.
29. **Mena Jorge, Ernesto Ferrer y García Suárez-Arango, Adrián.** Desarrollo del sistema SEDIM-SV utilizando la metodología Weiss-Kulikowski . [Online] Junio 2008. 94.
30. **Ivar Jacobson, Grady Booch,James Rumbaugh.** [Online] [Cited: junio 10, 2011.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.El Proceso Unificado de desarrollo de Software.

31. **Beck, K.** Extreme Programming Explained. [Online]
32. **Núñez Camallea, Noel L.** Gestión de Base de Datos con ADO.NET. . [Online] Científico- Técnica, 2004. <http://bibliodoc.uci.cu/pdf/reg04117.pdf>.
33. **RYBARCZYK, MARTIN.** PROYECTO FIN DE CARRERA. [Online] [Cited: junio 15, 2011.] <http://scalab.uc3m.es/~sfernandez/PFC/AI-LIVE/memorias/Memoria2-clips.pdf>.
34. tutorial de Prolog. [Online] [Cited: mayo 10, 2011.] <http://proton.ucting.udg.mx/tutorial/prolog/index.html>
35. ARQUITECTURA DE UN SISTEMA EXPERTO. [Online] [Cited: junio 15, 2011.] <http://maricelamaldonado.wordpress.com/2011/03/23/arquitectura-de-un-sistema-experto/>
36. Módulo de adquisición del Conocimiento ARQUITECTURA DE UN SISTEMA EXPERTO . [Online] [Cited: junio 15, 2011.] <http://maricelamaldonado.wordpress.com/2011/03/23/arquitectura-de-un-sistema-experto/>

GLOSARIO DE TÉRMINOS

TIC: Tecnologías de la Información y las Comunicaciones. Son todos los servicios, software y hardware que interconectados contribuirán a mejorar las condiciones de vida de las personas.

UCI: Universidad de las Ciencias Informáticas.

Aprendizaje: Proceso mediante el cual un individuo adquiere conocimiento y experiencia, de forma que le permita modificar su comportamiento y adaptarlo a nuevas condiciones de su entorno.

IA: Rama de la informática dedicada a analizar y desarrollar sistemas reproduzcan e imiten los procesos de pensamiento y razonamiento del hombre.

Internet: Red de redes. Sistema mundial de redes de computadoras interconectadas.

Software: Término en inglés para describir a los programas de computación. Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. Soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica.

HTTP: (*Hyper Text Transfer Protocol*), Protocolo estándar de comunicación y transferencia de información entre un navegador y un servidor Web. Es un protocolo de aplicación diseñado para la capa superior de protocolos TCP/IP.

API: (Interfaz de Programación de Aplicaciones), es un conjunto de funciones que permite al programador acceder a servicios de una aplicación a través del uso de un lenguaje de programación.

Open Source: Código abierto (OSS o *Open Source* en inglés) es software para el que su código fuente está disponible públicamente.

IDE: Entorno de Desarrollo Integrado. Es un programa compuesto por un conjunto de herramientas para facilitar la interacción del programador con el lenguaje en cuestión.