

Universidad de la Ciencias Informáticas

Facultad 15



**Propuesta de arquitectura para un sistema de
apoyo a la Gestión del Conocimiento.**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas.

Autores: Adilaraima Martínez Barrio.

Joan Abreu Amaro.

Tutor: Lic. Rolan Rober Bullain Dieguez.

Ciudad de La Habana, junio de 2010

“Año 52 de la Revolución”

“Por más difícil que se nos presente una situación, nunca dejemos de buscar la salida, ni de luchar hasta el último momento. En momentos de crisis, solo la imaginación es más importante que el conocimiento.”

ALBERT EINSTEIN

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 15 de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio y como estime conveniente.

Para que así conste firmamos la presente a los ____ días del mes de _____ de 2010.

Adilaraima Martinez Barrio

Autor

Joan Abreu Amaro

Autor

Lic. Rolan Rober Bullain Dieguez

Tutor

Dedicatoria

Especialmente dedicado a la Revolución por ser capaz de crear obras como la Universidad de las Ciencias Informáticas.

A la Universidad de las Ciencias Informáticas porque más de convertirnos en ingenieros nos convirtió en una mujer y un hombre con aspiraciones.

Dedico este trabajo y el esfuerzo depositado en él a mi papá y a mi mamá, por confiar y hacer de mí, la persona que soy; por no defraudarme y estar siempre ahí para apoyarme.

Dedicado a mis abuelos que son mis guías, mi luz y mi inspiración.

Dedicado a mi hermana que es mi razón de ser y mi esperanza, a mi novio por seguir a mi lado en los momentos buenos y malos.

De Adila

Dedico este trabajo a mis padres, a los que quiero mucho, por ser ellos el motor impulsor de mi vida, mi ejemplo a seguir, por ser mis guías, por su apoyo incondicional y su dedicación.

Gracias por todo.

A toda mi familia por la confianza depositada en mí.

De Joan

Agradecimientos

A mis abuelos María Mirtha Blanco y Olayo Martínez porque aunque no están físicamente presentes, mi día a día depende de todo lo que dejaron en mis padres.

A mi papá Daniel Martínez Castillo y a mi mamá Adis Barrio Blanco que son todo para mí, mi guía, mi inspiración y si no hubiera sido por su amor, dedicación y entrega no hubiera llegado hasta donde estoy, a ustedes les debo todo, los amo.

A Dariela, Papito y Elvira porque me dan la fuerza necesaria para seguir adelante.

A quien amo más de lo que yo pensaba, mi novio Abel, que siempre me ha apoyado y me ayuda a seguir adelante, compartiendo los momentos de estrés, de alegría y de tristeza.

A mis amigas en estos cinco años de carrera que han estado siempre para mí: Aliannys, Yinet, Yanet, Anilenis, Danelis y Yunia.

A quien siempre me confió su experiencia y apoyo: Luis Ernesto.

A mi compañero de tesis, Joan por soportarme y darme fuerzas cuando estaban perdidas.

A Irelis por estar siempre dispuesta a amanecer ayudándonos.

A mis compañeros de grupo que me han dado risas y amarguras, los del 3102, 3208, 34CD9 y 3506.

A mis profesores y amigas de la infancia, que me han guiado con paciencia y han moldeado mi personalidad.

A mi tutor que a pesar de tener poco tiempo siempre me enseñó mucho.

A todos los que de una forma u otra me han ayudado para llegar hasta aquí, a todos ustedes muchas gracias.

Adila

A mis padres, a quienes les debo tanto. A ustedes por confiar en mí, por el ánimo y apoyo que me brindan, por sus consejos, además de darme la fortaleza necesaria para seguir adelante. Los quiero mucho.

A mi hermana querida, que siempre ha estado a mi lado, por ser mi ejemplo y guía tanto en la vida personal como profesional.

A mi sobrina Amanda, la más pequeña de la familia, a ella por llenar la casa de alegría.

A Irellys (mi novia) gracias por tu cariño y comprensión. Por el amor que me has dado, por haber sido mi soporte principal durante estos años de carrera. Agradecerle también a tu familia por su apoyo y preocupación. Eres lo mejor que me ha pasado en la vida.

A mi primo, que más que primo, es mi hermano y amigo, porque hemos crecido juntos y compartido cada momento hasta llegar aquí.

A mi compañera de tesis por ser la mejor compañera del mundo, por su entusiasmo, su entrega al trabajo y por su profesionalismo ante todo. Lo que te propongas lo lograrás, no dudes nunca de eso.

A Iquiris por considerarla como uno de los pocos amigos que tengo, por estar presente en los buenos y malos momentos, por su ayuda desinteresada y su paciencia. Siempre estaré en deuda contigo.

A mis amigos de siempre, Raydel, Rafa, Osmin, Ormany, Jenny, Luisito, a Javier, Tomas y Carlos Rojas.

A todos los que de una forma u otra contribuyeron a mi formación y a este logro.

Joan

Resumen

La era moderna basa su desarrollo en el conocimiento, es por ello que las empresas se orientan a fortalecer sus negocios y competencias a través de lo que la capacidad humana sea capaz de entender y poner en práctica. En el presente trabajo de diploma se describe una propuesta de arquitectura para el posterior desarrollo de un sistema que apoye la Gestión del Conocimiento y que permita definir algunos de los subprocesos dentro de dicha actividad. La solución se ha pensado como una aplicación web, realizada con tecnologías libres destacándose el uso de Linux, Apache y PHP. Se presenta una arquitectura multiplataforma, robusta, flexible y extensible apta para que se le incorporen componentes que sean capaces de asistir varias actividades de la Gestión del Conocimiento no incluidas en el núcleo base.

Palabras Clave: Gestión del Conocimiento, Conocimiento, Arquitectura de Software.

Tabla de contenido

Índice de Figuras	IX
Índice de Tablas	X
Introducción	1
Capítulo 1. Fundamentación Teórica	4
1.1 Principios necesarios sobre Gestión del Conocimiento	4
1.2 Modelo para la Gestión del Conocimiento	7
1.2.1 Consideraciones generales de los modelos a seguir	14
1.3 Herramientas informáticas más usadas en la Gestión del Conocimiento	14
1.4 Aplicaciones informáticas usadas en el proceso de apoyo a la Gestión del Conocimiento	16
1.5 Selección de herramientas para el apoyo a la Gestión de Conocimiento	17
1.6 Metodología de desarrollo	19
1.7 Lenguaje Unificado de Modelado	22
1.8 Tecnologías	22
1.8.1 Tecnologías del lado del cliente	22
1.8.2 Tecnologías del lado del servidor	24
1.9 Framework proveedores de Mapeo de Objeto Relacional	24
1.9.1 Consideraciones del ORM a usar	25
1.10 Conclusiones Parciales	27
Capítulo 2 Propuesta de Arquitectura	28
2.1 Descripción del problema	28
2.1.1 Especificación de Requisitos	28
2.1.1.1 Requisitos Funcionales	29
2.1.1.2 Requisitos No Funcionales	32
2.2 Arquitectura de Software	32
2.3 Análisis de una propuesta de arquitectura	34
2.3.1 Patrones de Diseño Implementados	39
2.4 Vistas de la Arquitectura Propuesta	40
2.4.1 Vista de Casos de Uso	40
2.4.2 Vista Lógica	44
2.4.3 Vista de Implementación	48
2.4.4 Vista de Despliegue	50

2.5 Validación del Sistema	53
2.5.1 Validación por criterio de experto	54
2.6 Conclusiones Parciales	56
Conclusiones	57
Recomendaciones	58
Referencias Bibliográficas.....	59
Anexos	62
Anexo A Encuesta de Validación	62
Anexo B	64
Anexo C	64

Índice de Figuras

Figura 1 Proceso de Gestión del Conocimiento.....	6
Figura 2 Modelo de KMPG	8
Figura 3 Modelo de Arthur Andersen	8
Figura 4 Modelo de KMAT.....	9
Figura 5 Modelo de integración tecnológica propuesto por Larry Kerschberg.....	10
Figura 6 Modelo de integración propuesto por David J. Skyrmer para la Gestión del Conocimiento	12
Figura 8 Arquitectura del CMS Joomla.....	18
Figura 9 Propuesta de Arquitectura.....	35
Figura 10 Componentes de la Capa de Extensiones.....	36
Figura 11 Capa de Gestión del Conocimiento	37
Figura 12 Capa de Fuentes	39
Figura 13 Vista de Caso de Uso.....	41
Figura 14 Estilo Cliente-Servidor	45
Figura 15 Estilo Multicapas	45
Figura 16 Vista Lógica de las capas principales para el apoyo a la Gestión del Conocimiento	45

Figura 17 Vista de Implementación de un componente en Joomla	47
Figura 18 Vista de Implementación	49
Figura 19 Vista de despliegue	50
Figura 20 Modelo de Datos	52
Figura 21 Categoría científica de los expertos	55
Figura 22 Evaluación final de la investigación	56

Índice de Tablas

Tabla 1 Comparación entre metodologías	20
Tabla 2 Criterio obtenido para cada uno de los aspectos	55
Tabla 3 Competencia de los expertos sobre el tema	64
Tabla 4 Puntos de Corte	64

Introducción

A través de la evolución de la humanidad se ha buscado siempre conocer, no existe duda alguna que el conocimiento es un factor indispensable para lograr el éxito en cualquier ámbito. El conocimiento no es un recurso lineal, esto significa que un pequeño incremento de este recurso puede generar una proporción más que significativa de valor agregado, en el momento de su uso o aplicación.

Hacia mediados de la década de los noventa acontece un explosivo crecimiento de lo que constituye la gestión del conocimiento unido a las soluciones basadas en tecnologías de la información (Rubio, 2003).

Con el cambio de milenio se trata al conocimiento como proceso sistémico que atiende a problemas como el ciclo de vida del conocimiento, el vínculo entre innovación y conocimiento, la métrica del conocimiento y la lógica de negocio de la gestión del conocimiento. Surgen las primeras redes y asociaciones profesionales que favorecen su evolución de manera ordenada y convergente. Con ello, empiezan a desarrollarse las *Nuevas Teorías de la Firma*, es decir, la lógica de la creación de valor basada en el conocimiento (Rubio, 2003).

Dado que el conocimiento es un concepto que implica una búsqueda individual de su significación por parte de cada unidad de análisis (individuo, grupo, organización) ya que comprende tres funciones: el pensar (inteligencia), el querer (voluntad y componente emocional) y el hacer (acción), hay que admitir su carácter indisociable de la persona y de la sociedad en la que se desarrolla. En este sentido, las múltiples manifestaciones del conocimiento son complejas combinaciones de estados personales e interacciones sociales, que varían continuamente retroalimentándose en bucles múltiples (Rodríguez, 2008).

Ya para comienzos del siglo XXI muchas empresas implementaron sus sectores clave bajo una administración formal del capital intelectual y el conocimiento, con el fin de alcanzar objetivos organizacionales y convertirlas en más competitivas.

En el contexto nacional, puede observarse el incremento de la gestión de la información en disímiles esferas del país como son: el medio ambiente, el incremento de los eventos científicos, la gestión de los medios de difusión masiva, las acciones en la esfera de la educación, la salud pública, los deportes y otros (Fernández, 2009).

Es lógico plantear que la gestión del conocimiento en la organización es sólo el comienzo de un proceso necesario para la toma de decisiones concretas, relacionadas con el cumplimiento de los objetivos. Por ello con la implantación de un sistema de gestión del conocimiento puede contribuir al eficaz funcionamiento de otros sistemas de gestión, partiendo que: se maximiza el rendimiento del aprendizaje, se considera la información como un recurso más de la organización, se reutiliza la información almacenada y se incorpora en los procesos funcionales y operacionales de la organización, se garantiza la durabilidad de la información, el proceso de aprendizaje es continuo, la capacitación es efectiva porque las personas de forma individual y colectiva se sienten implicadas en su propio desarrollo de forma activa. Al encontrarse documentados los principales conocimientos: se elimina el peligro de pérdidas cuando un trabajador abandona la organización, los problemas no se vuelven repetitivos, se estimula la habilidad para relacionar ideas, conceptos y conocimientos (Rodríguez, 2008).

La dirección del país viene trabajando por la inserción en el mundo tecnológico y no está ajena a la realidad existente, en el que cada día se producen más conocimientos y los conocimientos científicos y técnicos han ampliado los horizontes de la humanidad hasta límites insospechados. La Universidad de las Ciencias Informáticas (UCI en lo adelante) no se encuentra exenta de esta situación, la tecnología ha dado paso a acumular gran cantidad de datos, el crecimiento de la información se hace exponencial y dicha saturación de información atenta contra la mejor explotación del conocimiento en función de los procesos tanto docentes como productivos.

A esto se le suma la diversidad de herramientas que responden de manera dispersa a categorías específicas dentro de la Gestión del Conocimiento. Se conoce que un buen empleo de dichas herramientas implica soluciones factibles sobre todo al problema de saturación de información y la adquisición de la misma por parte de los recursos humanos sabiendo diferenciar la que necesitan, si se gestiona adecuadamente. Concentrar en una única herramienta que de solución a varios subprocesos de la Gestión del Conocimiento, tanto para: visualizar, compartir y descubrir conocimientos, mitigaría en gran medida estos aspectos.

Considerando lo antes planteado se define el siguiente **Problema a resolver**: Las aplicaciones informáticas utilizadas en la Gestión del Conocimiento sólo soportan algunos subprocesos de la misma, implicando el uso de varios sistemas informáticos, lo que ocasiona que el proceso sea más complejo y superficial en ocasiones.

Como **objeto de estudio** se definieron las tecnologías y herramientas para la Gestión del Conocimiento y como **campo de acción**: Arquitectura de las herramientas utilizadas en la Gestión del Conocimiento.

Se trazó como **objetivo general**: Proponer la arquitectura de un sistema para el apoyo a la Gestión del Conocimiento.

Hipótesis: El desarrollo de una propuesta de arquitectura de software flexible y escalable, permitirá el posterior desarrollo de una aplicación informática que integre, en un solo sistema, los distintos subprocesos de la Gestión de Conocimiento con la capacidad de evolucionar, mitigando la complejidad y la superficialidad del proceso.

Para el desarrollo de este trabajo, se trazaron las siguientes **tareas de la investigación**:

1. Realización del estudio del estado del arte de los sistemas utilizados en la Gestión del Conocimiento.
2. Identificación de las principales metodologías para la extracción del conocimiento.
3. Definición de características deseables del sistema de gestión de conocimientos.
4. Definición de la arquitectura del sistema.
5. Validación de la solución propuesta.

El presente documento se estructura en: resumen, dos capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos. En el capítulo 1 se recogen características de la Gestión del Conocimiento, incluyendo las estrategias y herramientas que soportan actualmente los subprocesos de la Gestión del Conocimiento con sus características. En el capítulo 2 se lleva a cabo el desarrollo de la propuesta de arquitectura con una breve descripción de los elementos fundamentales, la validación y la explicación del método usado para la misma.

Capítulo1. Fundamentación Teórica

En este capítulo se aborda lo referido a conceptos fundamentales que se deben tener en cuenta en las herramientas informáticas usadas como apoyo a la gestión del conocimiento y elementos básicos para entender los términos. Se analizan los mecanismos existentes para gestionar el conocimiento y de acuerdo con las arquitecturas de estos sistemas se va perfilando lo que podría ser una propuesta de solución que permita esta actividad. Se estudian las metodologías, tecnologías y lenguajes de desarrollo que se usarán.

1.1 Principios necesarios sobre Gestión del Conocimiento

¿Qué es el conocimiento? En su dimensión más básica se define el conocimiento como *“Una verdad justificada”*. La definición mencionada se basa en raíces filosóficas donde tienen importante influencia los métodos para adquirir y concebir dicha verdad justificada (Nonaka, 1995).

Otra visión de conocimiento es la dada por Mazo y Ortiz: *“El conocimiento surge cuando una persona considera, interpreta y utiliza la información de manera combinada con su propia experiencia y capacidad”* (Mazo y Ortiz, 1998).

Pudiera verse además como: *“la información que transforma algo o a alguien, ya sea en función de lograr acciones, o en función de que el individuo o la institución tenga capacidad de adoptar una acción diferente o eficiente”* (Druker, 1983).

Es importante destacar que según el concepto que plantean Mazo y Ortiz se puede cambiar el entorno, con la utilización de la interpretación, a las situaciones de la cotidianidad basada siempre en el conocimiento y es ahí donde quiere llegar la sociedad actual. Desde otro punto de vista, vinculado a lo que se pretende lograr, se necesita transformar la sobresaturación de la información con acciones, de forma tal que se aproveche al máximo en el desarrollo productivo.

Como señalan Azpiazu y colegas *“la sociedad está emprendiendo una transformación fundamental desde la Era Industrial a la Era de la Información. El motor de la Era de la Información es el aprendizaje y los conocimientos”* (Azpiazu, 2001).

Para competir hay que innovar e innovar requiere contar con nuevos conocimientos o utilizar conocimientos ya disponibles, pero de una forma eficiente (Rivero, 2002). Es entonces cuando se gestiona el conocimiento, por lo nuevo del término existen varias definiciones:

Maestre define Gestión del Conocimiento como un “Conjunto de procedimientos, reglas y sistemas destinados a captar, tratar, recuperar, presentar y transmitir datos, informaciones y conocimientos de una organización” (Maestre, 2000).

“Es el proceso sistemático de buscar, organizar, filtrar y presentar la información con el objetivo de mejorar la comprensión de las personas en una específica área de interés”, es otro punto de vista según Thomas H. Davenport¹.

Otra visión dada por Dr. Yogesh Malhotra²: “encarna el proceso organizacional que busca la combinación sinérgica del tratamiento de datos e información a través de las capacidades de las Tecnologías de Información y las capacidades de creatividad e innovación de los seres humanos”.

Esta investigación se centra en la idea que expresa Thomas H. Davenport debido a que en la actualidad una organización que desee hacer frente a los cambios constantes del entorno, necesita crear conocimiento además de procesarlo eficientemente.

El auge de la gestión del conocimiento tuvo sus inicios en el desarrollo de modelos para captación, explicación, socialización, comunicación y herramientas de soporte para el intercambio de conocimiento que influyeran en los procesos de negocio. Evidentemente, las Tecnologías de la Información y las Comunicaciones (TIC) se encuentran detrás de los aumentos de productividad de empresas y países. Si se analizan las herramientas que permiten el desarrollo de los subprocesos de la gestión del conocimiento se puede percibir que son muy diversos.

Ruggles³, considera entre las tecnologías para la gestión del conocimiento: las intranets, las extranets, los repositorios de conocimientos, las herramientas de ayuda a la toma de decisiones, las herramientas de trabajo en grupo, las bases de datos y hasta la propia Internet.

Sin embargo, Meso y Smith⁴ califican las tecnologías con el término tecnologías del conocimiento y entre ellas, ubican las herramientas de trabajo en grupo, la mensajería, la videoconferencia, las

¹Working Knowledge, Thomas H. Davenport and Laurence Prusak, Harvard Business School Press, 1998.

²El Dr. Yogesh Malhotra es considerado uno de los líderes más influyentes del mundo, en la gestión del conocimiento.

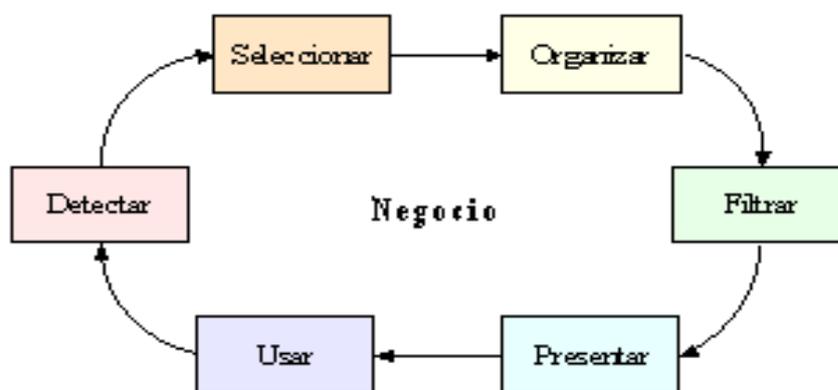
³Rouggles R. El estado de la noción: Gestión del conocimiento en la práctica. 1998;49(3): 80-9.

⁴ Meso P, Smith R. Un recurso basado en vista de los sistemas de gestión de la organización del conocimiento. 2000;4(3):224-34.

tecnologías de apoyo a la toma de decisiones, los navegadores, la tecnología web, la minería de datos, las herramientas de búsqueda, localización y los sistemas de gestión documental.

El proceso de gestión del conocimiento debe entenderse como los subprocesos necesarios para el desarrollo de soluciones orientadas a generar las bases del conocimiento de valor para la organización. Cabe destacar que responde a un flujo de procesos que interactúan como muestra la **Figura 1** (Salazar, 2000).

Figura 1 Proceso de Gestión del Conocimiento.



Fuente: Tomado de (Salazar, 2000).

- ✓ *Seleccionar*: es el proceso de evaluación y elección del modelo en torno a un criterio de interés.
- ✓ *Organizar*: es el proceso de almacenar de forma estructurada la representación explícita del modelo.
- ✓ *Filtrar*: una vez organizada la fuente, puede ser accedida a través de consultas automatizadas en torno a motores de búsquedas.
- ✓ *Presentar*: los resultados obtenidos del proceso de filtrado deben ser presentados a personas o máquinas.
- ✓ *Usar*: el uso del conocimiento reside en el acto de aplicarlo, al problema objeto a resolver. Que posibilite evaluar la utilidad de la fuente de conocimiento a través de una actividad de retroalimentación.
- ✓ *Detectar*: es el proceso de localizar modelos cognitivos y activos (pensamiento y acción) de valor para la organización, el cual radica en las personas. Son ellas, de acuerdo con sus

capacidades cognitivas (modelos mentales, visión sistémica, etc.), quienes determinan las nuevas fuentes de conocimiento de acción.

Los proyectos de Gestión del Conocimiento, según la caracterización realizada por los autores David De Long, Thomas Davenport y Mike Beers, pueden clasificarse según sus resultados en:

- ✓ *Capturar y reusar el conocimiento estructurado.*
- ✓ *Capturar y compartir lecciones aprendidas desde la práctica.*
- ✓ *Identificar redes y fuentes de experiencia.*
- ✓ *Estructurar y mapear las necesidades de conocimiento para mejorar el rendimiento.*
- ✓ *Medir y manejar el valor del conocimiento.*
- ✓ *Sintetizar y compartir conocimiento desde fuentes externas.*

Basado en la clasificación anterior es relevante que el resultado de la futura aplicación esté centrada en estructurar y mapear las necesidades de conocimiento para mejorar el rendimiento, aprovechando las fuentes externas de información y conocimiento, al proporcionar un contexto para el exceso de información disponible.

1.2 Modelo para la Gestión del Conocimiento

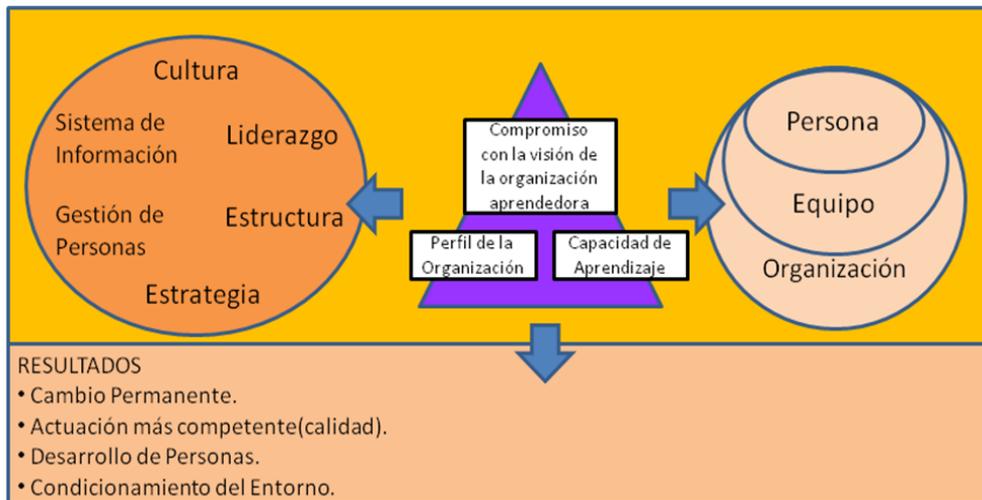
Un modelo para la gestión del conocimiento parte de la siguiente pregunta: ¿qué factores condicionan el aprendizaje de una organización y qué resultados produce dicho aprendizaje? Para responder la pregunta KPMG Consulting⁵ realiza un esfuerzo que produce un modelo cuya finalidad es la exposición clara y práctica de los factores que condicionan la capacidad de aprendizaje de una organización, así como los resultados esperados del aprendizaje (Tejedor y otros, 1998).

Una de las características esenciales del modelo es la interacción de todos sus elementos, que se presentan como un sistema complejo en el que las influencias se producen en todos los sentidos. La estructura organizativa, la cultura, el liderazgo, los mecanismos de aprendizaje, las actitudes de las

⁵ Nombre de la empresa cuyo acrónimo está formado a partir de las iniciales de los fundadores que dieron lugar a KPMG: Klynveld Main Goerdeler (KMG) y Peat Marwick International.

personas, la capacidad de trabajo en equipo; no son independientes, sino que están conectados entre sí (Tejedor y otros, 1998).

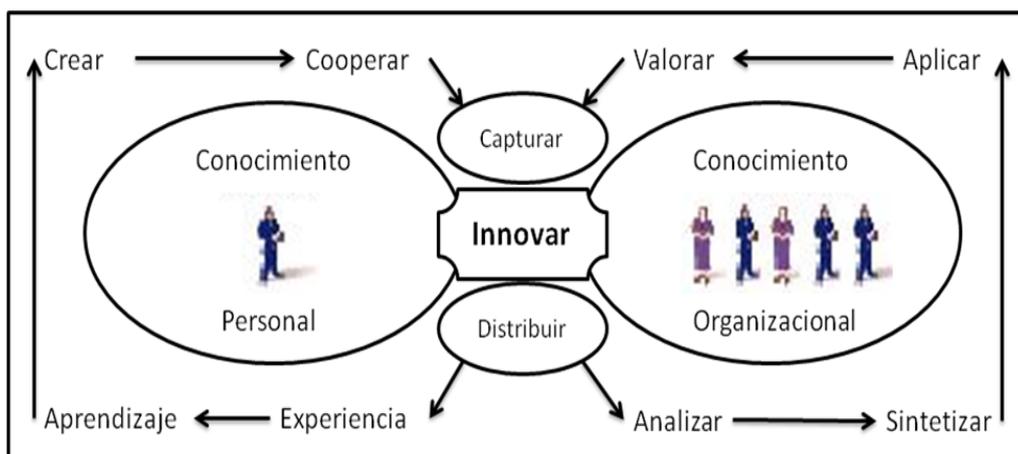
Figura 2 Modelo de KMPG



Fuente: Tomado de (Ramirez, 2000).

Sin embargo, el modelo de Andersen reconoce la necesidad de acelerar el flujo de la información que tiene valor, desde los individuos a la organización y de vuelta a los individuos, de modo que ellos puedan usarla creando valor para los clientes. Este modelo aporta la responsabilidad personal de compartir y hacer explícito el conocimiento para la organización (Andersen, 2000).

Figura 3 Modelo de Arthur Andersen



Fuente: Tomado de (Ramirez, 2000).

Desde la perspectiva organizacional, la responsabilidad de crear la infraestructura de soporte para que la perspectiva individual sea efectiva, instaurando los procesos, la cultura, la tecnología y los sistemas que permitan capturar, analizar, sintetizar, aplicar, valorar y distribuir el conocimiento (Andersen, 2000).

El Knowledge Management Assessment Tool (por sus siglas en inglés y en adelante KMAT) es un instrumento de evaluación y diagnóstico construido sobre la base del Modelo de Administración del Conocimiento Organizacional desarrollado por Arthur Andersen. El modelo propone cuatro facilitadores (liderazgo, cultura, tecnología y medición) que favorecen el proceso de administrar el conocimiento organizacional.

Figura 4 Modelo de KMAT



Fuente: Tomado de (Ramirez, 2000).

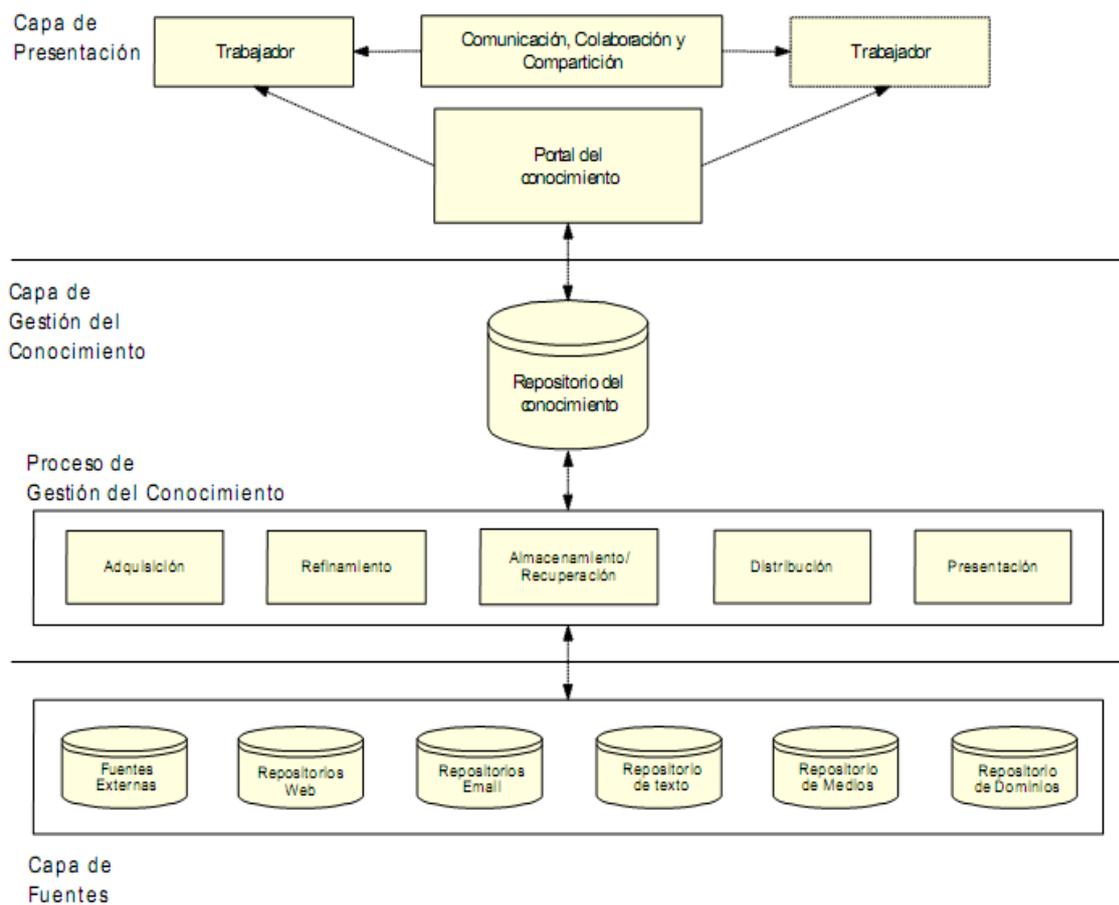
Este modelo vincula el desarrollo de la organización utilizando tecnologías, cosa que no lo hacen los anteriores e incluso hace hincapié en el liderazgo que comprende la estrategia de la organización para reforzar las competencias críticas.

Sin duda la utilización o no, de uno de estos modelos teóricos tiene que estar directamente relacionado con lo que se quiere lograr para apoyar la gestión del conocimiento, que se dirige a una mayor integración en todos los sentidos obteniendo valor agregado para la institución. El modelo de Andersen sólo se enfoca al flujo del conocimiento importante, destacando la poca relevancia para el desarrollo del futuro sistema. Lo positivo que aporta el modelo de KMPG es la incorporación de mecanismos de aprendizajes que van a actuar sobre las actitudes personales y orientados al trabajo en equipo, pero no asocia los mecanismos de aprendizaje con las tecnologías en las que se puede implementar, característica que si está presente en el modelo KMAT, que permite integrar todo lo que define aporte

en una institución con las posibles herramientas a utilizar, por lo que se considera éste, el modelo teórico más adecuado para la presente investigación.

Los modelos de integración tecnológica se definen a partir de dichos modelos teóricos, mostrando de manera gráfica y funcional la interacción entre sus componentes. Un modelo representativo de la integración tecnológica, es el presentado en el reporte de KPMG (Report, 2000) por Larry Kerschberg.

Figura 5 Modelo de integración tecnológica propuesto por Larry Kerschberg



Fuente: Tomado de (Salazar, 2000).

Este modelo (**Figura 5**) crea una dependencia en el concepto de repositorio del conocimiento, el cual viene dado con recursos compartidos y la colaboración de todos los interesados, orientado al manejo de documento con facultades agregadas, uso de filtros complejos y perfiles de usuario, para vincular el conocimiento en búsquedas, pero no deja visible la integración de las tecnologías con los procesos de

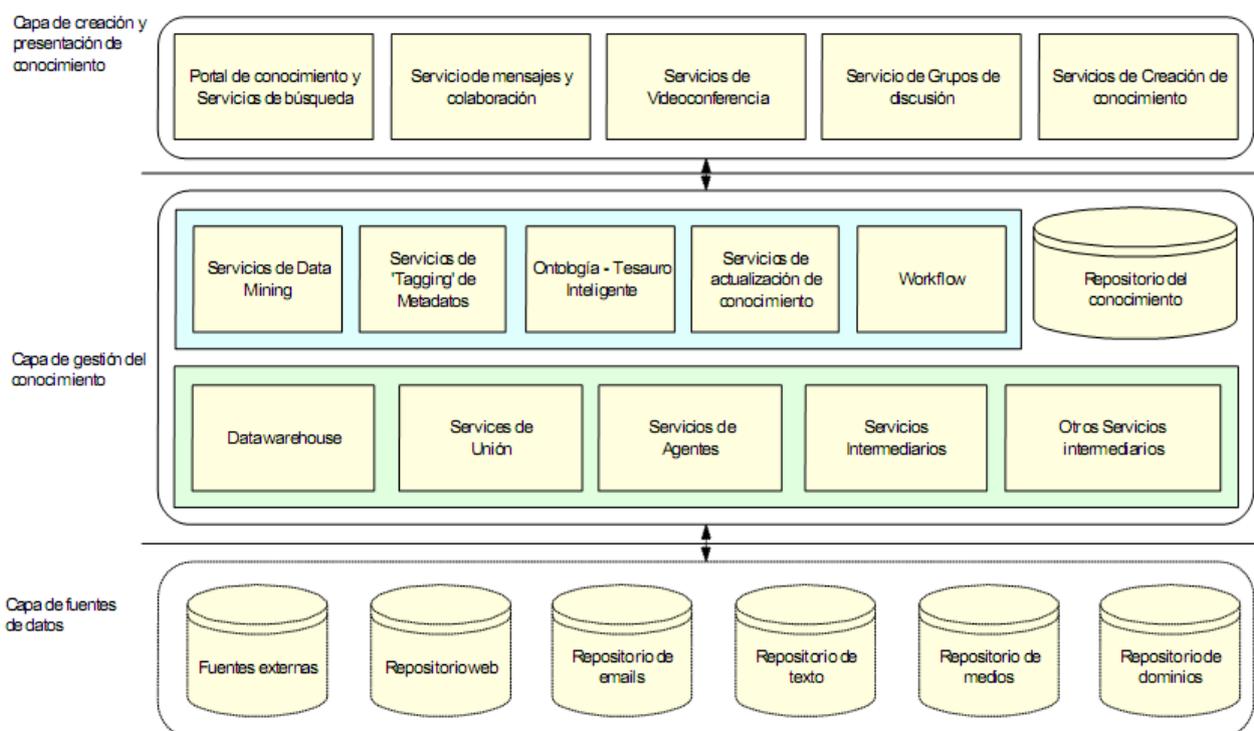
la gestión del conocimiento que se definen para la adquisición, el refinamiento, la distribución y la presentación.

Se menciona además las fuentes externas como repositorios; en la bibliografía consultada, repositorio es el nombre genérico con que se identifican colecciones abiertas de producción científica donde los usuarios o autores añaden directamente los contenidos, que habitualmente se relacionan con instituciones o áreas temáticas. Para un mayor entendimiento se clasifican en el modelo de acuerdo con el contenido que ha sido almacenado, es el caso de texto, de correos, de dominio, de medios y la web; los que están estrechamente vinculados a servicios que ofrecen en el campo de la información.

No es descabellada la idea de contar con todo tipo de clasificaciones de fuentes que se propone, lo que sí es un hecho, es la variedad de archivos que pueden servir los repositorios web sin necesidad de clasificar la información. En otras palabras se clasifica mucho la información en fuentes, característica que puede ser descartada con la implementación de una base de datos mixta y un mecanismo de filtrado de información.

Otro modelo de integración, es el propuesto por David J. Skyrmer (Skyrme, 2000), que muestra la **Figura 6**, establece los diferentes niveles y jerarquías de una infraestructura de conocimientos basada en Tecnologías de la Información.

Figura 6 Modelo de integración propuesto por David J. Skyrmer para la Gestión del Conocimiento



Fuente: Tomado de (Salazar, 2000).

Dicho modelo de integración se basa en la visualización de todos los componentes por capas para gestionar el conocimiento. El presente trabajo considera ambiguo el término que se utiliza en la capa de Presentación de Conocimiento y Gestión de Conocimiento: Servicio de Creación de Conocimiento y Repositorio de Conocimiento respectivamente, estando este muy vinculado a todo el proceso repetitivo de las actividades como flujo de valor. Cuenta además con la clasificación excesiva de la información en repositorios, necesitándose la renovación de estas características en el nuevo sistema. Es importante destacar que dicho modelo propone servicios en la capa de gestión del conocimiento, donde cada uno de ellos influye específicamente en algún proceso de dicha actividad, resaltando la mencionada particularidad en la futura propuesta de arquitectura del presente trabajo, la que se basará en la generalidad de un apoyo base a la gestión del conocimiento con sustanciales mejoras del modelo propuesto.

Govín propone una clasificación de herramientas, dividiéndolas en 7 clases principales (Govín, 2005):

1. *Herramientas de búsqueda y recuperación de la información:* que están formadas por programas que permiten localizar dentro de un conjunto, aquellos documentos que cumplen requisitos específicos. Actualmente, se encuentran disponibles en la red, motores capaces de indexar millones de páginas y

localizar “algo” en ellas en muy breve tiempo. Debido a su facilidad para la indexación es que se plantea su utilización para apoyar la gestión del conocimiento dígame Motores de búsqueda y Metabuscaadores.

2. *Herramientas de filtrado y personalización de la información:* se ocupan de entregar al usuario, la información que requiere sin que este se vea obligado a buscarla en la web. Para estos servicios son utilizadas las Tecnologías Push basándose en la diseminación selectiva de la información.

3. *Tecnologías de almacenamiento y organización de la información:* que son aquellos datos organizados para servir de soporte a diferentes aplicaciones, que evitan la redundancia y la duplicidad de las informaciones. Se denominan dentro de estos los Sistemas de Gestión de Bases de Datos (SGBD), Data Warehousing y Asignación de metadatos. Para la gestión del conocimiento, las bases de datos y repositorios posibilitan la organización del conocimiento en las instituciones.

4. *Herramientas de análisis de información:* utilizan potentes técnicas analíticas, que permiten descubrir relaciones, patrones y tendencias entre los datos que de otra manera no serían descubiertas y que constituyen información para identificar nuevas oportunidades o amenazas, debilidades o fortalezas que proporcionan ventajas competitivas. Dentro de ellas podemos citar minería de datos (Data Mining), minería de textos (Text Mining), árboles de decisión y sistemas expertos, razonamiento basado en casos, tecnologías de autorganización (redes neuronales y asociativas) y simulación.

5. *Sistemas de gestión de flujos y comunicación:* son las herramientas que permiten identificar, analizar, representar y modificar la estructura funcional de la organización, sus flujos de trabajo y de información, realizar la comunicación interna y el control de los procesos; relacionan a todas las personas, funciones y tareas de la organización, entre sí y con el entorno. El soporte tecnológico de las funciones son las aplicaciones informáticas de representación de estructuras y flujos, herramientas de Ingeniería de Software Asistida por Computadoras (CASE por sus siglas en inglés y en lo adelante) y de elaboración de mapas conceptuales, de comunicación lineal, interactiva o simultánea, en tiempo real o diferido; correo electrónico, chat corporativo, intranets y portales corporativos.

Se pueden mencionar otras dos clasificaciones, las herramientas de aprendizaje y comercio electrónico (sistemas de e-Learning y e-Commerce) y los sistemas de gestión empresarial (GE), pero ambos detallan herramientas para la gestión de conocimiento en un área determinada por lo que no son objeto de análisis para una gestión básica del conocimiento.

1.2.1 Consideraciones generales de los modelos a seguir

La creación y transferencia del conocimiento depende de la interacción entre las personas y otras fuentes de información. Como se explicó anteriormente KMAT es un modelo teórico, lo suficientemente consolidado para que responda al modelo de integración tecnológico propuesto por David J. Skyrmer el que proporciona la integración, organización de los elementos y el empleo de tecnologías, con los objetivos de crear un depósito, mejorar el acceso y crear un ambiente para el intercambio de conocimiento, alegando a modificaciones para la arquitectura del futuro sistema que tendrá sus bases en herramientas de gestión de flujos y comunicación, las de análisis de información, tecnologías de almacenamiento, organización de la información y herramientas de búsqueda. Entre ellas abarcan las actividades elementales de los procesos que define la gestión de conocimiento.

1.3 Herramientas informáticas más usadas en la Gestión del Conocimiento

Internacionalmente la Gestión del Conocimiento está tomando cada vez mayor relevancia en el desarrollo sustentable de las empresas, donde se percibe dicha actividad como una solución puramente tecnológica. Por ejemplo: la participación de la tecnología en las soluciones está marcada por el uso de Internet (93%), Intranet (78%), Data warehousing y Data Mining (63%), administración de documentos (61%), apoyo a decisiones (49%), Groupware (43%) y Extranets (38%), frente a un 44% de desarrollo de una estrategia de conocimiento, 33% de desarrollo de políticas y creación de redes formales en torno al conocimiento (Salazar, 2000). Sin duda la productividad de las empresas está dada por la utilización de herramientas en la vinculación con sus procesos de negocio. Estudios realizados por distintas consultoras Internacionales demuestran que empresas del nivel de Microsoft, Hewlett Packard, Ernst & Young, Chevron, Sun Microsystems, British Petroleum entre otras, han iniciado programas de gestión del conocimiento (Programas Knowledge Management) orientados a fortalecer sus negocios y competencias. Cabe destacar el caso de British Petroleum con un programa llamado Equipo de Trabajo Virtual orientado a compartir experiencias; además cuenta con una guía administrada por los empleados, tipo páginas amarillas, que contiene información de 10.000 personas. Basta consultarla para encontrar a la persona que tiene el conocimiento sobre una determinada actividad.

Otra empresa que ha tenido un alto resultado por la aplicación de técnicas para la gestión del conocimiento es la Microsoft que ha tenido su base en el desarrollo de una estructura de competencias. Factor interesante de resaltar es el desarrollo de un ranking de empleados basados en

dichas competencias, el cual está orientado a establecer un diálogo en torno a las capacidades de los empleados a través de toda la empresa.

El caso de Hewlett Packard es muy peculiar y llamativo debido a que la idea partió de generar una red de expertos que pudieran proveer de conocimientos a toda la compañía. De hecho, el desarrollo de productos se fortaleció a través de links de conocimiento, lo cual significa acceso a la documentación de las mejores prácticas establecidas por los expertos en la compañía.

Cuba ha dado sus primeros pasos para elevar la calidad de sus empresas, uno de ellos es descubrir la necesidad de redefinir estrategias basada en la gestión del conocimiento, estamos hablando de la Empresa de Telecomunicaciones de Cuba S.A. (ETECSA) que definió su estrategia de capacitación a través de las páginas web para materializar la gestión del conocimiento, definiendo los indicadores básicos de las posibles competencias, conocimientos y habilidades que debe poseer el capacitador.

Los mayores esfuerzos se han dirigido a la gestión de la información y no existen pruebas, según la bibliografía consultada, sobre la presencia de proyectos que favorezcan la integración de acciones orientadas a la gestión del conocimiento acumulado en las organizaciones y el empleo de métodos y herramientas que permitan su aprovechamiento, aunque existen entidades que gestionan el conocimiento como, por ejemplo, el Ministerio de Salud Pública (MINSAP) por medio de su red Infomed (Fernández, 2009).

Actualmente en la UCI existe un reto que ha revolucionado la manera de pensar de los líderes de proyecto, se hace necesaria una mayor productividad que implica que cada miembro conozca lo que hace y que aproveche su tiempo de trabajo; conocer lo que hace implica que la persona esté familiarizada con el ambiente, el entorno y como fluye cada proceso en el que está involucrado. Se encuentra definido lo que pudiera ser un modelo de gestión conocimiento para que en un proyecto productivo se logre vincular la producción con la gestión del conocimiento dado en el evento Uciencia 2008 (Pérez, 2008). Donde se plantea las siguientes estrategias:

- ✓ Creación de mapas de conocimientos.
- ✓ Creación de equipos autorganizables.
- ✓ Creación de sesiones de dialogo grupal, esto permite el intercambio de ideas y de soluciones, a través de esquemas, modelos, metáforas, propiciando la creación y expansión del conocimiento.

- ✓ Creación de almacenes del conocimiento, disponer de base de datos que beneficien a todos los implicados (digitalizar todos los documentos internos necesarios para el desarrollo del estudio y la investigación, documentos de profesores e investigadores) lo cual puede ser organizado con el uso de herramientas para este fin como es el caso de los mapas conceptuales.
- ✓ Actualización diaria de la información necesaria para agilizar y hacer más efectiva la toma de decisiones.
- ✓ Disponer de algún medio para el intercambio de información, conocimientos y experiencias.
- ✓ Disponer de un repositorio de información al que tengan acceso todos los trabajadores, complementados con herramientas de comunicación para intercambiar experiencia, conocimientos y casos de estudio, lo que implica una mejora en los procesos de gestión.

1.4 Aplicaciones informáticas usadas en el proceso de apoyo a la Gestión del Conocimiento

Para hacer efectiva una gestión del conocimiento dentro de las complejas organizaciones de hoy en día, es necesario disponer de herramientas que permitan generar procesos colaborativos, distribuir y sincronizar tareas en la organización, de forma que se pueda reducir el tiempo y aumentar la eficacia. Las herramientas engloban procesos que podrían incluirse dentro de funcionalidades de búsqueda o de distribución personalizada de la información, van mucho más allá de esto, para convertirse en paquetes altamente integrados, capaces de realizar una adecuada gestión del conocimiento de la organización (Grau, 2001).

Se pueden mencionar las herramientas de Simulación: las cuales simulan el esquema de coordinación de una tarea de trabajo, con lo que se puede corregir errores sin necesidad de que se hayan producido. Están basadas en los procesos de pensamiento humano con la finalidad de poder reproducir sus esquemas principales. Son de utilidad para organizaciones que desarrollan complejos proyectos con una gran cantidad de usuarios.

Otras de las herramientas empleadas son las de Búsqueda: que basan su funcionamiento en la exploración a través de documentos o palabras clave, información específica. El trabajo en un entorno dinámico exige la obtención de información sobre temas relacionadas en distintos momentos del tiempo, que se garantiza con la utilización de buscadores.

Además, se pueden analizar los almacenes de datos como una copia de datos específicamente estructurada para consultas y análisis, los que tienen significación en actividades de gestión de conocimiento debido a que permiten el posterior análisis de la información.

Cabe destacar en el marco de esta investigación la utilización de Portales Corporativos como herramientas, que permiten la utilización de estándares de Internet e integrar las distintas herramientas de gestión de datos e información. Mediante los portales, los individuos tienen acceso a contenidos personalizados y además, son un instrumento que ayudan a crear ambientes colaborativos. Como herramienta para el desarrollo de portales corporativos se analizan los Sistemas de Gestión de Contenido (Content Management Systems, en inglés y CMS en lo adelante). Sistemas que facilitan la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. Uno de los campos más interesantes es que permite la incorporación de estándares que mejoran la compatibilidad de componentes, facilitan el aprendizaje al cambiar de sistema, aportan calidad y estabilidad y se puede contar con componentes necesarios dentro del modelo de gestión del conocimiento que están probados. Otra particularidad de las herramientas de creación de portales es que, indirectamente, se permite el acceso a datos que no necesariamente estén ubicados en los repositorios de la organización. Además, que con el acceso personalizado permite que aumente la eficiencia en las búsquedas de información.

Es relevante para este trabajo que los CMS cuenten con un conjunto de procesos ya definidos para gestionar el conocimiento, que comparado con los almacenes de datos, las herramientas de simulación y de búsqueda solo se vinculan con uno, resulta conveniente que permitan la asociación de varias de estas herramientas, con el fin de lograr un buen apoyo a la gestión de conocimiento, partiendo entonces de la gestión de contenido que viene implementado por ende en estos sistemas; se les adiciona la característica de flexibilidad a la hora de expandirse, son sencillos e imitan la forma en que se relacionan las personas, permiten poner el conocimiento a disposición de todos, sin barreras formales u otras consideraciones además de poder construir sistemas propios para cada organización, permitir la colaboración, el intercambio y la comunicación; además que no necesitaría ningún software adicional instalado en las máquinas clientes, lo que elevaría su uso.

1.5 Selección de herramientas para el apoyo a la Gestión de Conocimiento

Primeramente es necesario definir las características importantes para la selección de una arquitectura base deseada, sus características estarían orientadas a incluir nuevas funcionalidades en la web. El sistema tiene que crecer y adaptarse a las necesidades futuras. Debe estar basado en código libre.

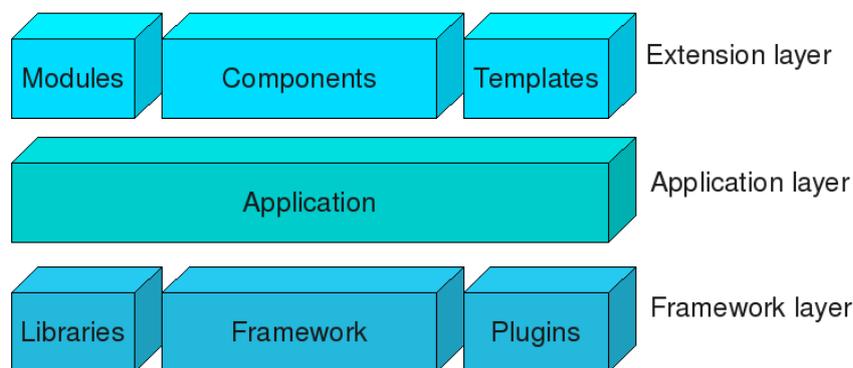
También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos. Controlar el acceso gestionando los diferentes permisos para las áreas, grupos o individuos. Tiempo requerido para el dominio de la herramienta, es decir, la facilidad de uso. De la misma forma influye el número de módulos existentes para cumplir las necesidades iniciales de la aplicación y la comunidad de usuario que den soporte al desarrollador. Teniendo en cuenta las características antes mencionada y la bibliografía consultada el interés inicial se inclinó por: Plone, Drupal y Joomla.

Sin duda Plone es un CMS muy potente que cuenta con un motor de búsqueda completo, indexación en tiempo real y motor Workflow integrado, pero el dominio de su lenguaje es un poco costoso para insertarle nuevos componentes y posee muy pocos de estos reutilizables para el nuevo sistema, comparado con Joomla o Drupal. Por su parte Drupal cuenta con alta complejidad en el desarrollo de sus complementos, por lo que Joomla con su vasta documentación y una amplia comunidad de colaboradores, diversidad de componentes ya programados y la modularidad que lo hace flexible lo convierte en superior a los antes analizados, de ahí que sea el seleccionado para asumir su arquitectura como base en el desarrollo del presente trabajo.

Joomla

Es una de los más populares CMS de código abierto. Tiene una gran comunidad de usuarios y también toda la documentación para crear tus propias aplicaciones. La arquitectura de Joomla se basa en 3 capas fundamentales:

Figura 7 Arquitectura del CMS Joomla



Fuente: Tomado de (Reference, 2009).

Capa de Extensión: conformada por tres elementos: módulos, componentes y los temas o plantillas

Capa de Aplicación: se compone de aplicaciones que amplían el Framework, JApplication, conformado por: JInstallation es responsable de la instalación de Joomla en un servidor web. JAdministrator es responsable de la parte final del Administrador. JSite por su parte se encarga de la front-end de la página web, incorporándose XML-RPC quien soporta la administración remota de la página web de Joomla.

Capa de Framework: cuenta con las librerías, que representan paquetes de código, las cuales proporcionan un grupo de funciones relacionadas con el Marco de Joomla; y con Plug-ins que son pequeñas secuencias de código que se ejecutan en cuanto se producen ciertos eventos en el sistema.

Resulta conveniente destacar que la arquitectura de Joomla tiene alta modularidad y flexibilidad implementada bajo el patrón Modelo-Vista-Controlador que hace la hace ideal para sistemas extensibles.

1.6 Metodología de desarrollo

En un proyecto de desarrollo de software, la metodología define “quién” debe hacer “qué”, “cuándo” y “cómo” debe hacerlo, para alcanzar un determinado objetivo. La finalidad de una metodología de desarrollo, es garantizar la eficacia y la eficiencia en el proceso de generación de software. En realidad no existe una metodología estándar, sino que las características de cada proyecto, las de su equipo de desarrollo, recursos disponibles y tiempo para su elaboración exigen la flexibilidad del proceso, adaptándose el mismo al entorno y teniendo como objetivo alcanzar la máxima calidad en lo que se produce (José H. Canós, 2004).

Tabla 1 Comparación entre metodologías

Metodologías Ágiles	Metodologías Tradicionales
Basada en heurística provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Procesos menos controlados, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura de software	La arquitectura de software es esencial y se expresa mediante modelos.

Fuente: Tomado de (José H. Canós, 2004).

Haciendo un análisis de la **Tabla 1**, las metodologías ágiles están preparadas para cambios en el transcurso del proyecto de desarrollo, son concebidas para equipos de trabajo pequeños y generan pocos artefactos pero trata con menos énfasis la arquitectura, siendo este un factor indispensable para el desarrollo de la presente investigación; no siendo así en las tradicionales que aunque generen más artefactos, el proceso es más controlado regido por normas y políticas. Es por ello que centraremos la atención en las metodologías tradicionales. El problema principal radica en como coordinar las actividades del desarrollo del sistema que brinden un enfoque hacia su arquitectura. Dentro de las más usadas están: Microsoft Solutions Framework (MSF) que se centra en los modelos de proceso, encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: modelo de arquitectura del proyecto, modelo de equipo, modelo de proceso, modelo de gestión del riesgo, modelo de diseño de proceso y finalmente el modelo de aplicación. Esta es una metodología flexible e

interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos.

Proceso Unificado de Rational (RUP, por sus siglas en inglés y en lo adelante) la que se rige por los principios de ser iterativo e incremental, basado en casos de uso y centrado en arquitectura.

Según el Proceso Unificado de Desarrollo de Software la arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Se representa mediante varias vistas que se centran en aspectos concretos del sistema (Rational, 2003).

La representación de la arquitectura de un software en RUP está formada por 4+1 vista o modelos parciales separados, relacionados entre sí, proponiendo para ello vistas obligatorias, las cuales son (Rational, 2003):

- ✓ *Vista de casos de uso*: sólo hay una vista de caso de uso del sistema, que ilustra los casos de uso y los escenarios que implican riesgos técnicos, de clase o de comportamientos arquitectónicamente significativos.
- ✓ *Vista lógica*: comprende las clases, interfaces y colaboraciones que forman el vocabulario del problema y su solución. Sólo hay una vista lógica del sistema, que ilustra las realizaciones de casos de uso clave, subsistemas, paquetes y clases que abarcan el comportamiento arquitectónicamente significativo. La vista lógica se perfecciona durante las iteraciones.

Incluyendo las restantes tres vistas de forma opcional: vista de implementación, vista de procesos y vista de despliegue.

Aunque no existe de forma explícita una vista arquitectónica, estas cinco vistas pretenden describir, en su conjunto, la arquitectura del sistema según lo propuesto por RUP.

Escogida para el desarrollo de este trabajo primeramente por la vasta documentación y soporte por herramientas CASE que posee, además de poseer una variedad de herramientas para soportar el modelado con UML (por sus siglas en inglés y en lo adelante) tanto libres como propietarias. Provee un marco de buenas prácticas de gestión de proyecto y centra el proceso de desarrollo en la arquitectura definiendo para ella actividades y artefactos.

1.7 Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Su objetivo es representar el conocimiento acerca de los sistemas que se pretenden construir y las decisiones tomadas durante su desarrollo, tanto los representados por diagramas estáticos (Casos de Uso, diagrama de clases) como los dinámicos (Diagramas de actividades, interacción).

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (G. Booch, 1999).

RUP propone el uso de este lenguaje de modelado para la descripción de los artefactos propuestos por sus flujos de trabajo en cada una de las distintas fases. El surgimiento de este lenguaje casi a la par de RUP y por los mismos creadores hace que fusionen perfectamente.

¿Por qué usar UML?

Las desventajas que presentan los Lenguajes de Descripción de Arquitecturas de Software (ADLs), implicando el estudio de una sintaxis especializada y la carencia de un estándar en su elaboración; pueden ser superadas si se utiliza un lenguaje de Modelamiento que sea conocido en la industria y que además esté apoyado por herramientas y metodologías de desarrollo, este lenguaje de Modelamiento es UML, que se está convirtiendo en una notación estándar de hecho en las empresas. UML permite que se represente de manera semi-formal la estructura general del sistema, con la ventaja de que este mismo lenguaje puede ser usado en todas las etapas de desarrollo del sistema y su representación gráfica puede ser usada para comunicarse con los usuarios.

1.8 Tecnologías

Se proponen varias tecnologías a usar para el desarrollo del futuro sistema que tributan a una estandarización en el proceso de implementación, orientadas a la selección previa del rediseño de un CMS como punto de partida para la nueva arquitectura.

1.8.1 Tecnologías del lado del cliente

HTML

HyperText Markup Language o Lenguaje de Marcas de Hipertexto es el lenguaje de marcado predominante para construir páginas web y usado normalmente en la World Wide Web. Define la estructura y el contenido de las páginas permitiendo combinar textos, imágenes, sonidos, videos y enlaces a otras páginas. Su nivel de complejidad es bajo y además permite embeber dentro de su código otros script escritos en lenguajes como PHP y JavaScript. Sus características y funcionalidades hacen propia el uso de este lenguaje en el marco de este trabajo.

JavaScript

Es un lenguaje orientado a objetos, interpretado, que se ejecuta del lado del cliente. Utilizado principalmente en páginas web y con sintaxis semejante a Java o C, sigue la programación basada en prototipos. Además, usa DOM (en el inglés Document Object Model) para acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML. Dentro de las principales tecnologías para interactuar con DOM que usa JavaScript se encuentran AJAX y DHTML.

Javascript además de ser un lenguaje con muchas posibilidades, permite la programación de pequeños y grandes programas, con funciones y estructuras de datos complejas. Tiene la ventaja que brinda al programador todo tipo de funciones con la que se puede administrar una página web, elevando su uso debido a que es interpretado por todos los navegadores actuales. Permite así que se considere su uso en el desarrollo del presente trabajo.

CSS

Cascade StyleSheet u Hoja de Estilos es una tecnología formal que define cómo se va a mostrar un documento escrito en HTML o XML. Establece la separación definitiva de la lógica (estructura) y el físico (presentación) del documento. El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores.

Permite el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. El documento HTML en sí mismo es más claro de entender y

se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea) es por las razones mencionadas que se utilizará en el desarrollo del trabajo.

1.8.2 Tecnologías del lado del servidor

El equipo de desarrollo ha decidido utilizar PHP, se darán las razones por las que usar dicho lenguaje.

Lenguaje PHP

PHP es un acrónimo recursivo que significa Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994. Es un lenguaje de propósito general usado para el desarrollo de aplicaciones web y puede ser embebido dentro de código HTML. Es ejecutado en una web, creando páginas web a partir del código PHP que toma como entrada. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno (Southwell, 2005).

Dentro de las facilidades que brinda se puede mencionar que es multiparadigma y multiplataforma. Posee capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones). Su capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad. Posee una amplia documentación en su página oficial entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Permite las técnicas de Programación Orientada a Objetos. Posee una biblioteca nativa de funciones sumamente amplias. No requiere definición de tipos de variables.

Su condición de lenguaje para desarrollar aplicaciones web, facilidad de uso, heterogeneidad para conectarse con cualquier gestor de bases de datos, debido a que posee juego de funciones que permiten esto, es un factor decisivo para ser usado como lenguaje en el desarrollo de este trabajo.

1.9 Framework proveedores de Mapeo de Objeto Relacional

El Mapeo de Objeto Relacional (ORM por sus siglas en inglés y en lo adelante) es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos utilizado y el sistema de base de datos relacional. El mapeo objeto-relacional ayudará precisamente a eso, a olvidarnos completamente de cómo convertir los objetos en datos primitivos para almacenarlos y viceversa. Proporcionan: lenguaje propio para realizar las consultas, rapidez en el desarrollo, abstracción de la base de datos, reutilización y seguridad.

Propel

El proyecto Propel se inició en 2005. Minuciosamente documentado, respaldado por muchos tutoriales en la web, también se beneficia de una comunidad que proporciona un apoyo rápido para el principiante y los programadores. Implementa todos los conceptos clave de capas ORM: el patrón ActiveRecord, validadores, los comportamientos, la herencia de tablas, una ingeniería inversa de bases de datos existentes, conjuntos anidados y las transacciones anidadas. La extensibilidad está en el centro de diseño de Propel, puede ser utilizado para cuando se necesita consultas personalizadas o transacciones hiper-optimizado. Posee IDE-amigable y es fácil de usar.

Doctrine

Doctrine es un ORM muy potente para PHP, que cuenta con una capa de abstracción a bases de datos (DBAL por sus siglas en inglés). Una de sus características principales es la posibilidad de escribir consultas orientadas a SQL, esto proporciona a los desarrolladores una poderosa alternativa que facilita la flexibilidad, sin necesidad de duplicación de código innecesario. Provee además métodos que pueden validar o modificar la entrada y salida de base de datos, a fin de posibilitar la exportación de una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos (POO) debido a que doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrado a los principales frameworks de desarrollo utilizados actualmente.

1.9.1 Consideraciones del ORM a usar

Ambos ORMs tienen características básicas similares, ya que soportan cualquier operación usual en un CRUD (Create, Retrieve, Update and Delete), ya sea desde crear un nuevo registro o actualizar los registros existentes. Además, los dos pueden generar las clases PHP de un modelo de datos, Propel basado en XML y Doctrine basado en YAML, y ambos soportan varios motores de bases de datos. Doctrine soporta migraciones, *caching*, *events*, *pagination*, *command line interface*, una comunidad de usuario que le aporta soporte y documentación al ORM, con lo que podríamos decir que con estas características avanzadas Doctrine supera a Propel. Se propone como framework en el desarrollo del futuro sistema para la implementación de la capa abstracción de bases datos pues brinda un rendimiento en ejecución y admite escribir de forma segura consultas muy complejas.

1.10 Herramientas CASE

El nombre de “Herramienta CASE” consiste en una o varias herramientas que permiten organizar y manejar cierta información de un proyecto informático. Están destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Sirven de apoyo en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática y documentación o detección de errores.

Rational Rose

Es una herramienta desarrollada por los creadores de UML que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los 42 componentes, transición a los usuarios y certificación de las distintas fases. Proporciona un lenguaje común de modelado que facilita la creación de software con calidad. Dentro de sus principales características se encuentran que posee soporte para análisis de patrones ANSI C++, Rose J y Visual C++. Soporte de ingeniería directa y/o inversa para algunos de los conceptos más comunes de Java 1.5. La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables y la capacidad de análisis de calidad de código. Rational Rose es una herramienta CASE de software propietario. El costo de una licencia para un usuario único es de 4,741 euros contando los valores agregados que significa el mantenimiento por 12 meses.

Visual Paradigm

Soporta notación UML 2.x, capacidades de generación de código, importación desde Rational Rose, generación de código e ingeniería inversa a la vez de los lenguajes: Java, C++, CORBA IDL, PHP, XML Schema, Ada y Python. Adicional, soporta la generación de código en: C#, VB .NET, Object Definition Language (ODL), Flash ActionScript, Delphi, Perl, Objective-C y Ruby.

Desarrolla un entorno de creación de diagramas para UML. Su diseño está centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Usa un lenguaje estándar, común a todo el equipo de desarrollo que facilita la comunicación. El modelo y el código permanecen sincronizados en todo el ciclo de desarrollo. Presenta disponibilidad de integrarse en los principales Entornos de Desarrollo Integrado (IDE en lo adelante) y en múltiples plataformas.

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, entre otros. Además, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en

detalles excesivos, también, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico.

Partiendo del estudio realizado a las principales herramientas CASE existentes, se decidió utilizar Visual Paradigm, por ser una herramienta multiplataforma, facilidad que pocas herramientas CASE brindan, permite exportar a varios formatos los diagramas elaborados, es robusta y de fácil uso. Además, se sustenta la elección de la misma en el hecho de que la UCI cuenta con la licencia para el uso de dicha herramienta.

1.10 Conclusiones Parciales

En el presente capítulo se trataron conceptos importantes de gestión de conocimiento, donde esta actividad es asistida por herramientas diversas. Por lo que se arribaron a las siguientes conclusiones:

- ✓ Con el estudio de los modelos de la gestión del conocimiento se escogió el modelo teórico KMAT, que permite integrar el flujo de información en una organización con las herramientas a utilizar.
- ✓ Se utilizará el modelo de integración tecnológica propuesto por David Skymer ya que cumple con las características deseadas.
- ✓ Se escogió como herramienta base un CMS cuyas características permitirán la escalabilidad y extensibilidad del nuevo sistema.
- ✓ Se optó por usar la metodología RUP que se centrará en el desarrollo de la arquitectura. Se propone la utilización de UML para el modelado de los procesos; como tecnologías CSS, HTML, Java Script; PHP como lenguaje de desarrollo y el framework Doctrine para la abstracción de bases de datos, los que asistirán el proceso lo más organizado posible.

Capítulo 2 Propuesta de Arquitectura

En este capítulo se abordarán elementos de definición de arquitectura, estilos arquitectónicos y patrones. Se propone una arquitectura capaz de soportar algunos de los subprocesos para lograr un sistema de apoyo a la gestión del conocimiento. Se irá describiendo dicha arquitectura mediante las actividades que propone RUP para este tipo de sistemas, dígase vistas arquitectónicas importantes para entender de manera abstracta la propuesta y se incluirá la validación de esta.

2.1 Descripción del problema

La arquitectura es la encargada de establecer los cimientos en los cuales se responde a los requisitos funcionales y no funcionales, además de basarse en el ambiente donde se desarrollará el sistema, la reutilización de componentes, la adopción de estándares como algunos de los aspectos a tener en cuenta en su confección.

El desarrollo en este ámbito es para solucionar la inexistencia de un sistema que permita el apoyo a la gestión del conocimiento, ya que en la actualidad solo se vinculan de forma aislada en subprocesos, como son: servicios de minería de datos, servicio de búsqueda, mapas conceptuales, publicación de contenidos, servicio de grupos de discusión, servicio de mensajería y colaboración, necesitando cohesionar algunos de estos que abarquen varias áreas de dicha actividad. Es por ello la necesidad de definir en una primera iteración, una arquitectura capaz de responder a categorías como las mencionadas.

Dentro de los requisitos que debe cumplir debe ser multiplataforma, soportado sobre herramientas libres, configurable para la sesión del administrador, extensible, fácil de administrar y por defecto que permita la publicación de contenidos.

2.1.1 Especificación de Requisitos

La Especificación de Requisitos de Software es una descripción completa del comportamiento del sistema que se va a desarrollar. También contiene requisitos no funcionales (o complementarios) que imponen restricciones en el diseño o la implementación. En este apartado se registran los requisitos capturados hasta el momento, y se especifican las capacidades operacionales y funcionales que el sistema deberá tener con el mayor detalle posible. A continuación se definen los requisitos funcionales y no funcionales que el sistema debe cumplir.

2.1.1.1 Requisitos Funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. A continuación se listan los requisitos funcionales significativos para la primera iteración del sistema.

RF1. Realizar autenticación de usuario: el sistema permitirá que los usuarios invitados puedan registrarse. Para lo cual deberá proporcionar: usuario que usará en el sistema y contraseña.

RF2. Adicionar usuario al sistema: el administrador del sistema podrá registrar nuevos usuarios proporcionando los datos pertinentes como son: nombre y apellido, usuario que usará en el sistema, dirección de correo y contraseña.

RF3. Eliminar usuario al sistema: el administrador podrá dar baja a los usuarios registrados en el sistema seleccionando el usuario deseado.

RF4. Actualizar datos de usuario del sistema: el administrador podrá actualizar cualquiera de los datos de los usuarios registrados al introducir el nombre del mismo.

RF5. Buscar contenido: un invitado podrá buscar contenido en un área específica o por etiquetas, introduciéndole la palabra clave.

RF6. Buscar imágenes: un usuario invitado podrá seleccionar un área específica o introducir una etiqueta para buscar una imagen.

RF7. Buscar conceptos: un invitado podrá realizar la búsqueda de conceptos introduciendo la palabra clave.

RF8. Guardar preferencias de búsqueda: el usuario registrado podrá guardar la búsqueda realizada, almacenando la palabra clave y la(s) dirección (es) URL(s) del (los) documento (s) seleccionado (s).

RF9. Adicionar área de contenido: el usuario registrado podrá crear una nueva área introduciendo datos como nombre del área y descripción.

RF10. Modificar área de contenido: el usuario registrado podrá modificar los datos del área al introducir el nombre.

RF11. Eliminar área de contenido: un usuario registrado podrá eliminar un área previamente listada.

RF12. Adicionar publicación: el usuario registrado podrá adicionar publicaciones insertando los siguientes datos: nombre, autor, descripción, publicación y fecha.

RF13. Modificar publicación: el usuario registrado podrá modificar la publicación correspondiente proporcionando los datos específicos.

RF14. Eliminar publicación: el usuario registrado podrá eliminar el contenido especificado con el nombre.

RF15. Adicionar tesoro: el usuario con privilegios podrá adicionar tesoro insertando datos como: tema del tesoro, lista de términos y jerarquía entre términos.

RF16. Modificar tesoro: el usuario con privilegios podrá modificar tesoro actualizando los datos correspondientes del mismo.

RF17. Eliminar tesoro: el usuario registrado podrá eliminar un tesoro asociado a un tema seleccionado.

RF18. Guardar término en Tesoro: un usuario registrado podrá, cuando encuentre un término, insertarlo en un tesoro existente, especificando el nivel de inserción y el tema donde se insertará.

RF19. Adicionar usuario de mensajería: un usuario registrado puede crear su propio usuario de mensajería introduciendo los siguientes datos: apodo, contraseña y correo electrónico.

RF20. Modificar usuario de mensajería: el usuario registrado podrá actualizar los datos correspondientes a apodo, contraseña y correo electrónico.

RF21 Eliminar usuario de mensajería: un usuario registrado podrá eliminar usuario de mensajería introduciendo el apodo.

RF22. Adicionar contacto de mensajería: un usuario registrado podrá adicionar contacto especificándole el apodo de este.

RF23. Modificar contacto de mensajería: el usuario registrado puede modificar los datos de un contacto seleccionado por el apodo.

RF24. Eliminar contacto de mensajería: el usuario registrado podrá eliminar un contacto especificando el apodo.

RF25. Adicionar tema de grupo de discusión: un usuario registrado podrá adicionar un tema para un grupo de discusión introduciendo los siguientes datos: nombre del tema y descripción.

RF26. Modificar tema de grupo de discusión: un usuario registrado podrá modificar lo referente a tema de grupo de discusión especificando los datos correspondientes.

RF27. Eliminar tema de grupo de discusión: el usuario registrado podrá eliminar un tema de los previamente listados.

RF28. Adicionar comentario a grupo de discusión: un usuario registrado podrá insertar un comentario en un tema de discusión especificando los datos siguientes: autor y comentario.

RF29. Eliminar comentario de grupo de discusión: el usuario registrado podrá eliminar un comentario dado por este.

RF30. Crear mapa conceptual: un usuario registrado podrá crear un mapa conceptual introduciendo datos como: nombre del mapa, términos que representan los conceptos y las relaciones entre estos.

RF31. Guardar mapas conceptuales: un usuario registrado podrá guardar un mapa, especificando el nombre y el directorio donde se almacenará.

RF32. Adicionar workflow: el usuario con privilegios podrá adicionar un workflow insertando las tareas, el estado de ellas, fecha de inicio, de modificación y de fin.

RF33. Modificar workflow: un usuario con privilegios podrá modificar un workflow siempre que introduzca los datos pertinentes.

RF34. Adicionar clasificación de tagging: un usuario con privilegio podrá adicionar una clasificación introduciendo el nombre de la clasificación.

RF35. Modificar clasificación de tagging: un usuario con privilegios podrá modificar una clasificación actualizando el nombre de esta.

RF36. Eliminar clasificación de tagging: un usuario con privilegios podrá eliminar clasificación especificada de las previamente listadas.

RF37. Adicionar tagging: un usuario con privilegios podrá insertar un nuevo tagging especificándole su clasificación.

RF38. Modificar tagging en una clasificación: un usuario con privilegios podrá actualizar de un tagging datos como el nombre y la clasificación a la que pertenece.

RF39. Eliminar tagging: un usuario con privilegios podrá eliminar un tagging de los previamente listados.

2.1.1.2 Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son esas características que posibilitan que el producto sea atractivo, usable, rápido y confiable. Normalmente, están vinculados a requerimientos funcionales, es decir, una vez que 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.

Requerimientos de Software:

RNF1: Navegador Internet Explorer v6.0 o superior, Mozilla Firefox v2.x.

RNF2: Servidor Web Apache v2.x o superior con módulo PHP 5.2.3 disponible.

Requerimientos de Usabilidad:

RNF-5: El sistema debe permitir acceso a los usuarios.

Requerimientos de Rendimiento:

RNF-6: El sistema deberá ser capaz de gestionar toda la información y dar respuestas a las solicitudes en menos de 5 segundos.

2.2 Arquitectura de Software

La arquitectura de software es la organización fundamental de un sistema y tuvo sus primeras definiciones en el 1960 por el ingenio de Edsger Dijkstra donde manifestaba que se establezca una estructuración correcta de los sistemas de software antes de lanzarse a programar (Reynoso, 2004). Así se orientó el camino a lo que es hoy la arquitectura de software.

Por lo nuevo del término, existen diversas definiciones de arquitectura de software, este trabajo se centra en la dada por la IEEE 1471-2000, donde plantea: “... es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

De forma general se puede decir que la arquitectura de software abarca decisiones importantes sobre:

- ✓ La organización del sistema software.
- ✓ Los elementos estructurales que compondrán el sistema, sus interfaces, comportamiento y colaboración.
- ✓ Evolución de los elementos estructurales en subsistemas.
- ✓ El estilo de arquitectura.

No podemos descartar los disímiles conceptos que existen a través de este término. Para detallar un poco se tratarán algunos conceptos de los que depende la arquitectura como son los estilos arquitectónicos, donde se asumirá la clasificación realizada por Reynoso: estilos de flujo de datos, centrados en datos, de llamada y retorno, de código móvil, Peer-To –Peer, arquitectura basada en eventos y arquitecturas orientadas a servicios (Reynoso, 2004). Para la propuesta de solución se asumirá la clasificación “Estilos de Llamadas y Retorno”, dado que el mismo enfatiza en la modificabilidad y escalabilidad. Este estilo arquitectónico incluye: arquitectura basada en capas, arquitectura basada en componentes, arquitectura Modelo – Vista – Controlador y arquitectura orientada a objetos, que han sido seleccionado por presentar ventajas como:

Modelo-Vista-Controlador (MVC)

Destaca la separación entre interfaz, lógica de negocio y de presentación. Evita poner el código indebido en la capa impropia, facilitando el despliegue en caso de modificaciones en el modelo de datos. Su sencillez para crear distintas representaciones de los mismos datos le dan una connotación positiva incorporándole la característica de reutilización de los componentes, simplicidad en el mantenimiento de los sistemas y la escalabilidad en su desarrollo.

Arquitectura en Capas

Brinda modularidad al sistema, facilitando la localización de errores, cualidad que mejora el soporte del sistema. Permite la reutilización de capas y a su vez la contención de cambios en ellas.

Arquitectura Basada en Componentes

Una de las características a destacar de los componentes es que son reutilizables y no son separables de la arquitectura (Jonás A., 2008). La ventaja de una arquitectura basada en componentes es la utilización de funcionalidades ya probadas que no tuvieran errores y eso permitiría la incorporación de

las mismas para la solución. De aquí, que tiende a tener reusabilidad contribuyendo en gran medida a la extensibilidad.

Arquitectura Orientada a Objeto

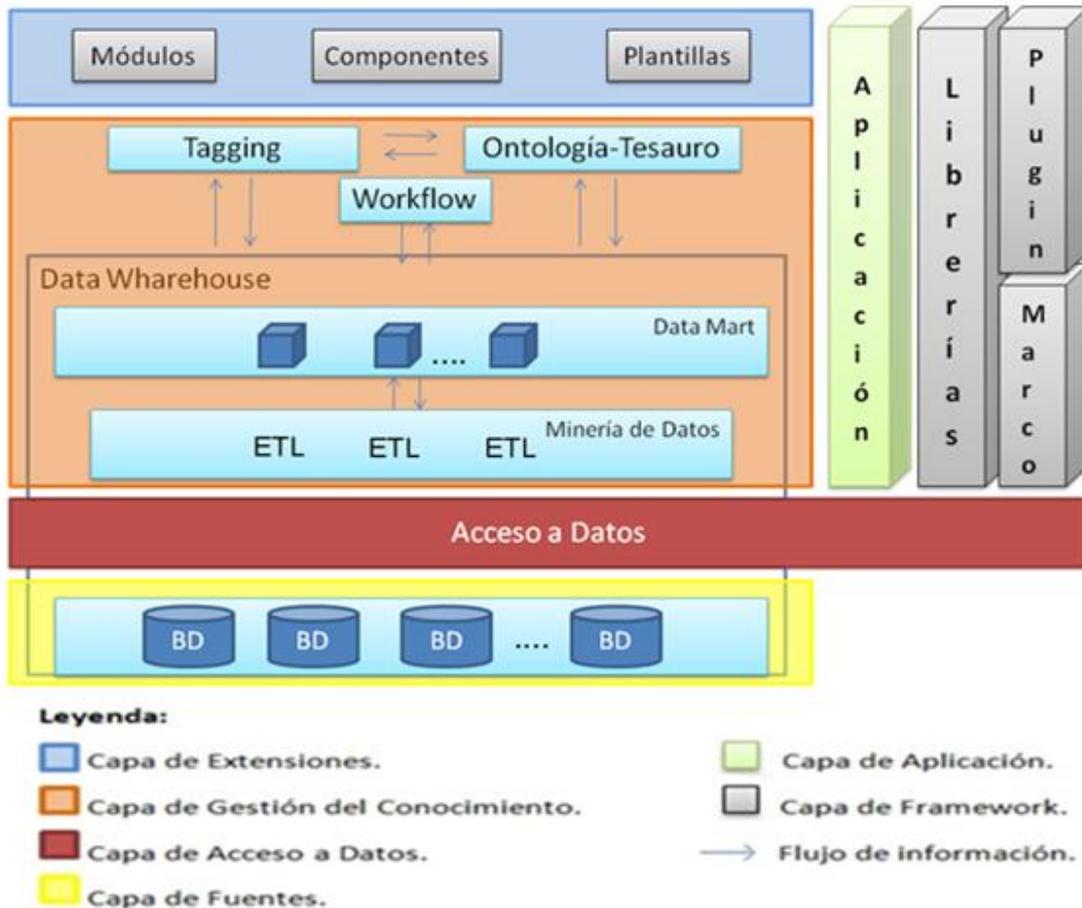
La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

2.3 Análisis de una propuesta de arquitectura

Para la propuesta brindada, a través de un estudio de diferentes procesos que proporcionen una gestión del conocimiento, se vincularon elementos que asocian los modelos clásicos de gestión de conocimiento y las herramientas que soportan dichos procesos, entre ellos los sistemas de gestores de contenidos, por facilitar componentes que pueden servir como base para el desarrollo de la propuesta con una mayor funcionalidad en la gestión de conocimiento.

La estructura está dada por un estilo de capas basado en componentes donde la arquitectura de capas es recomendada para construir aplicaciones distribuidas, escalables y los componentes son una vía para soluciones flexibles y de fácil mantenimiento que permitirá la extensibilidad y la operabilidad del sistema, mostrada en la **Figura 9**.

Figura 8 Propuesta de Arquitectura



Fuente: Elaboración propia

La capa de extensiones está compuesta por:

Plantillas: son responsables de cambiar la forma en la que será vista la aplicación.

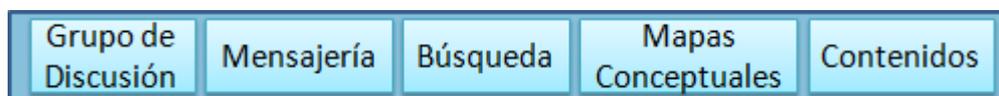
Módulos: que son elementos del sistema que muestran información en diferentes zonas de la plantilla.

Componentes: son programas independientes que pueden ser instalados posteriormente desde el panel de administración. Cada uno de ellos estructurados bajo la implementación del patrón arquitectónico MVC.

Mediante la estructura de esta capa se garantiza un sistema adaptable a diferentes ámbitos. Con la perspectiva orientada a la integración de nuevos componente se gana en modularidad y extensibilidad.

Dentro de los componentes propuestos para la solución se encuentran:

Figura 9 Componentes de la Capa de Extensiones



Fuente: Elaboración propia

Estos componentes (Grupo de discusión, Mensajería Instantánea, Búsqueda y Contenidos) que sirven de apoyo a la gestión de conocimiento son brindados por la comunidad de desarrollo de Joomla, se asumen como elementos decisivos para la propuesta, por lo que se le incorporan, con ajustes pertinentes, reutilizando el código por las ventajas que proporciona.

Los contenidos se distinguen a través de *tagging* como un atributo adicional y obligatorio a la hora de insertarlos, posibilitando posteriormente clasificarlos, procesarlos y almacenarlos en un *Data Warehouse*.

Mensajería instantánea: concebido desde la perspectiva de intercambio de información en tiempo real, importante para aclarar conceptos y generar conocimiento.

Búsqueda: realizado para la localización de contenidos disponibles sobre una palabra o frase en específico. Tiene una estrecha conexión con la capa de gestión del conocimiento en el mero hecho de que la operación Buscar, se implementa además con la utilización del parámetro *tagging*. Dichas búsquedas serán a través de estas palabras clave.

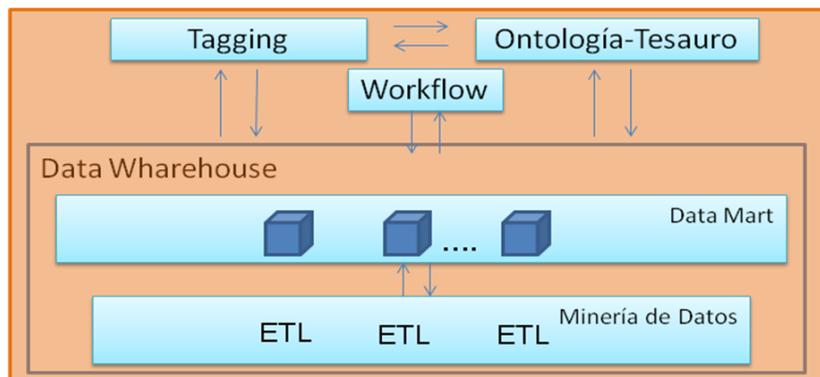
Grupo de discusión: a través de temas de discusión y los comentarios asociados a cada tema. Es una técnica para recopilar información relevante sobre un tema de investigación. De esa manera, se logra una especie de retroalimentación entre los distintos sujetos que son parte del grupo de discusión.

Creación de mapas del conocimiento: brinda la posibilidad de relacionar conceptos por su semántica. Está formado por un conjunto de tesauros capaz de generar un árbol de jerarquía con la que se relacionarán dichos conceptos. Este componente tendrá asociado una tabla en la base de datos donde se harán las consultas para extraer los tesauros correspondientes. Interactúa con el componente de Ontología y Tesauro de la capa de gestión del conocimiento.

Dicha capa de extensiones, mediante los contenidos y los mapas de concepto, se encuentra en constante retroalimentación para poder reutilizar el conocimiento guardado en los documentos,

necesitando herramientas de clasificación, búsqueda, almacenamiento y extracción de conocimiento. Por lo que se propone la capa de gestión del conocimiento, aportando tareas complejas que se agruparán en subsistemas.

Figura 10 Capa de Gestión del Conocimiento



Fuente: Elaboración propia

En la Capa de Gestión de Conocimiento se encuentra una relación de subsistemas que interactúan de manera combinada para responder a los requisitos del sistema:

Workflow: se ocupa de la circulación de información electrónica dentro del sistema, con notificaciones que mantienen actualizados a los usuarios sobre los diferentes eventos asociados a la gestión del conocimiento. Implementado con herramientas encargadas de la definición de los procesos y con aplicaciones que permiten el envío automático de notificaciones (fax y correo).

Minería de datos (DM, Data Mining): los sistemas Data Mining se desarrollan bajo lenguajes de última generación basados en la Inteligencia Artificial y utilizan métodos matemáticos tales como: Redes Neuronales principalmente para clasificar resultados. Estos sistemas están asociados a los algoritmos de extracción, transformación y carga, que necesita un Data Warehouse para el filtrado y la simplificación de redundancia en los datos.

Tagging: una estrategia de clasificación para ordenar y clasificar un determinado contenido mediante una o más palabras clave. Tiene asociado un componente Gestionar tagging encargado de manipular la clase tagging.

Ontología-Tesoro: los tesauros pueden ser usados para representar de forma explícita el significado de términos pertenecientes a un vocabulario a través de relaciones de sinonimia y antonimia mediante

las referencias entre ellos. Esto se logra con la asociación de un componente que permitirá generar un tesoro teniendo en base de datos listados de términos predefinidos, descriptores y sus jerarquías.

Data warehouse (DWH): se crea al extraer datos desde una o más bases de datos de aplicaciones operacionales. El proceso comienza con la conformación del DWH con archivos habituales, ya sea archivos de texto, hojas de cálculos, bases de datos transaccionales, imágenes y videos, que son llamados OLTP (Procesamiento Analítico Transaccional por sus siglas en inglés), los cuales almacenen datos históricos. Para poder extraer los datos desde la colección de OLTPs, para luego manipularlos, integrarlos, transformarlos y posteriormente cargar los resultados obtenidos en el DWH, es necesario contar con algún sistema que se encargue de ello, usando para ello los ETL (Extracción, Transformación y Carga por sus siglas en inglés) dentro de los que se encuentran métodos asociados a la minería de datos (Ver Minería de Datos). Los ETL extraen los datos de las diversas fuentes que se requieran, los transforman para resolver posibles problemas de inconsistencias entre los mismos y finalmente, luego de su depuración se procede a la carga en el depósito de datos.

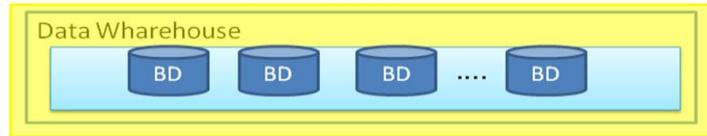
La Capa de Aplicación incluye cuatro utilidades fundamentales, las cuales responden a la instalación, administración, administración remota y la visualización de la parte frontal del sistema con la recodificación de lo propuesto por Joomla en su definición.

Capa Framework: almacena todas las librerías, plugins y clases necesarias para el uso del framework por otros desarrolladores.

Capa de Acceso a Datos: pretende encapsular las especificidades del proveedor de datos a la siguiente capa. Doctrine es un framework que cumple perfectamente con el objetivo de este nivel, dígase trabajo con procedimientos almacenados y métodos de persistencia o consultas. Esto facilita que el sistema pueda acceder a diferentes bases de datos. Inicialmente se propone clases para MySQL y PostgreSQL.

La Capa de Fuentes importante para almacenar todo el cúmulo de información disponible, con el propósito del posterior filtrado. Son bases de datos que se encuentran distribuidas y no precisamente en un mismo lugar físico.

Figura 11 Capa de Fuentes



Fuente: Elaboración propia

Extiende su función a lo que es una verdadera Biblioteca de Recursos online donde contiene información digital de todo tipo que se puede agregar, buscar, modificar y eliminar.

2.3.1 Patrones de Diseño Implementados

Existen diversas definiciones a lo que son los patrones arquitectónicos pero es acogido la definición que propone Buschmann en el marco de la investigación donde se ven: *“como descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. Provee un conjunto de subsistemas predefinidos, especificando sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Son plantillas para arquitecturas de software concretas, que especifican las propiedades estructurales de una aplicación (con amplitud de todo el sistema) y tienen un impacto en la arquitectura de subsistemas.”* La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de software (Reynoso, 2004).

A continuación se mencionan algunos patrones, para el desarrollo de la futura aplicación en una primera iteración de la propuesta:

Modelo-Vista-Controlador

Modelo Vista Controlador (MVC, en lo adelante) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El controlador: es el punto de entrada de la aplicación, se mantiene a la escucha de todas las peticiones, ejecuta la lógica de la aplicación, y muestra la vista apropiada para cada caso.

El modelo: contiene todo el código relacionado con el acceso a datos. Un código genérico que permite ser utilizado en otras situaciones y proyectos. Nunca se incluye lógica en el modelo, solamente consultas a la base de datos y validaciones de entrada de datos.

La vista: contiene el código que representará lo que se verá por pantalla, en este caso se trata de código HTML.

Para el desarrollo del sistema se propone el uso de este patrón de diseño por las ventajas que el mismo proporciona y las facilidades con que dota al CMS Joomla, teniendo en cuenta además su integración al mismo.

Singleton (Instancia única)

El patrón Singleton es uno de los más sencillos patrones de diseño, y es útil para limitar el máximo número de instancias de una clase en exactamente solo una. En este caso, si más de un objeto necesita utilizar una instancia de la clase Singleton, esos objetos comparten la misma instancia de la clase Singleton (Welicki).

Directamente a través de un método declarado como público y estático, que será el encargado de crear la instancia de la clase.

Observer (Observador)

Clasifica como patrón de comportamiento a nivel de objeto, el problema que lo motiva es la existencia de diferentes objetos desacoplados que deben mantenerse actualizados del estado de otro objeto, o de los determinados sucesos que ocurren en él, su propósito es garantizar que los interesados en el estado del objeto observado sean notificados convenientemente. Es utilizado en el diseño de componentes de interfaz de usuario de la capa de presentación.

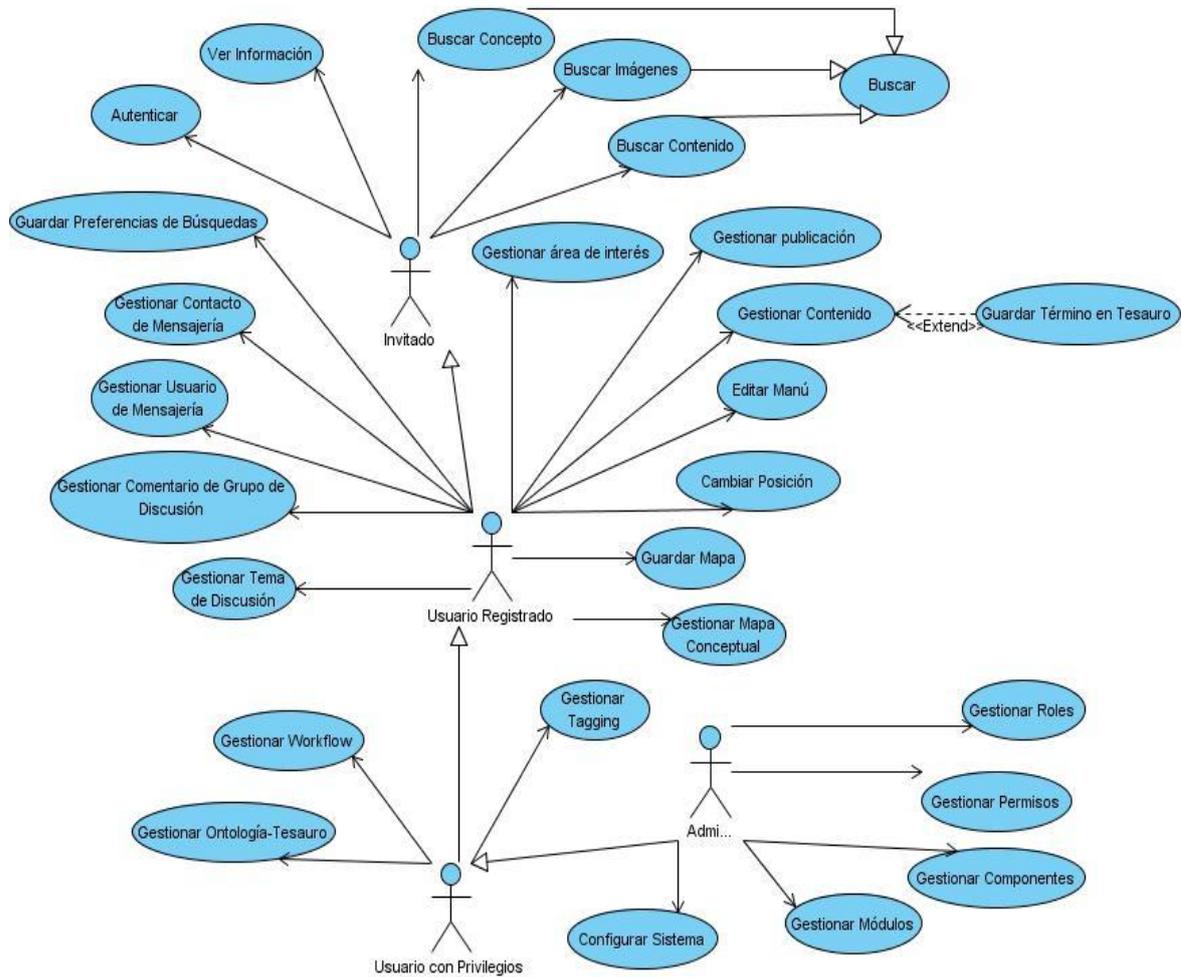
2.4 Vistas de la Arquitectura Propuesta

Se presenta la Arquitectura fundamentalmente a través de las vistas propuestas por RUP utilizando UML.

2.4.1 Vista de Casos de Uso

El modelo de caso de uso del sistema describe la funcionalidad propuesta de la nueva arquitectura a desarrollar, muestra la interacción entre los actores del sistema y los procesos a automatizar, representados a través de casos de uso y que responden a los requisitos funcionales del mismo.

Figura 12 Vista de Caso de Uso



La vista de caso de uso se dividirá por los actores iniciales que se proponen para el sistema, siendo los casos de usos arquitectónicamente significativos los que encapsulan parte de la funcionalidad central del sistema para dar apoyo a la gestión del conocimiento.

Casos de uso arquitectónicamente significativos

Uno de los roles que se proponen como iniciales es el de Invitado, que no es más que un usuario que tendrá una navegación básica con funcionalidades restringidas. Este usuario tendrá la posibilidad de realizar un conjunto de actividades tales como:

CU Autenticar: corresponde a un procedimiento para comprobar la identidad de un usuario para acceder a un determinado recurso o actividad.

CU Ver Información: tiene relación con la navegación básica y la obtención de la información disponible que se le brinda a un usuario sin autenticar.

CU Buscar: es una funcionalidad, donde el sistema devuelve de manera rápida la información necesaria, conociendo un determinado criterio. La búsqueda puede acotarse a contenidos, imágenes y conceptos, siendo los criterios básicos opcionales: buscar por tagging y en áreas específicas.

CU Buscar Conceptos: devuelve, a partir de una palabra o frase, el concepto asociado en caso de existir.

CU Buscar Imágenes: la búsqueda devuelve las imágenes con un vínculo al contenido asociado a la misma.

CU Buscar Contenido: mostrará todos los contenidos donde esté la palabra o frase buscada.

CU Buscar en área específica: caso de uso donde la dimensión de la búsqueda se reduce al contenido asociado a un área en específico.

CU Buscar por tagging: la búsqueda donde se seleccione este criterio, contemplará solo aquellos contenidos, imágenes o conceptos que tengan asociados la palabra clave como etiqueta.

Otro rol para el desarrollo del sistema es el usuario registrado, donde tendrá la posibilidad de hacer todas las actividades que realiza un invitado pero se le agregan nuevas por su condición de usuario del sistema. Aclarar que los casos de uso que se especifican se contemplan como subsistemas ya que involucran varios casos de uso dentro de ellos. Especificando dichas actividades:

CU Guardar preferencias de búsqueda: guardará la palabra clave por la que se realizó la búsqueda y el vínculo al documento seleccionado de los encontrados.

CU Gestionar Contenido: incluye casos de uso que cumplen el patrón de requisitos Create, Read, Update, Delete.

CU Gestionar Área de Interés: este caso de uso permite gestionar el trabajo con las áreas, las cuales abarcan contenidos referentes a temas comunes.

CU Gestionar Publicación: este caso de uso le permite al usuario gestionar una publicación de cualquier tipo de información, referente a un área específica de interés, teniendo la posibilidad de guardar término en tesoro.

CU Guardar término en Tesauro: permite que se incorpore o no un determinado término dentro de una publicación a un tesauro específico.

CU Gestionar Usuario de Mensajería: este caso de uso es el responsable de crear un usuario personal con datos que se requiere para ello.

CU Gestionar Contacto de Mensajería: posibilita gestionar los usuarios que se deseen, incluyendo una búsqueda del contacto que se va a añadir para verificar su existencia.

Se pueden incorporar otros casos de uso como editar perfil, cambiar contraseñas y crear salas de discusión, ya que estos no son imprescindibles para el trabajo con el subsistema de apoyo a la gestión del conocimiento.

CU Gestionar Tema de Discusión: este caso de uso manipula todo lo relacionado con temas a debatir para los usuarios.

CU Gestionar Comentario: acción para usuarios registrados en el sistema con el objetivo de responder a un tema en específico.

CU Gestionar Mapa: es un caso de uso que se encarga del diseño y eliminación de los mapas, incluye una búsqueda previa de los mismos.

CU Guardar Mapa: brinda la funcionalidad de guardar un mapa con una extensión que permita posteriormente cargarlo.

CU Editar Menú: aquí el usuario podrá determinar qué elementos quiere o no tener en los menús, ya sea el lateral u horizontal.

CU Cambiar posición: este caso de uso permitirá al usuario colocar los componentes de la vista principal donde le sea más visible para el trabajo.

Importante para el sistema es el rol de usuario con privilegios, el que permitirá hacer tanto lo que hace el usuario registrado como sus propias actividades. Este usuario se encargará de:

CU Gestionar workflow: relaciona la asignación de workflows a usuarios y contenidos determinados.

CU Gestionar tagging: en este caso de uso se permitirá hacer lo referente a la gestión de clasificaciones de etiquetas y la asociación de etiquetas a una clasificación determinada.

CU Gestionar Ontología-Tesoro: trabajará con tesauros básicos, añadiéndole términos y jerarquizándolos. Incluye revisar los tesauros existentes o crear nuevos.

El rol que más peso tiene es el administrador, ya que posee un control total sobre los componentes que integran el sistema.

CU Gestionar Usuario: relaciona la gestión de usuarios que tendrán acceso al sistema.

CU Gestionar Módulos: da la posibilidad de habilitar o deshabilitar nuevos módulos en la medida de las necesidades funcionales de la aplicación.

CU Gestionar Roles: en este caso de uso el sistema permite otorgar roles para el cumplimiento de determinadas actividades.

CU Gestionar Permiso: gestiona permisos necesarios para la realización de actividades en el sistema a usuarios existentes.

CU Gestionar Componente: gestiona la incorporación de nuevos componentes en el sistema.

CU Configurar Sistema: este caso de uso interviene para darle personalización al ambiente del sistema.

2.4.2 Vista Lógica

La vista lógica proporciona una visión abstracta entre la dependencia de los estilos empleados y la distribución de los componentes en ellos a nivel de paquetes, subsistemas y clases significativas.

Se presenta a través de tres niveles de arquitectura, cada uno de los cuales corresponde a un refinamiento del anterior. El primer nivel describe el estilo arquitectónico de la solución y el segundo nivel especifica la composición en módulos y la relación entre ellos.

Estilos arquitectónicos

Atendiendo a los objetivos y restricciones vistos anteriormente se optan por los siguientes estilos que tipifican la arquitectura del sistema:

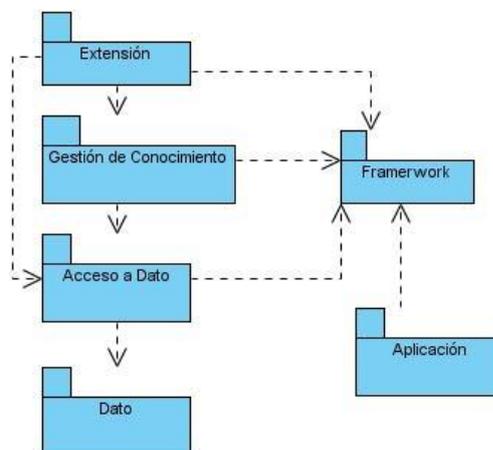
- ✓ *Cliente Servidor⁶*: se caracteriza por existir un nodo (o más) servidor donde reside el servicio que se expone y varios clientes que consumen dicho servicio. En el sistema este estilo responde al hecho de que se trate de una aplicación Web y se concreta con un servidor de bases de datos, un servidor Web (con función de servidor de aplicaciones, Apache) y varios clientes que acceden al sistema a través de un navegador.

Figura 13 Estilo Cliente-Servidor



- ✓ *Multicapas*: se caracteriza por organizar los componentes del sistema en capas con responsabilidades bien definidas y donde las capas del nivel más alto invocan los servicios de las del nivel inferior, en el sistema es posible identificar las siguientes capas: extensión, gestión de conocimiento, framework, aplicación, acceso a datos y datos.

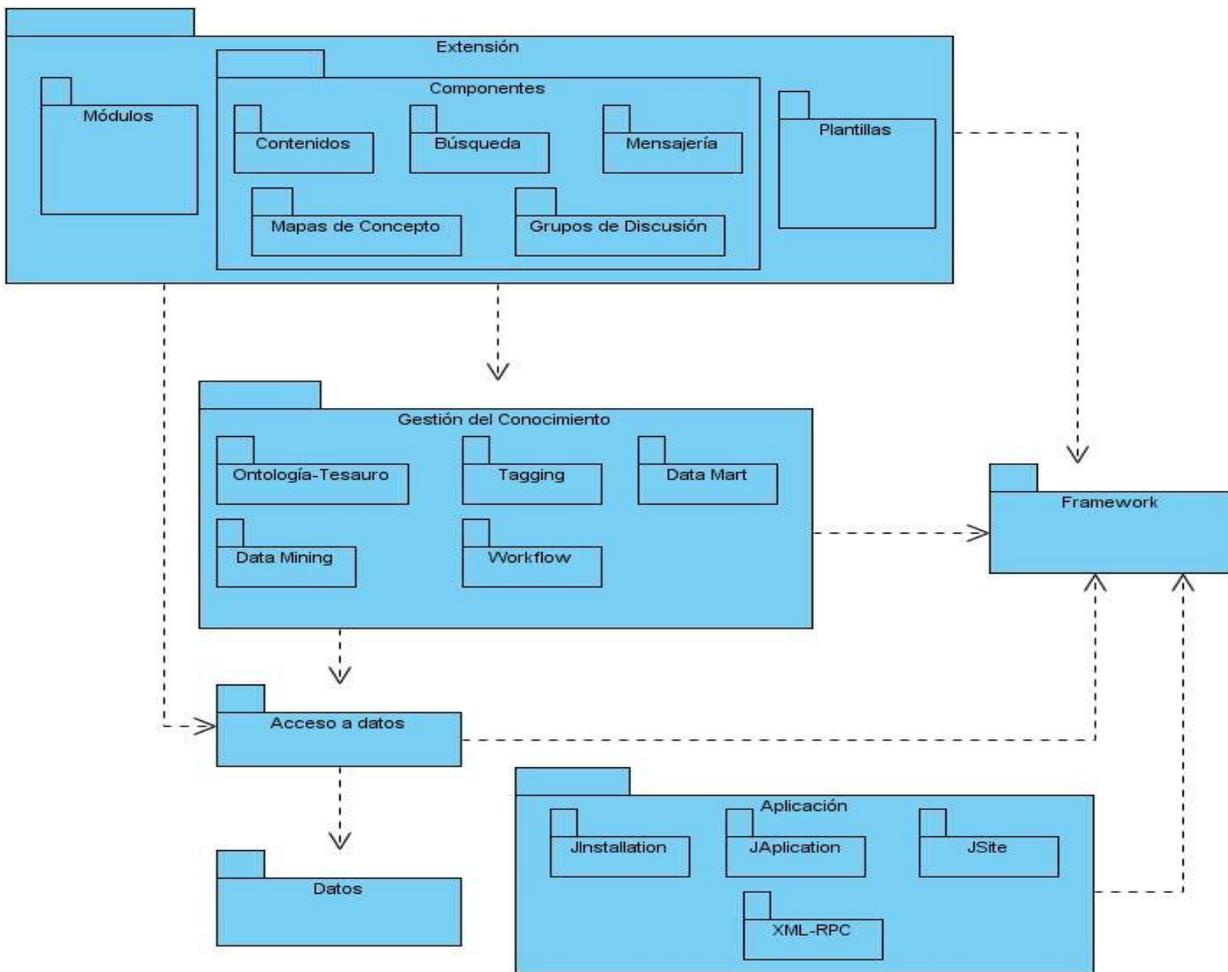
Figura 14 Estilo Multicapas



Para mayor comprensión de la vista lógica, se muestran los componentes en cada una de estas.

Figura 15 Vista Lógica de las capas principales para el apoyo a la Gestión del Conocimiento

⁶ Cliente servidor es una especialización del estilo en capas, se presenta como un estilo más por una cuestión de entendimiento y de que resulta coherente con algunas posiciones arquitectónicas presentes en la bibliografía respectiva, es necesario destacar que no está entre los estilos presentados en el capítulo 1 del trabajo.

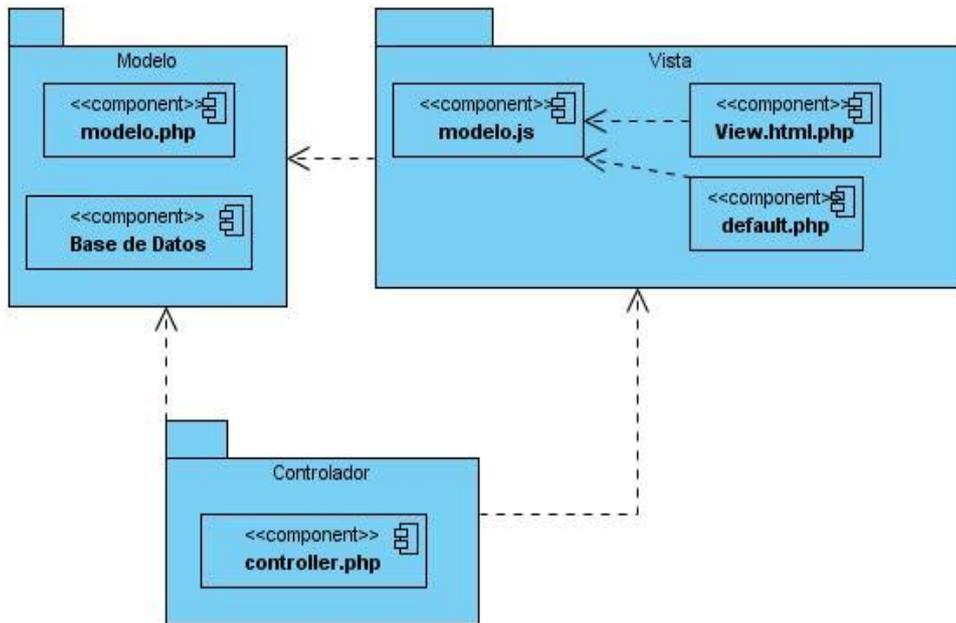


En la **Figura 16** está representados los subsistemas que permitirán un apoyo al proceso, cada uno de ellos encapsula sus correspondientes funcionalidades. Así mismo se aprecia la presencia del Framework y de Aplicación que contemplan funcionalidades de soporte al sistema. Visto la generalidad se hace necesario entrar en las particularidades de cada capa.

Capa de Extensiones

Los paquetes de la capa de extensiones responden al desarrollo empleando el patrón MVC donde separa la lógica de la vista y el modelo se encarga de la persistencia de la información.

Figura 16 Vista de Implementación de un componente en Joomla



Los paquetes de Búsqueda, Grupo de Discusión, Mapas de Concepto y Mensajería, contiene las interfaces de usuario y los formularios, como se había comentado mediante el desarrollo del patrón MVC, de dichos componentes.

Capa de gestión del conocimiento

En la capa de Gestión del Conocimiento los paquetes están vistos como subsistemas ya que involucran grandes responsabilidades y asocian varias clases para dar respuesta al apoyo que necesitamos para la gestión del conocimiento.

El paquete de Ontología-Tesoro contiene clases persistentes con valor para poder utilizar los tesauros.

El paquete de Tagging se corresponde con las clases para asignar los tagging a los contenidos y los atributos necesarios para realizar la búsqueda mediante este criterio.

Data Mining es un paquete que responde a métodos implementados bajo Inteligencia Artificial que permitirán el análisis de los datos de manera eficiente para la toma de decisiones.

Data Mart son aquellos datos que se guardan de manera dimensional, organizados y de donde actúan los métodos de Data Mining.

El paquete de Workflow contiene lo relacionado con el flujo de los datos, sus revisiones, procesos y tareas.

Capa de Acceso a Datos

La capa de acceso a datos estará conformada por aquellas clases del framework Doctrine que permitirán las conexiones con diferentes bases de datos. El trabajo real, es mayormente reenviado a otros componentes como la clase Doctrine_Table. Esta clase tiene la típica interfaz de mapeo de datos con métodos como: createQuery(), find(id), findAll(), findBy*(), findOneBy*().

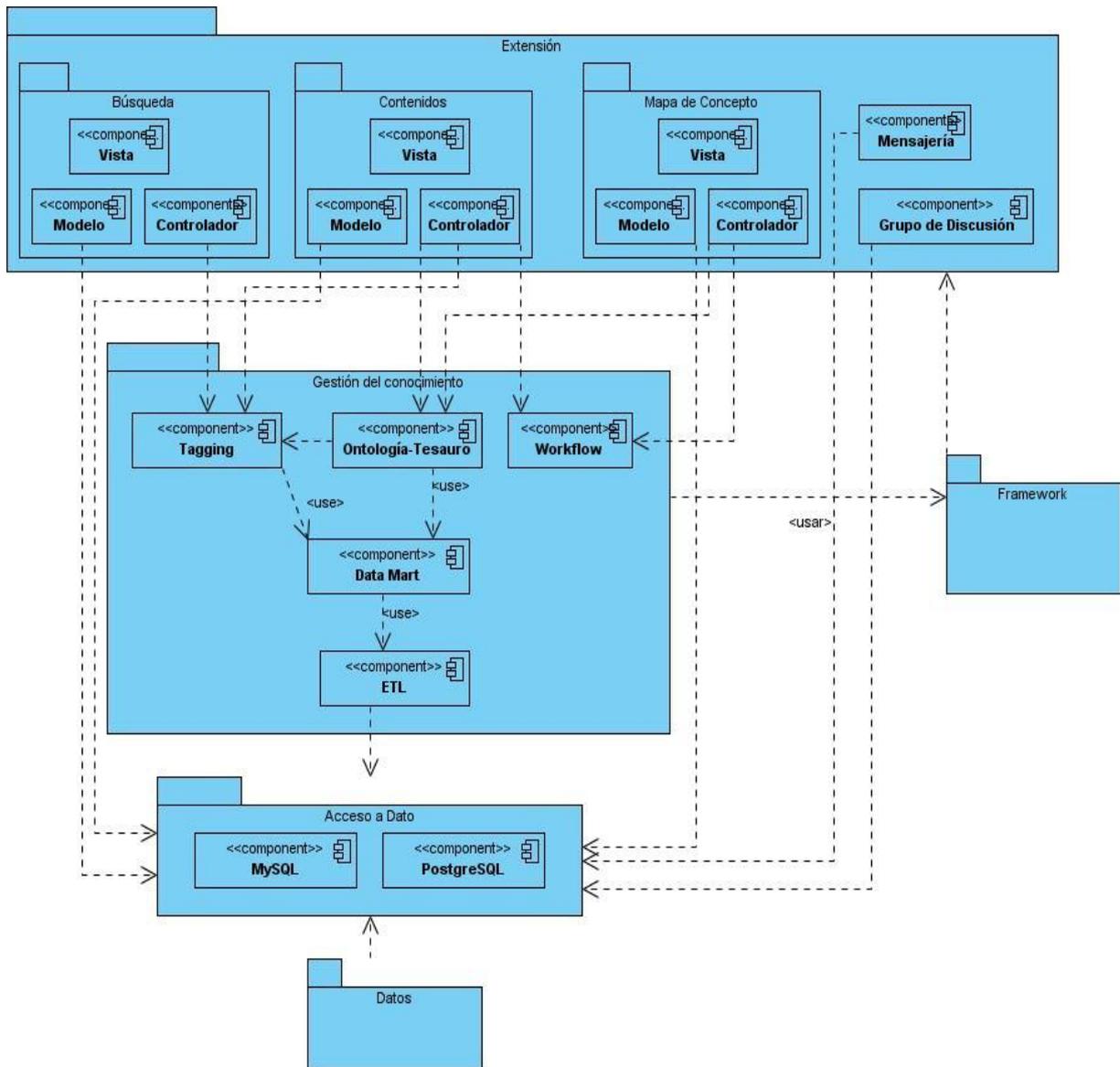
En la presente vista se resuelve la interacción de la Capa de Extensión con la capa de Gestión del Conocimiento e igualmente con la de Acceso a Datos según la implementación que se propone de los componentes.

2.4.3 Vista de Implementación

Dicha vista se centra en la organización real de los módulos en el ambiente de desarrollo del software. Describe la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación.

Para una primera iteración dicha vista de basa fundamentalmente en los rediseños que se incorporan en los componentes ya creados para el sistema (por permitir Joomla esta funcionalidad) y en los de la capa de Gestión de Conocimiento que apoyen dichos componentes.

Figura 17 Vista de Implementación



En la figura se evidencian las relaciones que tiene los componentes de la capa de extensiones con los de la capa de gestión del conocimiento.

La relación del componente de Búsqueda con el tagging es mediante la redefinición de su método básico *Buscar* para que sea capaz de la clasificación. Por su parte el controlador de Mapas de Concepto utiliza la Ontología-Tesoro para vincular la lista de término que devuelve un método de este subpaquete para relacionarlo mediante la graficación visual. En el controlador de contenido existe algo similar cuando se almacena algún contenido al sistema, ya que a este se le asocia un tagging para

facilitar la búsqueda, este tagging también está relacionado con un tesoro por medio de estas palabras clave. El contenido además tiene asociado un flujo de actividades para su publicación o simplemente para tareas asociadas a este por lo que se necesita el componente de Workflow.

Para el buen filtrado de la información, se cuenta con un método de Extracción, Transformación y Carga (ETL), que estará implementado por métodos de Inteligencia Artificial permitiendo la clasificación de la información y el almacenamiento en el Data Mart con datos bien específicos, los componentes que hacen uso de este son el de Tagging y el de Ontología-Tesoro.

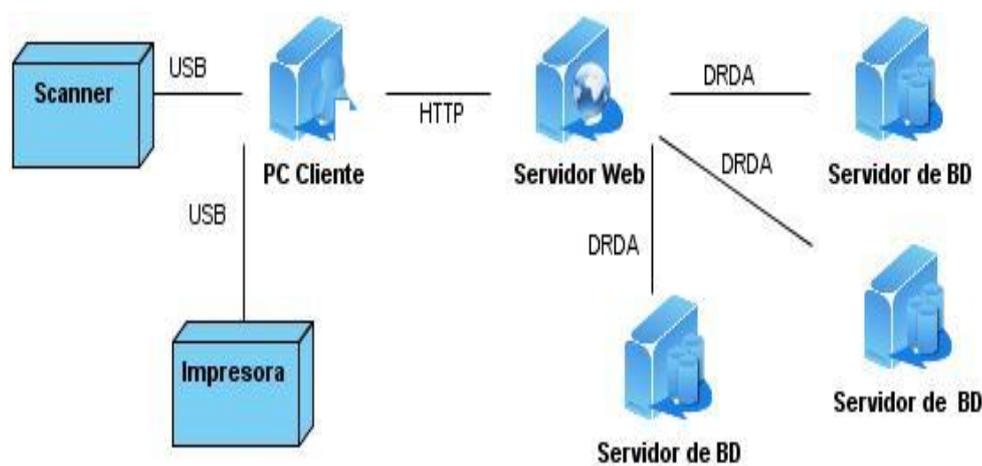
El acceso a las bases de datos incluye clases para cada uno de los gestores con que el sistema podrá comunicarse, inicialmente se cuenta con el Framework Doctrine que brinda la abstracción necesaria.

Con relación a la figura mostrada anteriormente se aprecian una serie de elementos que encapsulan las funcionalidades, que a su vez se agrupan en otros componentes con clases que se relacionan entre sí para conformar la utilidad del subsistema.

2.4.4 Vista de Despliegue

Debido a que el sistema de Gestión de Conocimientos maneja mucha información, está diseñado para que se instale una instancia del sistema en servidores y sea utilizado de acuerdo con sus necesidades teniendo en cuenta que su acceso es través de un navegador.

Figura 18 Vista de despliegue



El sistema se ha diseñado para que sea una aplicación distribuida por las facilidades que brindan, a su vez de estar conectadas a bases de datos relacionales distribuidas con diferente tipo de información mediante Arquitectura Distribuida de Base Datos Relacional (DRDA, por sus siglas en inglés), que es

un sistema de protocolos, o de reglas, que permiten a un usuario tener acceso a los datos distribuidos, sin importar donde residen físicamente. Proporcionando métodos para coordinar la comunicación entre localizaciones distribuidas. Esto permite que se tenga acceso a tablas alejadas en múltiples y varias localizaciones, y que al usuario del otro extremo le parezca como si estuvieran todas en un mismo ente lógico.

Impresora: este dispositivo garantiza las funcionalidades de impresión de reportes y documentos que genere el sistema en dependencia de las necesidades del usuario y la utilización de este a través de la red.

Scanner: hace posible la conversión a formato digital de cualquier documento impreso o escrito, en forma de imagen, importante para almacenar documentos legítimos que luego puedan ser consultados.

PC Cliente: este nodo se corresponde con la máquina donde se hará uso del sistema, recalcar que será una aplicación web y como requerimiento necesario que contenga un navegador web.

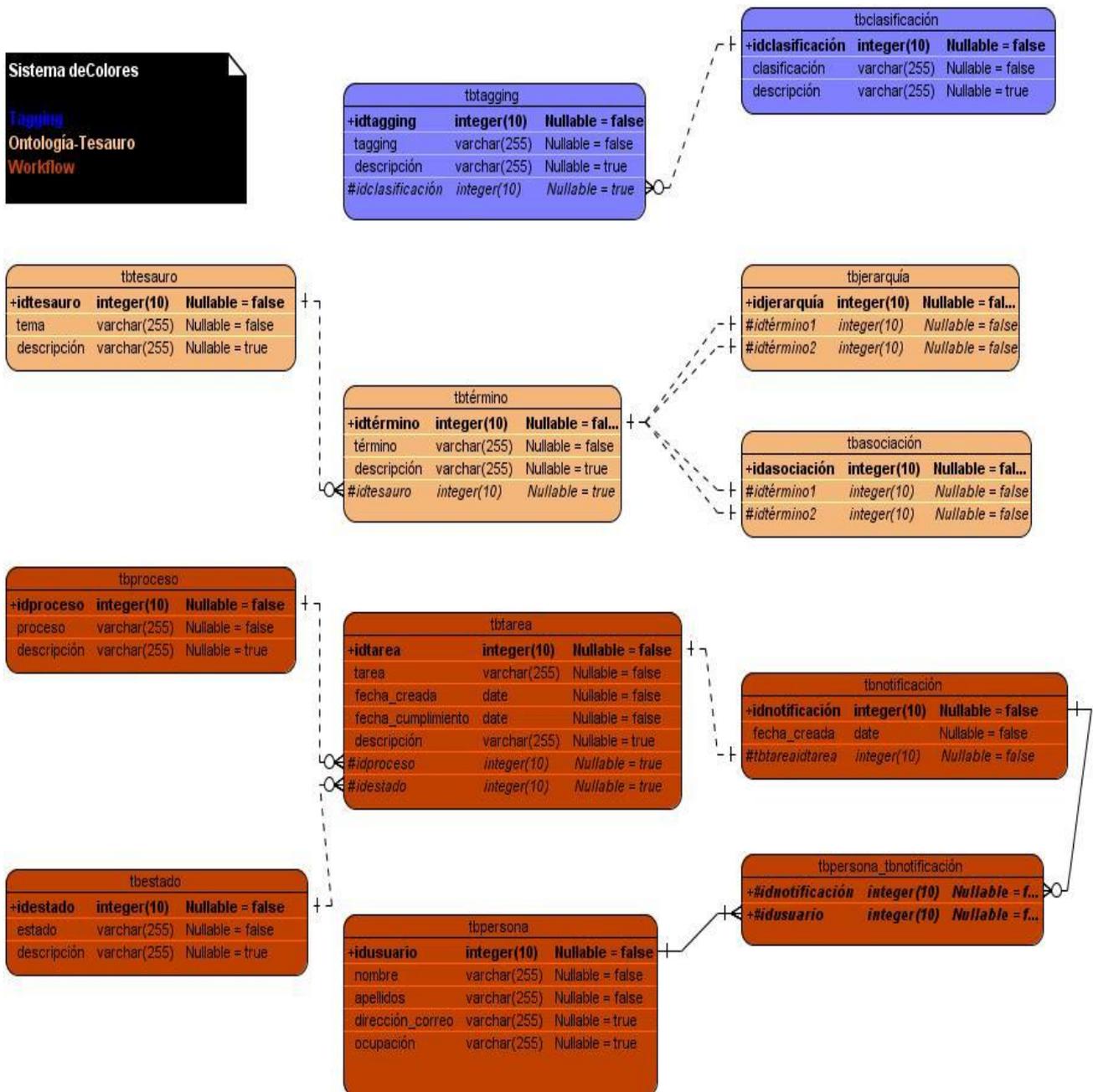
Servidor Web: es el encargado de tener instalado el sistema al que tendrán acceso las PCs Clientes, incluye en sus requerimientos cualquier sistema operativo.

Servidor BDs: estos servidores contienen información mixta de la que tendrá acceso la aplicación mediante el protocolo DRDA, por encontrarse de forma aislada entre sí.

2.4.5 Vista de Datos

La siguiente imagen muestra las tablas de la base de datos donde se almacenan algunas, correspondientes a los nuevos componentes que se agregan para el apoyo a la gestión del conocimiento, dichas tablas están generadas por componentes en la capa Gestión del Conocimiento, respondiendo a las clases persistentes que se crean. Por la magnitud del sistema que se propone, dichas tablas constituyen una primera versión del modelo final de datos, esto seguirá evolucionando en las posteriores iteraciones de la propuesta.

Figura 19 Modelo de Datos



La tabla tbtagging almacenará los tagging o etiquetas que existen y que pueden ser asociados finalmente con los contenidos. Toda etiqueta tiene una clasificación (tbclasificación) y una clasificación en específico puede tener varios tagging.

Para el componente Tesouro se hace necesario la tabla *tbtesouro* donde almacenará los datos del mismo. Un tesouro contendrá una lista de términos guardados en *tbtermino*, con sus jerarquías en *tbjerarquía* y sus sinónimos en *tbasociación*.

Para hacer consistentes los datos que manipulará el componente a realizar workflow se requiere de varias tablas: *tbproceso* que almacenará los procesos a automatizar, estos a su vez están compuestos por tareas (*tbtarea*) con el estado (*tbestado*) de cada una de ella y notificaciones (*tbnotificación*) que se lanzarán cuando empiece o caduque alguna tarea y a que persona (*tbpersona*) de la empresa u organización se le enviaría esta notificación. Creándose la tabla *tbpersona_notificación* a consecuencia de una relación de muchos a muchos entre las dos últimas tablas mencionadas.

Como consideraciones de las vistas de la arquitectura, solo están descritas las que permiten en esta iteración un claro entendimiento del problema y la solución que se propone, para ello se tienen en cuenta, mediante la metodología utilizada, las que son obligatorias, eso implica que la vista de procesos, que es opcional, al nivel en que se encuentra el desarrollo de la arquitectura y la definición de dichos procesos, sea omitida.

2.5 Validación del Sistema

En el presente epígrafe se realizará la validación de la propuesta de arquitectura. Antes es necesario conocer los métodos que se emplearon, resultados obtenidos y estadísticas que verifiquen estos resultados.

¿Qué es una Evaluación?

Es un estudio de factibilidad que pretende detectar posibles riesgos, buscando recomendaciones para contenerlos (Gustavo Andrés Brey, 2005).

La diferencia entre evaluar y verificar es que la evaluación se realiza antes de la implementación de la solución. La verificación es con el producto ya construido (Gustavo Andrés Brey, 2005).

Objetivos de Evaluar una Arquitectura

El objetivo de evaluar una arquitectura es saber si puede habilitar los requerimientos, atributos de calidad y restricciones para asegurar que el sistema a ser construido cumple con las necesidades de los stakeholders (Gustavo Andrés Brey, 2005).

La evaluación de una arquitectura de software es una tarea no trivial, puesto que se pretende medir propiedades del sistema sobre la base de especificaciones abstractas, como por ejemplo los diseños arquitectónicos. Por ello, la intención es más bien la evaluación del potencial de la arquitectura diseñada para la extensibilidad. Las mediciones que se realizan sobre una arquitectura de software pueden tener distintos objetivos, dependiendo de la situación en la que se encuentre el arquitecto y la aplicabilidad de las técnicas que emplea. Algunos de estos objetivos son: cualitativos, cuantitativos y de máximos y mínimos teóricos (Erica Camacho, 2004).

La medición cualitativa se aplica para la comparación entre arquitecturas candidatas. Este tipo de medición brinda respuestas afirmativas o negativas, sin mayor nivel de detalle. Es la empleada en la investigación por el avance que se ha alcanzado en esta primera iteración, ya que es necesario medir el diseño antes de comenzar a implementar. La técnica que se emplea para dicha validación es la evaluación por criterio de expertos dentro de las técnicas de cuestionarios.

2.5.1 Validación por criterio de experto

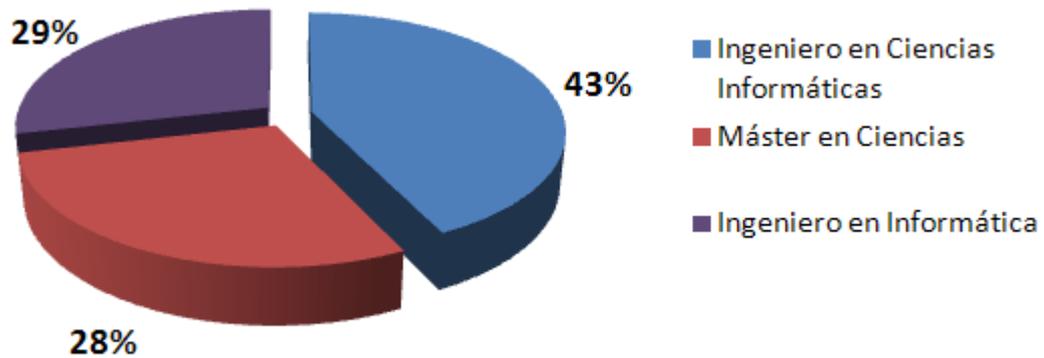
El criterio de expertos es un instrumento rápido y eficaz por el potencial que posee para conformar, valorar y enriquecer criterios, concepciones, modelos, estrategias y metodologías. Para validar la propuesta de arquitectura, se aplicó una encuesta (**Ver Anexo A**). Este instrumento, es una técnica de adquisición de información de interés, con un cuestionario previamente elaborado, permitiendo conocer la opinión y la valoración de los especialistas seleccionados en una muestra, sobre el asunto dado.

Nivel de competencia de los expertos

Para la selección de los expertos que harían la valoración de la propuesta de arquitectura se escogieron un conjunto de indicadores generales que permitieron obtener información más rica y actualizada. Estos fueron: la labor que desempeñan actualmente, la calificación profesional y la categoría docente y científica; cuidando que existiera una mayor representatividad de los arquitectos de proyecto.

De los especialistas se conoce que tres trabajan en el Centro de Estudio de Software Educativo de Pinar del Río, uno de ellos en DATEC, otros dos como arquitectos en el polo CEGEL y otro vinculado a Sistemas de Gestión Hospitalaria. La categoría científica se ilustra en la figura siguiente.

Figura 20 Categoría científica de los expertos



Fuente: Elaboración propia

El coeficiente de competencia de los expertos obtenido fue de un 42.86% medio y un 57.14% alto, el coeficiente general fue alto con un valor de 0.85 (**Ver Anexo B**), un conocimiento promedio de 7.14, un coeficiente de conocimiento de 0.71 y un coeficiente de argumentación de 0.98.

Una vez analizadas las evaluaciones de cada criterio, se obtuvo en el 37.5% de los aspectos un criterio de muy adecuado y el 62.5% se obtuvo un criterio de bastante adecuado. Lo que permite validar la hipótesis planteada.

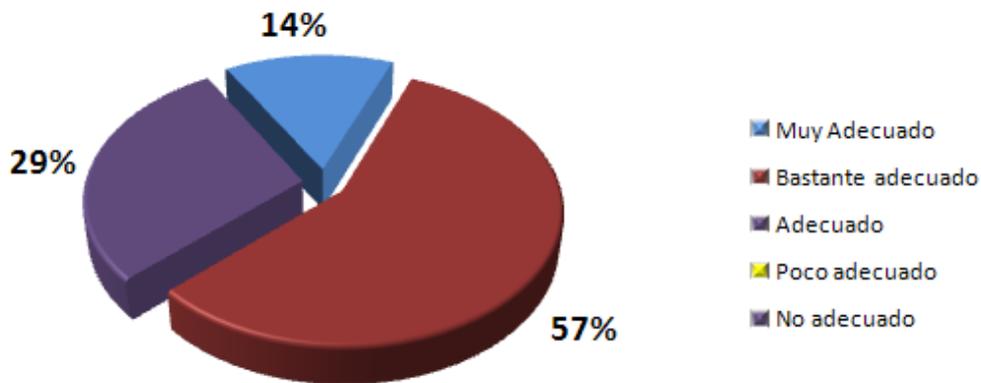
Tabla 2 Criterio obtenido para cada uno de los aspectos

Aspectos	Criterio
Calidad de la investigación.	Muy adecuado
Novedad Científica.	Muy adecuado
Aporte Científico.	Muy adecuado
Necesidad de la arquitectura propuesta.	Bastante Adecuado
Contribución al proceso de Gestión del Conocimiento.	Bastante Adecuado
Facilidades de utilización para gestionar conocimiento.	Bastante Adecuado
Posibilidad de uso al alcance de todos.	Bastante Adecuado
Posibilidades de extensión mediante la inclusión de nuevos componentes para la Gestión del Conocimiento.	Bastante Adecuado

Fuente: Elaboración propia

De forma general el 14% de los encuestados considera que la investigación es excelente y presenta una alta novedad científica, con aplicabilidad y resultados relevantes; otro 57% considera que la investigación es buena contando con novedad científica y el 29% considera que el resultado es aceptable, suficientemente bueno con reservas.

Figura 21 Evaluación final de la investigación



Fuente: Elaboración propia

2.6 Conclusiones Parciales

Mediante la realización de este capítulo se obtuvo como resultado:

- ✓ La especificación de los requisitos funcionales permitió definir los casos de usos y la arquitectura base del sistema.
- ✓ Se definió una arquitectura estable que permitirá la evolución de la aplicación sin afectar los componentes existentes.
- ✓ Se identificaron patrones, estilos arquitectónicos y los correspondientes elementos de la arquitectura respondiendo a un sistema flexible y escalable.
- ✓ Se realizó una validación de la propuesta mediante el criterio de expertos teniendo una aceptación de un 14% de excelente o Muy adecuado, un 57% de bueno o Bastante adecuado y un 29% de aceptable o Adecuado.

Conclusiones

Una vez obtenida la propuesta de arquitectura para una herramienta de apoyo a la gestión del conocimiento; fueron obtenidos resultados que permiten arribar a las siguientes conclusiones:

- ✓ Luego de la investigación realizada y el estudio de los principios, herramientas y modelos de la gestión del conocimiento, se obtuvo la base fundamental para la obtención de una arquitectura de apoyo a la gestión del mismo.
- ✓ A través de la arquitectura propuesta se logra un sistema flexible con los componentes que integran cada una de las capas, las funcionalidades e importancia que brindan a los clientes.
- ✓ A través del criterio de expertos se validó la solución, oscilando los resultados entre Muy Adecuado, Bastante Adecuado y Adecuado dando cumplimiento al objetivo planteado.

Recomendaciones

- ✓ Adicionar los restantes componentes para completar la propuesta de arquitectura dada por David J. Skyrmer.
- ✓ Continuar en próximas iteraciones actualizando el documento de Descripción de Arquitectura.

Referencias Bibliográficas

1. **2008.** Desarrollo Web. [En línea] 11 de 11 de 2008. [Citado el: 01 de 02 de 2010.] <http://www.desarrolloweb.com/articulos/que-es-un-cms.html>.
2. **2009.** Joomla! [En línea] 24 de 11 de 2009. http://api.joomla.org/li_Joomla-Framework.html.
3. **2009.** Mosaic tecnología y comunicación multimedia. [En línea] 23 de 12 de 2009. [Citado el: 01 de 02 de 2010.] <http://mosaic.uoc.edu/2009/12/23/una-aproximacion-a-los-gestores-de-contenidos-cms/>.
4. **A., Jonás A. Montilva y N. 2008.** *Desarrollo de Software basado en componentes*. Venezuela : s.n., 2008.
5. **Carrera, Juan Cristóbal López. 2006-2008.** *Guía sobre Gestión del conocimiento, Facultad de Filosofía y Letras*. Universidad Autónoma de Nuevo León : s.n., 2006-2008.
6. **Druker, Peter F. 1983.** *New Relatives*. 1983.
7. **Erica Camacho, Fabio Cardeso y Gabriel Nuñez. 2004.** *ARQUITECTURAS DE SOFTWARE (Guía de Estudios)*. 2004.
8. **Fernández, D.C. 2005.** *Gestión del conocimiento*. La Habana : s.n., 2005.
9. **Fernández, Dra. C. María Aurora Soto Balbón y Dra. C. Norma M. Barrios. 2009.** *Gestión del conocimiento. Parte I. Revisión crítica del estado del arte*. s.l. : Biblioteca Virtual de Salud en Cuba, 2009. 43.
10. **Fernández, M. A. Soto Balbón y N. M. Barrios. 2009.** *Gestión del conocimiento: Parte I. Revisión crítica del estado del arte. Biblioteca Virtual en Salud de Cuba*. 2009. 43.
11. **G. Booch, I. Jacobson y J. Rumbaugh. 1999.** *The UML Modeling Language User Guide*. s.l. : Addison-Wesley, 1999.
12. **Gómez, Salvador Gómez y Omar. 2007.** *Evaluando Arquitecturas de Software. Parte2. Métodos de Evaluación*. México : s.n., 2007. 1870-0888.
13. **Gómez, Salvador Gómez y Omar. 2007.** *Evaluando Arquitecturas de Software. Parte2. Métodos de Evaluación*. México : s.n., 2007. 1870-0888.
14. **Govín, D. C. 2005.** Wikilearning. [En línea] 10 de 05 de 2005. [Citado el: 27 de 01 de 2010.] http://www.wikilearning.com/monografia/propuesta_de_clasificacion_de_las_herramientas_software_para_la_gestion_del_conocimiento-el_software_para_la_gc/7727-19.
15. **Guerrero, I.I. 2003.** *Modelo para la creación de entornos de aprendizaje basados en técnicas de Gestión del Conocimiento*. Madrid, España : s.n., 2003.

16. **Gustavo Andrés Brey, Gastor Escobar, Nicolás Passerini y Juan Arias. 2005.** *Arquitectura de Proyectos IT. Evaluación de Arquitecturas. I.* Buenos Aires : Universidad Tecnológica Nacional. Facultad Regional de Buenos Aires, 2005.
17. [http://www.sei.cmu.edu/publications/documents/..](http://www.sei.cmu.edu/publications/documents/)
18. **José H. Canós, Patricio Letelier y M^a Carmen Penadés. 2004.** *Métodologías Ágiles en el Desarrollo de Software.* Universidad Politécnica de Valencia : s.n., 2004.
19. **K., Carlos Billy Reynoso y N. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* Buenos Aires : s.n., 2004.
20. **L. Bass, P. Clements y R. Kazman. 1998.** *Software Architecture in practice.* Addison-Wesley : s.n., 1998.
21. **Lentzm, Robin Schumacher y Arjen. 2008.** *MySQL: Dispelling the Myths.* 2008.
22. **M. Barbacci, M. Klein, T. Longstaff y C. Weinstock. 1995.** Software Engineering Institute. [En línea] 12 de 1995. [Citado el: 20 de 04 de 2010.]
23. **Microsystems., Sun. 2008.** *MySQL 5.0 Manual de Referencia.* 2008.
24. **Nonaka, Takeuchi &. 1995.** 1995.
25. **Pérez, Z. R. 2008.** *Gestión de conocimiento en los proyectos de software.* La Habana. : s.n., 2008.
26. **Pressman., R. Quinta Edición.** *Ingeniería de Software. Un Enfoque Práctico.* . Mc Graw Hill. : s.n., Quinta Edición.
27. **R. Kazman, P. Clements y M. Klein. 2001.** *Evaluating Software Architectures. Methods and case studies.* s.l. : Addison Wesley, 2001.
28. **Ramírez, Juan Carrión Maroto y Fabián. 2000.** Modelos de Gestion del Conocimiento. [En línea] 2000. [Citado el: 15 de 12 de 2009.] <http://www.gestiondelconocimiento.com/modelos.htm>.
29. **Rational, Ayuda del. 2003.** *Ayuda del Rational.* 2003.
30. **Reference, Joomla! 1.5 API. 2009.** Joomla! 1.5 API Reference. [En línea] 04 de 11 de 2009. [Citado el: 31 de 01 de 2010.] http://api.joomla.org/li_Joomla-Framework.html.
31. **Report, Knowledge Management Research. 2000..** KPMG. [En línea] 2000. [Citado el: 10 de 12 de 2009.] <http://www.kpmg.co.uk/kpmg/uk/services/manage/pubs/km2000.pdf>.
32. **Reynoso, Carlos Billy. 2004.** *Introducción a la Arquitectura de Software.* UNIVERSIDAD DE BUENOS AIRES : s.n., 2004.

33. **Rodríguez, Zulem Pérez. 2008.** *Gestión del Conocimiento en los proyectos de software*. La Habana : s.n., 2008.
34. **Salazar, A. A. 2000.** Gestión del Conocimiento. [En línea] 2000. [Citado el: 15 de 12 de 2009.] <http://www.gestiondelconocimiento.com/documentos2/apavez/gdc.htm>.
35. **Salazar, Alejandro Andrés Pavez. 2000.** *Modelo de implantación de Gestión del Conocimiento y Tecnologías de Información para la Generación de Ventajas Competitivas*. Santa María : s.n., 2000.
36. **Skyrme, Dr David J. 2000.** *Knowledge Management Solutions - The IT Contribution*. 2000.
37. **Southwell, Chris Snyder y Michael. 2005.** *Pro PHP Security*. New York : s.n., 2005.
38. symfony.es. [En línea] [Citado el: 2009 de 12 de 6.] <http://www.symfony.es/>.
39. **Welicki, León E. (s.f.).** *Implementando Extreme Programming en la plataforma*. (UPSAM, Departamento de Lenguajes, Sistemas Informáticos e Ingeniería del Software, Madrid España.) Obtenido de <http://www26.brinkster.com/lwelicki/articles/Implementando%20Extreme%20Programming%20en%20la%20plataforma%20NET.pdf>
40. **Z., Lucia I. Passoni y A. 2008.** *Un modelo de Gestión del conocimiento en los departamentos académicos*. 2008.

Anexos

Anexo A Encuesta de Validación

Marque con una X, según su criterio, sus conocimientos acerca de las Arquitecturas de Software:

1	2	3	4	5	6	7	8	9	10

Antes de proceder a la valoración de los criterios nos gustaría que autoevalúe su conocimiento acerca del tema en cuestión:

FUENTES DE ARGUMENTACIÓN	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted.			
Su experiencia obtenida.			
Trabajos de autores nacionales consultados.			
Trabajos de autores extranjeros consultados.			
Su propio conocimiento del estado del problema en el extranjero.			
Sus conocimientos empíricos sobre el tema.			

Debe valorar los aspectos que se relacionan a continuación utilizando una escala de 1 a 5, donde 5 indica el máximo, ajustado a los siguientes términos:

Puntos	Escala	
5	Excelente (E)	Totalmente de acuerdo (TA)
4	Bueno (B)	Muy de acuerdo (MA)
3	Aceptable (A)	De acuerdo (A)
2	Cuestionable (C)	Parcialmente de acuerdo (PA)
1	Malo (M)	En desacuerdo (D)

Marque con una X, según su criterio, la calificación otorgada para cada pauta.

Criterio	Calificación				
	E	B	A	C	M
Criterio de mérito científico					
1. Calidad de la investigación.					
2. Novedad Científica.					
3. Aporte Científico.					
Criterio de implantación	TA	MA	A	PA	D
4. Necesidad de la arquitectura propuesta.					
5. Contribución al proceso de Gestión del Conocimiento.					
Criterios de Generalización					
6. Facilidades de utilización para gestionar conocimiento.					
7. Posibilidad de uso al alcance de todos.					
8. Posibilidades de extensión mediante la inclusión de nuevos componentes para la Gestión del Conocimiento.					

Valoración Final:

Elementos a Suprimir:

Elementos a Mejorar:

Elementos a Añadir:

Categoría Final de la Investigación:

Marque con una X, la calificación otorgada para la investigación de manera general.

___ **Excelente:** *Alta novedad científica, con aplicabilidad.*

___ **Bueno:** *Novedad Científica.*

___ **Aceptable:** *Suficientemente bueno con reservas.*

___ **Cuestionable:** *No tiene relevancia científica.*

___ **Malo:** *No aplicable.*

Anexo B

Tabla 3 Competencia de los expertos sobre el tema

No	Nombre del Experto	Conoc	P1	P2	P3	P4	P5	P6	Ka	Kc	K	Competencia
1	1	7	0.2	0.4	0.05	0.1	0.1	0.1	1	0.7	0.825	ALTO
2	2	8	0.2	0.4	0.1	0.1	0.1	0.1	1	0.8	0.9	ALTO
3	3	6	0.2	0.4	0.1	0.05	0.1	0.1	1	0.6	0.775	MEDIO
4	4	5	0.2	0.4	0.1	0.05	0.1	0.1	1	0.5	0.725	MEDIO
5	5	10	0.2	0.5	0.1	0.05	0.1	0.05	1	1	1	ALTO
6	6	6	0.2	0.4	0.1	0.1	0.05	0.1	1	0.6	0.775	MEDIO
7	7	8	0.3	0.4	0.1	0.1	0.05	0.1	1.1	0.8	0.925	ALTO
	Total	7.14286	0.21	0.41	0.09	0.08	0.09	0.09	0.98	0.71	0.85	ALTO

Fuente: Elaboración propia

Anexo C

Tabla 4 Puntos de Corte

No	Aspectos	C1	C2	C3	C4	Suma	P	N-P	
1	Calidad de la investigación.	-0.57	3.72	3.72	3.72	10.59	2.65	-1.78	Muy adecuado
2	Novedad Científica.	-3.72	0.18	3.72	3.72	3.90	0.97	-0.11	Muy adecuado
3	Aporte Científico.	-3.72	-0.18	3.72	3.72	3.54	0.88	-0.02	Muy adecuado
4	Necesidad de la arquitectura propuesta.	-3.72	0.18	3.72	3.72	3.90	0.97	-0.11	Bastante Adecuado
5	Contribución al proceso de Gestión del Conocimiento.	-0.57	1.07	3.72	3.72	7.94	1.98	-1.12	Bastante Adecuado
6	Facilidades de utilización para gestionar conocimiento.	0.18	0.57	3.72	3.72	8.18	2.05	-1.18	Bastante Adecuado
7	Posibilidad de uso al alcance de todos.	-1.07	-0.57	1.07	3.72	3.15	0.79	0.08	Bastante Adecuado
8	Posibilidades de extensión mediante la inclusión de nuevos componentes para la Gestión del Conocimiento.	0.57	1.07	1.07	3.72	6.42	1.61	-0.74	Bastante Adecuado

Fuente: Elaboración propia